EHT – 10/EHT – 10/2

System Development Guide

OS Version 2.0

EPSON

Y24399100801
M013B

REMARKS

(2) The contents of this manual are subject to change without notice.

(3) All efforts have been made to ensure the accuracy of the contents of this manual. However, should any errors be detected, SEIKO EPSON would greatly appreciate being informed of them.

(4) The above notwithstanding, SEIKO EPSON can assume no responsibility for any errors in this manual or their consequences.

CONTENTS

PART 1 SOFTWARE

PART 2. HARDWARE

PART 3. BASIC

PART 4. APPENDIX

INTRODUCTION

< PURPOSE OF THIS MANUAL >

This manual describes the functions of the operating system for the SEIKO EPSON EHT-10/EHT-10/2 system. It is intended for system house users who are to develop applications programs which make the best of the EHT-10/EHT-10/2's capabilities.

The reader is assumed to be familiar with the following:

- Basic knowledge about the CP/M operating system
- General knowledge about machine-language programming
- Z80 instructions

< BEFORE READING THIS MANUAL >

This manual uses the following notational conventions:

(1) Types of the EHT-10 series

In EHT-10 series, there are three types as follows.

```
EHT-10    --------- Touch key type
EHT-10/2  --------- Keyboard type with backlight
EHT-10/2B ------- Keyboard type without backlight
```

In this manual, EHT-10/EHT-10/2 are the general names of the above three types unless there is the special note for the name.

(2) Data representation

This manual uses binary, decimal, and hexadecimal numbers. They are represented in the formats:

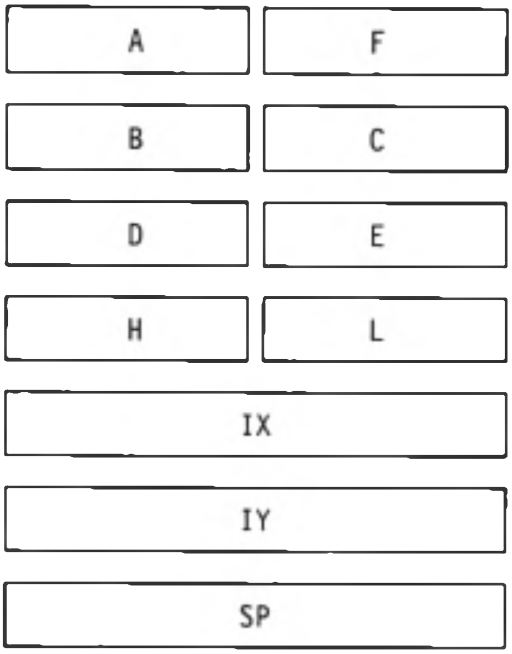
```
Binary:        00100011B (Numbers are followed by 'B')
Decimal:       35 (Only numerals)
Hexadecimal:   23H (numbers are followed by 'H')
```

Character constants are enclosed in apostrophes (') or double quotation (").

Example: 'ABC' or "ABC"

(3) Register Representation

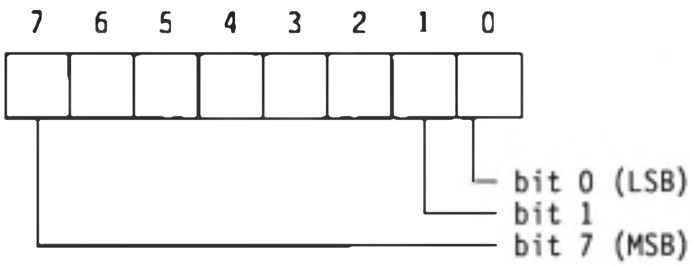
The EHT-10/EHT-10/2 registers are illustrated below.



Registers are expressed at A, B, DE, HL, and so on. They may sometimes be followed by the word "register" to clearly identified as the Z flag (or Z), the C flag (or C), and so on.

(4) Bit Representation

Bits are numbered 0, 1, and so on, from the lowest order bit (0) to the highest order bit. The lowest order bit is referred to as the least significant bit (LSB) and the highest order bit as the most significant bit (MSB).



(5) Address Representation

Addresses are generally represented in hexadecimal notation. I/O addresses are prefixed by "P".

Examples:

0010H	Memory address 10H
P10H	I/O port address 10H

Note that the contents of I/O addressed may differ during read and write operations.

< *THE CONSTITUTION OF THIS MANUAL* >

This manual consists of four parts, which are software, hardware, BASIC, APPENDIX.

Part 1 - SOFTWARE - describes the features of the EHT-10/EHT-10/2 and the information which are necessary to develop software like as BIOS, BDOS, I/O, system function, system hook, and etc.
Part 1 is divided into 11 chapters.The contents are as follows.

chapter 1, 2	Overview and basic operation
chapter 3 to 8	BIOS, BDOS, I/O, interrupt, and etc
chapter 9	System function
chapter 10, 11	How to use expansive function

Part 2 - HARDWARE - explains the hardware and interfaces. It also explains the necessary information to use the universal unit.
Part 2 is divided into 6 chapters. The contents are as follows.

chapter 1	Overview of the hardware
chapter 2 to 4	I/O registers, interfaces and etc
chapter 5	Power supply
chapter 6	Advice for designing the circuit

Part 3 - BASIC - is divided into 3 chapters.

chapter 1	Memory control of BASIC
chapter 2	Expansion of the commands including the commands analyzing routine.
chapter 3	Expansion of the sequential access devices.

Part 4 - APPENDIX - describes the various kinds of code tables and BIOS call list. And at the end of this manual is the index.

Because each section of this manual describes each subject independently, you don't have to read this manual from the beginning to the last. You can use the contents to search the part you want to read.
However, we recommend to read the chapter 1 and the chapter 2 of software part and the chapter 1 of the hardware part first because the overview of the EHT-10/EHT-10/2 is described there.

PART 1 SOFTWARE

CHAPTER 1 SYSTEM OVERVIEW

1.1 Characteristics of System

1.1.1 EHT-10/EHT-10/2 concepts

EHT-10/EHT-10/2 has the new functions in addition to the basic functions of PX-4/HX-40. EHT-10/EHT-10/2 is a hand-held computer much smaller than PX-4/HX-40.

Although the EHT-10/EHT-10/2 can be carried by one hand, EHT-10/EHT-10/2 has mass memory of 256-Kbyte (maximum) RAM, 128-Kbyte system ROM, and 128-Kbyte (maximum) application ROM, that is much more than general personal computers. EHT-10/EHT-10/2 is equipped with CP/M and BASIC in standard so that application programs can easily be created.

Further, EHT-10/EHT-10/2 has user-friendly functions such as the touch panel with LCD and IC card read/write functions so that the operators can easily handle the terminals without special education and training.

Small-sized and high-performance mobile computer EHT-10/EHT-10/2 is most appropriate for order issuing and accepting jobs for distribution industry, liaison jobs for financial institutions, electricity, gas, and water gauge examination, stock-taking and supply order issuing for supermarkets, cargo booking for carrying trade and agriculture industry, and production management jobs.

1.1.2 Characteristics of the system

(1) Extended CP/M version 2.2

1 The extended CP/M Ver. 2.2 is used as OS for EHT-10/EHT-10/2. This OS enables the users to easily create application programs. The application programs of other devices that operate under CP/M can be converted by taking into account of display screen and input method.

2 Main memory can be extended up to 256 Kbytes. A part of main memory (up to 232 Kbytes) can be used as RAM disk. A RAM disk can be handled in the same way as a floppy disk and can be accessed in high speed. Further an IC card can be used instead of a floppy disk.

3 Application programs input from the ROM drive can be loaded to the main RAM for execution as in the previous CP/M way. However, these application programs can be directly executed in the ROM drive to use memory efficiently and to have less power consumption. Up to 1-Mbit (128-Kbyte) ROM can be mounted in the ROM drive so that a large-size application programs can be handled. Further, an application program stored in an application ROM inserted in the ROM drive can be automatically executed at power on.

(2) Software development environment

1 Application programs in BIOS and BASIC level has the compatibility with PX-4/HX-40 and conversion can easily be done (however, the display size and input methods must be noted).

2 EPSON BASIC is supported so that programs can easily be created.

3 The development cartridge is supported in standard as an option. Using the development cartridge and development software(WAD), the users can easily create and debug application programs.

(3) Touch panel and keyboard

1 The input screen and the display screen are the same for EHT-10 for higher operationability and extendability. The touch panel has 70 (horizontal 5 x vertical 14) input points and OS supports normal, alphanumeric and kana input modes for the touch panel. The keys can be freely redefined in normal mode by application programs so that user-friendly software can be created with united input and display screens.

2 EHT-10/2 has 34 keys and OS supports normal, alphabet, and kana modes. Each mode can be identified from the lit LED. The Keys can be redefined in normal or alphabet mode.

(4) LCD and EL backlight

1 The size of EHT-10 real screen (LCD) is 12 columns x 14 lines and the size of EHT-10/2 real screen (LCD) is 20 columns x 4 lines. In addition, 12 columns x 42 lines of EHT-10 virtual screen and 20 columns x 25 lines of EHT-10/2 virtual screen are supported so that application programs requiring larger screen can be created and executed.

2 A model (EHT-10/2B) that has EL backlight is provided so that it can be used in a dark room.

(5) IC card equivalent to ISO

By using the IC card reader/writer equipped in standard, IC cards can be handled in the same way as floppy disks (storing collected data, distributing programs, etc.).

(6) Barcode reader

Barcode reader I/F is equipped in standard. OS supports reading barcodes that are JAN/EAN/UPC, 3 of 9, codabar, and interleaved 2 of 5 so that barcodes can be easily used.

(7) Kanji support

OS supports 174 characters of Japanese Characters (kana) and symbols, and 996 kanji characters. Further JIS level-1 kanji characters can be used when JIS level-1 ROM is mounted in the ROM drive. Up to 6144 external characters (gaiji) can be registered (the maximum number differs depending on the media size).

(8) Support devices

1 For EHT-10/EHT-10/2, all memory I/O devices are handled as disk drives.

a. RAM disk (A:)
A part of RAM can be used as a disk to realize a disk that can be accessed in high speed.

b. ROM drive (B:)
The ROM drive is supported so that a program can be executed in ROM.

c. IC card (C:)
IC card is supported so that it can be used as an R/O disk (ROM card) or R/W disk (RAM card).

2 The following device I/F are supported in addition:
Floppy disk (D: or E:)
Barcode I/F
RS-232C I/F
Cartridge I/F (printer unit)
Clock
Buzzer

(9) Timer management

1 Even when an application program is being executed, the alarm function displays the alarm screen to notify the user that specified time came.

2 The wake function turns on the main frame power to execute the specified program at the specified time.

(10) Other functions

1 The MENU function enables the user to select programs easily.

2 The DLL function can receive and execute programs sent from the host computer. The received program is directly initiated. The protocol used for transmission can be freely changed. Further, the protocol can be extended to the one exclusively used by an application.

3 The system menu function can set the system environment, send or receive files, manage disks, and supports the test function.

4 The calculator function can perform arithmetic operations and the result of arithmetic operation can be fetched as data to be input to an application. Arithmetic operations are performed in BCD so that results with less errors can be obtained.

5 If the power is turned off in continue mode, processing can be continued when the power is turned on again.

6 Self-test is performed at power on for higher system reliability.

7 The sleep mode and automatic power off functions prevent unnecessary power consumption.

1.2 System Configuration

Figure blow shows the devices supported by EHT-10/EHT-10/2.

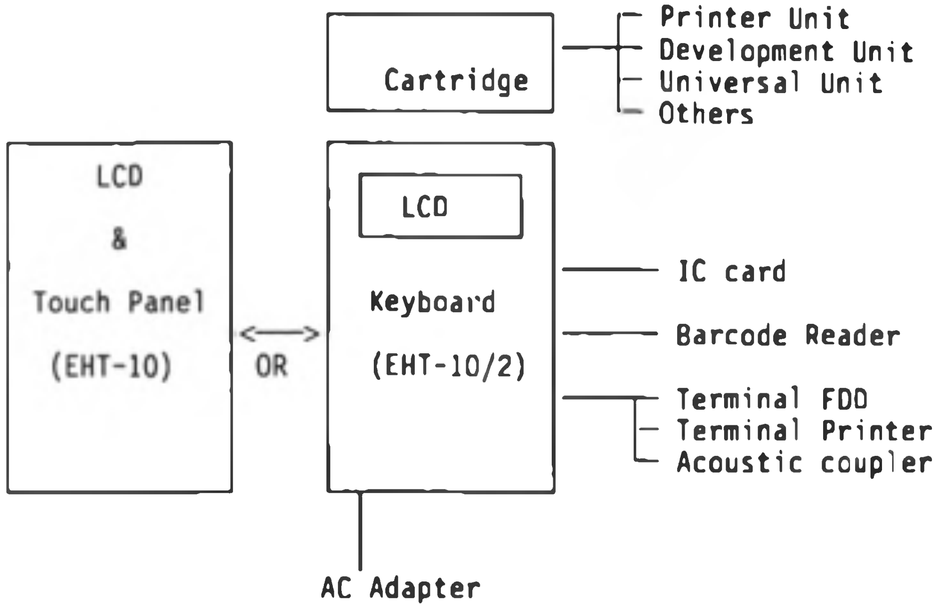


Fig. 1.1 EHT-10/EHT-10/2 System Configuration

EHT-10/EHT-10/2 consists of two CPUs, using Z-80 as the main CPU and 7508 as the slave CPU.

Memory space are used as four main memory banks and subbanks. 64-Kbyte RAM and 128-Kbyte system ROM are equipped in standard. RAM can be extended up to 256 Kbytes. See "Section 1.4 Memory Map" for details on banks.

Slave CPU 7508 controls the keyboard, clock and power supply. Main CPU Z-80 supports RS-232C, cartridge I/F, IC card I/F, and barcode I/F.

1.3.1 Overview

The EHT-10/EHT-10/2 system software is stored in the 128-Kbyte system ROM. Since the system software corresponds to EHT-10, EHT-10/2, and EHT-10/2B, the common system ROM is mounted in EHT-10, EHT-10/2, and EHT-10/2B. The system ROM consists of the OS, utility, BASIC, and kanji sections.

```
1FFFFH +---------------------+
       | Kanji               |   --- The kanji section consists of
       |(Chinese Char.)      |       996-kanji and 174-nonkanji
       | Part                |       character codes.
18000H +---------------------+
       | BASIC               |   --- BASIC interpreter section
       | Interpreter         |
       |                     |
10000H +---------------------+
       | Utility             |   --- The utility section consists
       | Part                |       of a part of extended BIOS
       |                     |       and utility programs such as
08000H +---------------------+       system menu and calculator.
       | OS                  |   --- This is the core section of
       |                     |       EHT-10/EHT-10/2 OS.
       |                     |
00000H +---------------------+
```

- The size of each section is 32 Kbytes.

Fig. 1.2 Structure of system ROM

1.3.2 Structure

Figure 1.3 shows the structure of EHT-10/EHT-10/2 operating system functions.

```
                      ┌System ┬─ Kernel ──┬─ Starter
                      │Section│  Section  ├─ Relocater
                      │       │           ├─ Interrupt Process
                      │       │           ├─ Resident Process
                      │       │           ├─ MENU Function
                      │       │           └─ DLL Function
EHT-10/EHT-10/2     ─ │       └─ CP/M ────┬─ CCP
Operating System      │          Section  ├─ BDOS
                      │                   └─ BIOS
                      ├Utility ┬──────────── Calculator
                      │        └System ───┬─ CONFIG Function
                      │         MENU      ├─ UL/DL Function
                      ├BASIC              ├─ DISK Function
                      └Kanji              └─ TEST Function
```

Fig. 1.3 OS Structure

(1) Structure of system section

The system section consists of the kernel and CP/M sections.

1 Starter
The starter is the boot loader that initiates the system and performs initialization.

2 Relocater
The relocater is a group of processing routines that relocate programs resided in RAM. The relocater is initiated by the starter or BIOS WBOOT.

3 Interrupt processing
Interrupt processing performs various interrupts such as 7508 (key input, alarm, 1-second interrupt, and power), ART (serial receiving), ICF, OVF, and EXT (printer, interval, and IC card back panel).

4 Resident processing
Resident processing is structured by a part of interrupt processing, bank switch processing, and the BIOS and BDOS interface sections.

5 MENU
MENU displays executable program files in the menu screen so that the user can select and execute a program.

6 DLL
DLL down-loads an execution program from the host computer through a communication line and executes the program.

7 CCP section
The CCP section receives data related to program initiation from MENU and DLL and initiates a program.

8 BDOS section
The BDOS section is the CP/M disk file management section. The BDOS section can be used by an application program calling BDOS.
EHT-10/EHT-10/2 supports the ROM drive, IC card, and a part of RAM as disks.

9 BIOS section
The BIOS section is the CP/M I/O handler. The standard CP/M BIOS is extended for EHT-10/EHT-10/2.

(2) Utility section

The utility section consists of the calculator and system menu functions. Refer to EHT-10/EHT-10/2 Operating Manual for details.

(3) BASIC section

The BASIC section contains the EPSON BASIC interpreter. The BASIC interpreter is directly executed in ROM instead of expanded in RAM, so that memory space in main RAM can be used efficiently. Refer to "BASIC PART" and EHT-10/EHT-10/2 BASIC Reference Manual for details.

(4) Kanji section

The kanji section stores 996-kanji and 174-nonkanji fonts.

1.4 Memory Map

1.4.1 Memory space

System ROM (128 Kbytes), RAM (up to 256 Kbytes), application ROM (up to 128 Kbytes), in total of up to 512 Kbytes, are supported as EHT-10/EHT-10/2 memory space.

However, main CPU Z-80 can directly access only 64-Kbyte memory space starting from 0000H to FFFFH.

Because of this, the bank method is used for EHT-10/EHT-10/2 so that main CPU Z-80 can directly access 512-Kbyte memory space.

In the EHT-10/EHT-10/2 bank method, main banks and subbanks are used in memory map shown in FIgure 1.4.

Generally, OS switches banks so that application programs do not require to take into account of bank switching. However, high- level application programs can be created by taking into account of the concept of bank switching and using this concept for application program execution.

[Notes]
Banks are specified as follows in this manual:
[main bank]#[subbank]

Example
[bank 2#1]: Subbank 1 in bank 2. 2#1 starts from 8000H to FFFFH in application ROM and is mapped from 6000H to DFFFH in the memory map.

However, the system bank is called [system bank] because the system bank does not have any subbank.

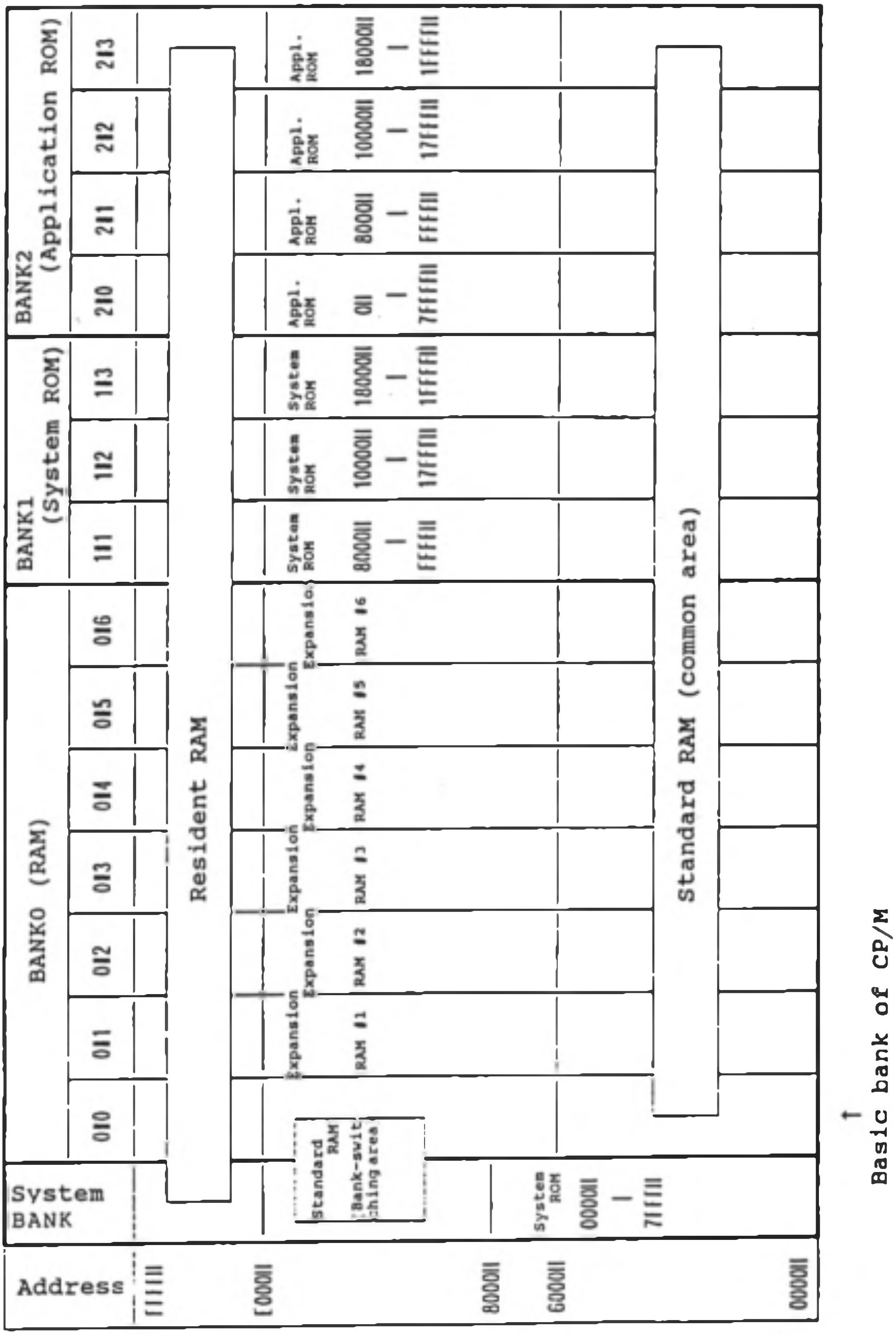


Fig. 1.4 Memory space

(1) System bank

The system bank does not have any subbank. The system bank is structured as follows:
0000H to 7FFFH: 00000H to 07FFFH (OS section) in system ROM
8000H to DFFFH: 24 Kbytes in standard RAM (8000H to DFFFH in bank 0#0)
E000H to FFFFH: Resident RAM section (a part of standard RAM that can be directly accessed by CPU Z-80 regardless of bank switching.)

(2) Bank 0 (bank in all RAMs)

All RAMs have bank 0. Bank 0 has the following subbanks depending on the size of extended RAM:

Size of extended RAM	Subbanks						
	0#0	0#1	0#2	0#3	0#4	0#5	0#6
0KB(only for Standard RAM)	0						
64KB	0	0	0				
128KB	0	0	0	0	0		
192KB	0	0	0	0	0	0	0

An empty field indicates that there is no subbank.

Bank 0 is structured as follows:
0000H to 5FFFH: This is a part of standard RAM. This can be accessed regardless of main bank and subbank switching.
8000H to DFFFH: This is switched with a part of another extended RAM depending on the subbank. When bank 0#0 is specified, this becomes a part (24 Kbytes) of standard RAM.
E000H to FFFFH: Resident RAM (same as the system bank)

The BIOS and BDOS entries are relocated in bank 0#0. Bank 0#0 is used as the base of CP/M. Also, programs in load and execute mode are loaded in bank 0#0.

Generally an application program does not switch the bank in extended RAM to the one from 0#0 to 0#6 because the extended RAM is used as a disk by the system.

(3) Bank 1 (bank in system ROM)

Bank 1 is the extension of system bank and is bank 0 6000H to DFFFH replaced as a part of system ROM.

Bank 1 has the following subbanks:
Bank 1#1: 07FFFH to 0FFFFH (utility section) in system ROM are mapped.
Bank 1#2: 10000H to 17FFFH (BASIC section) in system ROM are mapped.
Bank 1#3: 18000H to 1FFFFH (kanji section) in system ROM are mapped.

(4) Bank 2 (bank in application ROM)

Bank 2 is bank 0 6000H to DFFFH replaced as a part of application ROM (ROM drive). Bank 2 has the following subbanks depending on the size of application ROM:

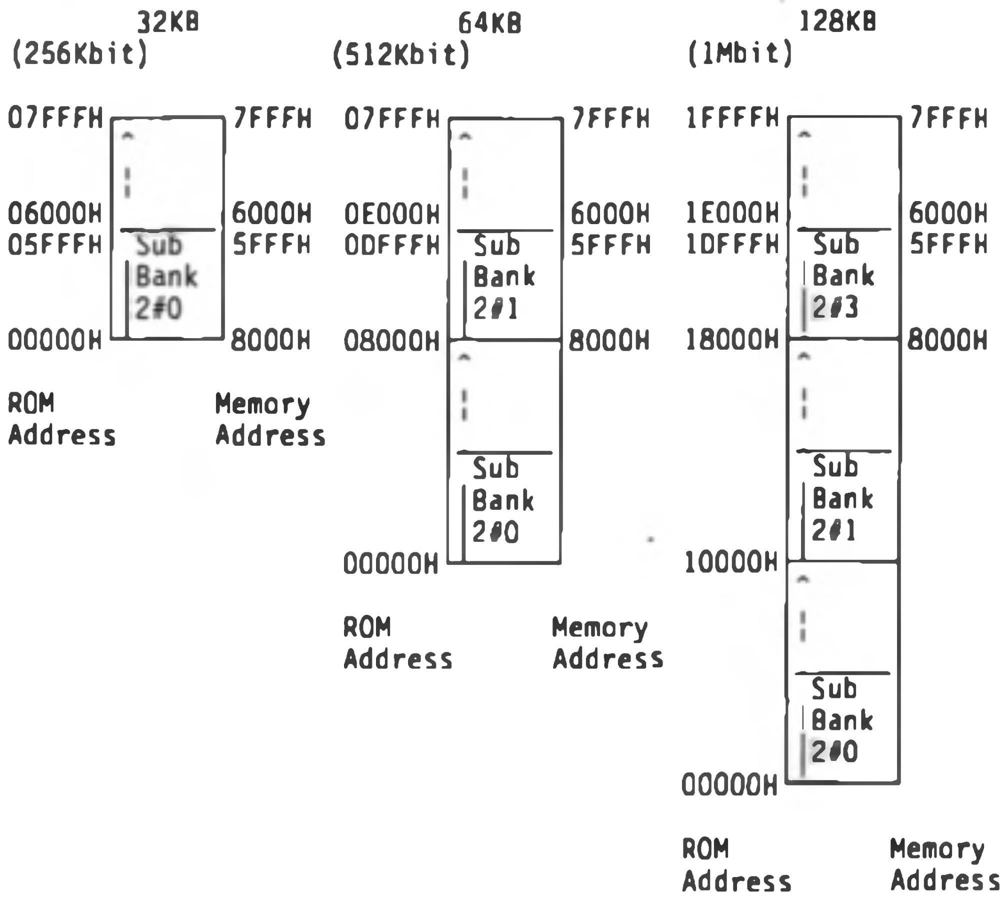


Fig. 1.5 Structure of Application ROM

(Note 1) For application ROM, ROM physical addresses differ from the logical addresses mapped in memory.

1.4.2 Memory map

The resident RAM area (E000H to FFFFH) and bank 0#0 contains the system parameters, entries of resident processing routines BIOS and BDOS. Bank 0#0 is used as the base of CP/M.

The memory map of bank 0#0 for EHT-10 differs from that of EHT-10/2. Figures 1.6 and 1.7 show the memory maps of EHT-10 and EHT-10/2.

(1) RBDOS and RBIOS

RBDOS1 is functionally same as RBDOS2 and RBIOS1 is functionally same as RBIOS2. An application program in load and execution mode (loaded and initiated starting from 100H in RAM) uses RBIOS1 and RBDOS1 (by calling BIOS and BDOS).

An application program in ROM-based mode (operates at 6000H to DFFFH in bank 2) uses RBIOS2 and RBDOS2 (both have fixed addresses) because RBIOS1 and RBDOS1 may be placed at the different bank in ROM.

(2) RSYSPR system area

RSYSPR is a system program in resident section that performs bank switching interrupt processing. The system area consists of work area, jump table, and hook used by the system.

(3) Gaiji definition area and key redefinition area

Gaiji font is defined in the gaiji definition area and key codes are redefined in the key definition area.

(4) Virtual screen and system screen

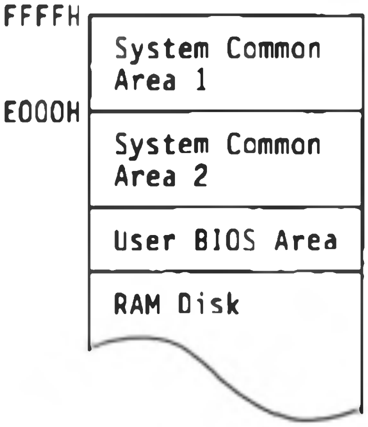
The virtual and system screens are the display data storage areas. The size of virtual screen is fixed and the map is not changed even if the virtual screen size is modified.

(5) VRAM

For EHT-10, VRAM is used as the VRAM data save area. For EHT-10/2, VRAM consists of VRAM1, VRAM2, and VRAM3. VRAM1 is used as the virtual screen 1 VRAM, VRAM 2 is used as the virtual screen 2 VRAM, and VRAM 3 is used as the system screen VRAM.

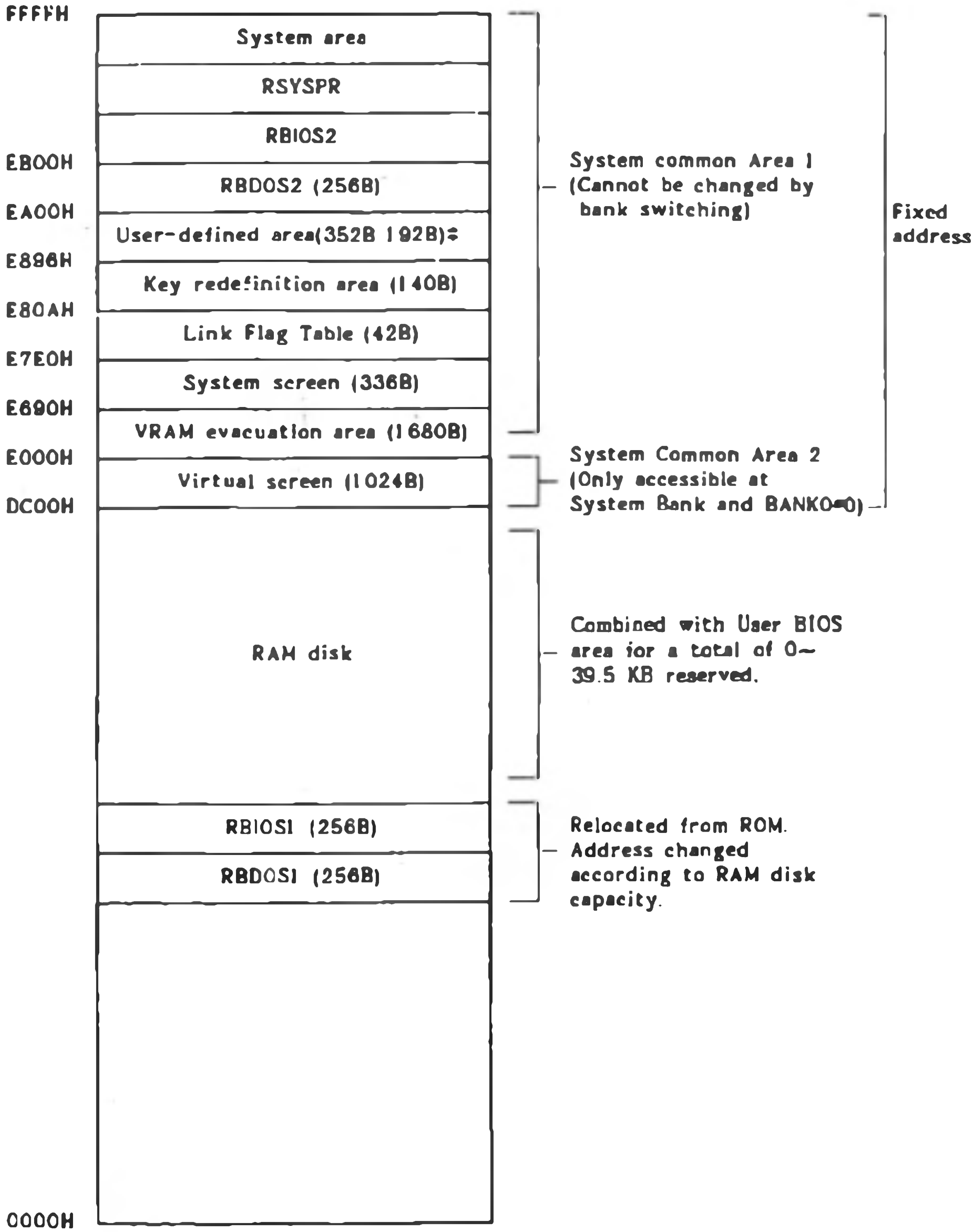
(6) User BIOS area

The default size of user BIOS area is 0. However, the user BIOS area is reserved between the RAM disk and system common area 2 when there is user BIOS.



(Note) The position of user BIOS area differs depending on EHT-10 or EHT-10/2.

The user BIOS area is used when a part of the user program is made to remain resident after termination of user program.



* *352 bytes are required if the screen will be used from top to bottom, or 192 bytes for side-to-side use.*

Fig. 1.6 EHT-10 Bank 0#0 Memory Map

Note 1: DLL is executed when forced-load is specified.
Note 2: A specific program is executed when forced-selection is specified. The above two items are specified by CONFIG utility Exec type.
Note 3: Application is initiated by the wake string when Wake is used for starting.
Note 4: At system initialization or reset, menu processing is executed even if there is only one execute-mode program in the disk.

2.2.2 Status change

Figure 2.2 shows the changes of EHT-10/EHT-10/2 status.

EHT-10/EHT-10/2 has the following status:
(1) Power off (restart) status
(2) Power off (continue) status
(3) Application operation status
(4) MENU screen
(5) Alarm/Wake screen
(6) System menu screen
(7) Calculator screen
(8) DLL screen

Power off status is in either restart or continue mode. The mode is determined according to the condition when the power is turned off.

EHT-10/EHT-10/2 is in MENU or DLL screen status when the power is turned on in restart mode. EHT-10/EHT-10/2 is in the status held at power off when the power is turned on in continue mode.

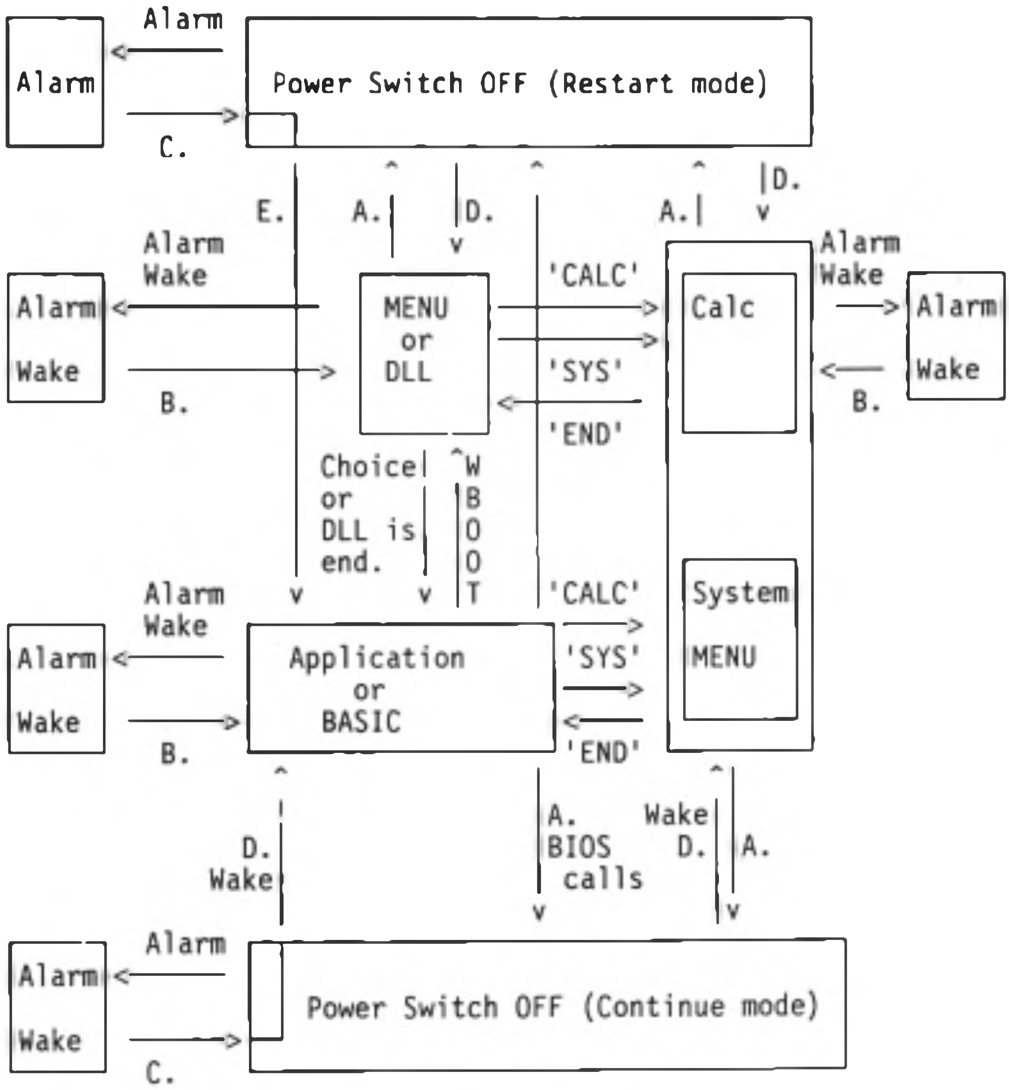


Fig. 2.2 Changes of System Section Status

('END' means the Return key for EHT-10/2.)

A Power switch off, power failure, automatic power off
B Power switch off, power failure, 50 seconds elapse, END
C Power switch off, power failure, 50 seconds elapse, END, power switch on
D Power switch on
E Wake

2.3.1 System initialization

(1) Overview

If program execution enters wild run status, 7508 (slave CPU) operation enters wild run status, RAM contents are destroyed due to lowered battery voltage, or the system is initialized for the first time, system initialization is performed to reset the EHT- 10/EHT-10/2 environment (the contents of ROM are protected in general when the battery voltage is lowered).

(2) Initiation

The system initialization is initiated under one of the following conditions:

1 7508 operation hangs up or the battery voltage is lowered. Make the AC adapter enter ready status and open the back panel to press the reset switch.
2 The power is turned off during system initialization and then the power is turned on again (during CONFIG in system initialization).
3 The result of system area sum check performed at power on is not equal to the result of sum calculated at power off.
4 An error occurs as a result of the following check operations during reset processing and initialization is specified (see "Section 2.5 Self-Test):

  (a) The total of the internal RAM disk size and the BIOS size exceeds the maximum. The maximum values are as follows:
    EHT-10: 39.5 Kbytes
    EHT-10/2: 40.0 Kbytes
  (b) The BDOS or BIOS loading address in RAM is not equal to the value calculated from the RAM disk size during reset processing.

(3) Processing

The system is initialized in the following way:
1 A keyboard reset command is sent to CPU 7508 to initialize the keyboard.
2 The interrupt mask status and the bank status are initialized.
3 The system work area (RSYSAR1) is initialized.
4 System device cold boot (note 1) is performed.
5 The display screen is turned on to display system initialization messages. When the cause of initiating system initialization is 3 or 4 explained in "Section (2) Initiation", a system error message is displayed for about one second before system initialization message.
6 The default value is set as the size of RAM disk and the disk is formatted.
7 Reset processing is performed (explained later).
8 BIOS BOOT processing is performed.

9 System menu CONFIG is initiated so that the user can perform system initialization. Table 2.1 shows the system default values.

(Note 1) Device cold boot and device warm boot

(1) Device warm boot

1 The I/O registers are initialized.
2 LCD is initialized.
3 Whether ROM is mounted in the ROM drive is checked.
4 Whether an extended RAM is mounted is checked.
5 Whether the cartridge device is mounted is checked and the cartridge device is initialized.

(2) Device cold boot

1 The system parameters (RSYSAR2 and RSYSAR3) are initialized and the resident processing routines are loaded.
2 1-second interrupts are allowed and the alarm is suppressed.
3 The DIP switches are read to determine the corresponding country.
4 Device warm boot is performed.
5 The display and keyboard are initialized.

(4) End of system initialization

System initialization ends when CONFIG ends. Then, MENU or DLL processing is initiated.

Table 2.1 Default Values of System Parameters

Parameter	Contents	Default Value
Self Test	Self-test function ON/OFF	ON
BASIC	Parameters of BASIC	Open file 3 Record 128 bytes Buffer size 256 bytes
Calculate	Display mode of Calculator operations result	Floating
Date/Time	Date Time	00/00/00 (note 3) 00:00:00
Disk size	Size of internal RAM disk Size of User BIOS	31KB 0
DLL	Communication parameters for down loading.	Bit rate 4800 bps Bit length 8 bits Parity NON Stop bit 2 bits Protocol Filink Connector RS-232C
RS-232C	Communication parameters for RS-232C	Bit rate 4800 bps Bit length 8 bits Parity NON Stop bit 2 bits Control NON SI/SO Disable
Exec type	Initiation	Auto
Power OFF	Power control time	Main Power 5 min. Printer Unit 180 sec. Back light 3 min.
Printer	Printer output	Printer unit
Country	Character set for country	ASCII (note 1) or Japan

(Note 1) Japan or overseas is specified according to DIP switch setting.
(Note 2) The output destination for Calculate, Country, DLL, and Printer is initialized by reset processing.
(Note 3) The default values of date and time are set only during CPU 7508 reset processing.

2.3.2 Reset

(1) Overview

If program execution enters wild run status, press the reset switch at the side of main frame to reset data residing in RAM.

(2) Initiation

Reset is initiated under one of the following conditions:

1 The reset switch is pressed once in power on status.
2 After the power failed in power off status, the power switch is turned on or an alarm/wake occurs.
3 The result of user area (TPA or user BIOS area) sum check is not equal to the value calculated at power off.

(3) Processing

1 A keyboard reset command is set to CPU 7508 to initialize the keyboard.
2 Device cold boot is performed (see the section of system initialization).
3 The sizes of the standard RAM section and user BIOS area are added. System initialization is performed if the total exceeds the maximum value (39.5 Kbytes for EHT-10 and 40 Kbytes for EHT-10/2).
4 System initialization is performed if the result of adding the RBDOS1 or RBIOS2 load address and its size is not equal to the beginning of the standard RAM in the RAM disk.
5 The disk parameter blocks are set for RAM and ROM disks.
6 RAM disk sum check is performed. If the result unmatches, the user selects whether to perform RAM disk formatting.
7 User area sum check is performed if reset processing is initiated due to power failure in power off status.

(4) End of reset processing

Reset processing is automatically ended after the above operations and BIOS BOOT processing is started (however, control is returned to system initialization if reset processing is initiated from system initialization).

2.3.3 Restart mode

(1) Overview

The power is turned off in restart mode when the power switch is turned off during MENU or DLL processing.

(2) Initiation from power off status in restart mode

1 Power switch on
When the power switch is turned on, MENU or DLL processing is restarted or an application is directly initiated (see "Section 2.2.1 Control flow").

2 Alarm time
When the alarm time comes, the alarm is sounded and the alarm screen is displayed. Operation 1 is performed if the power switch is turned on at this point.
3 Wake time
Operation 1 is performed when the wake time comes. In this case, an application can be initiated by assuming the wake string as key input data.

(3) Restart mode conditions

1 The power switch is turned off in MENU or DLL status or the power is turned off automatically.
2 Restart mode is set by BIOS CONTINUE in an application and the power is turned off by the power switch or by BIOS POWER OFF in restart mode.

(4) Processing

1 Device warm boot is performed (see the section of system initialization).
2 The self-test is performed (see "Section 2.5 Self-Test").
3 High-pitched tone is sounded to indicate power off in restart mode.
4 BIOS WBOOT is performed.
5 Control is returned to MENU, DLL, or an application according to the initiation condition (power switch on, wake) and the system status (existence of execute-mode file, execution type).

(5) End

Processing ends after initiating MENU, DLL, or an application.

2.3.4 Continue mode

(1) Overview

Continue mode is used as the default during application execution. When the power is turned on after turned off in continue mode, the program can be continued from the status held at power off.

(2) Continue mode initiation from power off

1 Power switch on
The program is continued from the status held at power off.
2 Alarm time
When the alarm time comes, the alarm is sounded and the alarm screen is displayed. Operation 1 is performed if the power switch is turned on at this point.
3 Wake time
When the wake time comes, alarm is sounded, the wake screen is displayed, and then the program is continued from the status held at power off. In this case, the wake string is ignored.

(3) Continue mode conditions

1 The power switch is turned off during application execution.
2 The power is turned off because the automatic power off time comes during application execution.

3 The power is turned off by the BIOS POWER OFF in continue mode.
4 The power is turned off during calculator or system menu operation because the power switch is turned off or the automatic power off time comes.
5 The power switch is turned off when the power failed during operation other than system initialization, MENU, or DLL operation.

(4) Processing

1 Device warm boot is performed (see the section of system initialization).
2 Self-test is performed (see "Section 2.5 Self-Test").
3 A high-pitched tone is sounded to indicate power off in continue mode.
4 The register status held at power off is recovered and control is returned to the execution address used at power off.

(5) End

Control is returned to the address used at power off and then processing ends.

(6) Remarks

Generally, processing is continued from the status held at power off when the power is turned on in continue mode. However, the cartridge options are initialized at power off and cannot be continued. For the printer unit, complete continuation operation is not possible because specifications are reset during I/O operation at power on and the country specification and gaiji definition data are initialized.

See "Section 7.3 Cartridge Interface" to continue cartridge options.

2.3.5 Sleep function and automatic power off function

(1) Overview

To have less power consumption, EHT-10/EHT-10/2 supports the sleep and automatic power off functions. Further, EHT-10/2B also supports the automatic backlight off function.

(2) Sleep function

Sleep mode is enabled by using a Halt command to stop CPU when processing is waiting for input data during BIOS CONIN operation. When an interrupt occurs during Halt command execution, sleep mode is cleared and interrupt processing is executed. Processing enters in sleep mode again after interrupt processing when the interrupt is not a key input interrupt. When the interrupt is a key input interrupt, input operation is executed and CONIN is terminated.

The automatic power off, alarm, power switch off, and power failure functions can operate in sleep mode.

(3) Automatic power off function

The power is automatically turned off if any data is not input from the keyboard within the fixed time (default is 5 minutes) while CONIN operation is waiting for input data. Processing is continued from the status held at power off when the power is subsequently turned on (the power switch is turned on or the wake time comes). However, the power is turned on in restart mode if the power is automatically turned off during MENU or DLL processing.

The time for automatic power off is set by CONFIG.

(4) Automatic backlight off function

For EHT-10/2B, the backlight is automatically turned off if any data is not input from the keyboard within the fixed time (default is 3 minutes) while BIOS CONIN operation is waiting for input data. The backlight is turned on again when data is input from the keyboard or the main power supply is turned off and on. The automatic backlight off function is effective only when the backlight switch is on.

The time for automatic backlight off is set by CONFIG.

(5) Using the automatic power off, automatic backlight off, and sleep functions in an application

Some application programs input data by using CONST to poll the keyboard status. The automatic power off, automatic backlight off, and sleep functions do not operate for such application programs.

In order to use these functions in such cases, processing shown in FIg. 2.3 is required in the application programs. (See APPENDIX 9 SAMPLE 25)

```
                    (START)
                       |
                       v
   | The automatic power off time is set. |  (1)
                       |
                       v
   | The automatic backlight off time is set. |  (2)
                       |
 +-------------------->|
 |                     v
 |            | BIOS CONST |  (3)
 |                     |
 |                     v                      (4)   Yes
 |   < Is data input from the keyboard? >------------------
 |                     |                                   |
 |                     | No                                |
 |                     v                 (5)               v  (10)
 | +--< Has the automatic power off time come? >     | BIOS CONIN |
 | |                   |                                   |
 | | No                | Yes                               v  (11)
 | |                   v                             | BIOS BACKLIGHT |
 | |          | BIOS POWEROFF |  (6)                 |      (ON)      |
 | |                   |                                   |
 | +------------------>|<------ Power ON                   |
 |                     v                     (7)           v
 | +--< Has the automatic backlight off time come? >   ( RET )
 | |                   |
 | |  No               | Yes
 | |                   v
 | |         | BIOS BACKLIGHT(OFF) |  (8)
 | |                   |
 | +------------------>|
 |                     v
 |                 | HALT |  (9)
 |                     |<------ Interrupt
 +---------------------+
```

Fig. 2.3 Automatic Off Operation

(Note 1) Omit steps (2) , (7) , (8) , and (11) for EHT-10/EHT-10/2 since EHT-10/EHT-10/2 does not have the automatic backlight off function. If the automatic backlight off function is not required although EHT-10/2B is used, the above steps should be omitted.

Step Explanation

(1) The automatic power off time is set.
The automatic power off time is set. TIMEEND is set in two bytes as follows:

```
          (ATSOTIM1) + (TIMERO) -> (TIMEEND)
          (F021H)      (F02DH)     (F5F9H)
```

(2) The automatic backlight off time is set.
The automatic backlight off time is set. ELOFFEND is set in two bytes as follows:
(ELOFTIME) + (TIMERO) -> (ELOFEND)
(F023H) (F02DH) (F387H)

(3)&(4) Whether data is input from the keyboard is checked during BIOS CONIN.

(5)&(6) The automatic power off time is checked.
Whether the automatic power off time has come is checked. If the time has come, the power is turned off. Checking is done according to the expression below. The time has come if the result of calculation is negative.
(TIMEEND) - (TIMERO)
(F5F9H) (F02DH)

(7)&(8) The automatic backlight off time is checked.
Whether the automatic backlight off time has come is checked. If the time has come, the backlight is turned off. The following expression is used:
(ELOFEND) - (TIMERO)

(9) HALT
Sleep mode is enabled by a Halt command. Processing jumps to operation (3) by a Jump command after the Halt command.

(10) & (11) Data input from the keyboard is fetched and the backlight is turned on. Data input from the keyboard is fetched by BIOS CONIN and the backlight is turned on.

2.3.6 Power failure

(1) Overview

Power failure occurs if main-battery voltage is lowered (to about 4.7 V or less). If power failure occurs, the power failure screen is displayed to notify the user of power failure.

(2) Cause

Processing for power failure starts when a power failure interrupt is sent from the slave CPU (7508) to the main CPU.

1 Slave-CPU (7508) operations
If the voltage of the main battery is lowered to about 4 V or less, a power failure interrupt is sent to the main CPU and the power is also supplied from the subbattery. A power failure interrupt is sent every 1 second. The power is forcibly turned off at the main CPU if power off processing is not executed at the main CPU within 50 seconds. In this case, reset processing is performed when the power is turned on again.

2 Main-CPU operations
After accepting a power failure interrupt, the main CPU continues I/O operation until it can be left off if I/O operation is in progress, and then performs power failure processing.

(3) Processing

If power failure occurs, power off processing is executed (excepting the actual power off operation) in continue mode and then the power failure screen shown in Figure 2.4 is displayed.

```
Charge
Battery
```

EHT-10

```
Charge Battery
```

EHT-10/2

Fig. 2.4 Power Failure Screen

The power failure screen is displayed on and off every second and then the power is turned off under one of the following conditions:

1 30 seconds elapse after displaying the power failure screen is started.
2 The user turns off the power switch after displaying the power failure screen is started.
3 50 seconds elapse after power failure occurs.

The power failure screen is displayed by using the system screen so that displaying the power failure screen does not affect the user screen.

(4) Others

1 Recovering after the power is turned off
 - Connect the charger to charge sufficient electricity.
 - Replace the main battery.
 Perform one of the above operations and turn the power switch on to recover from power failure.
2 Continuation processing after power failure
 Processing can be continued after one of the operations explained in 1 if the power is turned off after the power failure screen is displayed. However, if the slave CPU forcibly turns off the power before the power failure screen is displayed (I/O operation is not ended within 50 seconds after power failure occurs), reset processing (same as processing executed when the reset switch is pressed) is executed but not continuation processing even if one of the operations explained in 1 is performed.
3 Power failure occurrence in power off status
 Power is consumed to backup 7508 and RAM even in power off status. The subbattery is supplying the required power if the voltage of main battery is lowered to the fixed value or less. In such a case, the power is not turned on even if the power witch is turned on. Therefore, perform one of the operations explained in 1 to execute reset processing and to recover the system. In this case, the system area sum check is executed and the result is compared with the value calculated at power off. If the result is not equal to the calculated value, it is assumed that the memory contents are destroyed in power

off status and system initialization is executed.

4 Alarm/wake at power failure

When power failure occurs, the alarm/wake function is disabled and the status held at power failure is recovered at the subsequent power switch on. Because of this, the alarm/wake operation to be performed between power failure occurrence and the subsequent power on may be omitted.

2.4.1 Overview

EHT-10/EHT-10/2 has the alarm function that sounds alarm and displays the alarm screen when the set time comes, and the wake function that automatically turns the power on if the power is off and performs the specified operation when the set time comes (however, either alarm or wake can be specified at the same time).

The alarm/wake operation differs depending on the power on or off status. Table 2.2 shows the outline of alarm/wake functions.

Type	Power off status		Power on status
	Restart mode	Continue mode	
Alarm	(1) The power is turned off in restart mode after the alarm screen is displayed. (2) Power on processing in restart mode is executed if the power switch is turned on while the alarm screen is displayed.	(1) The power is turned off in continue mode after the alarm screen is displayed. (2) Power on processing in continue mode is executed if the power switch is turned on while the alarm screen is displayed.	(1) The alarm screen is displayed.(at display timing.)
Wake	(1) Alarm is sounded and power on processing in restart mode is executed. Processing specified by the wake string is executed if it is specified.	(1) Alarm is sounded and power on processing in restart mode is executed. A wake string is ignored even if it is specified.	(1) The wake screen is displayed (at display timing.)

Table 2.2 Alarm/Wake Functions

2.4.2 Alarm

(1) Function

When the set time comes, alarm is sounded and the alarm screen shown in Figure 2.5 is displayed.

```
<Alarm>
MM/DD hh:mm
<Alarm mess-
 age (20 char
 acters)>

        Press
```

EHT-10

```
<Alarm> MM/DD hh:mm
<Alarm message (18 chr)>

Press <- key
```

EHT-10/2

Fig. 2.5 Alarm Screen

When alarm occurs in power off status and the power switch is turned on while the alarm screen is displayed, power switch on processing is executed.

(2) Setting and resetting

The alarm is set or reset by calling BIOS TIMDAT. See "Section 4.2 Overview of BIOS Commands" for TIMDAT.

(3) Initiation

1 Alarm time comes in power off status.
2 Alarm time comes in power on status.
  In this case, the alarm function is initiated at a timing so that I/O execution is not affected.

(4) End

1 If the alarm function is initiated in power off status, processing returns back to the status held before initiation (power off status in restart or continue mode) under one of the following conditions:
  (a) 50 seconds elapse after initiation.
  (b) The 'Press' (EHT-10) or 'Return' (EHT-10/2) key is pressed.
  (c) The power switch is turned off.
  (d) Power failure is detected.
  (e) The power switch is turned on. In this case, power switch on processing is executed.

2 If the alarm function is initiated in power on status, processing returns back to the status held before initiation under one of the following conditions:
  (a) 50 seconds elapse after initiation (see Note 1).
  (b) The 'Press' (EHT-10) or'Return' (EHT-10/2) key is pressed.
  (c) The power switch is turned off.
  (d) Power failure is detected.

(Note 1) The alarm display time can be modified by modifying the contents of system area ALRMTIME (F1CAH). The value is indicated in the units of seconds.

(5) Remarks

1 The alarm screen is displayed on the system screen and the system screen returns back to the status held before alarm occurrence after alarm processing ends.
2 In the alarm screen, control codes (00H to 1FH) are moved to "^" + "@" (40H) and displayed.
3 Alarm (or wake) occurrence is suppressed temporarily during one of the following operations:
 (a) Data is being received during DLL or DL.
 (b) Data is being input in alphabet or kana mode for EHT-10.
 (c) Power failure occurs.

2.4.3 Wake

(1) Function

The following operations are executed at the set wake time:

1 When processing is initiated in power off status in restart mode
 The power is turned on, alarm is sounded, and power on processing in restart mode is executed. If a wake string is set, the wake string is passed to application initiation processing and used as the program initiation parameters (for CP/M CCP).
2 When processing is initiated in power off status in continue mode
 The power is turned on, alarm is sounded, power on processing in continue mode is executed. A wake string is ignored even if it is specified.
3 When processing is initiated in power on status
 Alarm is sounded and the wake screen shown in Figure 2.6 is displayed.

```
<Wake>
MM/DD hh:mm
<Wake string
(20 character
 s)>
        [Press]
```

EHT-10

```
<Wake>  MM/DD hh:mm
<Wake  string  (18 chr)>

Press <- key
```

EHT-10/2

Fig. 2.6 Wake Screen

(2) Setting and resetting

Wake is set or reset by calling BIOS TIMDAT. See "Section 4.2 Overview of BIOS Commands" for TIMDAT.

(3) Initiation

1 Wake time comes in power off status.
2 Wake time comes in power on status.
 In this case, the wake function is initiated at a timing so that I/O execution is not affected.

(4) End

1 If the wake function is initiated in power off status, the same operations executed when the power switch is turned on and the wake function is initiated is executed.

2 If the wake function is initiated in power on status, processing is returned to the status held before initiation under one of the following conditions:

(a) 50 seconds elapse after initiation.
(b) The 'Press' (EHT-10) or 'Return' (EHT-10/2) key is pressed.
(c) The power switch is turned off.
(d) Power failure is detected.

2.5.1 Overview

EHT-10/EHT-10/2 performs RAM sum check at power on to improve system reliability and compares the result with the check sum value calculated at power off.

The entire EHT-10/EHT-10/2 RAM areas are classified into the following three groups for RAM sum check:

1 System area
2 User area
3 RAM disk

Checking the user area and RAM disk can be skipped at power on when no checking is specified by CONFIG.

2.5.2 System area sum check

(1) Areas to be checked

The following system areas are sum-checked:

1 RBDOS1 area (80H bytes)
2 RBIOS1 area (A5H bytes)
3 The address after the end of user BIOS area (RAM disk when size 0 is specified) to the address before the stack area used by the system
For EHT-10: DC00H to FCDFH
For EHT-10/2: ED00H to FCDFH
This area contains resident processing routines and system parameters.
4 Area (FF60H to FFFFH) containing system hook, system jump vector, and interrupt vector

(2) Checking sum

Simple sum is obtained for each byte in the units of 1 Kbytes. The sum up to the end of the area or remaining area is used as data for sum check if the size of an area or remaining area is less than 1 Kbytes.

(3) Check sum storage area

Check sum data is stored starting from the address determined from the following expression: Start address of RBDOS1 + C0H

(4) Handling a sum check error

If a sum check error occurs in a system area, a message indicating "SYSTEM ERROR" is displayed unconditionally and system initialization is executed.

2.5.3 User area sum check

(1) Areas to be checked

The following areas are sum-checked:
1 From address 0 to the address before RBDOS1
2 User BIOS area

(2) Checking sum

Checking sum is executed in the same way as system area check sum.

(3) Check sum storage area

Check sum data is stored starting from the address determined from the following expression:
Start address of RBDOS1 + 80H

(4) Handling a sum check error

If a sum check error occurs in an user area, message indicating "RAM SUM CHECK ERROR" is displayed unconditionally and reset processing is executed.

2.5.4 RAM disk sum check

(1) Overview

The RAM disk has a check sum area. Generally sum is determined for each sector when data is written in the RAM disk, and the sum is checked when data is read from the RAM disk.

However, since a part of main RAM is used as the RAM disk, data may be destroyed by an application program or RAM may be destroyed in power off status. Because of this, EHT-10/EHT-10/2 performs sum check of entire RAM disk at system initiation (power on or reset).

(2) Handling a sum check error

A message requesting to format the RAM disk is output if a sum check error occurs in the RAM disk.

In this case, either Yes or No can be selected. Application processing is continued if the status is power on in continue mode. "ERR #nnnn" in the message indicates the sector number (sector number is counted starting from track 0 and sector 0 in ascending order) of the error sector.

(3) Handling a sum check error by an application program

A serious error may occur if application program execution is continued after a sum check error occurs in the RAM disk. Therefore, application programs must detect and handle any RAM disk sum check errors.
The sector number of an error sector is stored in system area ERRSEC (F6B7H to F6B8H). An application program can find out error occurrence by referencing this system area. However, the contents of ERRSEC is unknown until the first error occurs. Because of this, the application program must initialize this area in advance with a sector number that does not exist (for example, FFFFH).

2.6.1 Overview

EHT-10/EHT-10/2 OS is the extended CP/M version 2.2. The EHT-10/EHT-10/2 CP/M is structured as shown in Figure 2.7 and has extended device and BIOS functions.

(1) RS-232C, IC card reader/writer, barcode reader, cartridge device (printer unit is supported for the system), and touch panel are supported as the extended devices.

Various BIOS functions are provided to support these devices and unique BIOS functions that takes into account of hand-held terminals are also provided.

(2) RAM disk, ROM socket, IC card, terminal floppy disk drives are supported as the disk drives.

(3) The most part of EHT-10/EHT-10/2 OS is executed in ROM so that a larger user RAM area can be used. Because of this, up to 59.5-Kbyte (default is 28.5-Kbyte) CP/M for EHT-10 and up to 60- Kbyte (default is 29-Kbyte) CP/M for EHT-10/2 can be structured.

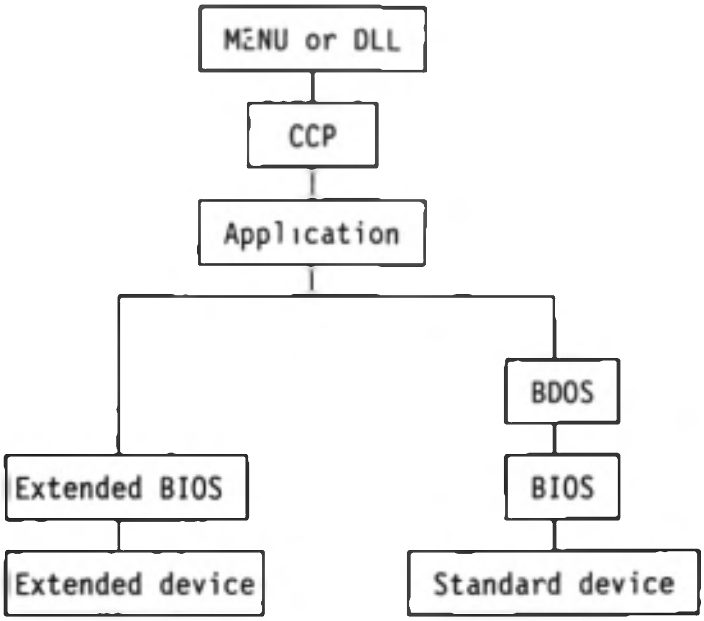


Fig. 2.7 CP/M Structure

2.6.2 CP/M structure

(1) Overview

The RAM (bank 0#0) memory map of EHT-10 slightly differs from that of EHT-10/2. These RAM memory maps are shown in Figures 1.6 and 1.7 in Chapter 1.

For EHT-10/EHT-10/2, the main sections of BDOS, BIOS, and CCP are in the system ROM and are not relocated in RAM. Only the BDOS and BIOS entries are relocated in RAM. Entry point RBDOS1 (RBDOS2) or RBIOS1 (RBIOS2) is called when the user calls BDOS or BIOS.

When BDOS or BIOS is called, OS switches the bank to pass control to the main part of BDOS or BIOS in the system ROM. See Chapters 4 and 5 for details on calling BDOS and BIOS.

(2) RBDOS and RBIOS

RBDOS1 is functionally same as RBDOS2 and RBIOS1 is functionally same as RBIOS2. RBDOS1 is the jump table to RBDOS2 and RBIOS1 is the jump table to RBIOS2.

RBDOS2 or RBIOS2 calls a processing routine in the main section of BDOS or BIOS stored in system ROM, by using the bank switching routine stored in RSYSPR.

Application programs in load and execute mode (loaded and executed in addresses starting from 100H in RAM) use RBDOS1 and RBIOS1 (general BIOS and BDOS calling).

Application programs in ROM-based mode (operates at addresses from 6000H to DFFFH in bank 2) use RBIOS2 and RBDOS2 (they are at the fixed addresses) because RBDOS1 and RBIOS1 may be in a different bank.

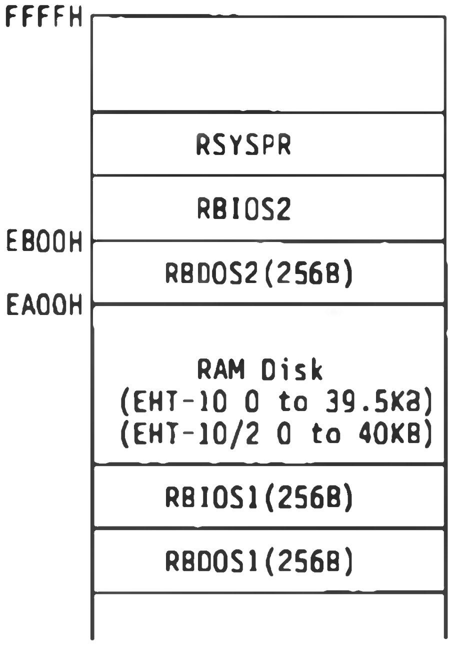


Fig. 2.8 RBDOS and RBIOS

2.6.3 CP/M Initiation and Termination

(1) CP/M initiation

Figure 2.9 shows CP/M initiation processing.
(Note) RSYSAR1: System area initialized at system initialization
RSYSAR2: System area initialized at reset
RSYSAP3: System area initialized at WBOOT

7508 reset

System Initialization

1. RSYSAR1 is loaded.
2. System environment is set.

Reset switch

Reset

1. RSYSPR is loaded.
2. RSYSAR2 is loaded.
3. RBIOS1 and 2 are loaded.

Power on

Power on

1. Device mount check

Jump 0

BOOT

1. The current drive (A:) is set.
2. The I/O byte is set.

WBOOT

1. The display parameters are initialized.
2. Key table is initialized.

Processing common to BOOT and WBOOT

1. I/O close processing
2. BDOS and WBOOT jump addresses are set
3. RBDOS1 and 2 are loaded.
4. RSYSAR3 is loaded.

(Note 1) In continue mode, the previous processing is continued after power on processing.

MENU or DLL processing

Application program initiation

Fig. 2.9
CP/M Initiation

(2) CP/M interrupt and termination

1 When the key defined with key code F6H is pressed:

<Aim>

I/O processing is immediately interrupted.

<Processing>

The keyboard buffer is cleared and I/O interrupt flag (CSTOPFLG:F220H) is turned on.

I/O processing (RS-232C send or receive operation or processing waiting for the printer to become ready) is immediately interrupted according to the I/O interrupt flag.

2 The power switch is turned off:

<Aim>

The power is turned off in restart or continue mode and processing is terminated or interrupted.

<Processing>

I/O processing is continued until processing can be left off, I/O processing is interrupted, and the power is turned off. The subsequent power on is executed in restart or continue mode.

3 When the automatic power off time comes:

<Aim>

The power is automatically turned off to save electricity if the user forgets to turn the power off.

<Condition>

Processing must be waiting for key input operation at the specified time.

<Processing>

The power is turned off in continue mode. The previous processing is continued when the power is turned on again. However, processing is not continued if the power is automatically turned off during MENU or DLL processing.

4 When power failure is detected:

<Cause>

The battery voltage is lowered to the specified voltage or less.

<Processing>

I/O processing is continued until processing can be left off, I/O processing is interrupted, a message indicating "power off" is displayed, and the power is turned off in continue mode.

5 When the reset switch is pressed:

<Aim>

The system is cold-started if wild run occurs. However, CP/M may not correctly be initiated if the CP/M size, RAM disk, etc are destroyed.

<Processing>

Reset processing is executed. The RAM disk and user BIOS area are saved but the BIOS entries are initialized.

6 When 7508 (slave CPU) is reset:

<Aim>

The system is initiated if 7508 enters wild run.

<Processing>

The entire system including 7508 is initialized.

2.6.4 CCP

(1) Overview

For EHT-10/EHT-10/2, the CCP execution program initiation section is extended to have the following characteristics:

1 ROM-based programs can be initiated.
2 Execution programs (including BASIC files) received by using DLL can be initiated.
3 CCP command input operations are deleted so that operations are simplified. Differing from the previous CP/M CCP, EHT-10/EHT-10/2 CCP is not relocated to RAM but the CCP section is operated in the OS ROM.

(2) Error handling

The two errors listed below may occur when CCP initiates a file. If an error occurs, error handling displays an error message and executes warm boot.

1 "Bad load error"
Cause: The size of the specified file to be initiated was too large and the file could not be stored in the TPA area.
User response: Press the 'Press' or 'Return' key to perform warm boot.

2 "Command error"
Cause: The specified file was not in the specified disk or there was an error in the command line.
User response: Press the 'Press' or 'Return' key to perform warm boot.

2.6.5 I/O byte

The I/O byte is used to assign physical devices to logical devices. Table 2.3 shows the I/O byte assignment. A device is assigned by replacing the contents of RAM address 3.

Logical Device	LST:	PUN:	RDR:	CON:	
Bit Location	7,6	5,4	3,2	1,0	
Bit Value	Output	Output	Input	Output	Input
0 0	–	–	Keyboard (Touch key)	RS-232C	Keyboard (Touch key)
0 1	* Cartri-dge	LCD	–	* LCD	Keyboard (Touch key)
1 0	RS-232C	*RS-232C	*RS-232C	LCD	RS-232C
1 1	-	-	-	RS-232C	RS-232C

Table 2.3 I/O Byte Assignment

(Note 1) An asterisk (*) indicates the default.
(Note 2) If a physical device is assigned to a logical device indicated by a negative sign (-) in the PUN: filed and if characters are output to the device in ready status, nothing is output.
If a physical device is assigned to a logical device indicated by a negative sign (-) in the RDR: filed, if the device is ready, and if a character input request is issued, 1AH (CTRL/Z) is sent back.

2.6.6 Setting system parameters

Table 2.4 shows the conditions under which system parameters are set.

Continue: Power on in continue mode
WBOOT: Warm boot (JMP 0000H is executed by an application program.)
Restart: Power on in restart mode
Reset: The reset switch is pressed.
Initialization: The 7508 reset switch is pressed.
System menu: Modification using system menu CONFIG

The characters in the table have the following meanings:

- S: The default value is set by the system.
- (S): The default value is set by the system conditionally.
- u: This parameter is set by the user.
- v: The value specified by the user is set by the system.
- -: No modification

Parameter	Default value	Continue	WBOOT	Restart	Reset	Initialization	System menu
Date Time	00/00/00 00:00:00	-	-	-	-	u	u
Self-test	ON	-	-	-	-	u	u
BASIC parameters	(See note 1.)	-	-	-	-	u	u
Calculator parameters	Floating-point representation	-	-	-	s	u	u
RAM disk size	31KB	-	-	-	-	u	u
User BIOS size	0KB	-	-	-	-	u	u
Character set	The default value differs depending on the DIP switch setting (see note 2)	-	v	v	v	u	u
DLL parameters	(See note 3.)	-	-	-	s	u	u
RS-232C parameters	(See note 3.)	-	-	-	-	u	u
Program execute mode	AUTO	-	-	-	-	u	u
Power control	(See note 5.)	-	-	-	-	u	u
Printer output	Cartridge	-	-	-	s	u	u
Alarm/Wake	No setting	-	-	-	-	s	-
I/O byte	xx101001 (xx differs depending on the printer output parameters.)	-	-	-	s	s	-
Key constant table		-	s	s	s	s	-
Display parameters	(See note 6.)	-	s	s	s	s	-
Keyboard status	Normal mode	-	s	s	s	s	-
Gaiji (E0H to FFH)	(See note 4.)	-	s	s	s	s	-
Subroutine key table	Only for return operation	-	s	s	s	s	-
Current drive	Drive A	-	-	-	s	s	-
User BIOS entry	Only for return operation	-	-	-	s	s(s)	

Fig. 2.4 System Parameter Set Timing

(Note 1) The BASIC parameters specify the following items:
Number of files that can be opened at the same time (3 files)
Maximum record length to be used for random file (128 bytes)
RS-232C receive buffer size (256 bytes)

(Note 2) The default value is changed according to the DIP switch.
When Japan is specified: JAPAN
When overseas is specified: USA ASCII

(Note 3) The following initial values are used for communication parameters: 4800 BPS, 8 bits, nonparity, and 2 stop bits

(Note 4) The gaiji default value differs depending on Japan or overseas specification as follows:
Japan: Yen, year, month, and day (The remaining area is filled with blanks.)
Overseas: (The entire area is filled with blanks.)

(Note 5) The following parameters can be specified for power control:
Automatic power off time (5 minutes)
Printer power save time (3 minutes)
Automatic backlight off time (3 minutes) (only for EHT-10/2B)
A value enclosed in a pair of parentheses ( ) is the default value.

(Note 6) The following parameters and default values are used as the display parameters:
Screen size: 12 columns x 28 lines (20 columns x 25 lines for EHT-10/2)
Scroll mode: Follow mode
Cursor type: Block and blink
Attribute: None (normal status)

CHAPTER 3 APPLICATION PROGRAM CREATION

3.1 Overview

This chapter explains how to create EHT-10/EHT-10/2 application programs. For EHT-10/EHT-10/2, the BASIC interpreter is generally stored in ROM so that application programs can be written in BASIC. Further, CP/M Ver. 2.2 is used as the OS so that programs can easily be created in machine language or in a high-level language.

EHT-10/EHT-10/2 application programs are created in the development machine and loaded to EHT-10/EHT-10/2 for execution and debug operations.

Application programs are debugged by using the development tool that is an option. To use this development tool, a computer that supports CP/M or MS-DOS (PC-DOS) as the OS is required as the development machine (refer to EHT-10/EHT-10/2 Development Tool User's Guide for details).

Figure 3.1 shows the overview of application system creation procedure.

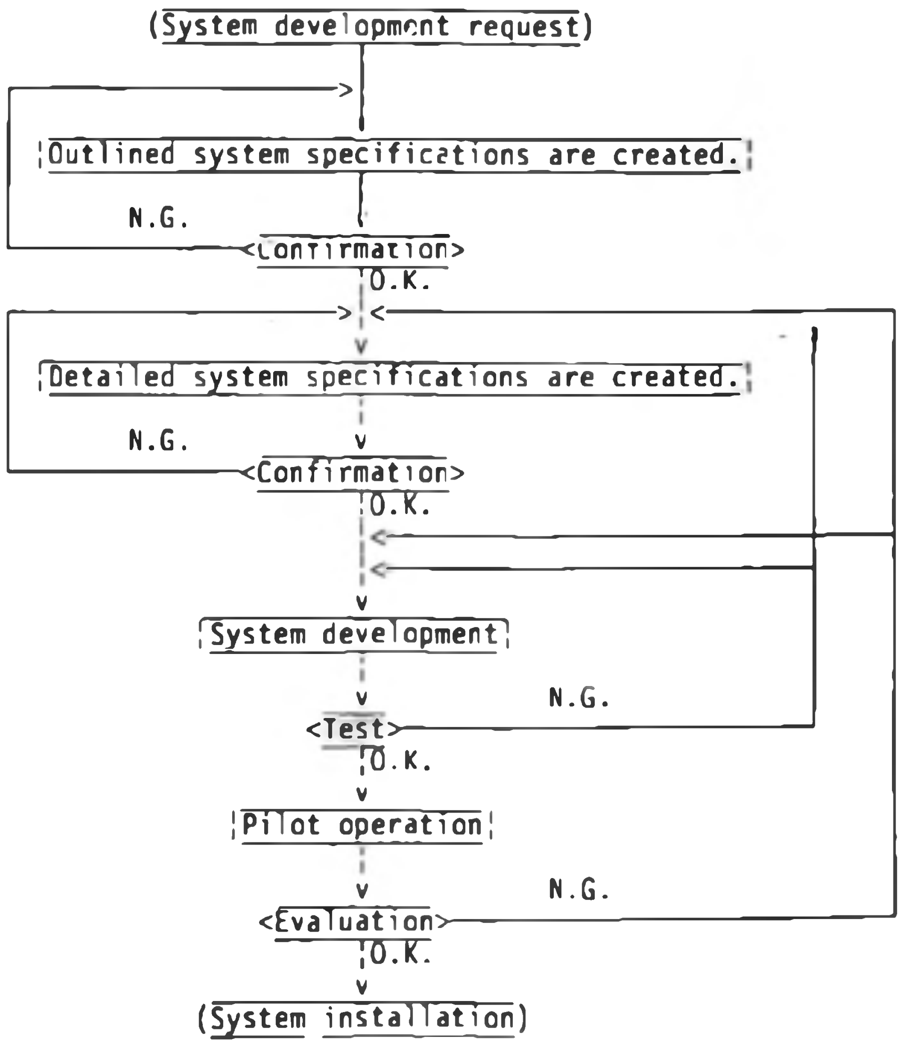


Fig. 3.1 System Development Procedure

3.2.1 Programming language selection

For EHT-10/EHT-10/2, application programs can be written in machine language, BASIC, and high-level compiler-type language that can be executed under CP/M. A programming language is selected according to the processing contents of application programs.

Application programs in machine language can be stored in ROM in the format that enables the application programs to be executed in ROM. In this case, RAM space is saved, less power is consumed, and execution is done in high speed (see "Section 3.4 Creating ROM-Based Application Programs" for details).

3.2.2 Program distribution and maintenance

The method of program distribution and maintenance must be determined when application programs are designed. There are the following three ways to distribute and maintain programs:

(1) Programs are distributed and maintained by down-loaded to RAM (TPA or

RAM disk) by DLL or DL using communication lines.
In this case, a communication program that down-loads application programs must be created at the host computer side. EHT-10/EHT-10/2 supports the Filink protocol and No protocol as the communication protocols. A protocol can be extended so that the user can create a user protocol. See "Section 11.2 Extending a Communication Protocol" for details.
See Appendix for the Filink protocol.

(2) Application programs are stored in ROM for distribution.

By storing fixed data (such as kanji file) and processing in ROM, RAM space is saved and maintenance becomes easier.

(3) Application programs are distributed from IC card.

The master file and programs are stored in an IC card. Programs in IC card can easily be executed because EHT-10/EHT-10/2 can use an IC card as a disk.

Further, the files and programs in an IC card can be directly maintained from the host computer through the communication line.

3.3.1 Computer used to create application programs

Application programs are created by a development machine (computer) and are down-loaded to EHT-10/EHT-10/2 for debug operations.

The EHT-10/EHT-10/2 debugger (development tool) requires a CP/M or MS-DOS (PC-DOS) machine as a terminal. Refer to EHT-10/EHT-10/2 Development Tool User's Guide for the names of the computers that can be used.

3.3.2 Development software and development procedure

Table 3.1 shows the types of development software required to create application programs.

Language	Type of software		Example
BASIC	Development tool (WAD)		Programs can also be edited by the general text editor.
High-level language	Editor and compiler		BDS C Compiler (BD Software Co.) and other compilers that operate under CP/M Ver. 2.2.
Machine language	When the host OS is CP/M	Editor	Word Master (Micropro Co.) or ED
		Assembler	MACRO80 (Microsoft Co.)
		Linker	L80 (Microsoft Co.)
		Development tool (WAD)	
	When the host OS is MS-DOS	Editor	Word Star (Micropro Co.) or EDLIN
		Assembler	XMACRO80 (Nikkei Co.)
		Linker	XLINK80 (Nikkei Co.)
		Development tool (WAD)	

Table 3.1 Application Development Software

(1) BASIC

Since the BASIC interpreter is internally built in the OS, a program in BASIC can be directly created and debugged in EHT-10/EHT-10/2 using the development tool. In this case, screen edit operations can be performed by using a development machine as the terminal.

(2) Machine language

A program in machine language is created, assembled, and linked in the development machine to be generated as an object module (COM file) in execute form. As other CP/M programs, a program in machine language is generally assembled starting from address 100H. However, note that the

execution start address of a ROM- based program is not address 100H (see Section 3.4).
The development tool is used to down-load and debug a program in machine language.

(3) High-level language

The high-level languages that satisfy the following conditions can be executed in EHT-10/EHT-10/2:

1 Compiler-type languages that can be executed under standard CP/M Ver. 2.2.
2 The compilers can generate an object module (COM file) in execute form.

As a program in machine language, the object module of program in high-level language is down-loaded and executed in EHT-10/EHT-10/2.

3.3.3 Storing program data in ROM

To store completed programs and data in ROM, the programs and data are converted in the ROM format by development tool WPROMFRM. If the development machine and the ROM writer are connected through RS-232C, WPROMFRM can directly transfer data to the ROM writer.

3.4.1 Overview

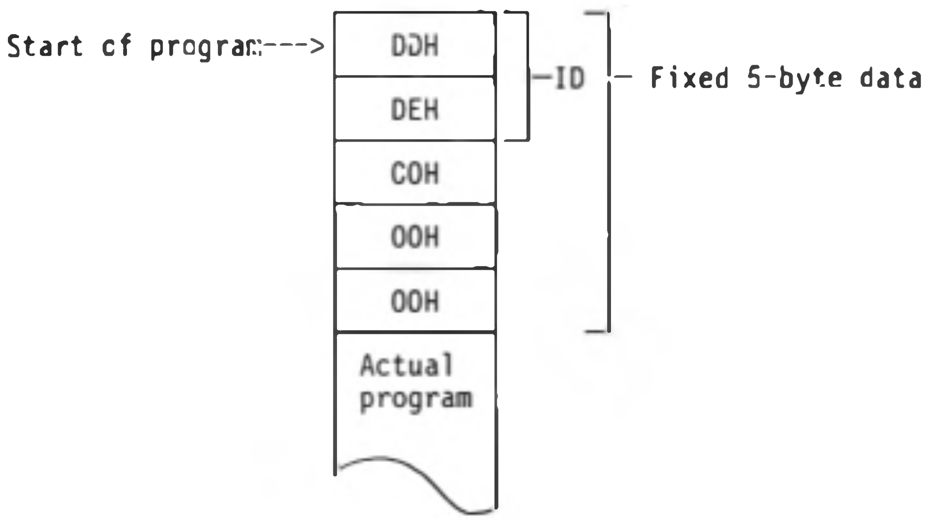
An application program in ROM socket can be loaded to address 100H in ROM for execution in the same way as the previous CP/M. However, EHT-10/EHT-10/2 can also execute the application program directly in ROM to save memory space and to have less power consumption.

The rest of Section 3.4 explains how a ROM-based program is executed, how to set a ROM-based program, and notes on using a ROM-based program.

3.4.2 Setting

To identify a program that is directly executed in ROM (ROM-based program) from a program that is loaded and executed in ROM (program in load and execute mode), a 5-byte ID must be placed at the beginning of a ROM-based program.

The system determines a ROM-based program from this ID and performs processing for ROM-based program.



3.4.3 Processing flow

Figure 3.2 shows the processing flow until control is passed to a ROM-based program.

```
(File in execute form is initiated.)
                 |
                 v
|The first one sector of the file is|
|read into address 100H.            |
                 |
                 v
 / The ROM-based program ID  \
 \ (DDH and DEH) is checked. /------------------+
                 |                               |
                 |                     |Load and execution|
                 |                     |                  |
                 |                     |  General CP/M    |
                 |                     |  processing      |
                 v
|The program execution start |
|address is determined.      |-------> Program in ROM
                 |                 (Beginning of file) + 0005H
                 v
  |Control is passed to the|
  |application program.    |
```

Fig. 3.2 Initiating a ROM-based program

3.4.4 Program start address

Since a program in load and execution mode is loaded starting from address 100H in RAM, addressing is started from 100H as a general CP/M file.

On the other hand, a ROM-based program is directly executed in ROM so that addressing must be started from the program start address determined by taking into account of ROM format and ROM in the memory map.

See "Section 6.4 ROM Drive" for ROM format.

The start address of a ROM-based program can be determined from the following expressions:

- Number of programs stored in ROM: n
- Size of each program: Fk (k=1, 2,...n) Rounded up to the unit of 1 Kbytes.
- Number of directories for each program: Fk/16384 (16384 = 16 Kbytes)
- Total number of directories: Dnum = F1 + F2 +...Fn
- Size of directory area: Dsize = (1 + Dnum) x 32 (Rounded up to the unit of 128 bytes.)

Header section (32 bytes).

The execution start address of i-th program can be determined as follows:
{Dsize + (F1 + F2 + ...Fi-1)}/32768 (32768 = 32 Kbytes)

Address = (remainder) + 6000H
Bank data = (quotient x 10H) + 02H (Quotient is the subbank data.)

Figure 3.3 shows the address map used when four files are stored in a 32-Kbyte ROM.

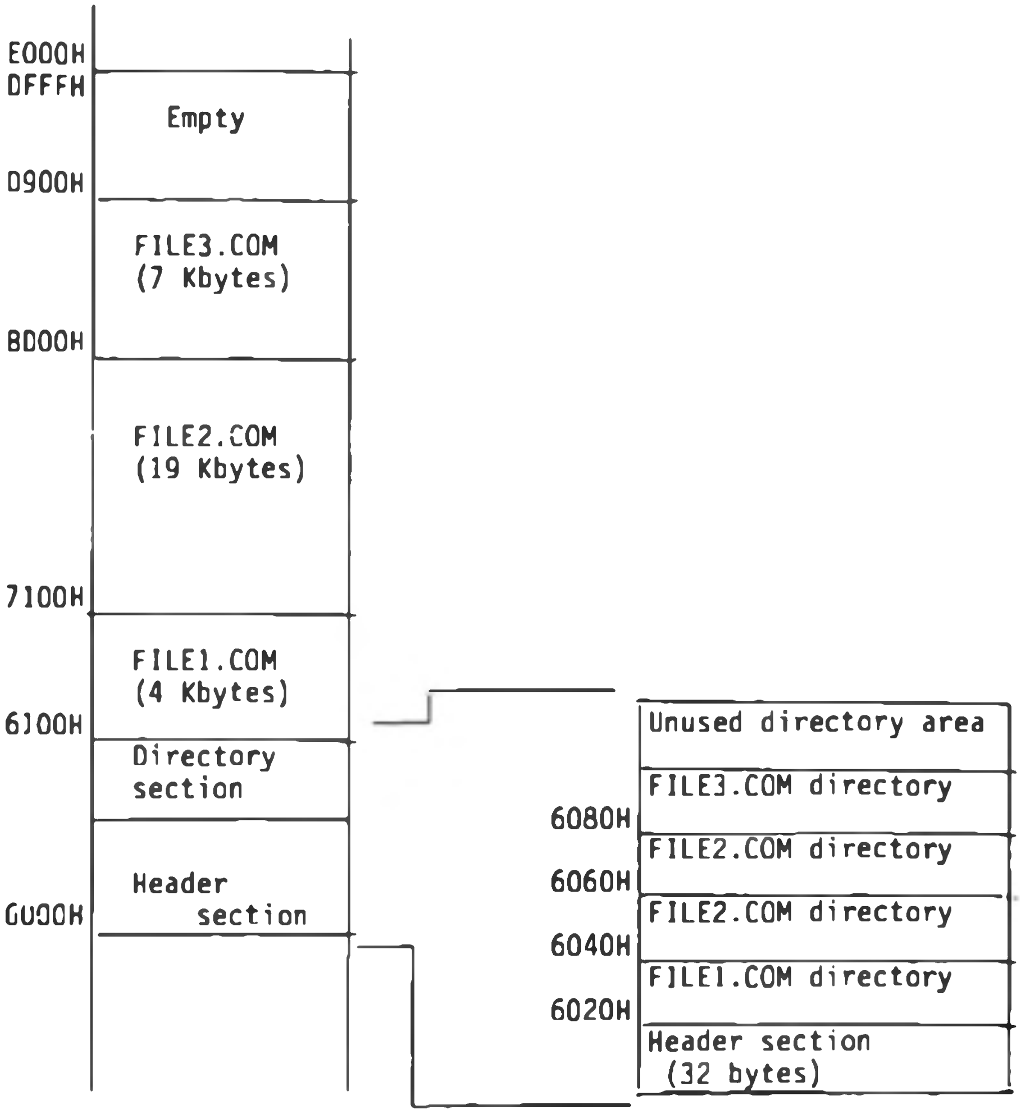


Fig. 3.3 Example of ROM structure (in EHT-10/EHT-10/2)

3.4.5 Calling BDOS/BIOS

A program in load and execute mode generally uses 0005H to 0007H to call BDOS and 0000H to 0002H to call BIOS. However, BDOS and BIOS may be placed at the back of the ROM capsule for a ROM -based program. Because of this, a ROM-based program must call BDOS and BIOS (RBDOS2 and RBIOS2) in the resident section.

```
RBDOS2 entry address..................FF90H
                                      (JP RBDOS2)

RBIOS2 BOOT address...................EB00H
       WBOOT address..................EB03H
       CONST address..................EB06H

       INFORM address.................EBA5H
```

3.4.6 ROM storing ROM-based programs

Formats P and M are provided as the ROM formats of ROM drive for EHT-10/EHT-10/2. ROM to be used to store ROM-based programs must be created in format P.

ROM to be used to store programs in load and execute mode can be either in format P or M.

3.4.7 Work area

A ROM-based program uses a work area different from the one used by a program in load and execute mode.

A program in load and execute mode can freely use areas inside the program and after the program as the work areas. A ROM-based program uses a work area starting from address 100H.

The upper limit of work area of ROM-based program is obtained as follows:
MIN {(ROM start address), (RBDOS1 start address)}

The ROM start address is 6000H and the RBDOS1 start address is determined from the contents of addresses 6 and 7.

3.4.8 32-Kbyte or more ROM-based program

In EHT-10/EHT-10/2, ROM is allocated on the memory map for each subbank in the units of 32 Kbytes (see "Section 1.4 Memory Map"). One ROM-based program must not be extended over two or more subbanks.
If the program size is more than 32 Kbytes, the following processing is required:

(1) Each module in the program must not be extended over bank end address (DFFFH in memory map).

(2) Dummy data must be inserted so that the end address of the last module in a bank is the subbank end address.

(3) The module next to the one explained in (2) must be addressed starting from address 6000H.

(4) JSCALLX (FF99H) must be used to perform subroutine call operations between banks (parameters same as that of BIOS CALLX are used).

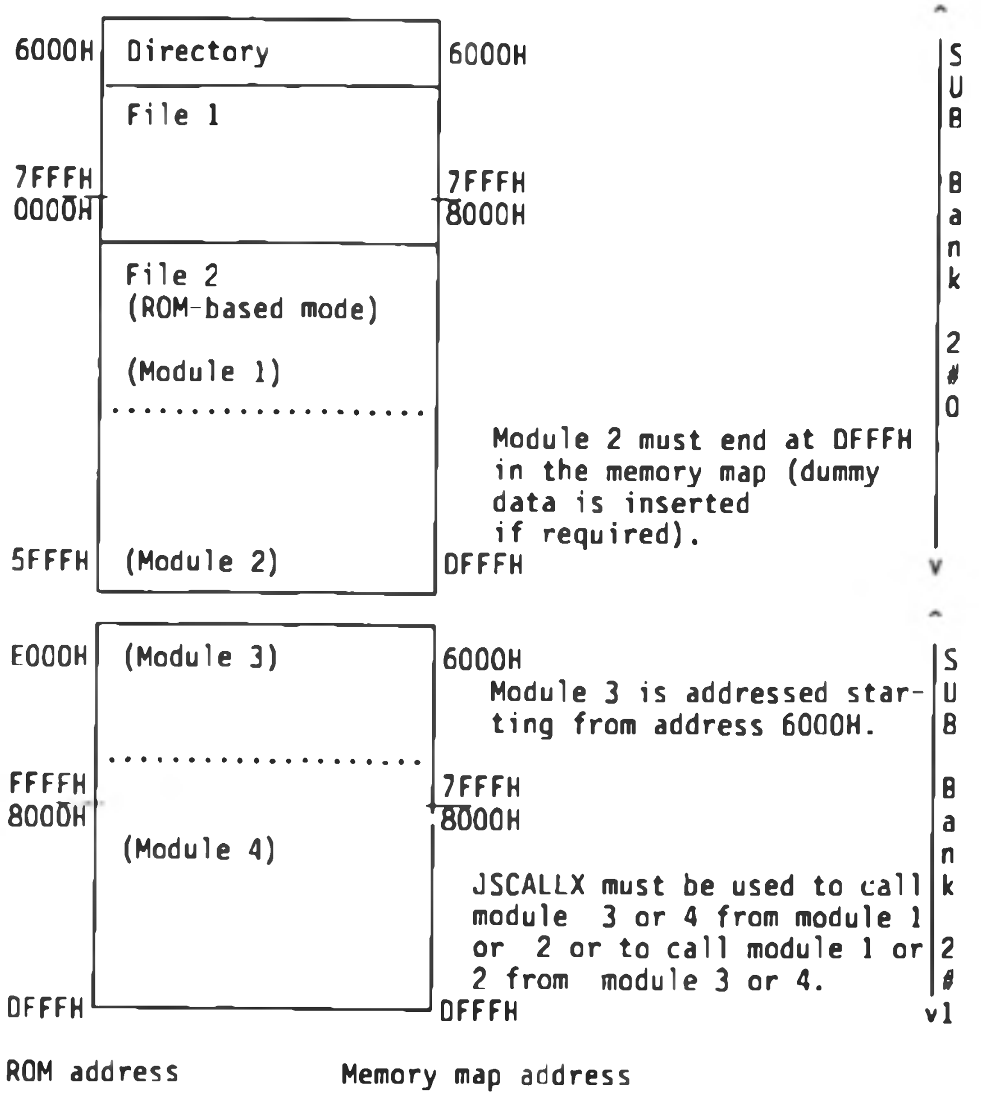


Fig. 3.4 Structure of ROM-based file stored in 2banks

CHAPTER 4 BIOS OVERVIEW

4.1 Overview

4.1.1 Characteristics of EHT-10/EHT-10/2 BIOS

As BDOS, the EHT-10/EHT-10/2 BIOS processing is performed mainly in system bank OSROM. Two BIOS RAM entries are provided so that an ROM-based program can use BIOS without taking into account of banks.

For EHT-10/EHT-10/2, the standard BIOS is extended so that communication with serial devices, touch panels, IC cards, and public lines can easily be done. Further, BIOS can be extended by the user since the user BIOS and BIOS hook are provided. See Section 4.6 for the user BIOS and Chapter 10 for BIOS hook.

4.1.2 BIOS processing

(1) Processing flow

Processing flows as follows (Figure 4.1) when an application program calls EHT-10/EHT-10/2 BIOS:

1 The bank is switched to the system bank at the BIOS section in RAM.
2 The actual BIOS in OS ROM is called.
3 After BIOS processing, various return information and data are saved and control is returned to the bank used before BIOS is called.

An application program calls BIOS as follows:

1 A load-and-execute program obtains the BIOS address from the JMP WBOOT placed at address 0000H to call BIOS.
2 A ROM-based program directly calls the BIOS jump table placed in RBIOS2 (EB00H to EBFFH).

The functions of BIOS are the same for programs in load and execute mode and for programs in ROM-based mode.

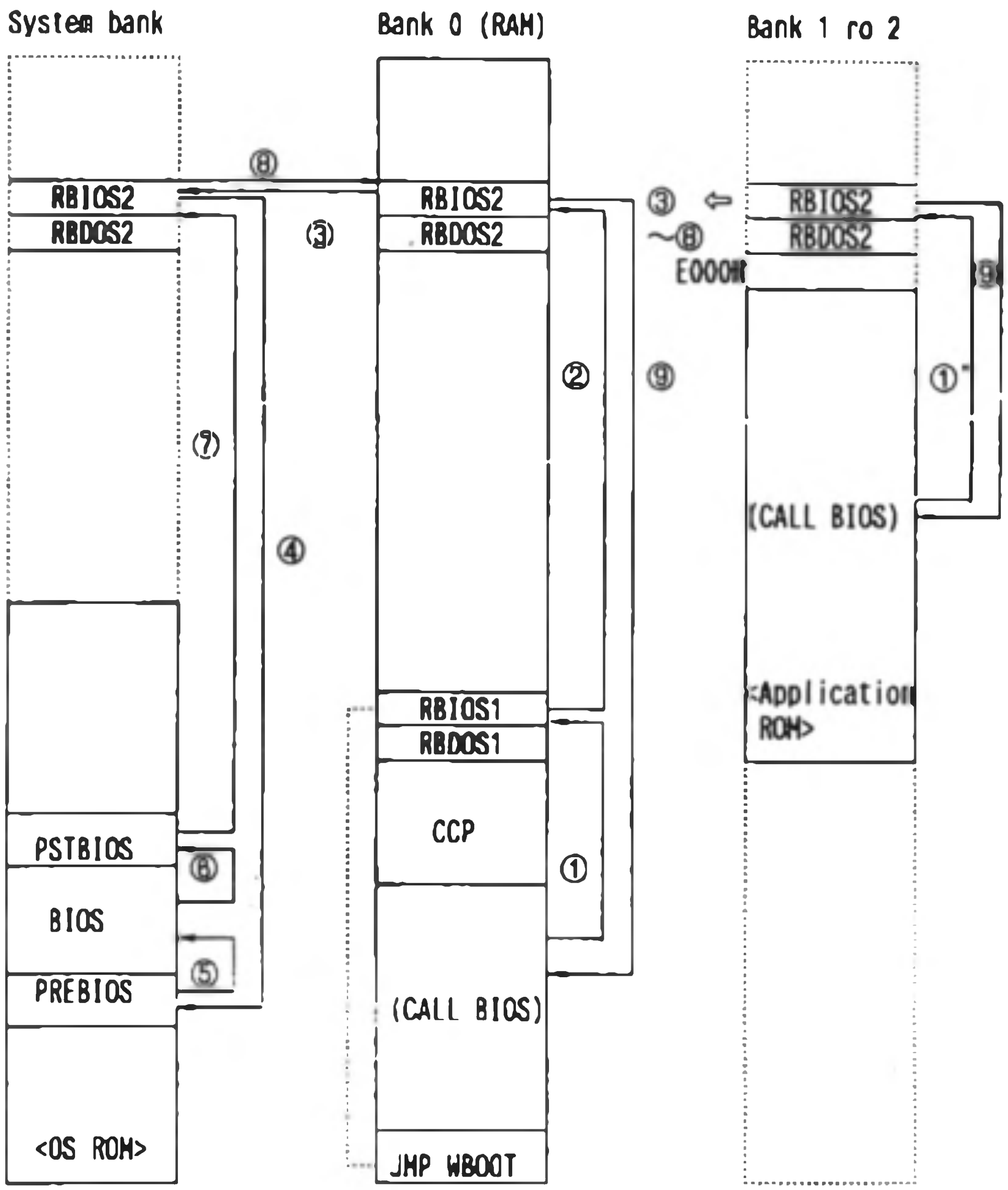


Fig. 4.1 BIOS Processing Flow

(2) PREBIOS and PSTBIOS

EHT-10/EHT-10/2 BIOS processing is based on the PRE/PST BIOS concept to have higher system reliability.

If interrupt processing is executed immediately after an interrupt occurs during BIOS processing, the program execution may not be continued after control returns from interrupt processing. Therefore, interrupt processing is suppressed before actual BIOS processing is started and interrupt processing for interrupts that occurred during BIOS processing is executed after BIOS processing is completed. PRE/PST BIOS controls these operations.

PRE/PST BIOS is automatically executed when BIOS is called from a BIOS entry placed in RAM.

1 PREBIOS
PREBIOS sets the flag indicating BIOS processing in progress, the flag indicating alarm suppressed, and the flag indicating power off suppressed.

2 PSTBIOS
PSTBIOS resets the flags set by PREBIOS and performs alarm or power off processing if alarm or power off has been requested during BIOS processing.

4.1.3 Notes on using BIOS

(1) The contents of registers other than the one used to store return parameters are not guaranteed unless other wise stated. The contents of the required registers must be saved in advance.

(2) The addresses of two BIOS entries are used as follows:

1 To indicate a BIOS entry address with an offset value based from WBOOT, the contents (WBOOT entry address) of addresses 001H and 0002H are fetched to determine the BIOS entry address.

2 There is no specific problems when a BIOS entry address is indicated with a fixed address. However, study the maps in RAM carefully to convert previous CP/M application programs.

4.2 Overview of BIOS Commands

This section explains each BIOS command. Each command is explained according to the conventions explained below. See Appendix 5 for the list of BIOS commands.

(1) Command name
The command name is written with upper-case letters at the top left of a page. The commands are explained in the ascending order of addresses. If a command has subcommands, the subcommand names are written enclosed in a pair of parentheses ( ) after the command name.

(2) Entry addresses
The offset address from WBOOT and the absolute address are written at the top right of a page.

(3) [Function]
[Function] explains the overview of a BIOS command.

(4) [Entry parameters]
[Entry parameters] explains the input parameters required to call a BIOS routine.

(5) [Return parameters]
[Return parameters] explains the parameters set when BIOS routine execution is ended.

(6) [Saved registers]
[Saved registers] lists the registers of which the contents are saved even if BIOS is used. [Saved registers] is not written unless there is one or more saved registers.

(7) <Explanation>
<Explanation> explains the functions and usages of each BIOS command.

(8) <Relations>
<Relations> explains other related BIOS command.

(9) <Reference>
<Reference> explains sections to be referenced.

[Function]
BOOT performs CP/M cold boot.

[Entry parameters]
None

[Return parameters]
C=00H

<Explanation>

(1) The following operations are performed when BOOT is executed:
1 The current drive is changed to A:.
2 The I/O bytes are initialized.
3 The keyboard standard (for Japan or overseas) and the display character set are initialized according to the DIP switches.
4 RBIOS1 and 2 are loaded.

The rest of operations are the same as warm boot (see WBOOT).

(2) BOOT is used by the system for system initialize, reset, or 7508 reset operation but not by the user.

<Relations>

WBOOT (WBOOT+0)

<Reference>

Section 2.6 CP/M Operations

[FUNCTION]
WBOOT performs CP/M warm boot.

[Entry parameters]
None

[Return parameters]
C=drive number

<Explanation>

(1) The following operations are performed when WBOOT is executed:
1 The display parameters are initialized.
2 The key tables are initialized.

The rest of operations are the same as BOOT.

3 I/O close processing is executed.
4 The jump address of BDOS WBOOT is set.
5 RBDOS1, and 2 are loaded.
6 RSYSAR3 is loaded.
7 MENU or DLL is initiated.

(2) WBOOT is initiated to branch out of an application when address JP=0000H is executed or the power is turned on in restart mode.

<Relations>

BOOT (WBOOT-03H)

<Reference>

Section 2.6 CP/M Operations
APPENDIX 9 SAMPLE-LIST

[Function]
CONST checks whether data is input from the device allocated as the current CON: (default is the keyboard or touch panel).

[Entry parameters]
None

[Return parameters]
A=00H: Console input
A=FFH: No console input

<Explanation>

(1) The current CON: is determined according to the I/O bytes.

(2) "No console input" is determined for undefined keys or keys without codes if the keyboard or touch panel is allocated as the current CON:. For EHT-10, BIOS TOUCH or PUTPFK must be executed at first to define keys in order to use CONST and CONIN because all keys are defined ineffective (function code=FFH) after an application program is initiated.

(3) If the keyboard or the touch panel has been allocated as the current CON: and a function call key is pressed, the key operation corresponding to the key is executed, key input is checked, and then the status is set in register A. The function call keys have the key function codes from E0H to FFH used to call subroutines. See "Section 4.3 Key Input" for details.

(4) The status of receive buffer is returned if the current CON: input is to RS-232C.

(5) If the current CON: input is RS-232C, and if the user has opened the cartridge serial by using BIOS RSIOX or has opened an IC card by using BIOS ICCARD, A=00H is set as the return parameter of CONST.

(6) If CON: input is RS-232C and the user is using the RS-232C, the transmission-mode parameter specified by the user at open operation is used.

(7) See Section explaining PUNCH to modify the RS-232C transmission mode.

<Relations>

CONIN (WBOOT+06H)
PUNCH (WBOOT+0FH)
PUTPFK(WBOOT+6CH)
TOUCH (WBOOT+93H)
KEYIN (WBOOT+99H)

<Reference>

Section 2.6.5 I/O bytes
Section 4.3 Key Input (Touch Panel/Keyboard)

[FUNCTION]
CONIN inputs one character from the device allocated as the current CON: (default is the keyboard or touch panel) and sets the character in register A. Processing waits until data is input if there is no input data.

[Entry parameters]
None

[Return parameters]
A=input data (function code)
C=position code

<Explanation>

(1) The current CON: is determined according to the I/O bytes.

(2) For EHT-10, BIOS TOUCH or PUTPFK must be executed at first to define keys in order to use CONST and CONIN because all keys are defined ineffective (function code=FFH) after an application program is initiated.

(3) If CON: input is the keyboard or the touch panel, the sleep mode and automatic power off functions are operated. See "Section 2.3.5 Sleep function and Automatic Power Off function" for details.

(4) If a subroutine call key that has a function code from E0H to FFH is pressed, the corresponding subroutine is executed and then processing waits for key input operation.

(5) The return parameter in register C is meaningless if CON: input is RS-232C.

(6) If CON: input is RS-232C, and if the user has opened the cartridge serial by using BIOS RSIOX or the IC card by using BIOS ICCARD, control returns without executing any operation. In this case, the return parameters are meaningless.

(7) If CON: input is RS-232C and the user is using the RS-232C, the transmission-mode parameter specified by the user at open operation is used.

(8) See Section explaining PUNCH to modify the RS-232C transmission mode.

<Relations>

CONST (WBOOT+03H)
PUNCH (WBOOT+0FH)
PUTPFK(WBOOT+6CH)
TOUCH (WBOOT+93H)
KEYIN (WBOOT+99H)

<Reference>

Section 2.3.5 Sleep function and Automatic Power Off function
Section 2.6.5 I/O bytes
Section 4.3 Key Input (Keyboard/Touch Panel)
APPENDIX 9 SAMPLE 2

[Function]
CONOUT outputs one character to the device allocated as the current CON: (default is LCD).

[Entry parameters]
C=output data

[Return parameters]
None

<Explanation>

(1) The current CON: is determined according to the I/O bytes.

(2) Display operations can be controlled with control codes (00H to 1FH) and the ESC sequences if CON: output is LCD. See "APPENDIX 6 DISPLAY CONTROL FUNCTIONS" for details on control code and ESC sequence functions.

1 The control codes, ESC sequences, and character codes are classified as follows:
Control codes: 00H to 1FH
ESC sequences: 1BH (ESC) + n1 + n2 +...nk
Character codes: 20H to FFH

2 In an ESC sequence, CONOUT is called as may times as the number of parameters.

3 Control codes and ESC sequences can be used to control not only LCD but also keyboard (touch panel) LED and buzzer.

4 If an ineffective control code or an ESC sequence parameter error is detected, no operation is executed and the original status is guaranteed.
ESC sequence parameters are checked after all parameters are received.
See "Section 4.4 LCD Display" for the LCD display functions.

(5) Processing same as PUNCH is executed if the CON: device is RS-232C.

<Relations>

PUNCH (WBOOT+0FH)
GRAPHICS(WBOOT+90H)
TOUCH (WBOOT+93H)
KANJI (WBOOT+99H)

<Reference>

Section 2.6.5 I/O bytes
Section 4.4 LCD Display
APPENDIX 6 DISPLAY CONTROL FUNCTIONS

[Function]
LIST outputs one character to the device allocated as the current LST: (default is the cartridge I/F).

[Entry parameters]
C=output data

[Return parameters]
Only when a printer unit is mounted,
(LSTERR)=00H: Normal termination
≠00H: Abnormal termination (indicates a printer unit not mounted or forced system termination.)

<Explanation>

(1) The current LST: is determined according to the I/O bytes.

(2) Processing waits until the LST: device becomes ready if it is not ready. However, if LST: is a cartridge I/F and a printer unit is not mounted, error information is set in LSTERR and control returns.

(3) See Section explaining BIOS INFORM for the LESTER address.

(4) Print and other operations can be executed concurrently by using the spooling function if a printer unit has been allocated as the LST: device. See "Section 4.5 Printer" for details.

(5) If LST: device is RS-232C, and if the user has opened the cartridge serial by using BIOS RSIOX or the IC card by using BIOS ICCARD, control returns without executing any operation.

(6) If the LST: device is RS-232C and the user is using the RS-232C, the transmission-mode parameter specified by the user at open operation is used.

(7) See Section explaining PUNCH to modify the RS-232C transmission mode.

<Relations>

PUNCH (WBOOT+0FH)
LISTST(WBOOT+2AH)
SCRNDUMP(WBOOT+33H)
RSIOX (WBOOT+51H)
ICCARD(WBOOT+96H)
KANJI (WBOOT+9CH)
INFORM(WBOOT+A2H)

<Reference>

Section 2.6.5 I/O bytes
Section 4.5 Printer
APPENDIX 9 SAMPLE 3

[Function]
PUNCH outputs one character to the device allocated as the current PUN: (default is RS-232C).

[Entry parameters]
C=output data

[Return parameters]
None

<Explanation>

(1) The current PUN: is determined according to the I/O bytes.

(2) If PUN: device is RS-232C, and if the user has opened the cartridge serial by using BIOS RSIOX or the IC card by using BIOS ICCARD, control returns without executing any operation.

(3) If the PUN: device is RS-232C and the user is using the RS-232C, the transmission-mode parameter specified by the user at open operation is used.

(4) Transmission mode default value and how to modify the mode when the physical I/O device is RS-232C, the same transmission mode is used for BIOS CONIN (CONST), CONOUT, and LIST (LISTST, SCRNDUMP, KANJI printout) because a transmission mode is defined in the same system area. The transmission mode for these BIOS commands is modified by CONFIG. The transmission mode can also be modified by a user program rewriting the following system area:

(Example)
Address: Variable name(Byte length)

F010H : SRSADR (2+2)
First 2 bytes specifies the Start address of receive buffer (initial value: COMBUF F95AH)
Second 2 bytes specifies the Size of receive buffer (initial value: 0100H)

F014H : SRSPAK (5)
Each byte specifies the following parameter.
Transmission speed (initial value 0DH .....4800 BPS)
Bit length (initial value: 03H .....8 bits)
Parity (initial value: 00H .....nonparity)
Stop bit (initial value: 03H .....2 stop bits)
Special parameter (initial value: FFH)

1 The parameter is structured in the same way as RSIOX open operation.

2 RS-232C must be closed by BIOS RSIOX after modification when this parameter is modified by an application program.

<Relations>

```
CONST  (WBOOT+03H)
CONIN  (WBOOT+06H)
CONOUT(WBOOT+09H)
LIST   (WBOOT+0CH)
LISTST(WBOOT+2AH)
SCRNDUMP(WBOOT+33H)
RSIOX  (WBOOT+51H)
ICCARD(WBOOT+96H)
KANJI  (WBOOT+9CH)
```

<Reference>

Section 2.6.5 I/O bytes
Section 7.2 Serial Interface
APPENDIX 9 SAMPLE3

[Function]
READER inputs one character from the device allocated as the current RDR: (default is RS-232C).

[Entry parameters]
None

[Return parameters]
A=input data

<Explanation>

(1) The current RDR: is determined according to the I/O bytes.

(2) Processing waits until data is input if there is no input data.

(3) Processing is interrupted to perform power off or alarm operation if power off or alarm status occurs while processing is waiting for input data.

(4) The return parameter output when the RDR: device is RS-232C is same as the one output by RSIOX Get.

(5) The transmission mode used when RS-232C is specified is same as the transmission mode used for PUNCH. The default values of transmission mode are 4800 BPS, 8 bits, and nonparity.
The communication parameters are modified by system menu CONFIG.

(6) 1AH (EOF) is set in register A if an I/O device (DTR or UR2) not supported for RDR: is allocated as the RDR:.

<Relations>

PUNCH (WBOOT+0FH)

<Reference>

Section 2.6.5 I/O bytes
Section 7.2 Serial Interface
APPENDIX 9 SAMPLE 3

HOME

WBOOT+15H
EB18H

[Function]
HOME sets the disk seek track to 0.

[Entry parameters]
None

[Return parameters]
None

<Explanation>

(1) For a floppy disk, the contents of blocking buffer are written and then the track is set to 0.

(2) HOME does not actually move the head to track 0.

<Reference>

CHAPTER 6 OVERVIEW OF DISK SYSTEM

[Function]
SELDSK specifies a drive.

[Entry parameters]
C=00H: Drive A (RAM disk)
=01H: Drive B (application ROM)
=02H: Drive C (IC card)
=03H: Drive D (external floppy disk drive)
=04H: Drive E (external floppy disk drive)
E=access information (only for floppy disk)
Bit 0=0: Initial disk selection
=1: Second or later disk selection

[Return parameters]
HL=0: Parameter error
≠0: Disk parameter address

<Explanation>

(1) A parameter error occurs if a disk is not connected.

(2) The correspondence between logical and physical drives can be freely changed for EHT-10/EHT-1C/2. See "CHAPTER 6 OVERVIEW OF DISK SYSTEM" for details.

<Reference>

CHAPTER 6 OVERVIEW OF DISK SYSTEM
APPENDIX 9 SAMPLE 4

[Function]
SETTRK specifies a track for read/write operations.

[Entry parameters]
BC=track number

[Return parameters]
None

<Explanation>

The parameter is not checked. However, an error occurs during read/write operation if a value out of the allowed range is specified. The following tack numbers are allowed for drives:

Physical drive	Logical drive	Range	Remarks
RAM disk	A:	$0 \leq BC \leq 28$	Maximum may vary.
ROM Socket	B:	$0 \leq BC \leq 15$	Maximum may vary.
IC card	C:	$0 \leq BC \leq 7$	Maximum may vary.
External disk drive	D: E:	$0 \leq BC \leq 39$	Maximum may vary.

Table 4.1 Track numbers for drives

<Reference>

CHAPTER 6 OVERVIEW OF DISK SYSTEM
APPENDIX 9 SAMPLE4

[Function]
SETSEC specifies a sector for read/write operations.

[Entry parameters]
C=sector number

[Return parameters]
None

<Explanation>

(1) The allowed range of sector numbers is $0 \leq C \leq 63$.

(2) An error occurs during read/write operation if a sector number out of the allowed range is specified.

<Reference>

CHAPTER 6 OVERVIEW OF DISK SYSTEM
APPENDIX 9 SAMPLE 4

[Function]
SETDMA specifies the start address of 128-byte data area to be read or written.

[Entry parameters]
BC=DMA start address

[Return parameters]
None

<Explanation>

(1) A DMA buffer address is specified. The DMA buffer must be reserved in main RAM.

(2) The DMA buffer receives or sends data during read/write operation.

<Reference>

CHAPTER 6 OVERVIEW OF DISK SYSTEM
APPENDIX 9 SAMPLE 4

[Function]
READ reads 128-byte data.

[Entry parameters]
None

[Return parameters]
A=return information
=00H: Normal termination
≠00H: Abnormal termination

<Explanation>

(1) READ reads 128-byte data from a disk according to the parameters set by SELDSK, SETTRK, SETSEC, and SETDMA.

(2) The return parameter output at FDD access abnormal termination has the following meaning:

A=FAH : read error
FBH : write error
FCH : select error
FDH : read only disk or write protect

(3) The error information of disk read/write operation executed by the called BIOS can also be referenced from the following area:

Address : Variable name(Byte length)

F471H : BIOSERROR (1)
BIOS return code
=00H:Normal termination
=01H:Read Error
=02H:Write Error
=03H:Write Protect Error
=04H:Time out or Communication Error
=FEH:Others

See Section explaining BIOS INFORM for the BIOSERR address.

<Relations>

INFORM (WBOOT+A2H)

<Reference>

CHAPTER 6 OVERVIEW OF DISK SYSTEM
APPENDIX 9 SAMPLE 4

[Function]
WRITE writes 128-byte data.

[Entry parameters]
C=writing specification (only for floppy disk)
=00H: Standard writing (blocking)
=01H: Forced writing (no blocking)
=02H: Sequential file writing

[Return parameters]
A=return information
=00H: Normal termination
≠00H: Abnormal termination

<Explanation>

(1) As READ, WRITE writes 128-byte data in a disk according to the set parameters.

(2) The return parameters indicating abnormal termination and error information returned in BIOSERROR are same as the ones of BIOS READ.

<Relations>

READ(WBOOT+24H)

<Reference>

CHAPTER 6 OVERVIEW OF DISK SYSTEM
APPENDIX 9 SAMPLE 4

[Function]
LISTST checks the use status of the device allocated as the current LST: (default is printer unit).

[Entry parameters]
None

[Return parameters]
A=printer status
=FFH: Ready (Data can be output to the LST: device.)
=00H: Busy (Data cannot be output to the LST: device.)
B=buffer status (only for printer unit. Details are explained later.)
=FFH: No data (empty)
=00H: There is data.
(LSTERR)=error status
=00H: Normal
≠00H: Printer unit not mounted

<Explanation>

(1) The current LST: is determined according to the I/O bytes.

(2) When the current LST: device is a printer unit, 128-byte printer buffer is provided for LST: device output operation so that multiprocessing with printer output operation can be executed. The buffer status indicates whether data to be output is in the printer buffer.

(3) When the current LST: is a printer unit and the output buffer is not full, printer ready is set in register A and control returns.

(4) See the explanation of BIOS INFORM for the LSTERR address.

<Relations>

INFORM(WBOOT+A2H)

<Reference>

Section 2.6.5 I/O bytes
Section 4.5 Printer
APPENDIX 9 SAMPLE 3

[Function]
SECTRN translates a logical sector to a physical sector.

[Entry parameters]
BC=logical sector

[Return parameters]
HL=physical sector

<Explanation>

SECTRN only copies the register contents because logicalsectors are equal to the physical sectors for EHT-10/EHT-10/2.

[Function]
SCRNDUMP dumps the contents of current VRAM to the LST:device (default is a printer unit). This command is effective only for EHT-10/2. No operation is executed for EHT-10.

[Entry parameters]
None

[Return parameters]
(LSTERR)=00H: Normal termination
≠00H: Interrupted

<Explanation>

(1) The current LST: is determined according to the I/O bytes.

(2) The contents of the current LCD section in VRAM are output to LST: in bit image. For EHT-10, no operation is is executed and control returns.

(3) See the explanation of BIOS INFORM for the LSTERR address.

(4) Processing waits until the LST: device becomes ready if it is not ready. However, if LST: is a cartridge I/F and a printer unit is not mounted, error information is set in LSTERR and control returns.

<Relations>

INFORM(WBOOT+A2H)

<Reference>

Section 2.6.5 I/O bytes
Section 4.5 Printer

[Function]
BEEP sounds the buzzer.

[Entry parameters]
There are the following three ways to specify the BEEP entry parameters:
(1) Software beep
 1 Tone specification
  B=tone ($13 \leq B \leq 60$. No sound for B=0)
  C=period ($1 \leq C \leq 255$. The unit is 100 ms.)
 2 Frequency specification (Turn the MSB of register B to 1.)
  C=period ($1 \leq C \leq 255$. The unit is 100 ms.)
  D=small loop counter
  E=large loop counter (No sound for DE=0)
(2) Hardware beep
  B=tone
  =100 (64H): 512 Hz
  =101 (65H): 1024 Hz
  =102 (66H): 2048 Hz
  C=period ($1 \leq C \leq 255$. The unit is 100 ms.)

[Return parameters]
A=00H: Normal termination
=FFH: Forced termination due to alarm or power off.

<Explanation>

(1) EHT-10/EHT-10/2 supports the software and hardware beeps. For software beep, the sound in the specified frequency is generated by the software timer and processing cannot branch out of the BEEP routine until the sound stops. Details on specification are explained later. For hardware beep, the sound in the fixed frequency is generated by hardware and processing can branch out of the BEEP routine after the buzzer is turned on and the timer is set. Beep sound generation is monitored by using 1-ms/8-ms timer.

(2) Software beep is specified as follows:
1 The small and large loop values are set in registers D and E. The following expressions are used to determine values D and E:
$T1 = \{13 * (D-1) + 8\} / (3.68 * 10^6)$ Sec
$T2 = \{3307 * (E-1) + 240\} / (3.68 * 10^6)$ Sec
Period $T = 2(T1 + T2)$ Sec
Frequency $f = 1 / T$ Hz

2 Buzzer has the compass range of 200 to 4000 Hz.

3 The table below shows the relationship between tone specification value and actual tone. The values enclosed in a pair of parentheses ( ) are the large and small loop values.

Table 4.2 Software Beep Tone (large and small loops)

	0	1	2	3	4
C	1 (05H, 30H)	13 (03H, 0FH)	25 (01H, FCH)	37 (01H, 75H)	49 (01H, 31H)
C#	2 (04H, F1H)	14 (02H, EFH)	26 (01H, EDH)	38 (01H, 6DH)	50 (01H, 2DH)
D	3 (04H, B8H)	15 (02H, D2H)	27 (01H, DFH)	39 (01H, 66H)	51 (01H, 2AH)
D#	4 (04H, 82H)	16 (02H, B7H)	28 (01H, D1H)	40 (01H, 5FH)	52 (01H, 26H)
E	5 (04H, 4FH)	17 (02H, 9DH)	29 (01H, C4H)	41 (01H, 59H)	53 (01H, 23H)
F	6 (04H, 1EH)	18 (02H, 85H)	30 (01H, B8H)	42 (01H, 53H)	54 (01H, 20H)
F#	7 (03H, EFH)	19 (02H, 6EH)	31 (01H, ADH)	43 (01H, 4DH)	55 (01H, 1DH)
G	8 (03H, C4H)	20 (02H, 59H)	32 (01H, A2H)	44 (01H, 48H)	56 (01H, 1BH)
G#	9 (03H, 9BH)	21 (02H, 44H)	33 (01H, 98H)	45 (01H, 43H)	57 (01H, 18H)
A	10 (03H, 75H)	* 22 (02H, 31H)	34 (01H, 8FH)	46 (01H, 3EH)	58 (01H, 16H)
A#	11 (03H, 51H)	23 (02H, 1FH)	35 (01H, 85H)	47 (01H, 39H)	59 (01H, 13H)
B	12 (03H, 2FH)	24 (02H, 0EH)	36 (01H, 7DH)	48 (01H, 35H)	60 (01H, 11H)

Note) An asterisk (*) indicates 440-Hz tone.

(3) By changing the system area, interrupt suppressed/allowed status for buzzer sounding period can be modified. The address can be found out by the BIOS INFORM.

Address : Name(Byte length)

F23BH : BPINTEBL (1)

Interrupt control flag for buzzer sounding period
Bit 7: 1-sec interrupt 1 = suppressed, 0 = no change
Bit 6,5: Fixed to 0
Bit 4: EXT interrupt 1 = suppressed, 0 = no change
Bit 3: Fixed to 0
Bit 2: ICF interrupt 1 = suppressed, 0 = no change

Bit 1: ART interrupt 1 = suppressed, 0 = no change
Bit 0: Key interrupt 1 = suppressed, 0 = no change
OVF interrupt is always suppressed.

(4) No sound is generated when 0 is specified in register B for tone.

(5) No sound is generated when 0 is specified in registers D and E for frequency.

(6) Processing is terminated when function key F6H is pressed and alarm or power off occur.

(7) While buzzer is sounded, the cursor is not blinked to prevent the tone from warping caused by OVF interrupt. Cursor blinking can be controlled for other interrupts for the same reason.

<Reference>

APPENDIX 9 SAMPLE 5

[Function]

TIMDAT supports the clock functions. The following nine functions are provided depending on the value of register C:

C=00H: Reads time. (Read Time)
 =FFH: Sets time. (Set Time)
 =80H: Allows alarm/wake. (A/W enable)
 =81H: Suppresses alarm/wake. (A/W disable)
 =82H: Sets alarm/wake. (Set A/W)
 =83H: (TMDT83 hook)
 =84H: Reads alarm/wake. (Read A/W)
 =85H: (TMDT85 hook)
 =86H: (TMDT86 hook)

No operation is executed when register C contains a value other than above values.

Details on each function are explained later.
Registers D and E are saved for each function.

See "Section 10.2 System Hook" for TIMDAT hook (TMDT83, TMDT85, TMDT86).

<Explanation>

(1) The time descripter is used as the parameters when time, or alarm/wake is set or read.

Figure 4.2 Time descripter

+0	Year (lower two digits) 2-digit BCD	1 byte Year: 00 to 99
1	Month 2-digit BCD	1 byte Month: 01 to 12
2	Day 2-digit BCD	1 byte Day: 01 to 31
3	Hour 2-digit BCD	1 byte Hour: 00 to 23
4	Minutes 2-digit BCD	1 byte Minute: 00 to 5916
5	Second 2-digit BCD	1 byte Second: 00 to 59
6	Day of the week	1 byte Day of the week (00: Sun, 01:Mon,...06: Sat)
7	Type (Note 1)	1 byte
8	Address (Note 2)	2 bytes
9		
10	Status (Note 3)	1 byte

(Note 1) Type.....The alarm/wake type is set.
 =00H: No setting.
 =01H: Alarm setting
 =02H: Wake setting

(Note 2) Address.....The start address of alarm message or wake string is specified.

```
|<--1 byte-->|<--Up to 32 bytes-->|
 ------------------------------------
| Length    | Message or string    |
 ------------------------------------
^
|
└─ Start address
```

The length of data that can be displayed is 20 bytes for EHT-10 and 18 bytes for EHT-10/2. A control code (00H to 1FH) is counted as two bytes.

(Note 3) Status.....The alarm generation status is indicated.
=00H: No generation
=01H: Generated

(2) The lower two digits of dominical year is set as the year. 24-hour clock is used for time indication.

(3) Once set correctly, the parameters from year to the day of week are automatically adjusted for 1901 to 2099 including leap years.

(4) The system uses the following area to manage alarm/wake data:

Address : Name(Byte length)

F028H : ALRMTP (1)
Alarm/wake set type
=00H: No setting (default)
=01H: Alarm setting
=02H: Wake setting
- This area is referenced by alarm/wake execution processing.

F029H : ALRMAD (2)
Start address of alarm/wake message
- This indicates ALRMMSG (F32FH).

F02BH : ALRMST (1)
Alarm/wake generation status
=00H: No generation (default)
=01H: Generated
- 00H is set by TIMDAT "Set Alarm/Wake" or "Read Alarm/Wake" and 01H is set at alarm/wake interrupt occurrence.

F32FH : ALRMMSG (34)
Alarm/wake message storage area
The first byte contains the message length.
- The message specified by TIMDAT "Set Alarm/Wake" is stored.

<Reference>

Section 10.2 System Hook
APPENDIX 9 SAMPLE 6

[Function]
TIMDAT (Set Time) sets time.

[Entry parameters]
C=FFH: Function number
DE=start address of time descripter

[Return parameters]
None

[Saved registers]
DE

<Explanation>

(1) Time is set according to the 7-byte data indicating year, month, day, hour, minute, second, and day of week specified by the time descripter.

(2) When 1111B is specified in the time descripter digit that is not required to be updated, the previously set value of this digit remains effective.

(3) Since parameter validity is not checked, the clock contents are not guaranteed if logically incorrect data is specified.

<Reference>

APPENDIX 9 SAMPLE 6

[Function]
TIMDAT (Read Time) reads the current time.

[Entry parameters]
C=00H: Function number
DE=start address of time descripter (A 7-byte area is required.)

[Return parameters]
The current time is set in the time descripter.

[Saved registers]
DE

<Explanation>

The current time (year, month, day, hour, minute, second, and day of week) is returned in the time descripter beginning from the start address.

<Reference>

APPENDIX 9 SAMPLE 6

[Function]
TIMDAT (Alarm/Wake Enable) enables the alarm/wake function.

[Entry parameters]
C=80H: Function number

[Return parameters]
None

[Saved registers]
DE

<Explanation>

(1) This command executes the alarm/wake operation at the set time.

(2) The system automatically enables the alarm/wake function when Set Alarm/Wake operation is executed.

<Relations>

TIMDAT (Set Alarm/Wake)

[Function]
TIMDAT (Alarm/Wake Disable) disables the alarm/wake function.

[Entry parameters]
C=81H: Function number

[Return parameters]
None

[Saved registers]
DE

<Explanation>

(1) Alarm/wake operations can no longer be executed after this command is executed.

(2) The alarm/wake operation data is saved even if Alarm/Wake Disable is executed. This alarm/wake data becomes effective when Alarm/Wake Enable is executed again.

<Relations>

TIMDAT (Alarm/Wake Enable)

[Function]
TIMDAT (Set Alarm/Wake) sets the time for alarm/wake operations.

[Entry parameters]
C=82H: Function number
DE=start address of time descripter

[Return parameters]
None

[Saved registers]
DE

<Explanation>

(1) 10-byte data in time descripter starting from year to address is set. However, 2-digit year and the lower digit of 2-digit second are ignored (10 is the minimum value that can be set for seconds).

(2) If all 1 is set as a BCD digit, the digit is assumed equal to any number from 0 to 9. For example, all 1 is set to the month, day, and day of week digits and specific values are set to the hour, minute, and second digits, alarm/wake operations are executed at the specified hour, minute, and second every day.

(3) The alarm/wake function is automatically enabled after this command is executed.

(4) Since parameter validity is not checked, alarm/wake processing is not guaranteed if logically incorrect data is input. Especially, data other than 00H to 02H must not be specified as the type.

[Function]
TIMDAT (Read Alarm/Wake) reads the alarm/wake time.

[Entry parameters]
C=84H: Function number
DE=start address of time descripter (11-byte area is required).

[Return parameters]
The alarm/wake status is set in the time descripter.

[Saved registers]
DE

<Explanation>

When this command is executed, the current alarm/wake time is set in the time descripter.

However, all 1 is set as the year and as the lower digit of the second because these are not included in the set alarm/wake time.

[Function]
MEMORY checks the current bank information.

[Entry parameters]
None

[Return parameters]
C=bank information

<Explanation>

An application program calls this routine to find out the bank in which the application program is being executed. The bank information is indicated as follows:

Bank information

```
                 7   6   5   4   3   2   1   0
               +---+---+---+---+---+---+---+---+
Register C     |   |   |   |   |   |   |   |   |
               +---+---+---+---+---+---+---+---+
               |<-- Subbank -->|<- Main bank ->|
               |  information  |  information  |
```

Bank information	Bank number	Explanation
FFH	System bank	0 to 7FFFH in system ROM
00H	0#0	Bank in Standard RAM
10H 20H 30H 40H 50H 60H	0#1 0#2 0#3 0#4 0#5 0#6	Banks in extended RAM
11H 21H 31H	1#1 1#2 1#3	Banks in system ROM
02H 12H 22H 32H	2#0 2#1 2#2 2#3	Banks in application ROM

Fig. 4.3 Correspondence between bank information and bank

<Reference>

Section 1.4 Memory Map

[Function]

RSIOX performs serial communication. RSIOX has the following 10 functions depending on the value in register B:

```
B=1XH:  Opens device. (Open)
        Lower 4 bits:  Device specification
                  =0:  RS-232C
                  =3:  Cartridge SIO
 =20H:  Closes device. (Close)
 =30H:  Checks whether data has been received in the receive buffer.
(Insts)
 =40H:  Checks whether sending is possible.  (Outst)
 =50H:  Receives 1-byte data from the receive buffer. (Get)
 =60H:  Sends 1-byte data. (Put)
 =70H:  Reads the control line status. (Ctlin)
 =80H:  Sets a control line. (Setctl)
 =90H:  Reads error status and clears error flags. (Ersts)
 =F0H:  Checks the current serial device use status. (Sens)
```

Details on each function are explained later.

<Explanation>

(1) Two types of interfaces (RS-232C and cartridge SIO) are provided for EHT-10/EHT-10/2 serial communication. However, these two interfaces cannot be used at the same time and only one device can be used. Because of this, OS uses one device efficiently. See "Section 7.2 Serial Interface" for details.

(2) The list below shows the system areas used by RSIOX.

Address : Variable name(Byte length)

F00DH : RSXON (1)
XON code used when XON/XOFF is specified. The initial value is 13H.

F00EH : RSXOFF (1)
XOFF code used when XON/XOFF is specified. The initial value is 11H.

F604H : RSPSTS (1)
RSIOX status flags
- Bit 7: DSR status (0: Active)
- Bit 6: Framing error status (1: Error)
- Bit 5: Receive overrun status (1: Overrun)
- Bit 4: Parity bit error status (1: Error)
- Bit 3: CD status (1: Active)
- Bit 2: Receive buffer overflow status (1: Overflow)
- Bit 1: Receive buffer status (1: Full)
- Bit 0: Open status (1: Open)

F605H : RSPRBGP (2)
Receive buffer get pointer

F607H : RSPRBPP (2)
Receive buffer put pointer

F609H : RSPRBAD (2)
Start address of receive buffer

F60BH : RSPRBSZ (2)
Size of receive buffer

F60DH : RSPBITR (1)
Bit rate (parameter set at open operation)

=02H:110bps	=0AH:1200bps	=10H:38400bps
=04H:150	=0CH:2400	=80H:75/1200
=05H:200	=0DH:4800	=81H:1200/75
=06H:300	=0EH:9600	
=07H:600	=0FH:19200	

F60EH : RSPBITL (1)
Bit length (Parameter set at open operation)
=02H:7bits =03H:8bits

F60FH : RSPPAR (1)
Parity (parameter set at open operation)
=00H:NON =01H:ODD =02H:EVEN

F610H : RSPSTOPB (1)
Stop bit (parameter set at open operation)
=01H:1bit =02H:2bits

F611H : RSPSPP (1)
Special parameter (parameter set at open operation)
Bit 7: Unused
Bit 6: RTS/CTS control (0: Control)
Bit 5: DTR/DSR control (0: Control)
Bit 4: XON/XOFF control (0: Control)
Bit 3: Unused
Bit 2: SI/SO control (1: Control)
Bit 1: RTS control (1: Active)
Bit 0: DTR control (1: Active)
- XON/XOFF, RTS/CTS, and DTR/DSR are used for buffer control. Only one of these can be specified.

F612H : RSRBFAD (2)
Receive buffer end address + 1

F614H : RSXONSZ (2)
Receive data length at the beginning of XON code sending (Receive buffer size/4)

F616H : RSXOFSZ (2)
Receive data length at XOFF code sending (Receive buffer size x 3/4)

F618H : CHR5MSK (1)
Receive data mask pattern
7FH...6-bit length, FFH...8-bit length

F619H : RSSSPP (1)
Special parameter for system reference
Bit 7: XOFF receive flag (1: Received)
Bit 6: RTS/CTS control (1: Active)
Bit 5: DTR/DSR control (1: Active)
Bit 4: XON/XOFF control (1: Active)
Bit 3: XOFF send flag (1: Sent)
Bit 2: SI/SO control (1: Active)
Bit 1: RTS control (1: Active)
Bit 0: DTR control (1: Active)

F61AH : RSPAKAD (2)
Address of RSIOX parameter packet (Value of RSIOX parameter HL)

F61CH : RSRDL (2)
Length of data stored in receive buffer

F61EH : SISOCNT (2)
Number of SI and SO codes in PSRDL. Actual data length can be determined by the following expression: RSRDL - SISOCNT

F620H : SSXMODE (1)
SI/SO send status
=00H: SI send status
=01H: SO send status
=02H: SI and SO unsend status (initial status)

F621H : RSXMODE (1)
SI/SO receive status
=00H: SI status (initial status)
=80H: SO status

F622H : RSFDEV (1)
RSIOX parameter 4-bit LSB in register B

F623H : RSODEV (1)
Device mode used by RSIOX
=00H: RS-232C
=03H: Cartridge SIO
=FFH: Nothing has been opened.

(3) Among the device specifications, only the serial communication mode switching is controlled by the system. Because of this, the cartridge mode (DB, IO, HS, or OT) is not changed even if the cartridge SIO is specified as the device. See "Section 7.3 Cartridge Interface" for details.

(4) The following terms are used for the RSIOX functions:
Serial: Generic for RS-232C and cartridge SIO
RS-232C: Communication through RS-232C connector
Cartridge SIO: Communication through cartridge connector

<Reference>

Section 7.2 Serial Interface
Section 7.3 Cartridge Interface
APPENDIX 9 SAMPLE 7

[Function]

RSIOX (Open) opens a device to be used.

[Entry parameters]

B=function number
 =10H: Opens RS-232C.
 =13H: Opens cartridge SIO.
HL=Start address of parameter block (9-byte area is required. Details are explained later.)

[Return parameters]

A=return information
 =00H: Normal termination (z-flag=1)
 =02H: The device had been opened. (z-flag=0)
HL=start address of return information block (Value is the same as entry HL. Details are explained later.)

<Explanation>

(1) This command is executed to switch the device to the specified one, set the communication status according to the specified parameters, allow serial interrupts. and enable sending and receiving.

(2) A parameter block is structured as follows:

Fig. 4.3 RSIOX parameter block

(HL)->	Start address of receive buffer
+1	
+2	Size of receive buffer (bytes)
+3	
+4	Bit rate
+5	Character length
+6	Parity check
+7	Stop bit
+8	Special parameter

1 Start address of receive buffer
This parameter specifies the start address of receive buffer.

2 Size of receive buffer (bytes)
This parameter specifies the size of receive buffer.

3 Bit rate
This parameter specifies the transmission speed.

Value	Send bit rate	Receive bit rate
02H	110 bps	110 bps
04H	150	150
05H	200	200
06H	300	300
08H	600	600
0AH	1200	1200
0CH	2400	2400
0DH	4800	4800
0EH	9600	9600
0FH	19200	19200
10H	38400	38400
80H	75	1200
81H	1200	75

4 Character length
This parameter specifies 1-character length for communication.
02H: 7 bits/character
03H: 8 bits/character

5 Parity check
This parameter specifies parity checking.
00H: No parity checking
01H: Odd parity
03H: Even parity

6 Stop bit
This parameter specifies the stop bit length for communication.
01H: 1 bit
03H: 2 bits

7 Special parameter
Each bit of this parameter specifies condition or status for serial communication

Bit position	Contents
0	Data-terminal-ready (DTR) line control 0: Inactive 1: Active
1	Request-to-send (RTS) line control 0: Inactive 1: Active
2	Shift in/sift out (SI/SO) control 0: Control 1: No control
3	Unused
4	XON/XOFF control 0: Control 1: No control
5	DTR/DSR control 0: Control 1: No control
6	RTS/CTS control 0: Control 1: No control
7	Unused

DTR and RTS are effective only when RS-232C is specified.
XON/XOFF, DTR/DSR, and RTS/CTS are used for buffer control. Only one of them can be specified.

(3) A return information block is structured as follows:

Fig. 4.4 RSIOX return information block

Offset	Contents
(HL) ->	Status flag
+1	Receive get pointer
+2	
+3	Receive put pointer
+4	
+5	Start address of receive buffer
+6	
+7	Size of receive buffer
+8	

1 Status flag
Each bit indicates serial status.

Bit position	Contents
0	Open status 0: Not open 1: Open status
1	Receive buffer status 0: Buffer sufficient 1: Buffer full
2	Overflow status of receive buffer 0: No overflow 1: Overflow
3	Carrier detector (CD) status 0: Inactive 1: Active
4	Parity error status 0: No error 1: Parity error
5	Receive overrun error status 0: No error 1: Receive overrun error
6	Framing error status 0: No error 1: Framing error
7	Data set ready (DSR) status 0: Active 1: Inactive

CD and DSR are effective only when RS-232C is specified.
Error information indicated by bits 2 and 4 to 6 is held once an error occurs until RSIOX (Ersts) is called.

2 Receive buffer get pointer
This pointer is used to get data from the receive buffer.

3 Receive buffer put pointer
This pointer is used to put receive data in the receive buffer.

4 Start address of receive buffer
Start address of receive buffer

5 Size of receive buffer
This is the byte length of receive buffer.

(4) Buffer control is ineffective when the size of receive buffer is less than 16 bytes.

(5) SI/SO is specified to send 8-bit data when 7 bits/character is used. Codes 0EH and 0FH cannot be sent when SI/SO processing is specified. Because of this, binary data cannot be sent. See "Section 7.2 Serial Interface" for details.

(6) XON/XOFF, DTR/DSR, or RTS/CTS is specified to synchronize the receive and send sides when the communication speed is faster than the processing speed at the receive side. Codes 11H and 13H cannot be sent when XON/XOFF processing is specified. Because of this, binary data cannot be sent. See "Section 7.2 Serial Interface" for details.

<Relations>

RSIOX (Ersts)

<Reference>

Section 7.2 Serial Interface
APPENDIX 9 SAMPLE 7

[Function]
RSIOX (Close) closes the used device.

[Entry parameters]
B=20H: Function number

[Return parameters]
None

<Explanation>

(1) RSIOX (Close) closes the serial device and prohibits interrupts at the receive side.

(2) The serial device is automatically closed when WBOOT processing is executed.

<Reference>

APPENDIX 9 SAMPLE 7

[Function]

RSIOX (Insts) checks whether there is receive data in the receive buffer.

[Entry parameters]

B=30H: Function number
HL=start address of return information block (9-byte area is required.)

[Return parameters]

Z-flag=1: Normal termination
A=FFH: There is receive data in receive buffer.
=00H: There is no receive data in receive buffer.
BC=byte length of receive data
HL=saved (return information in the same structure as Open)
Z-flag=0: Abnormal termination
A=03H: Open operation had not been executed.

<Explanation>

(1) RSIOX (Insts) checks whether there is receive data in the receive buffer.

(2) XON or XOFF codes (when XON/XOFF is specified) or SI or SO codes (when SI/SO is specified) are not included in the byte length of receive data.

(3) The current send or receive status is set in the return information block.

[Function]

RSIOX (Outst) checks whether sending can be done.

[Entry parameters]

B=40H: Function number
HL=start address of return information block (9-byte area is required.)

[Return parameters]

Z-flag=1: Normal termination
A=FFH: Sending possible
=00H: Sending not possible
HL=saved (return information in the same structure as Open)
Z-flag=0: Abnormal termination
A=03H: Open operation had not been executed.

<Explanation>

(1) Whether sending is possible is determined by checking TXReady. Sending is not possible under one of the following three conditions:
- XON/XOFF control is specified and XON is received.
- DTR/DSR control is specified and DSR is inactive.
- RTS/CTS control is specified and CTS is inactive.

(2) The current send or receive status is set in the return information block.

<Reference>

APPENDIX 9 SAMPLE 7

[Function]
RSIOX (Get) fetches 1-byte data from the receive buffer.

[Entry parameters]
B=50H: Function number
HL=start address of return information block (9-byte area is required.)

[Return parameters]
Z-flag=1: Normal termination
A=receive data
HL=saved (return information in the same structure as Open.)
Z-flag=0: Abnormal termination
A=03H: Open operation had not been executed.
=04H: Processing was forcibly terminated.
=05H: Receive-buffer-overflow occurred.

<Explanation>

(1) RSIOX (Get) fetches 1-byte data from the receive buffer and sets it in register A.

(2) When data is not yet received, processing waits until data is received.

(3) Power off or alarm/wake operation is executed if power off or alarm/wake occurs while waiting for data to be received. Processing waits for data to be received after the power off or alarm/wake operation.

[Function]

RSIOX (Put) sends the specified 1-byte data.

[Entry parameters]

B=60H: Function number
C=send data
HL=start address of return information block (9-byte area is required.)

[Return parameters]

Z-flag=1: Normal termination
HL=saved (return information in the same structure as Open)
Z-flag=0: Abnormal termination
A=03H: Open operation had not been executed.
=04H: Processing was forcibly terminated.

<Explanation>

(1) RSIOX (Put) checks the send possible (ready) status and sends the specified data if sending is possible.

(2) Whether sending is possible is determined in the same way as RSIOX (Outst). If sending is not possible, processing waits until sending becomes possible.

(3) If forced termination occurs while processing is waiting for sending to become possible, A=04H is set.

(4) Power off or alarm/wake operation is performed if power off or alarm/wake occurs while processing is waiting for sending to become possible. Processing waits again for sending to become possible after the power off or alarm/wake operation.

<Relations>

RSIOX (Outst)
APPENDIX 9 SAMPLE7

[Function]
RSIOX (Ctlin) reads control line information.

[Entry parameters]
A=70H: Function number

[Return parameters]
Z-flag=1: Normal termination
A=control line status
Z-flag=0: Abnormal termination
A=03H: Open operation had not been executed.

<Explanation>

The control line status is structured as follows in the return parameter:

Bit position	Contents
7	Data set ready (DSR) status 0: Active 1: Inactive
6	Unused
5	Clear to send (CTS) status 0: Inactive 1: Active
4	Unused
3	Carrier detect (CD) status 0: Inactive 1: Active
2,1,0	Unused

CTS, CD, and DSR are effective only when RS-232C is specified.

[Function]

RSIOX (Setctl) sets the control line in the specified status.

[Entry parameters]

B=80H: Function number
C=control line status

[Return parameters]

Z-flag=1: Normal termination
Z-flag 0: Abnormal termination
A=03H: Open operation had not been executed.

<Explanation>

(1) The control line status is structured as follows in the return parameter:

Bit position	Contents
From 7 to 2	Unused (0)
1	Request to send (RTS) control 0: Inactive 1: Active
0	Date transmit ready (DTR) control 0: Inactive 1: Active

RTS and DTR are effective only when RS-232C is specified.

(2) This command is used to reset the control line status specified at open operation.

[Function]
RSIOX (Ersts) reads error status and clears the error flags.

[Entry parameters]
B=90H: Function number

[Return parameters]
Z-flag=1: Normal termination
A=error information
Z-flag=0: Abnormal termination
A=03H: Open operation had not been executed.

<Explanation>

(1) The error information is structured as follows in the return parameter:

Bit position	Contents
7	Data set ready (DSR) status 0: Active 1: Inactive
6	Framing error status 0: No error 1: Error
5	Receive overrun error status 0: No error 1: Error
4	Parity error status 0: No error 1: Error
3	Carrier detect (CD) status 0: Inactive 1: Active
2	Receive buffer overflow error status 0: No error 1: Error
1,0	Unused

The same information is set for CD and DSR regardless of the device specification.

(2) The status of errors is reset by the software after the current error status is read.

(3) If a receive buffer overflow occurs, receive data is discarded until data is fetched from the receive buffer by RSIOX(Get).

<Reference>

APPENDIX 9 SAMPLE 7

[Function]
RSIOX (Sens) checks the use status of the current serial device.

[Entry parameters]
B=F0H: Function number

[Return parameters]
A=serial device use status

7	6	5	4	3	2	1	0
-	-	-	-				

Bits 3–0 → Used device
=0H: RS-232C
=3H: Cartridge SIO
=FH: No device is used.

<Explanation>

RSIOX (Sens) checks the use status of the current serial device and sets the information in register A.

[Function]
MASKI sets interrupt mask and checks the current mask status.

[Entry parameters]
B=interrupt mask data
C=7508-related interrupt mask data (explained later)

[Return parameters]
B=old interrupt mask status
C=old 7508-related interrupt mask status

[Saved registers]
AF, DE, HL

<Explanation>

(1) MASKI controls five types of interrupts for EHT-10/EHT-10/2 and three types of interrupts related to 7508.

(2) The entry parameters are structured as follows:

1 Register B

```
  7   6   5   4   3   2   1   0
+---+---+---+---+---+---+---+---+
|   | - | - |   |   |   |   |   |
+---+---+---+---+---+---+---+---+
  |           |   |   |   |   |
  |           |   |   |   |   +--> 7508 (Slave 4bit CPU)
  |           |   |   |   +------> ART (Serial RxRDY)
  |           |   |   +----------> ICF (BCRD signal change)
  |           |   +--------------> OVF (FRC overflow)
  |           +------------------> EXT (external interrupt)
  |                (Each of above indicates interrupt
  |                 suppressed when the bit is 0 and interrupt
  |                 allowed when the bit is 1.)
  Function specification
   =0:  Sets a mask (interrupt suppressed/allowed status is changed).
   =1:  Reads the current mask status (the mask status is not changed)
```

2 Register C

```
     7   6   5   4   3   2   1   0
   +---+---+---+---+---+---+---+---+
   |   |   |   | - |   |   |   | - |
   +---+---+---+---+---+---+---+---+
     |   |   |       |   |   |
     |   |   |       |   |   +-----> Key input        -+
     |   |   |       |   +-----> Alarm interrupt       |-- Related to
     |   |   |       +-----> 1-second interrupt       -+   7508
     |   |   |
     |   |   +--------> IC card cover          -+
     |   |              panel interrupt          |
     |   +-----> Cartridge I/F interrupt         |-- External (EXT)
     +--> 1-ms/8-ms interval                    -+   interrupts
```

(Each of these indicates interrupt suppressed when the bit is 0 and interrupt allowed when the bit is 1.)

3 The interrupt mask status held when MASKI is called is set in the return parameter. Registers B and C are structured in the same way as entry parameters.

4 The current interrupt status is stored in the following system areas:

Address : Variable name(Byte length)

F031H : ISTS7508 (1)
The current interrupt mask status related to 7508
Bits 7 to 4: Unused
Bit 3: 1-second interrupt (0: Suppressed, 1: Allowed)
Bit 2: Alarm interrupt (0: Suppressed, 1: Allowed)
Bit 1: Key interrupt (0: Suppressed, 1: Allowed)
The default value is 0BH.

F42EH : RZIER (1)
Current interrupt mask status
Bits 7 to 5: Unused
Bit 4: EXT interrupt (0: Suppressed, 1: Allowed)
Bit 3: OVF interrupt (0: Suppressed, 1: Allowed)
Bit 2: ICF interrupt (0: Suppressed, 1: Allowed)
Bit 1: ART interrupt (0: Suppressed, 1: Allowed)
Bit 0: 7509 interrupt ((0: Suppressed, 1: Allowed)
The default value is 19H.

(3) When the interrupt status is modified, the current interrupt status is read and the bits required to be modified are set or reset.

(4) Return the interrupt status back to the one held before modification when user processing ends.

<Reference>

CHAPTER 8 INTERRUPT PROCESSING
APPENDIX 9 SAMPLE 18

[Function]
LOADX reads 1-byte data at the specified address from the specified bank.

[Entry parameters]
C=bank information
HL=data address

[Return parameters]
A=read data

[Saved registers]
BC, DE, HL, IX, IY

<Explanation>

(1) Bank information is in the same format as the return parameter of MEMORY.

(2) Since parameter validity is not checked, operation is not guaranteed if incorrect parameters are specified.

(3) The data address is the address of data mapped in memory. The data address is not equal to ROM address when data is in a bank in application ROM.

<Relations>

MEMORY (WBOOT+4EH)

<Reference>

Section 1.4 Memory Map

[Function]
STORX writes 1-byte data at the specified address in the specified bank.

[Entry parameters]
C=bank information
A=data
HL=data address

[Return parameters]
None

[Saved registers]
AF, BC, DE, HL, IX, IY

<Explanation>

(1) All registers are saved.

(2) Since parameter validity is not checked, operation is not guaranteed if incorrect parameters are specified.

(3) Data can be actually written only in RAM.

<Relations>

MEMORY (WBOOT+4EH)
LOADX (WBOOT+5AH)

<Reference>

Section 1.4 Memory Map
APPENDIX 9 SAMPLE 19

[Function]
LDIRX moves data as much as the specified number of bytes from the specified bank to another bank.

[Entry parameters]
BC=byte length of data to be transferred
DE=start address of data in the destination bank
HL=start address of data in the original bank
(SRCBNK)=information of the original bank (same as MEMORY)
(DISBNK)=information of the destination bank (same as MEMORY)

[Return parameters]
BC=0000H
DE=entry parameter DE + entry parameter BC
HC=entry parameter HL + entry parameter BC
(Same as the ones at Z-80 instruction LDIR execution)

[Saved registers]
AF, IX, IY

<Explanation>

(1) This command is used to take into account of the bank used by instruction LDIR.

(2) See the explanation of BIOS INFORM for the SRCBNK and DISBNK addresses.

(3) Since parameter validity is not checked, operation is not guaranteed if incorrect parameters are specified.

<Relations>

MEMORY (WBOOT+4EH)
INFORM (WBOOT+A2H)

<Reference>

APPENDIX 9. SAMPLE 19

[Function]
JUMPX jumps to the specified address in the specified bank.

[Entry parameters]
IX=jump address
(DISBNK)=bank information (same as MEMORY)

[Return parameters]
None

[Saved registers]
AF, BC, DE, HL, IX, IY (when control jumps to the destination)

<Explanation>

(1) At the jump destination, all registers remain in the same status as the one held when control is passed to JUMPX.

(2) See the explanation of BIOS INFORM for the DISBNK address.

(3) Since parameter validity is not checked, operation is not guaranteed if incorrect parameters are specified.

(4) The BIOS stack provided by the system is used after this command is executed. Therefore, the processing routine to be executed after jump operation must set the stack again. If this is not done, wild run may occur when BDOS/BIOS is used.

(5) The system is in "BIOS in progress" status after this command is executed. Call JPSTBIOS (FF96H) to make the system out of "BIOS in progress" status.

<Relations>

MEMORY (WBOOT+4EH)
INFORM (WBOOT+A2H)

<Reference>

Section 1.4 Memory Map

[Function]
CALLX calls the specified address in the specified bank.

[Entry parameters]
IX=call address
(DISBNK)=bank information (same as MEMORY)

[Return parameters]
Among the registers returned from call destination, the ones other than IX and IY are saved.

<Explanation>

(1) The registers held when this routine is called are passed to the call destination.

(2) See the explanation of BIOS INFORM for the DISBNK address.

(3) Since parameter validity is not checked, operation is not guaranteed if incorrect parameters are specified.

(4) After this command is executed, the BIOS stack provided by the system is used and the system is in "BIOS in progress" status. Therefore, the notes (4) and (5) in the explanation of JUMPX must also be taken into account.

(5) When control returns, the original stack must be set again if the stack is modified at the call destination.

(6) This command is used to call a utility stored in system ROM.

<Relations>

MEMORY (WBOOT+4EH)
JUMPX (WBOOT+63H)
INFORM (WBOOT+A2H)

<Reference>

Section 1.4 Memory Map
APPENDIX 9 SAMPLE 21

[Function]
GETPFK reads the code of the current set key.

[Entry parameters]
C=function code
=01H: Reads the key table in normal mode.
=02H: Reads the key table in alphabet mode (effective only for EHT-10/2 but not for EHT-10).
B=position code
≠FFH: Reads the key code of the key corresponding to the position code.
=FFH: Reads all codes.
When B=FFH,
HL=start address of buffer used to store key code data

[Return parameters]
(1) When GETPFK is called with B≠FFH:
A=key code

(2) When GETPFK is called with B=FFH:
(HL)=key code data

[Saved registers]
BC, HL

<Explanation>

(1) All the key codes beginning from the address specified in HL are read from the key table when GETPFK is called with B=FFH.

(2) A 70-byte buffer (for EHT-10) or a 32-byte buffer (for EHT-10/2) must be reserved after address indicated by HL when B=FFH is specified. The read data is stored in this buffer in the ascending order of the key position codes.

<Relations>

CONST (WBOOT+03H)
CONIN (WBOOT+06H)
TOUCH (WBOOT+93H)

<Reference>

Section 4.3 Key Input (Touch Panel/Keyboard)
APPENDIX 9 SAMPLE 8

[Function]
PUTPFK stores a key code in the user key table.

[Entry parameters]
C=function code
=00H: Initializes the user key table.
=01H: Stores a key code in the key table in normal mode.
=02H*: Stores a key code in the key table in alphabet mode
(* is effective only for EHT-10/2 but not for EHT-10)
B=position code
≠FFH: Stores the key code of the key corresponding to the position code.
When B≠FFH,
A=key code
=FFH: Sets the entire user key table.
When B=FFH,
HL=start address of key code buffer

[Return parameters]
None

[Saved registers]
BC, HL

<Explanation>

(1) The contents of register B are meaningless when the user key table is initialized.

(2) Data stored at addresses starting from the one indicated by HL is set in the key table when the entire key table is set. The required length of data is 70 bytes for EHT-10 or 32 bytes for EHT-10/2. Data must be stored in the ascending order of the key position codes.

(3) The default values are stored in normal or alphabet mode key table when warm boot is executed.

<Relations>

CONST (WBOOT+03H)
CONIN (WBOOT+06H)
TOUCH (WBOOT+93H)

<Reference>

Section 4.3 Key Input (Touch Panel/Keyboard)
APPENDIX 9. SAMPLE 9

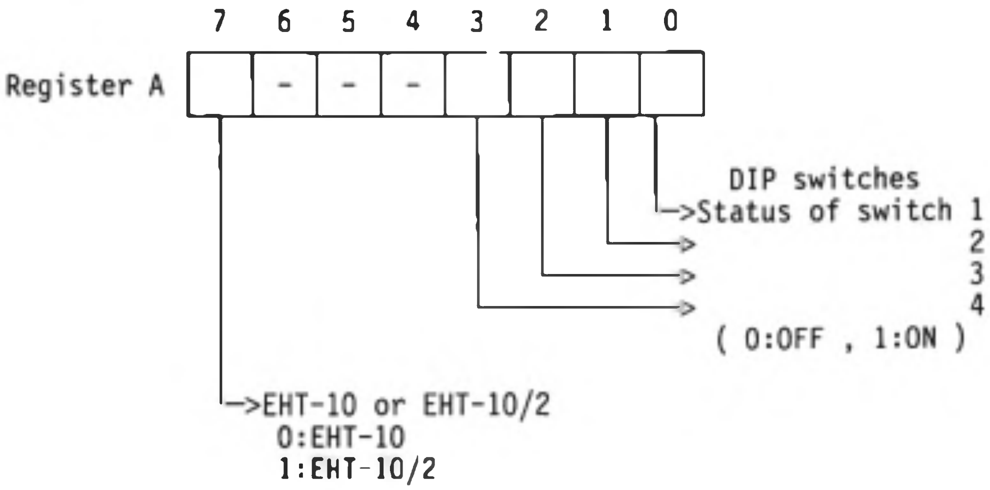
[Function]
READSW reads the status of various switches.

[Entry parameters]
C=02H: Reads the DIP switches.
=04H: Reads the power switch.

[Return parameters]
(1) When the DIP switches are read,
A=DIP switch status (explained later)
(2) When the power switch is read,
A=01H: Power switch on
=00H: Power switch off

<Explanation>

When the DIP switches are read, the return parameter isstructured as follows:



<Reference>

APPENDIX 9 SAMPLE LIST

[Function]
RDVRAM reads one character from the screen.

[Entry parameters]
B=read column position
C=read line position

[Return parameters]
A=00H: Normal termination
When A=00H,
H=character attribute (explained later)
L=character code (ASCII code)
A=FFH: Parameter error

[Saved registers]
BC

<Explanation>

(1) RDVRAM reads one character from the character buffer of the current active screen. The values that can be specified in the entry parameters are limited as follows:

1 EHT-10 (vertical display)

(i) Virtual screen section

$1 < B < 12$

$1 \leq C \leq$ virtual screen size

(ii) Fixed display section

$1 < B < 12$

$1 \leq C \leq 14$

2 EHT-10 (horizontal display)

$1 < B < 25$

$1 \leq C \leq$ virtual screen size

3 EHT-10/2

$1 < B < 20$

$1 \leq C \leq$ virtual screen size

(2) The character attributes are indicated in the same way as BIOS CONOUT Set Attribute (ESC+D9H) as follows:

```
               7   6   5   4   3   2   1   0
             +---+---+---+---+---+---+---+---+
Register H   | - | - |   |   |   |   |   |   |
             +---+---+---+---+---+---+---+---+
                       |   |   |   |   |   |
                       |   |   |   |   |   +->Horizontal ruler
                       |   |   |   |   |        line (top)
                       |   |   |   |   +->Horizontal ruler line
                       |   |   |   |       (bottom)
                       |   |   |   +->Vertical ruler line
                       |   |   +->Comma
                       |   +------>Reverse
                       +---------->Secret
                                    (0:Not Specified, 1:Specified)
```

(Note 1) Only the reverse and secret attributes are effective for EHT-10 horizontal display.

(Note 2) The contents of register H are meaningless for EHT-10/2 because EHT-10/2 does not have any character attributes.

<Relations>

CONOUT (WBOOT+09H)

<Reference>

Section 4.4 LCD Display
APPENDIX 6 FUNCTION DISPLAY CONTROL
APPENDIX 9 SAMPLE 22

[Function]
POWEROFF turns the system power supply off.

[Entry parameters]
C=power off mode
=00H: Turns the power off in continue mode.
=01H: Turns the power off in restart mode.

[Return parameters]
None

<Explanation>

(1) POWEROFF saves the current system status and turns the system power supply off.

(2) When the power is turned off in restart mode, the power may be turned off in continue mode instead of restart mode unless continue mode is reset by BIOS CONTINUE in advance.

<Relations>

CONTINUE (WBOOT+87H)

<Reference>

Section 2.3 Sleep function and Automatic Power-off function
APPENDIX 10 SAMPLE 10

[Function]
USERBIOS registers a user BIOS.

[Entry parameters]
Parameters differ depending on the user.

[Return parameters]
Parameters differ depending on the user.

<Explanation>

(1) USERBIOS is used to extend a user BIOS. A user BIOS area is reserved and the user BIOS is stored in this area.

(2) The execution start address of the user BIOS is stored as the entry address.

(3) The default value is the address indicating only return operation.

(4) The default value is set at reset or system initialize operation.

<Reference>

Section 4.6 User BIOS

[Function]
CONTINUE sets/resets the continue flag.

[Entry parameters]
C=00H: Reset
=01H: Set

[Return parameters]
None

<Explanation>

If the power switch is turned off after the continue flag is reset, the power is turned on in restart mode during subsequent power on operation. The default is continue mode.

<Reference>

APPENDIX 9 SAMPLE 11

[Function]

BARCODE supports the barcode reader functions. The following four functions are provided depending on the contents of register C:

C=00H: Allows to use barcode reader. (Open)
=01H: Prohibits to use barcode reader. (Close)
=02H: Reads one character. (Read)
=03H: Indicates the buffer status. (Status)

Details on each function are explained later.

<Reference>

Section 7.5 Barcode Reader Interface
Section 11.5 Addition of Bar Code Decoder
APPENDIX 9 SAMPLE 12

[Function]
BARCODE (Open) turns the barcode reader power on and enables the barcode reader operations.

[Entry parameters]
C=00H: Function number
A=code type
=01H: EAN/UPC-A/UPC-E/JAN
=02H: 3 of 9
=03H: Codabar
=04H: Interleaved 2 of 5
D=option (explained later)
E=delimiter (explained later)

[Return parameters]
None

<Explanation>

(1) The following options can be set:

Bit in register D	Corresponding code	Explanation
b7	All codes	Buzzer for normal reading 0: Sounds the buzzer. 1: Suppresses the buzzer. The buzzer is sounded for about 100 ms at normal reading when 0 is specified
b6	All codes	Delimiter 0: Uses the delimiter. 1: Suppresses delimiter. A delimiter is a special code (one character) inserted between two codes to indicate the end of a code. When the delimiter is specified, the code specified in register E is used as the delimiter.
b5	All codes	LED control for barcode reader 0: Performs LED control. 1: Suppresses LED control. LED remains lit when LED control is suppressed. LED blinks to have less power consumption when LED control is on.
b4,b3	Unused	

b2	Only for UPC-E	Zero addition 0: Zero is not added. 1: Zeros are added.
b1	Only for 3-of-9	Full ASCII conversion specification 0: Suppresses full ASCII conversion. 1: Performs full ASCII conversion.
b0	Only for 3-of-9	Check digit 0: Does not assume the last data as the check digit. 1: Assumes the last data as the check digit. When the check digit is specified, the last data is assumed as the check digit and is not indicated as a character. When the check digit does not match, the entire code is assumed invalid and ignored.

(2) The delimiter is specified as follows:
The value specified in register E is used as the delimiter when option b6 is 0. A value from 00H to FFH can be specified but this value must not be equal to any codes used for the barcodes in the character set. Other wise, the code is not effective as the delimiter.

<Reference>

APPENDIX 9 SAMPLE 12

[Function]
BARCODE (Close) turns the barcode reader power off and disables the barcode reader operations.

[Entry parameters]
C=01H: Function number

[Return parameters]
None

<Reference>

APPENDIX 9 SAMPLE 12

[Function]
BARCODE (Read) reads one character from the barcode reader.

[Entry parameters]
C=02H: Function number

[Return parameters]
Z-flag=0: There is data.
When Z-flag=0,
A=data
Z-flag=1: There is no data.

<Explanation>

(1) A barcode is read during interrupt processing and stored in a buffer. This command only fetches one character from data stored by interrupt processing.

(2) If 0 is specified as option b6 and delimiter is specified at Open operation, the delimiter is inserted between barcodes.

<Reference>

CHAPTER 8 INTERRUPT PROCESSING
APPENDIX 9 SAMPLE 12

[Function]

BARCODE (Status) indicates error information and the data stored in buffer by interrupt processing.

[Entry parameters]

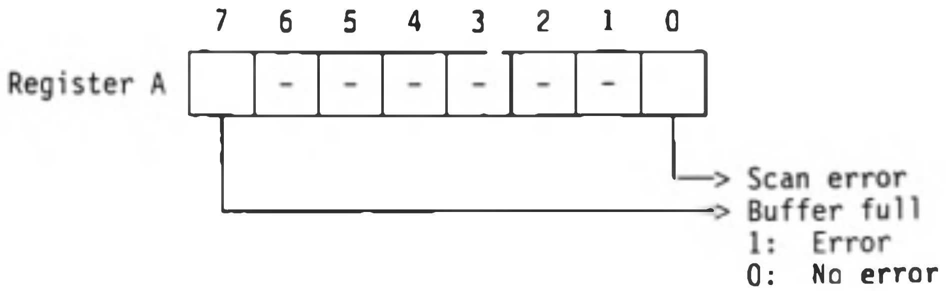
C=03H: Function number

[Return parameters]

BC=byte length of data stored in buffer
DE=size of empty area in buffer (byte length)
(BC + DE = buffer size. The buffer size is fixed to 80 bytes.)
A=error information (explained later)
 =00H: Normal termination
 ≠00H: Abnormal termination

<Explanation>

(1) Error information is indicated as follows in register A:



(2) Data in buffer is guaranteed even if A≠00H.

(3) Error information is reset once this routine is called.

<Reference>

APPENDIX 9 SAMPLE 12

[Function]

TCAM sends or receives data through public line. TCAM is the abbreviation of telecommunication access method. TCAM has he following four functions depending on the contents of register A:

A=01H: Connects to the host computer. (Connect)
=02H: Sends data to the host computer. (Send)
=03H: Receives data from the host computer. (Receive)
=04H: Disconnects from the host computer. (Disconnect)

Details on each function are explained later.

<Explanation>

(1) No protocol and Filink are the usable protocols. Further, a protocol such as BSC protocol can be extended (extended protocol).

(2) See "APPENDIX 8 FILINK PROTOCOL" for the Filink protocol.

(3) The serial line and transmission protocol are specified by system menu CONFIG.

(4) TCAM uses the following work areas:

Address:Variable name(Byte length)

F1CDH : TCAMPRM(7)

This area stores the initiation conditions used at TCAM open operation. The initiation conditions are set and modified by CONFIG but they can also be modified by an application program.

+0	Type
+1	Line
+2	Bit rate
+3	Character length
+4	Parity check
+5	Stop bit
+6	Special parameter

Type: Protocol type
=0: protocol direct-C
=1: protocol direct-B
=2: Filink (default)
=3: Extended protocol

Line: Serial line to be used for sending and receiving
=0: RS-232C (default)
=3: Cartridge I/F

The bit rate, character length, parity check, stop bit, and special parameter are same as that of BIOS RSIOX. The default values are 4800 Bps, 8 bits, Nonparity, and 2 stop bits.

F1D4H : TDFLTCNT (1)

This area is used to specify the number of retry operations executed when the Filink protocol does not match with the host protocol. (The default is 3.)

F1D5H : T1STTIME (2)
F1D7H : T2NDTIME (2)

This area is used to specify the timer used for timeout during data receiving (unit: Seconds).

T1STTIME: Timer for the first data receive operation (30 seconds)
T2NDTIME: Timer for the second data receive operation (3 seconds)

In the first data receive operation, data up to file name is received when Filink protocol is used, or 1-byte data is received when protocol is not used (Direct-B or -C).

<Reference>

Section 11.2 Extending Communication Protocol
APPENDIX 8 FILINK PROTOCOL
APPENDIX 9 SAMPLE 13

[Function]

TCAM (Connect) allows serial interrupts and connects the line to the host computer.

[Entry parameters]

A=01H: Function number
B=connection information

[Return parameters]

CY=0: Normal termination
=1: Abnormal termination
When CY=1,
A=error code
A=01H: Parameter error
=02H: Open operation had been executed.
=03H: Open operation was not executed.
=04H: Forced termination
=05H: Receive buffer overflow
=06H: Timeout
=07H: Protocol error
=08H: Communication error

<Explanation>

(1) This command allows serial communication interrupts and opens a serial line. Processing differs depending on the protocol as follows:

1 For No protocol
This command ignores the contents of register B and opens a serial line.

2 For Filink
The send or receive file operation is specified in entry parameter register B.
B=00H: Send file
=01H: Receive file
The file name must be transmitted by using TCAMAREA. See the explanation of BIOS INFORM for the TCAMAREA address. When send file is specified, set a file name in TCAMAREA and then perform the connect operation.

When receive file is specified, the name of a file to be received is stored when connect operation is normally terminated.
TCAMAREA is structured by a file name (8 bytes) and a file type (3 bytes).

3 For an extended protocol
Processing differs depending on the extended protocol. In this case, registers BC, DE, and HL can be used as parameters.

<Relations>

INFORM (WBOOT+A2H)

<Reference>

APPENDIX 9 SAMPLE 13

[Function]
TCAM (Send) sends data to the host computer.

[Entry parameters]
A=02H: Function number
BC=byte length of send data
HL=start address of send data

[Return parameters]
CY=0: Normal termination
=1: Abnormal termination
When CY=1,
A=error code (same as Connect)

<Explanation>

(1) The sent data specified by register HL is sent as much as the number of bytes specified in register BC.
BC=128(80H) is used when the Filink protocol is specified.

(2) No operation is performed when the Filink protocol is specified and "Receive File" is specified by Connect.

[Function]
TCAM (Receive) receives data sent from the host computer.

[Entry parameters]
A=03H: Function number
HL=start address of receive buffer

[Return parameters]
CY=0: Normal termination
When CY=0,
BC=byte length of receive data
CY=1: Abnormal termination
A=error code (same as Connect)

<Explanation>

(1) TCAM (Receive) stores receive data in the receive buffer specified in register HL. The byte length of receive data is set in return parameter register 3C.

(2) When No protocol or Filink protocol is specified, up to 128-byte data can be received by one Receive operation. Receive end is determined from BC=0.

(3) No operation is performed when the Filink protocol is specified and "Send File" is specified in Connect.

(4) When the Filink protocol is specified, register BC indicates 11 (0BH) at the start of receiving the second or later file to indicate that two or more files are received. In this case, a file name is stored in TCAMAREA in the same way as Open.
Note that register BC contains 128 (80H) when used as a return parameter for general data receive operation.

<Reference>

APPENDIX 9 SAMPLE 13

[Function]
TCAM (Disconnect) disconnects the line from the host computer.

[Entry parameters]
A=04H: Function number

[Return parameters]
CY=0: Normal termination
=1: Abnormal termination
When CY=1,
A=error code (same as Connect)

<Explanation>

TCAM (Disconnect) closes the serial line.

<Reference>

APPENDIX 9 SAMPLE 13

[Function]

GRAPHICS supports 12 graphics functions depending on thecontents of register C as follows:

C=00H: Initializes a graphic package. (Init)
=01H: Sets a view port. (View)
=02H: Clears the contents of view port. (Cls)
=03H: Sets a dot. (Pset)
=04H: Indicates the attribute of a dot. (Point)
=05H: Draws a line. (Line)
=06H: Draws a circle. (Circle)
=07H: Paints an enclosed area. (Paint)
=08H: Fills an enclosed area with tile pattern. (Tile)
=09H: Saves the drawing information of the specified area in the specified storage area. (Get)
=0AH: Loads the drawing information from the specified storage area to the specified area. (Put)
=0BH: Draws the specified kanji in the specified area. (Kanji)

Details on each function are explained later.

<Explanation>

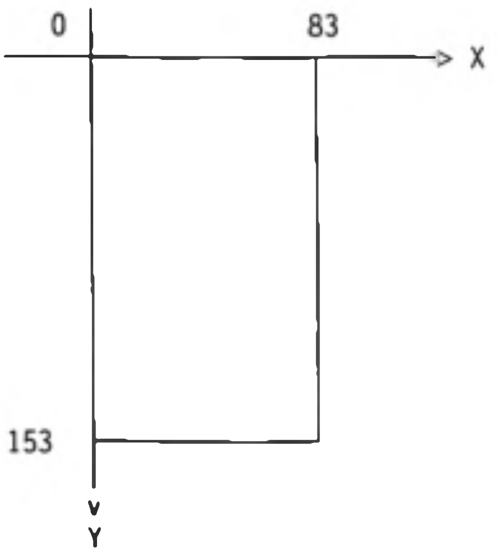
(1) GRAPHICS is a group of functions used to draw graphic images with dots and lines in LCD.

1 Coordinates

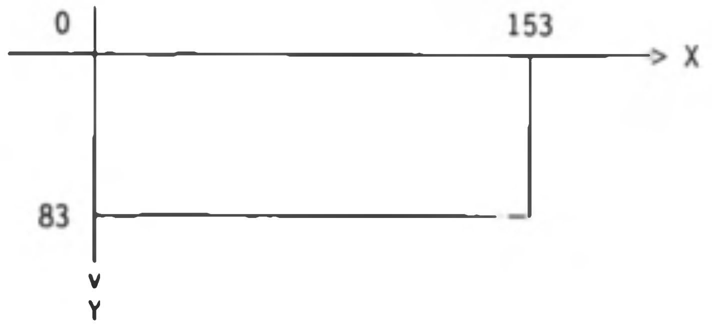
Coordinates for EHT-10 differ from that of EHT-10/2. For both EHT-10 and EHT-10/2, the origin is placed at the left top (0,0). This is same for EHT-10 horizontal display.

Fig. 4.5 Coordinates for EHT-10 and EHT-10/2

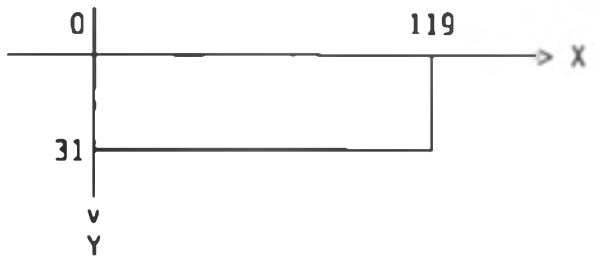
i) Coordinates for EHT-10 vertical display



ii) Coordinates for EHT-10 horizontal display



iii) Coordinates for EHT-10/2



2 View port
A view port is a defined drawing area in actual screen. Drawing by using the functions is allowed only in a view point. Coordinates are specified with X and Y. Values indicating out of a view port or exceeding the physical LCD sizes can be specified for the Pset, Line, Circle functions but drawing can only be done in a view port. Coordinates can be specified with integers from -32768 to 32767. The specified coordinates must be within a view port for the Point, Paint, Tile, Get, Put, and Kanji functions.

3 Data package
A data package is a group of data used to draw lines and circles to be displayed in LCD. The data package structure differs depending on the function. The start address of data package must be set in register HL when GRAPHICS is called.

4 Attribute
This is the dot drawing attribute and is specified with 0 or 1 as follows:
0: Dot reset
1: Dot set

5 Data format
2-byte data (such as coordinate values) used in GRAPHICS is specified in the order of low and high unless other wise stated.

(2) The cursor must be turned off before this BIOS command is used.

[Function]
GRAPHICS (Init) sets a view port containing the entire LCD. The screen is not changed.

[Entry parameters]
C=00H: Function number

[Return parameters]
A=end condition
=00H: Normal termination

<Explanation>

(1) This routine is automatically called during warm boot operation. This is one of the routines that initialize EHT-10/EHT-10/2. The user does not require this routine excepting to reset a view port containing the entire LCD.

<Relations>

WBOOT (WBOOT+0)

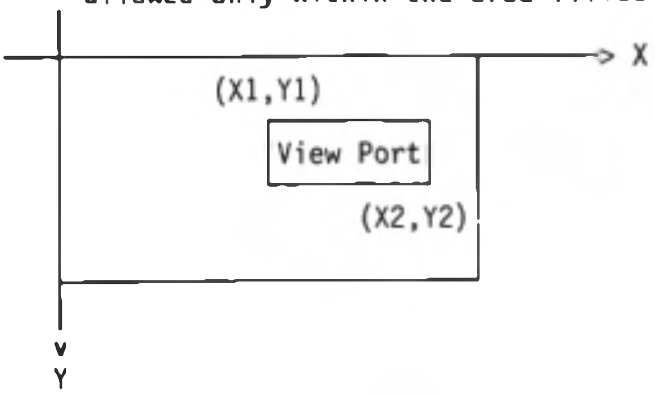
[Function]
GRAPHICS (View) specifies a view port. This can also paint the view port or draw the outer frame of the view port.

[Entry parameters]
C=01H: Function number
HL=start address of data package (explained later)

[Return parameters]
A=end condition
=00H: Normal termination
=01H: Parameter error
=02H: (X,Y) was not in LCD.

<Explanation>

(1) After this command is executed, drawing is allowed only within the view port specified by this command. In the example below, drawing is allowed only within the area filled with oblique lines.



(2) The data package is structured as follows:

	+0	+2	+4	+6	+8	+9	+10
HL ->	X1	Y1	X2	Y2	Cp	Cb	

X1: X coordinate at the left top of the view port
Y1: Y coordinate at the left top of the view port
X2: X coordinate at the right bottom of the view port
Y2: Y coordinate at the right bottom of the view port
(X1, Y1) and (X2, Y2) must be in the LCD.

Cp: Attribute used to paint the view port
00H: Dot reset
01H: Dot set
FFH: No painting]

Cb: Attribute used to draw the outer frame of the view port

    00H: Dot reset
    01H: Dot set
    FFH: No drawing

[Function]
GRAPHICS (Cls) clears the contents of view port.

[Entry parameters]
C=02H: Function number
HL=start address of data package (explained later)

[Return parameters]
A=end condition
=00H: Normal termination

<Explanation>

The data package is structured as follows:

```
     +0
HL ->| C |
```

C: Attribute
00H: Dot reset
01H: Dot set

[Function]
GRAPHICS (Pset) sets a dot at the specified coordinates.

[Entry parameters]
C=03H: Function number
HL=start address of data package (explained later)

[Entry parameters]
A=end condition
=00H: Normal termination

<Explanation>

The data package is structured as follows:

	+0	+2	+4	+5
HL ->	X	Y	C	

X: X coordinate
Y: Y coordinate
C: Attribute
00H: Dot reset
01H: Dot set

[Function]
GRAPHICS (Point) indicates the attribute of the specified coordinates.

[Entry parameters]
C=04H: Function number
HL=start address of data package (explained later)

[Return parameters]
A=end condition
=00H: Normal termination
=02H: (X, Y) was not in the view port.
C=attribute
=00H: Dot reset
=01H: Dot set

<Explanation>

The data package is structured as follows:

	+0	+2	+4
HL ->	X	Y	

X: X coordinate
Y: Y coordinate

[Function]

GRAPHICS (Line) draws a line between two specified points or draws a rectangle that has a diagonal line placed between two specified points.

[Entry parameters]

C=05H: Function number
HL=start address of data package (explained later)

[Return parameters]

A=end condition
=00H: Normal termination
=01H: Parameter error

<Explanation>

The data package is structured as follows:

	+0	+2	+4	+6	+8	+9	+10	+12
HL ->	X1	Y1	X2	Y2	C	F	P	

X1: X coordinate of drawing start point
Y1: Y coordinate of drawing start point
X2: X coordinate of drawing end point
Y2: Y coordinate of drawing end point
C: Attribute
  00H: Dot reset
  01H: Dot set
F: Drawing code
  01H: Line drawing
  02H: Rectangle drawing
  03H: Rectangle painting (P is ignored when this is specified.)

P: Line type (line pattern)
  0000H to FFFFH

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

¦<--- Left line data -->¦<--- Right line data ->¦

The attribute specified by C is set to the dot corresponding to the bit when a bit is 1. When the bit is 0, nothing is done.

<Reference>

APPENDIX 9. SAMPLE 14

[Function]

GRAPHICS (Circle) draws an ellipse or a circle according to the specified center and the specified radii on X and Y directions.
GRAPHICS (Circle) can also draw an arc or fan shape when start and end points are specified.

[Entry parameters]

C=06H: Function number
HL=start address of data package (explained later)

[Return parameters]

A=end condition
=00H: Normal termination

<Explanation>

(1) The data package is structured as follows:

	+0	+2	+4	+6	+8	+9	+10	+11	+13	+15
HL ->	CX	CY	RX	RY	C	F1	F2	TS	TE	

CX: X coordinate of center
CY: Y coordinate of center
RX: Radius in X direction
RY: Radius in Y direction
C: Attribute
00H: Dot reset
01H: Dot set
F1: Fan shape flag

7	6	5	4	3	2	1	0
	-	-	-	-	-	-	

(bit 0) -> Flag indicating whether to connect the start point and the center with a line segment.

(bit 7) -> Flag indicating whether to connect the end point and the center with a line segment.
(1:Connected 0:Not Connected)

A fan shape can be drawn when 1 is set to b0 and b7.

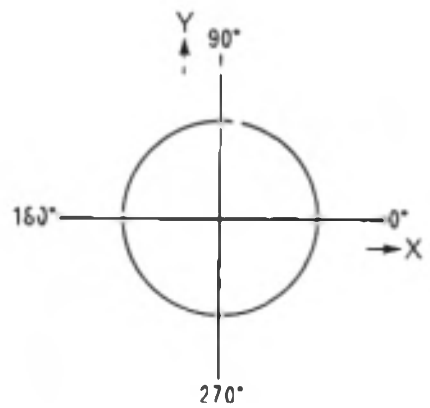
F2: This specifies what to draw when the start point is the end point.
00H: Draws a full circle.
01H: Draws the point only.
This specification is meaningless when the start point is not the end point.

TS: Start point (0000H to FFFFH)

TE: End point (0000H to FFFFH)

(2) The relationship between the start and end points and the specified values can be determined by the following expression where the angle of the start (end) point is t radian:
65536 x t/(2 x PI)
where PI is 3.141592.
(0 degree is indicated as 0000H and 180 degrees is indicated as 8000H.)

The left figure shows the angle relation between X and Y axes.



<Reference>

APPENDIX 9. SAMPLE 15

GRAPHICS (Paint) WBOOT+90H
EB93H

[Function]
GRAPHICS (Paint) paints the area enclosed in boundary color and including the specified point, with the specified color.

[Entry parameters]
C=07H: Function number
HL=start address of data package (explained later)

[Return parameters]
A=end condition
=00H: Normal termination
=01H: Parameter error
=02H: (X, Y) was not in the view point.
=03H: Insufficient work area

<Explanation>

The data package is structured as follows:

	+0	+2	+4	+5	+6	+8	+10
HL ->	X	Y	Cp	Cb	WS	WL	

X: X coordinate of paint start point
Y: Y coordinate of paint start point
Cp: Area color (attribute)
00H: Dot reset
01H: Dot set
Cb: Boundary color (attribute)
00H: Dot reset
01H: Dot set
WS: Work area start address
WL: Work area size
At least 12 bytes are required for the work area.
A larger work area is required to paint a complicated figure.
If the work area becomes insufficient during paint operation, processing is terminated and control error-returns.

[Function]

GRAPHICS (Tile) fills the area enclosed in the boundary color and including the specified point, with the specified tile pattern.

[Entry parameters]

C=08H: Function number
HL=start address of data package (explained later)

[Return parameters]

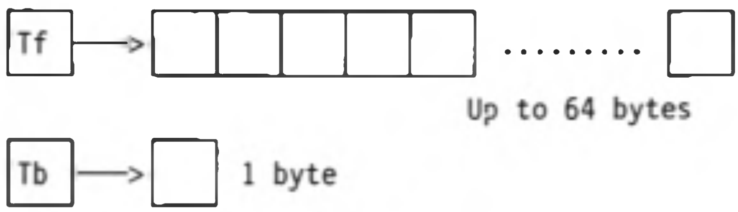
A=end condition
=00H: Normal termination
=01H: Parameter error
=02H: (X, Y) was not in the view port.
=03H: Insufficient work area

<Explanation>

(1) The data package is structured as follows:

	+0	+2	+4	+5	+7	+9	+10	+12	+14
HL ->	X	Y	L	Tf	Tb	Cb	WS	WL	

X: X coordinate of paint start point
Y: Y coordinate of paint start point
L: Length of tile pattern (1 to 64 bytes)
Tf: Tile pattern storage address
Tb: Background tile pattern storage address
Cb: Boundary color (attribute)
00H: Dot reset
01H: Dot set
WS: Work area start address
WL: Work area size



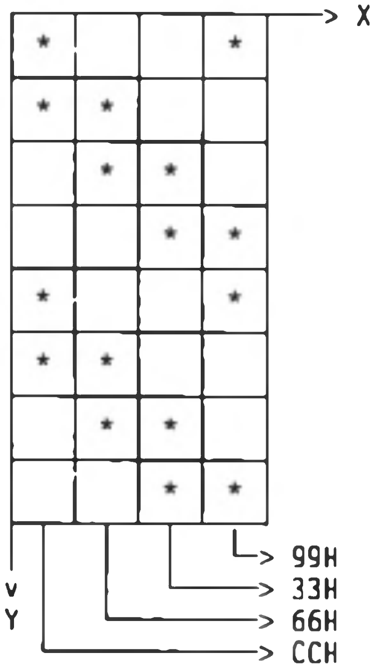
At least 12 bytes are required for the work area. A larger work area is required to paint a complicated figure. If the work area becomes insufficient during paint operation, processing is terminated and control error-returns.

(2) The tile pattern differs depending on the EHT-10/EHT-10/2 mode. In vertical display mode, 8 dots in Y direction are defined as 1 unit and units are defined as much as required in X direction. In EHT-10 horizontal display mode or EHT-10/2 mode, 8 dots in X direction are defined as 1 unit and units are defined as much as required in Y direction.

Mode	X direction	Y direction
EHT-10 Vertical	n	8 dots
EHT-10 Horizontal	8 dots	n
EHT-10/2	8 dots	n

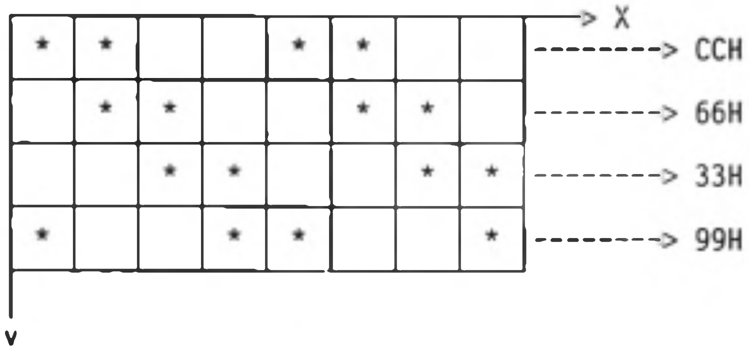
A tile pattern is specified with up to 64 bytes. Each byte contains a binary value (0: white, 1: black) in 8 dots.

Example) In EHT-10 vertical display mode



The tile pattern shown at the left figure is specified as follows:
CCH, 66H, 33H, 99H

Example) In EHT-10 horizontal mode or in EHT-10/2 mode



Y The pattern shown at the above figure is specified as follows:
CCH, 66H, 33H, 99H

<Reference>

APPENDIX 9 SAMPLE 16

[Function]

GRAPHICS (Get) saves the drawing information of the specified area in the specified storage area.

[Entry parameters]

C=90H: Function number
HL=start address of data package (explained later)

[Return parameters]

A=end condition
=00H: Normal termination
=01H: Parameter error
=02H: (X, Y) was not ir the view port.
=03H: Insufficient work area

<Explanation>

(1) The data package is structured as follows:

	+0	+2	+4	+6	+8	+10	+12
HL ->	X1	Y1	X2	Y2	WS	WL	

X1: X coordinate of the left top of specified area
Y1: Y coordinate of the left top of specified area
X2: X coordinate of the right bottom of specified area
Y2: Y coordinate of the right bottom of specified area
WS: Storage area start address
WL: Storage area size

The required size of the storage area is determined as follows:

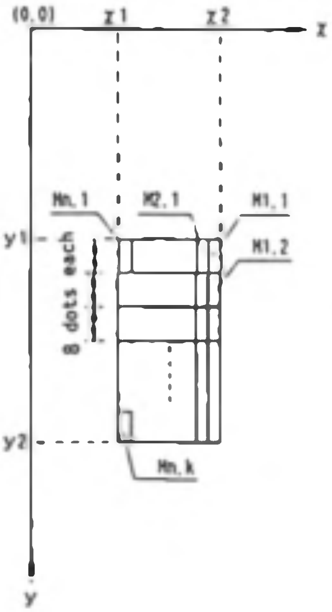
Number of required bytes =
4 + ((DY + 7) /8) * DX : EHT-10 vertical display mode
4 + ((DX + 7) /8) * DY : EHT-10 horizontal display mode or EHT-10/2

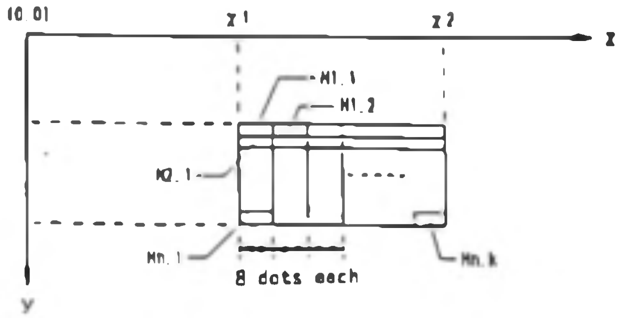
where DX = (X2 - X1) + 1
DY = (Y2 - Y1) + 1

"/" indicates division of which the decimal positions in results are truncated.

(2) Data read into the storage area by Get operation is in the following format:

1 Image in screen

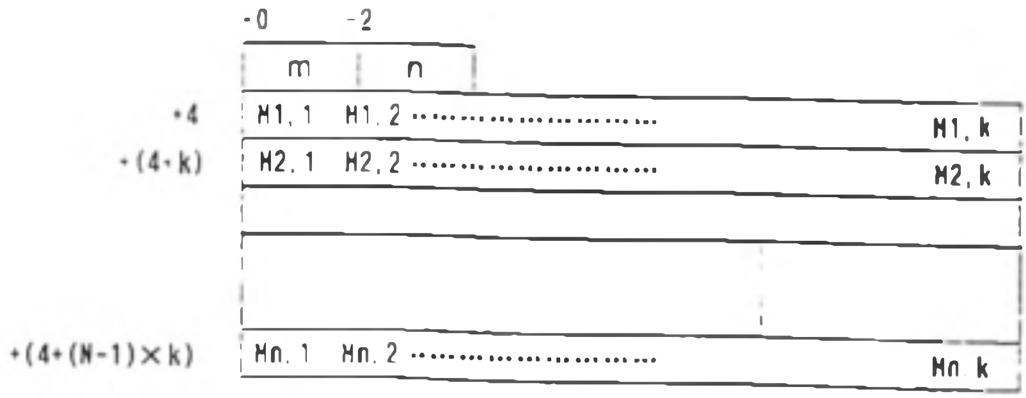




EHT-10 Horizontal display mode
EHT-10/2

EHT-10 Vertical display mode

2 Store order in storage area



A numeric value indicates the number of bytes.

where
m = DY, n = DX, k = (DY + 7)/8 (EHT-10 vertical display mode)
m = DX, n = DY, k = (DX + 7)/8 (EHT-10 horizontal display mode)
(EHT-10/2 mode)

/ indicates division of which the decimal positions in results are truncated.

[Function]

GRAPHICS (Put) loads drawing information from the specified storage area to the specified area.

[Entry parameters]

C=0AH: Function number
HL=start address of data package (explained later)

[Return parameters]

A=end condition
=00H: Normal termination
=01H: Parameter error
=02H: (X, Y) was not in the view port.

<Explanation>

(1) The data package is structured as follows:

	+0	+2	+4	+6	+7
HL ->	X1	Y1	WS	F	

X1: X coordinate of the left top of drawing area
Y1: Y coordinate of the left top of drawing area
WS: Storage area start address
F: Drawing mode
=0:PSET
1:PRESET
2:OR
3:AND
4:XOR

(2) The data format in the storage area must be equal to the format of data read into storage by Get operation.

[Function]
GRAPHICS (Kanji) draws the specified kanji in the specified area.

[Entry parameters]
C=0BH: Function number
HL=start address of data package (explained later)

[Return parameters]
A=end condition
=00H: Normal termination
=01H: Parameter error
=02H: (X, Y) was not in the view port.

<Explanation>

The data package is structured as follows:

	+0	+2	+4	+6 +7
HL ->	X	Y	C	F

X: X coordinate of the left top of drawing area
Y: Y coordinate of the left top of drawing area
C: Kanji code
See "APPENDIX 4 LIST OF KANJI CODES".
F: Drawing mode
=0 : PSET
1 : PRESET
2 : OR
3 : AND
4 : XOR

<Relations>

KANJI (WBOOT+9CH)

<Reference>

APPENDIX 9 SAMPLE 17

[Functions]
TOUCH sets and displays key blocks of touch-panel keys. This command is effective only for EHT-10. For EHT-10/2, no operation is performed and control returns.

[Entry parameters]
C=number of key blocks to be set
DE=start address of key block descripter (explained later)

[Return parameters]
None

<Explanation>

(1) If the same code is set to two or more adjoining touch-panel keys, these touch keys are assumed as one key and are called a key block. The TOUCH command sets key blocks and displays the specified key blocks in LCD.

(2) The key block descripter is structured as follows:

```
      Key block
      position | Number of keys
      |<----->|<----->|
     +0   +1   +2   +3   +4   +5   +6   +7                 +N
    ----------------------------------------------------------------------
DE->| X  | Y  | NX | NY | K  | A  |C/G| Display Data     | X  | Y  |......
    ----------------------------------------------------------------------
    |<------- Key block 1 ----------------------->|<-- Key block 2

                    K = Key code
                    A = Attribute
```

1 Key block position (X, Y)
This specifies the position of the key at the left top of the key block to be set. The values of X and Y must be as follows:
$1 \le X \le 5$, $1 \le Y \le 14$

The following table shows the coordinates of touch keys:

y \ x (1–12)	3–4	5–6	7–8	9–10	11–12
1	(1, 1)	(2, 1)	(3, 1)	(4, 1)	(5, 1)
2	(1, 2)	(2, 2)	(3, 2)	(4, 2)	(5, 2)
3	(1, 3)	(2, 3)	(3, 3)	(4, 3)	(5, 3)
4	(1, 4)	(2, 4)	(3, 4)	(4, 4)	(5, 4)
5	(1, 5)	(2, 5)	(3, 5)	(4, 5)	(5, 5)
6	(1, 6)	(2, 6)	(3, 6)	(4, 6)	(5, 6)
7	(1, 7)	(2, 7)	(3, 7)	(4, 7)	(5, 7)
8	(1, 8)	(2, 8)	(3, 8)	(4, 8)	(5, 8)
9	(1, 9)	(2, 9)	(3, 9)	(4, 9)	(5, 9)
10	(1, 10)	(2, 10)	(3, 10)	(4, 10)	(5, 10)
11	(1, 11)	(2, 11)	(3, 11)	(4, 11)	(5, 11)
12	(1, 12)	(2, 12)	(3, 12)	(4, 12)	(5, 12)
13	(1, 13)	(2, 13)	(3, 13)	(4, 13)	(5, 13)
14	(1, 14)	(2, 14)	(3, 14)	(4, 14)	(5, 14)

Key block position. The values enclosed in a pair of parentheses ( ) are the coordinates of key block.

Position where a character is displayed.

Fig. 4.6 Touch key display positions

2 Number of keys (NX, NY)
This specifies the number of keys in the X direction and the number of keys in the Y direction for the key block to be set. The total number of keys is NX multiplied by NY. The values of NX and NY must be as follows:
$1 \leq X + NX - 1 \leq 5$, $1 \leq Y + NY - 1 \leq 14$

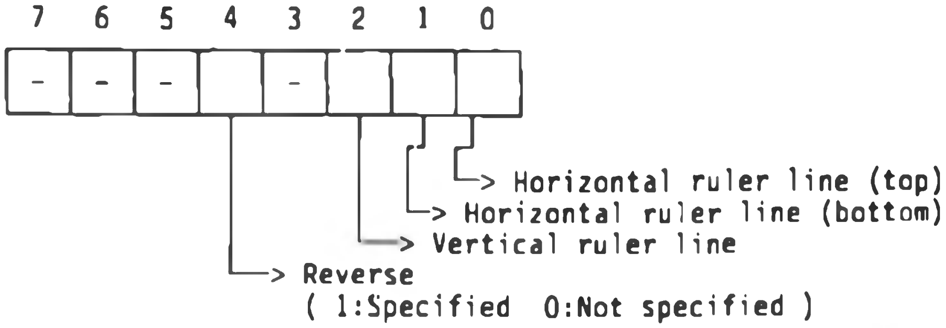
3 Key code
This specifies a key code defined to the key block. In other words, the specified key code is used in the entire key block. a value from 00H to FFH can be specified. However, fonts supported by the system in standard can be used if a value from F6H to FEH is specified.

The following fonts are supported by the system in standard:

F6H	F7H	F8H	F9H	FAH	F5H	FDH	FEH
ST OP	FE ED		000	S S Y	NO RM	AL PH	CA LC

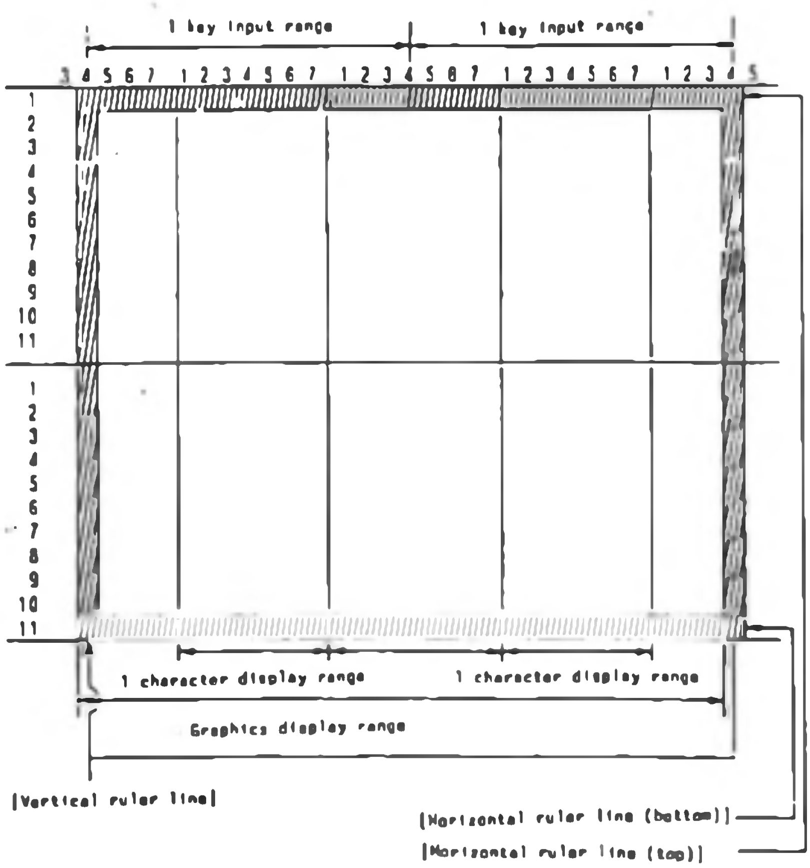
4 Attributes

This specifies the display attributes of the key block.



The attribute specification is effective for the entire block to be set. The following figure shows the display status specified by attributes in an example of key block (vertical 2 x horizontal 2):

Fig 4.7 Key block attributes



<Reverse> inverts data in the area enclosed in the vertical and horizontal ruler lines (including the frame).

5 C/G

This specifies whether to display the key block in character or graphics mode. The structure of data in the display data section differs depending on the C/G specification.

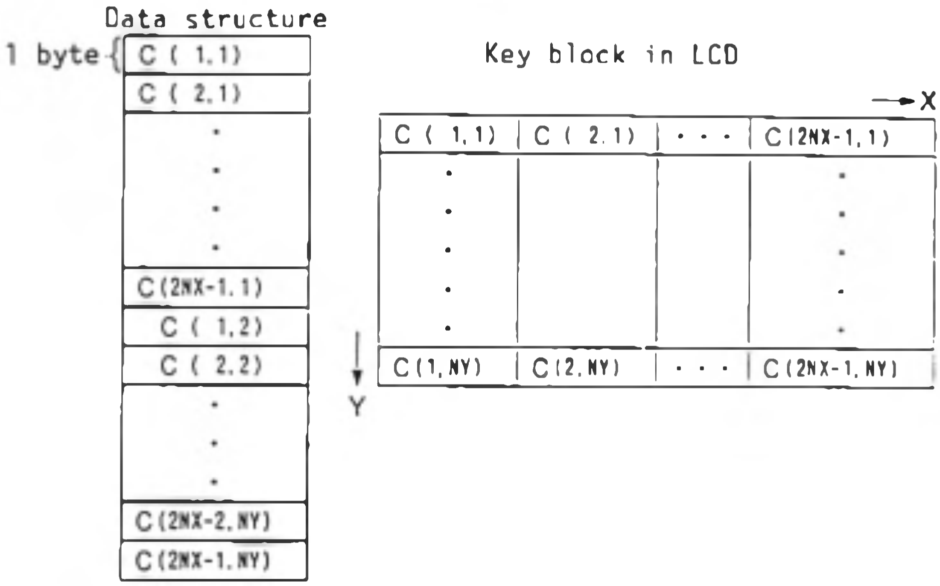
C/G=00H: Character mode
=01H: Graphics mode
=FFH: Uses the system fonts (this is effective only when the key code is a value from F6H to FEH).

6 Display data

The structure of data differs depending on the C/G specification (character mode (including using system fonts) and graphics mode).

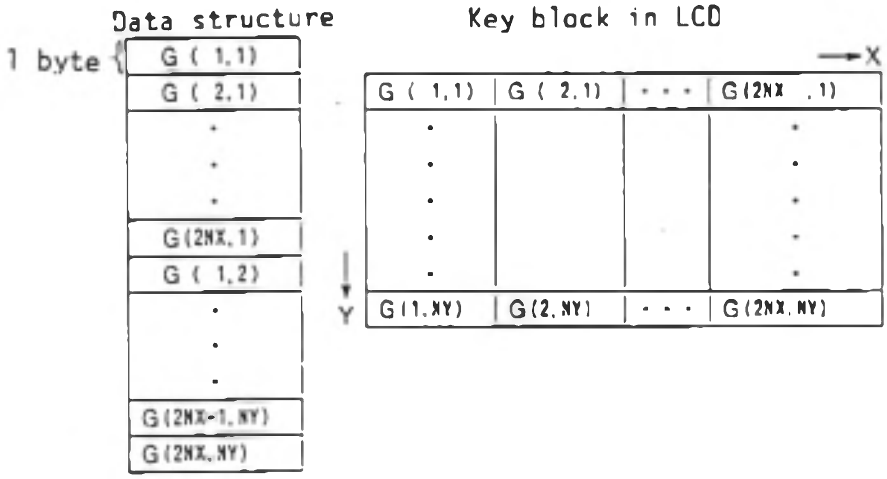
i) Character mode

Data as much as the number of bytes determined by the following expression is required when the key block size is NX multiplied by NY: NY x (2NX - 1)

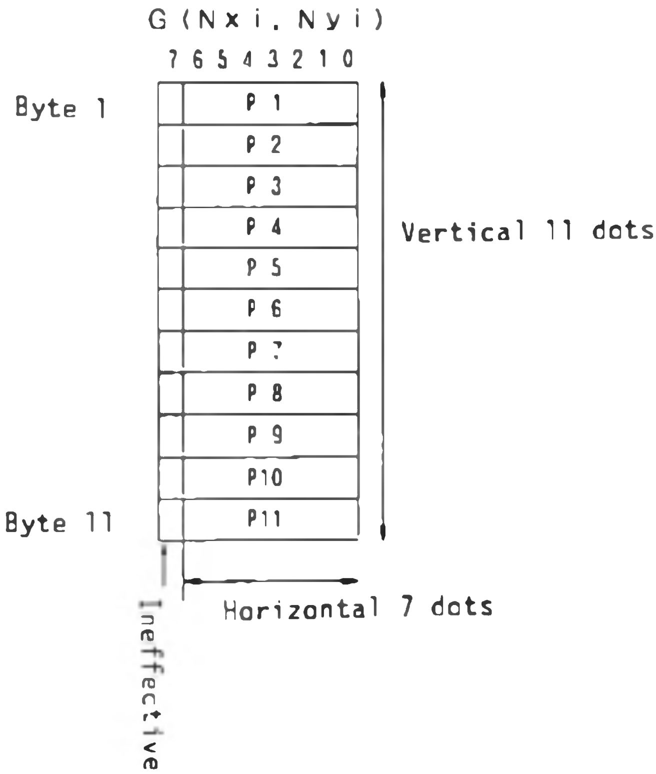


ii) Graphics mode

Graphic data as much as the number of bytes determined by the following expression is required when the key block size is NX multiplied by NY: NY x NX x 2 x 11

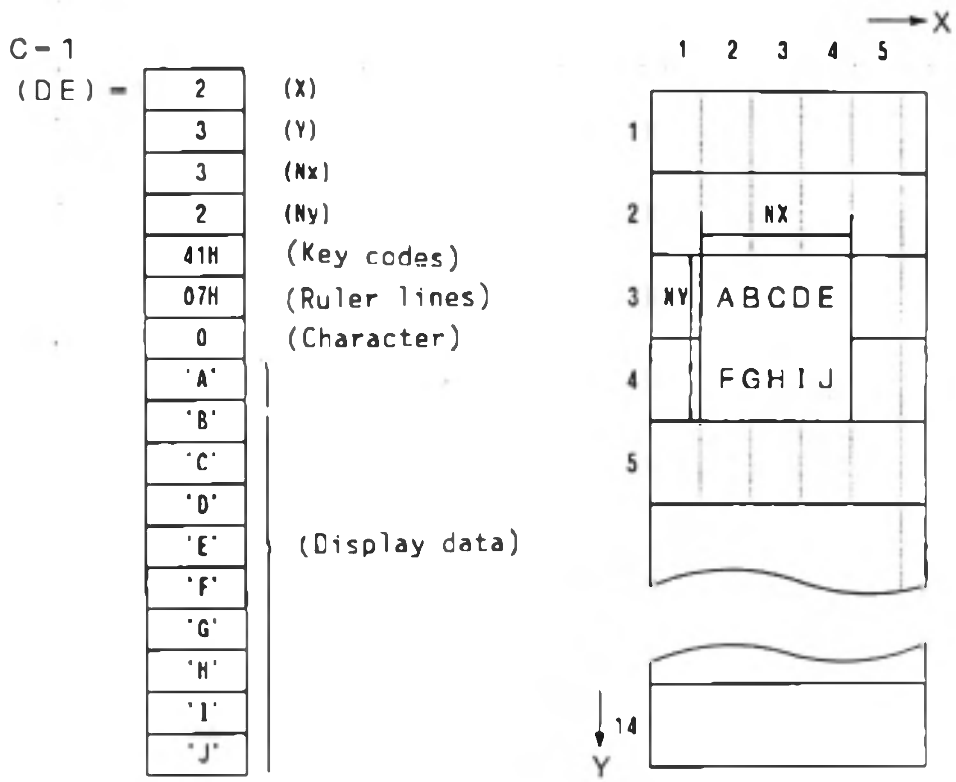


The contents of 11 bytes in key block G (Nxi, Nyi) is structured as follows:



(Example)

Data is displayed in LCD as shown in lower right when this BIOS routine is called with the following data descripter:



(3) Data is directly written in VRAM instead of the character buffer for key block display. All data written in a key block is replaced with key block display data.

(4) Data written by this BIOS routine is handled in the same way as graphic data in the view of CONOUT.

<Reference>

Section 4.3.6 Touch Panel
APPENDIX 9 SAMPLE LIST

[Function]

ICCARD inputs data from or outputs data to an IC card. ICCARD has the following five functions depending on the value in register A:

A=00H: Opens an IC card. (Open)
 =01H: Closes an IC card. (Close)
 =02H: Reads 1-character data. (Read)
 =03H: Writes 1-character data. (Write)
 =04H: Checks the status of data received from an IC card. (Status)

Details on each function are explained later.

<Explanation>

(1) Data is sent to or received from an IC card in serial communication by SIO I/F with 9600 bps, 8 bits, and even parity. The receive buffer (256 bytes) is provided by the system.

(2) Another serial interface (RS-232C) cannot be used at the same time because data is sent to or received from an IC card in serial communication by SIO I/F (terminal floppy disks can be used).

(3) The following system areas are used by ICCARD:

Address : Variable name(Byte length)

F1B1H : ICCDTIME (2)
    Timeout time used when data is received from an IC card

F1B3H : ICCDPRM (5)
    Line open parameters used when ICCARD open is specified.

F1E9H : ICTSDT (1)
    Data used for matching when IC card reset response is received.

F1AEH : ICRSTPNT (2)
    Points the stroage area of the Reset response. The default is to point ICRSTDT.

F663H : ICRSTDT (10)
    Area to store data at reset response.

<Relation>

RSIOX (WBOOT + 51H)

<Reference>

11.3 IC card protocol expansion
APPENDIX 9 SAMPLE 23

[Function]
ICCARD (Open) opens an IC card.

[Entry parameters]
A=00H: Function number

[Return parameters]
CY=0: Normal termination
=1: Abnormal termination
When CY=1,
A=error code
=01H: Parameter error
=02H: Open operation had been executed.
=06H: Timeout
=07H: Protocol error
=08H: Communication error
=0AH: Over current
=0BH: A serial device has already been used.
=0CH: The IC card was being used by disk processing.

<Explanation>

(1) The ICCARD (Open) enables sending data to and receiving data from an IC card using SIO I/F in serial communication. Open processing is executed as follows:

1 For the IC card, VCC CLK is supplied, RST is freed, and then "Answer to Reset" is received from the IC card.

2 Allows receive interrupts from the IC card.

(2) Close an IC card with the function CLOSE when any error is occurred.

<Reference>

APPENDIX 9 SAMPLE 23

[Function]
ICCARD (Close) closes an IC card.

[Entry parameters]
A=01H: Function number

[Return parameters]
None

<Explanation>

Receive interrupts sent from IC card is prohibited and the power supply to the IC card is turned off.

<Reference>

APPENDIX 9 SAMPLE 23

[Function]
ICCARD (Read) reads 1-byte data from an IC card.

[Entry parameters]
A=02H: Function number

[Return parameters]
CY=0: Normal termination
When CY=0,
A=read data
CY=1: Abnormal termination
When CY=1,
A=error code
=01H: Parameter error
=03H: Open operation was not executed.
=04H: Forced termination
=05H: Receive buffer overflow
=06H: Timeout
=08H: Communication error
=09H: Power off end
=0AH: Over current
=0BH: A serial device has already been used.

<Explanation>

(1) 1-byte data received from an IC card and stored in receive buffer is read.

(2) If data is not received for a fixed time (default is 3 seconds), a timeout error occurs and control returns.

(3) The communication error code (08H) is set for subsequent data if a communication error occurs in receive data.

<Reference>

APPENDIX 9 SAMPLE 23

[Function]
ICCARD (Write) writes 1-byte data to an IC card.

[Entry parameters]
A=03H: Function number
C=write data

[Return parameters]
CY=0: Normal termination
=1: Abnormal termination
When CY=1,
A=error code
=01H: Parameter error
=03H: Open operation was not executed.
=09H: Power off end
=0AH: Over current

<Explanation>

ICCARD (Write) sends data specified in register C to the IC card.

<Reference>

APPENDIX 9 SAMPLE 23

[Function]
ICCARD (Status) checks the status of data received from an IC card.

[Entry parameters]
A=04H: Function number

[Return parameters]
CY=0: Normal termination
When CY=0,
A=status
=FFH: There is receive data.
=00H: There is not receive data.
BC=byte length of receive data
CY=1: Abnormal termination
When CY=1,
A=error code
=01H: Parameter error
=03H: Open operation was not executed.
=09H: Power off end
=0AH: Over current

<Explanation>

(1) ICCARD (Status) checks whether there is data received from an IC card, and indicates the byte length of data stored in the receive buffer when there is data received from the IC card.

(2) To continue processing after control returns due to a communication error, insert the following operations to reset the flag:

```
RSPSTS  EQU  F604H
        LD   A,(RSPSTS)
        AND  10001011B
        LD   (RSPSTS),A
```

<Reference>

APPENDIX 9 SAMPLE23

[Function]

KEYIN initiates the system input function and receives entered data. KEYIN has the following four functions depending on the contents of register B.

B=00H: Initiates the calculator. (Calc)
=01H: Initiates the alphabet input function. (Alph)
=02H: Initiates the kana input function. (KANA)
=03H: Initiates the normal input function. (Norm)

Details on each function are explained later.

<Explanation>

This command initiates the calculator, alphabet, kana, or normal input function supported by the system in standard.

The calculator can perform simple arithmetic operations and input the results of arithmetic operations. The alphabet function can input alphanumeric (EHT-10) or alphabet (EHT-10/2) characters.

The kana function can input kana characters. The normal input function does not perform anything but enables normal input operations for EHT-10/2.

<Reference>

APPENDIX 9 SAMPLE 2

[Function]
KEYIN (Calc) initiates the calculator and enables arithmetic operations and inputting the results of arithmetic operations.

[Entry parameters]
B=00H: Function number
C=data format of arithmetic operation results
 =00H: BCD format
 =FFH: ASCII format
HL=start address of area used to store arithmetic operation result

[Return parameters]
A=byte length
(HL)=result of arithmetic operation

[Saved registers]
HL

<Explanation>

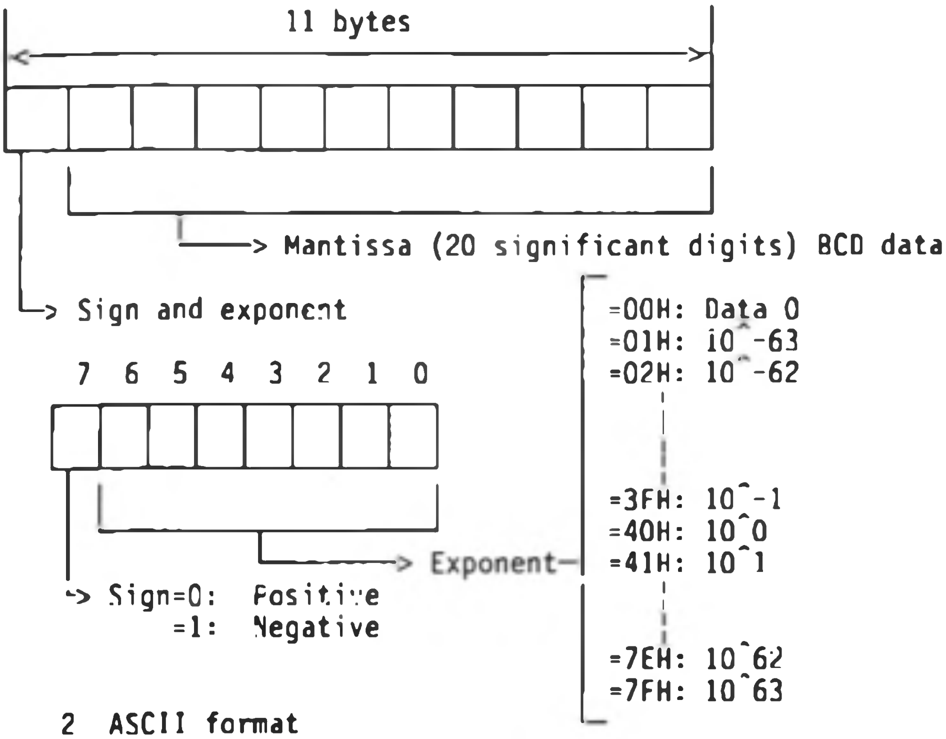
(1) KEYIN (Calc) displays the calculator screen to perform arithmetic operations. The result of arithmetic operation is set when the SAVE or QUIT key is pressed. When control is returned by the SAVE key, the arithmetic operation result held before return operation is set in the result storage area. When control is returned by the QUIT key, register A contains 00H. The BCD or ASCII format can be specified for the arithmetic operation results.

(2) The size of result storage area differs depending on the result data format as follows:
 11 bytes for BCD format
 13 bytes for EHT-10 ASCII format
 21 bytes for EHT-10/2 ASCII format

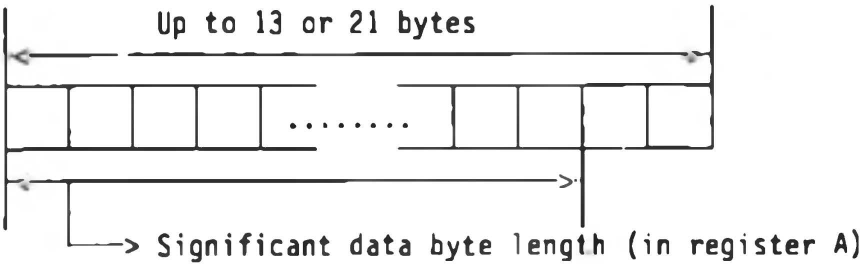
(3) The byte length is set in a return parameter (register A) as follows:
 When control is returned by the SAVE key,
  A=11 for BCD format
  A=1 to 13 for EHT-10 ASCII format
  A=1 to 21 for EHT-10/2 ASCII format
 When control is returned by the QUIT key,
  A=0

(4) The following formats are used for arithmetic operation results:

1 BCD format



2 ASCII format



CR (0DH) is added at the end of data for the ASCII format.

Example of -4.321 BCD data and ASCII data

	1	2	3	4	5	6	7	8	9	10	11
BCD	C1H	43H	21H	00H	00H	00H	00H	00H	00H	00H	00H

ASCII	'-'	'4'	'.'	'3'	'2'	'1'	0DH	A=07H

[Function]
KEYIN (Alph) initiates the alphanumeric (EHT-10) or alphabet (EHT-10/2) input function. Processing differs depending on EHT-10 or EHT-10/2 (details are explained later).

[Entry parameters]
B=01H: Function number
HL=start address of input alphanumeric character storage area
(This is not required for EHT-10/2.)

[Return parameters]
A=byte length (0 for EHT-10/2)
(HL)=input alphanumeric characters (only for EHT-10)

[Saved registers]
HL

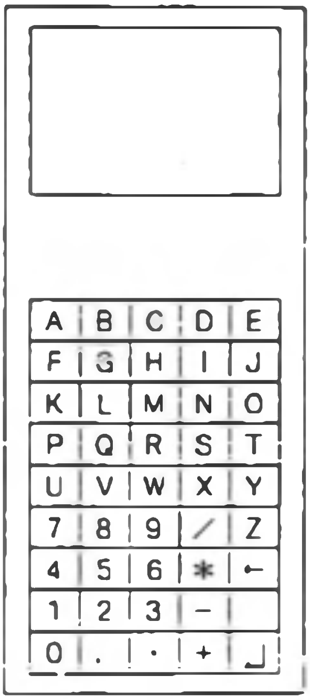
<Explanation>

(1) Differences between EHT-10 and EHT-10/2 processing

1 EHT-10

The current screen status is saved and alphanumeric input screen (shown in the next page) is displayed. At the top three lines of the screen, display data is copied so that the cursor position held at alphabet specification is placed at line 3. (However, data is copied so that the cursor position is in line 1 or 2 when the cursor position is in line 1 or 2 of the wind.)

Fig. 4.8 EHT-10 alphanumeric input screen



Input keys are displayed sequentially from the current cursor position. Up to 31 characters can be input.

'Return'key: This key ends input operation and stores input data in the buffer specified by register HL. CR (0DH) is added at the end of data. The number (1 to 32) of input keys is set in register A.

'<-' key: This key is used to correct input data. This key deletes the latest column and shifts the cursor one column back.

When this BIOS routine is called in EHT-10 horizontal display mode, the top three lines are cleared with blanks and the cursor is moved to the home position. Display and operations are equivalent to vertical display mode.

2 EHT-10/2

This command makes the keyboard enter alphabet mode and turns on the alphabet LED. The display screen does not change. The input alphabet character storage area is not required because this command only modifies the mode. 0 is set in register A (return parameter)

[Function]
KEYIN (KANA) initiates the kana input function. Processing differs depending on EHT-10 and EHT-10/2. This routine can be called only when Japan is specified by the DIP switch.

[Entry parameters]
B=02H: Function number
HL=start address of input kana character storage area (This is not required for EHT-10/2.)

[Return parameters]
A=byte length (0 for EHT-10/2)
(HL)=input kana character (only for EHT-10)

[Saved registers]
HL

<Explanation>

(1) Differences between EHT-10 and EHT-10/2 processing

1 EHT-10

The kana input screen shown in the next page is displayed. The screen is displayed in the same way as Alph.

Fig. 4.9 EHT-10 kana character input screen

2 EHT-10/2

This command makes the keyboard enter kana mode and turns on the kana LED. The display screen does not change. The input kana character storage area is not required because this command only modifies the mode. 0 is set in register A (return parameter). Data is input in Roman letters and converted from Roman letters.

<Reference>

4.3.7. Keyboard

[Function]
KEYIN (Norm) makes the keyboard enter normal mode and turns on the normal LED. The screen does not change. This command is effective only for EHT-10/2. No operation is executed for EHT-10.

[Entry parameters]
B=03H: Function number

[Return parameters]
A=00H

[Saved registers]
HL

[Function]

KANJI creates a gaiji (external character) or displays and prints a kanji. This command has the three functions depending on the contents of register B as follows:

B=00H: Specifies a gaiji file. (Gaiji)
=01H: Displays a kanji (gaiji). (Disp)
=02H: Prints a kanji (gaiji). (Print)

Details on each function are explained later.

<Explanation>

(1) Kanji codes indicate the place where the font is stored as follows:

Kanji codes

0000H to 00ADH: Font (non kanji) in system ROM
03E8H to 07CBH: Font (kanji) in system ROM
0800H to 1FFFH: Font in gaiji (external character) file
2000H to 4FFFH: Font in JIS level-1 ROM

1 Use a gaiji file that was specified by the Gaiji function.
2 The JIS level-1 ROM can be inserted to the application ROM drive of EHT-10/EHT-10/2. The code is same as the JIS code.
3 All kanji codes must be specified in the order of low and high.

(2) All the Kanji and Gaiji characters are made by 16 * 16 dots.

(3) The cursor must be turned off before this BIOS routine is called.

<Relations>

GRAPHICS (Kanji) (WBOOT+90H)

<Reference>

APPENDIX 9 SAMPLE 17

[Function]
KANJI (Gaiji) specifies a gaiji file.

[Entry parameters]
B=00H: Function number
HL=start address of the file name (explained later)

[Return parameters]
None

<Explanation>

(1) A gaiji is a font designed by the user and is accessed in a file. If a gaiji file is specified, the font fetched from the file can be displayed in LCD or output to the printer. In order to display or print a gaiji, this routine must be called to specify a gaiji file in advance.

(2) The file name is structured as follows:

```
      +0   +1                                 +9           +12
HL -> | Dr | File  Name                       | Extension  |
```

Dr: Drive name

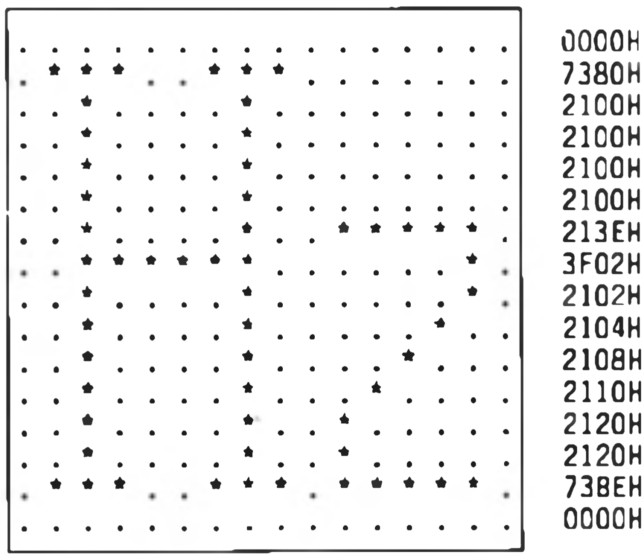
(3) Format of gaiji file
The font in a gaiji file is structured with 32 bytes/character.
The first character corresponds to kanji character code 800H, the second character corresponds to 801H, the third character corresponds to 802H, and so on.

```
| Code 800H | ..... 32 bytes
| Code 801H | ..... 32 bytes
| Code 802H | ..... 32 bytes
| Code 803H | ..... 32 bytes
|     |     |
```

The font structure of one character is horizontally scanned. In the example shown below, one character must be stored in the order of 00, 00, 73, E0,...

Fig. 4.10 Gaiji file structure

(Example)



[Function]
KANJI (Disp) displays kanji (gaiji) in LCD.

[Entry parameters]
B=01H: Function number
HL=start address of data package (explained later)

[Return parameters]
None

<Explanation>

The data package is structured as follows:

	+0	+1	+2	+3				
HL ->	X	Y	N	Code 1	Code 2	......	Code N

Kanji (gaiji) characters are displayed as much as N starting from the specified coordinates (X, Y) (dot position) toward right in the order of kanji codes 1, 2, 3,...

<Reference>

APPENDIX 9. SAMPLE 17

[Function]
KANJI (Print) prints kanji (gaiji).

[Entry parameters]
B=02H: Function number
HL=start address of data package (explained later)

[Return parameters]
None

<Explanation>

(1) The data package is structured as follows:

	+0	+1	+2	+3			
HL ->	X	Y	N	Code 1	Code 2	......	Code N

Kanji (gaiji) characters are printed starting from the specified position X toward right in the order of kanji codes 1, 2,... Position Y is ignored even if specified. Position X must be a multiple of 8. If it is not a multiple of 8, the largest multiple of 8 but less than the specified value is used.

(2) The maximum value of N is 9 for a printer.
Value N differs depending on the number of printer columns for a printer connected to RS-232C. However, N must satisfy the following expression:
Number of dots that can be printed starting from the current head position $\geq 8 \times X + 16 \times N$

<Reference>

Section 4.5 Printer
APPENDIX 9 SAMPLE 17

[Function]
BACKLIGHT performs the backlight software control. This command is effective only for EHT-10/2B. No operation is executed for EHT-10 and EHT-10/2.

[Entry parameters]
C=backlight control
=00H: Backlight off
=01H: Backlight on

[Return parameters]
None

<Explanation>

(1) The following table shows the relationship between EHT-10/2B power switch and backlight:

Switch status	1	2	3
Power status	OFF	ON	ON
Backlight	X	X	0

0 indicates that software control is allowed.
X indicates that software control is not allowed.

(2) System default is backlight on.

(3) The system supports automatic backlight off. Backlight is automatically turned off when data is not input by key operation for a fixed time (default is 3 minutes) in key input wait status. See "Section 7.8.3 Backlight" for details.

<Reference>

APPENDIX 9 SAMPLE 10

[Function]
INFORM checks the address of work area used by the system.

[Entry parameters]
C=system information code (explained later)

[Return parameters]
HL=address of system information

<Explanation>

When this routine is called with register C containing one of the following system information codes, this routine sets the address of the system information in HL register:

System information code	Name	Contents
0	BIOSERROR	Error information for disk access operations. See BIOS READ.
1	BPINTEBL	Interrupt status for BEEP processing See BIOS BEEP.
2	CONFMOD	Specification of whether CONFIG is allowed to set each item. See "Section 9.4 System Menu Functions"
3	DISBNK	Bank information for bank switching
4	FUNC_TBL	Subroutine call table for key input operations. See BIOS CONIN.
5	LSTERR	Error information for printer output See BIOS LIST.
6	SKEYFLG	Specification of whether to suppress or allow calling a subroutine by key input operations
7	SRCBNK	Bank information for bank switching
8	SYSMMOD	Specification of whether to suppress or allow setting each item by system menu operations
9	TCAMAREA	Area used to pass the parameters for communication. See BIOS TCAM.

<Relations>

```
CONIN (WBOOT+06H)
LIST  (WBOOT+0CH)
READ  (WBOOT+24H)
BEEP  (WBOOT+36H)
TCAM  (WBOOT+BDH)
```

<Reference>

Section 9.4 System Menu Functions
APPENDIX 9 SAMPLE LIST

4.3 Entering Data with Touch Panel or Keyboard Keys

4.3.1 Overview

Data is input for EHT-10 and EHT-10/2 by using the touch panel and keyboard, respectively. Both the touch panel and keyboard are controlled by the slave 7508 CPU when commands and data are exchanged with the main CPU through serial communications.

(1) Entering data with touch-panel keys

1 The 7508 CPU checks at every 30 msec whether any key has been pressed and, if one or more keys have been pressed, sets 80H in the key buffer of the 7508 CPU and sends an interrupt signal to the main CPU.

2 During interrupt processing, the main CPU scans the touch panel and generates a position code only for the first key that it scanned. The main CPU then generates a function code from the position code and the key state, and stacks the position and function codes in the key buffer.

3 The stacked key buffer data items are fetched and processed one by one by BIOS functions CONIN, CONST, and KEYIN.

(2) Entering data with EHT-10/2 keyboard keys

1 The 7508 CPU checks all keys at every 30 msec and , if one or more keys have been pressed, stacks the corresponding scan codes in the key buffer of the 7508 CPU and sends an interrupt signal to the main CPU.

2 During interrupt processing, the main CPU receives the scan codes from main CPU, generates position and function codes, and stacks them in the key buffer.

3 The stacked key buffer data items are fetched and processed one by one by BIOS functions CONIN and CONST.

(3) 7508 CPU functions for controlling data input through keys

The 7508 CPU has the functions listed below to control data input through keys.

See Section 7.6.3 for 7508 CPU commands.

1 When a key is pressed
In EHT-10, 80H is output as a scan code. (The actual scan code is generated during OS interrupt processing.) In EHT-10/2, the matrix position is output as the scan code (see Figure 4.11).

2 When two or more keys are pressed sequentially in EHT-10/2, corresponding scan codes are output sequentially.

3 When two or more keys are pressed simultaneously, only the code of the key scanned first is output.

4 All keys are scanned once in approximately every 30 msec.

5 The automatic repeat function is supported.
The prohibition or permission of the automatic repeat function can be specified in a command or the BIOS CONOUT ESC sequence (the default is prohibition). The automatic repeat start time and repeat interval can be modified by a command.

6 The prohibition or permission of key interrupt can be specified in a command or BIOS MASKI.

7 A seven-byte area is allocated as a key input buffer. Seven bytes or one byte can be specified as the size of this key input buffer (the default is seven bytes).

8 The keys pressed after the key input buffer becomes full of data are ignored.

4.3.2 Functions supported by BIOS

The following BIOS functions are supported.

(1) CONST : Checks the CON: input status.
(2) CONIN : Inputs one character from the CON: device.
(3) READER: Inputs one character from the RDR: device.
(4) GETPFK: Reads the key code currently set.
(5) PUTPFK: Registers a key code in the user table.
(6) TOUCH : Sets the indication for setting the key block of the touch-panel keys (only in EHT-10).
(7) KEYIN : Initiates the input function that the system has, and receives data.

See Section 4.2 for details on the BIOS functions.

4.3.3 Position and Function Codes

(1) Position and function codes

EHT-10 and EHT-10/2 use two types of key codes: position codes and function codes.

A position code is a physical code which indicates the position where the key was pressed. This position code is set in the C register when BIOS function CONIN is executed. Figure 4.11 shows the relationship between key positions and position codes.

A function code is a logical value assigned to a position code. This function code is set in the C register when BIOS function CONIN is executed. Function codes can also be set by BIOS function PUTPFK or TOUCH (only in EHT-10).
Table 4.4 lists the functions of function codes.

01	02	03	04	05

06	07	08	09	0A
0B	0C	0D	0E	0F
10	11	12	13	14
15	16	17	18	19
1A	1B	1C	1D	1E
1F	20	21	22	

EHT-10/2 keyboard

Note : Values are represented in hexadecimal notation.

01	02	03	04	05
06	07	08	09	0A
0B	0C	0D	0E	0F
10	11	12	13	14
15	16	17	18	19
1A	1B	1C	1D	1E
1F	20	21	22	23
24	25	26	27	28
29	2A	2B	2C	2D
2E	2F	30	31	32
33	34	35	36	37
38	39	3A	3B	3C
3D	3E	3F	40	41
42	43	44	45	46

EHT-10 touch-panel keys

Fig. 4.11 Position code array

Table 4.4 Functions of function codes

Function code (hex)	Function
00 - DF	Returns the specified function code to the user without performing any processing.
E0 - EF	Calls a user function. The default function is only to return control. This function also enables an application program to modify the subroutine call address and add operations.
F0 - F5	Calls a system function. The default function is only to return control.
F6	I/O forcible termination: If this function key is pressed, the I/O operation being performed can be suspended. The following I/O operations can be suspended: 1. Waiting for sending or receiving through an RS-232C interface 2. Waiting for printer ready status 3. Beep processing

F7	Paper feed (FEED): Feeds paper of the printer.
F8	Print (COPY): Outputs the data being displayed on the screen to printer as bit image data. In EHT-10, this function code performs no processing.
F9	000: If this function key is pressed, code 0 (30H) is returned three times consecutively.
FA	System MENU: If this function key is pressed, the system menu screen is initiated.
FB	Normal table (NORMAL): If this function key is pressed, the key status is set to normal mode.
FC	Kana table: If this function key is pressed, the keys status is set to kana mode. If overseas mode has been specified, this function key performs no processing. (In EHT-10, string data is input.)
FD	Alphabetic character table (ALPH): If this function key is pressed, the key status is set to letter (alphabetic character) mode. (In EHT-10, string data is input.)
FE	Calculator (CALC): If this function key is pressed, the calculator is initiated.
FF	Invalid: If this function code is defined, the key is made ineffective and, even if the key is pressed, no operation is performed.

(2) Subroutine call system

The subroutine call system has been incorporated in function codes E0H to FEH so that a specific service program can be executed by pressing the corresponding key.
Table 4.4 lists the system default functions.
The called subroutine is executed when BIOS function CONIN or CONST is called.

To execute a specific user service program, set the bank information and address of the service program in the address of the corresponding function code shown in the function table FUNC_TBL) listed in Table 4.5.

Note the following when registering user service programs:

1 To use a stack frequently, set a new stack. In this case,return the stack to its original state when control is returned.

2 BIOS and BDOS cannot be called during service program execution. If required, directly call the module stored in ROM.

Notes:

A function table consists of three bytes: one byte for bank information and two bytes for address information.
The top address of the table is determined by FUNC_TBL of BIOS INFORM.

Table 4.5 Function table

FUNC_TBL --> (F45AH)

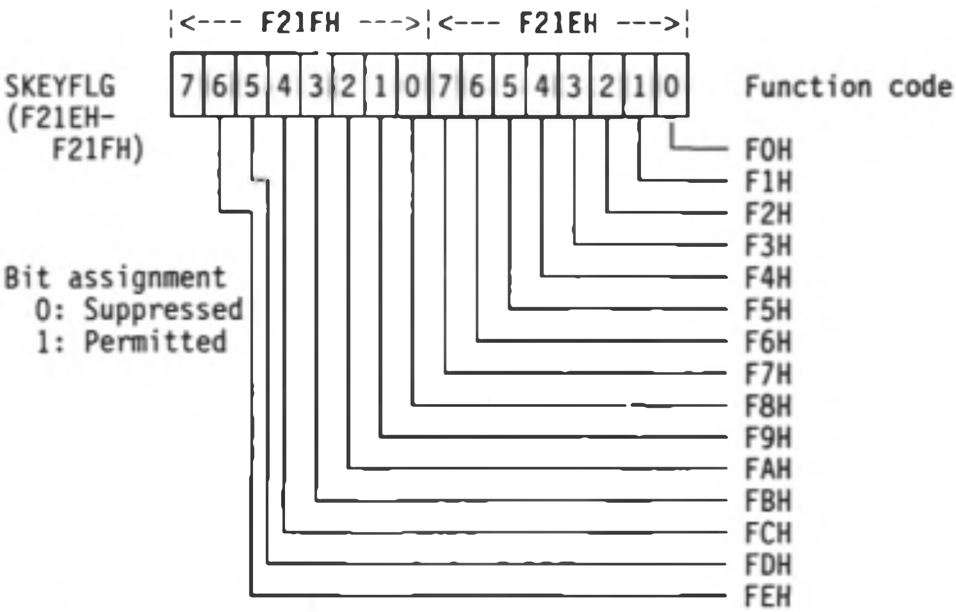
Function code (HEX)	Relative position from the table		Remarks
	Bank	Address	
E0	+0	+1 ,+2	
E1	3	4 , 5	
E2	6	7 , 8	
E3	9	10,11	Default is return
E4	12	13,14	
E5	15	16,17	
E6	18	19,20	
E7	21	21,22	
E8	23	24,25	
⋮	⋮	⋮	
EE	42	43,44	
EF	45	46,47	
F0	48	49,50	
F1	51	52,53	
F2	54	55,56	Default is return
F3	57	58,59	
F4	60	61,62	
F5	63	64,65	
F6	66	67,68	I/O forcible termination
F7	69	70,71	Paper feed
F8	72	73,74	Print
F9	75	76,77	000
FA	78	79,80	System menu
FB	81	82,83	Normal table
FC	84	85,86	Kana table
FD	87	88,89	Letter table
FE	90	91,92	Calculator

(3) Function code check mode

By setting the function check mode flag (YPFCMFLG: F21DH) to FFH, subroutine calling is not performed for function codes E0H to FEH and function codes can be returned by CONIN.

(4) Suppressing system key functions

The functions of function codes F0 to FEH can be suppressed by setting the corresponding system key flag (SKEYFLG) bits to 0. If suppression is set it is assumed, unlike the function key check mode, that the key has not been pressed.



:.3.4 Key buffer

In EHT-10 and EHT-10/2, the slave 7508 CPU has a key buffer and the main CPU has another key buffer. The main CPU also has a special buffer for inputting string data.

(1) 7508 CPU key buffer

Seven characters (default) or one character can be specified. Output a command directly for the 7508 CPU when making specification. (See Section 7.6.3 for details.)

Key scan codes are stacked in the 7508 CPU key buffer, and these codes are fetched one by one by interrupt processing performed by the main CPU. In EHT-10, however, the scan code is fixed to 80H.

(2) Main CPU key buffer

A 66-byte area capable of buffering the codes for 32 keys is allocated as the key interrupt buffer.

Two bytes are used for one key: one byte for a position code and one byte for a function code. The key buffer constitutes the ring structure.

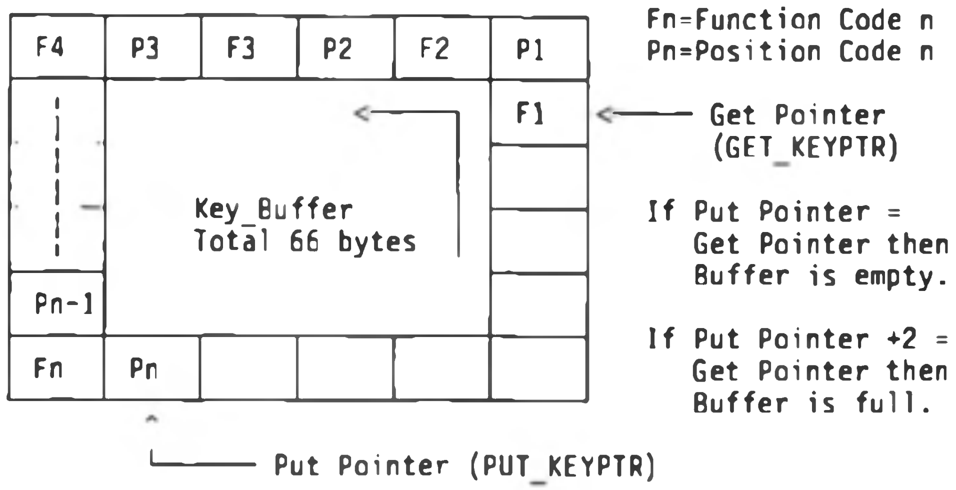


Fig. 4.12 Key buffer structure

Address : Variable name (Number of bytes)

F7E0H : KEY_BUFFER (66)
Key input buffer

F22DH : PUT_KEYPTR (2)
Key buffer put pointer
The key codes obtained by key interrupt are stored in the order of a function code and a position code. This pointer is pre-incremented by key interrupt so that it always points the head of the empty area.

F22FH : GET_KEYPTR (2)
Key buffer get pointer
This pointer is used to fetch key codes. This pointer is post-incremented by the key input routine so that it always points the next data to be fetched. No input data is assumed to be left when PUT_KEYPTR=GET_KEYPTR.

(3) String input buffer

A 32-byte area is allocated as the string input buffer used to store input data when the calculator function or the function of input in letter or kana mode (only in EHT-10) is called.

The data stored in the string input buffer is fetched one character at a time by BIOS COIN. When data resides in the string input buffer, the string input buffer data is always fetched first even if data also resides in the key buffer.

Position code FFH is always returned for string input.

Address : Variable name (Number of bytes)

F434H : YPFKBUF (32)
String input buffer

F231H : YPFKPTR (2)
Pointer which points the head address of the data .
This pointer is incremented whenever a character is fetched.

F229H : YPFKCNT (2)
Counter which indicates the data length .
This counter is decremented whenever a character is fetched.

4.3.5 Key input beep

When pressing a key is accepted, key input beep of 1024 Hz sounds for approximately 100 ms.

Output, nonoutput, frequency, and the sounding time of the key input beep can be specified or modified by directly modifying the system area contents.

If an external interrupt is suppressed while a key input beep is sounding, the sounding time is made longer because an external interrupt continues for 8 ms.

Address : Variable name (Number of bytes)

F094H : BUZ_FLG (1)
Specifies output or nonoutput of key input beep.
0: Nonoutput
Not 0: Output (default)
The default value is set when reset is performed.

F095H : KDFLTBZ (2)
Specifies the key input beep sounding time in units of milliseconds.
The default value is set when reset is performed.

F097H : BUZZHZ (1)
Specifies the beep frequency.
C0H: 2048 Hz
80H: 1024 Hz (default)
40H: 512 Hz
00H: Nonoutput
The default value is set when reset is performed.

4.3.6 Touch panel (EHT-10)

(1) Overview

In EHT-10, data is input from the transparent lattice-shaped keyboard consisting of five horizontal and 14 vertical grids (input points) mapped on the LCD screen. This keyboard is called a touch-panel keyboard.

An application program assigns keys (see the explanation of BIOS TOUCH for function key codes and key displaying) to these touch grids and use them. By using this keyboard, data can be input at the very position where the entered data is displayed, and which enables the users to create applications flexible to their needs. EHT-10 supports letter and kana modes so that alphanumeric and kana characters can be input.

EHT-10 also supports calculator mode so that simple four fundamental arithmetic operations can be performed by the application program. See Section 9.5 for calculator mode.

(2) Positional relationship between LCD display areas and touch panel

A total of 70 switches (five horizontal and 14 vertical switches) have been set on the 12-column x 14-line LCD display possible area.

The size of one touch grid (name of one rectangular part) is the same as that of an area for displaying two characters, and a character is displayed at the center of the touch grid.

Figure 4.13 shows the positional relationship between the display areas and the touch panel.

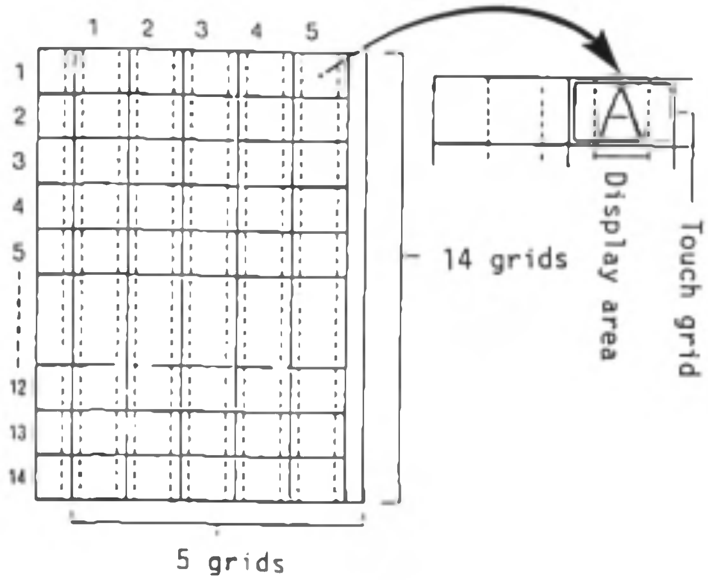


Fig. 4.13 Positional relationship between display areas and touch panel

(3) Normal mode

1 Overview

Normal mode is a standard mode in which the user can use the keyboard by defining arbitrary function codes for position codes. (See the explanation of BIOS functions TOUCH and PUTPFK.)

2 Setting conditions

The key status is set to normal mode under the following conditions:

(a) When the application program is initiated (including initiation by Menu. DLL) immediately after warm boot or restart power on In this case, the user must define function codes because all function codes are defined as invalid keys (FFH). Also, no data is displayed on the LCD screen.

(b) When letter, kana, or calculator mode is released
When the normal mode is restored from one of the above modes, the LCD display status is returned to the status set before the mode was switched to the above mode.

3 Function codes

The user can define desired function codes in normal mode.

Figure 4.14 lists the system default values.

Notes: 1. The values are presented in hexadecimal notation.
2. Code FFH specifies suppression of the key function.

1	FF	FF	FF	FF	FF
2	FF	FF	FF	FF	FF
3	FF	FF	FF	FF	FF
4	FF	FF	FF	FF	FF
5	FF	FF	FF	FF	FF
6	FF	FF	FF	FF	FF
7	FF	FF	FF	FF	FF
8	FF	FF	FF	FF	FF
9	FF	FF	FF	FF	FF
10	FF	FF	FF	FF	FF
11	FF	FF	FF	FF	FF
12	FF	FF	FF	FF	FF
13	FF	FF	FF	FF	FF
14	FF	FF	FF	FF	FF

Fig. 4.14 Function codes (default values) in normal mode

(4) Letter mode

1 Overview

Letter mode is a standard input mode supported by the system for inputting alphanumeric characters. Up to 31 characters can be input by using keys in this mode.

2 Setting conditions

The key status is set to letter mode under the following conditions:

(a) Function key F0 is pressed.
(b) Letter mode is specified as a parameter of BIOS KEYIN.
(c) Key 'EI' is pressed in kana mode.

3 Display format

(a) Figure 4.15 shows the letter input screen.
(b) Three lines of screen data are copied so that the current cursor line becomes the third line on the LCD screen. If the cursor is positioned in the first or second line of the window, however, the cursor line data is copied into the first or second line on the LCD screen. (In this case, the cursor is positioned in the first or second line.)

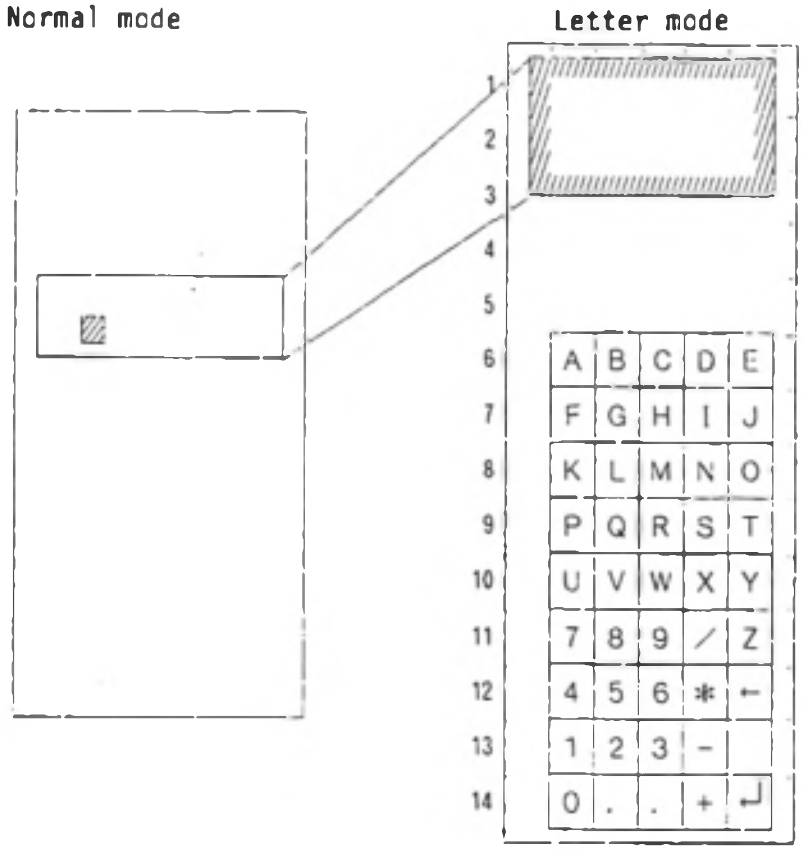


Fig. 4.15 Screen display in letter mode

4 Input procedure

(a) Up to 31 characters can be input in line-input mode. In this case, the input character string is processed as follows:

- When function key FD is pressed: The input character string is stored in the key input buffer (YPFKBUF: F434H) and is input one character at a time by CONIN.

- When BIOS KEYIN is executed: The input character string is stored in the buffer specified by the user.

(b) The pressed key is displayed at the current cursor position. If normal mode is restored after letter mode termination, the pressed key is not displayed.

(c) The functions of special keys pressed in letter mode are as follows.

'Return' key: Terminates inputting.
Pressing this key stores the input character string in the buffer and returns the current mode to normal mode. Value 0DH is added to the end of the character string.

'<--' key: Corrects entered data.
Pressing this key deletes the character entered last, moving the cursor to the deleted character position.

'kana' key: Sets the input mode to kana mode and display the kana input screen. This key is only effective when Japanese has been specified by the DIP switch.

5 Termination procedure

Pressing the 'Return' key terminates letter mode and returns to the original screen.

6 Function codes

Figure 4.16 shows the function codes used in letter mode.

Note : The values are represented in hexadecimal notation.

1	FF	FF	FF	FF	FF
2	FF	FF	FF	FF	FF
3	FF	FF	FF	FF	FF
4	FF	FF	FF	FF	FF
5	FF	FF	FF	FF	FC
6	41	42	43	44	45
7	46	47	48	49	4A
8	4B	4C	4D	4E	4F
9	50	51	52	53	54
10	55	56	57	58	59
11	37	38	39	2F	5A
12	34	35	36	2A	08
13	31	32	33	2D	20
14	30	2C	2E	2B	0D

Fig. 4.16
Function codes in letter mode

7 Remarks

(a) In letter (alphabetic character) mode, data is entered from the system screen. In this case, the data of the original user screen is saved.
(b) The input screen for letter mode is called the window which uses LCD lines 1 to 3.

(6) Kana mode

1 Overview

Kana mode is a standard mode supported by the system for inputting kana characters. Up to 31 characters can be entered by using keys in this mode.
If overseas mode has been specified, this mode is ineffective.

2 Setting conditions

The key status is set to kana mode under the following conditions:
(a) Function key FC is pressed.
(b) Kana mode is specified as a parameter of BIOS KEYIN.
(c) The 'kana' key is pressed in letter mode.

3 Display format

(a) Figure 4.16 (is omitted in this manual) shows the kana input screen.
(b) Three lines of screen data are copied so that the current cursor line becomes the third line on the LCD screen. If the cursor is positioned in the first or second line of the window, however, the cursor line data is copied into the first or second line on the LCD screen. (In this case, the cursor is positioned in the first or second line.)

4 Input procedure

(a) Up to 31 characters can be input in line-input mode. (See letter mode.)
(b) The pressed key is displayed at the current cursor position.
(c) The functions of special keys used in kana mode are as follows.

'Return' key: Terminates inputting. (See letter mode.)
'<-' key: Corrects entered data. (See letter mode.)
'EI' key: Sets the input mode to letter mode and displays the alphanumeric character input screen.
'Sho'key: Sets small character input mode. If the key is pressed, the small-sized character of it is entered, and the entered character is inverted and displayed.

5 Termination procedure

Pressing the 'Return' key terminates kana input mode and returns to the original screen.

6 Function codes

Figure 4.18 shows the function codes used in kana mode.

Note : The values are represented in hexadecimal notation.

1	FF	FF	FF	FF	FF
2	FF	FF	FF	FF	FF
3	FF	FF	FF	FF	FF
4	B1	B2	B3	B4	B5
5	B6	B7	B8	B9	BA
6	BB	BC	BD	BE	BF
7	C0	C1	C2	C3	C4
8	C5	C6	C7	C8	C9
9	CA	CB	CC	CD	CE
10	CF	D0	D1	D2	D3
11	D4	D5	D6	DF	DE
12	D7	D8	D9	DA	DB
13	DC	A6	DD	B0	A5
14	FD	00	20	08	0D

Fig. 4.18 Function codes in kana mode

4.3.7 Keyboard (EHT-10/2)

(1) Overview

In EHT-10/2, data is input from the keyboard.

The EHT-10/2 keyboard configuration is as follows:
1 Number of keys: 34
2 Number of LEDs: 3

All keys can be redefined arbitrarily by the application program (see BIOS PUTPFK).

Although the LEDs are used by the system for displaying the key mode, they can also be used by the user for BIOS CONOUT execution.

EHT-10/2 supports normal, letter, and kana modes as input modes. In kana mode, the Roman-character-to-kana-character conversion system has been incorporated.

EHT-10/2 also supports calculator mode so that simple four fundamental arithmetic operations can be performed by the application program. See Section 9.5 for calculator mode.

(2) Key arrangement

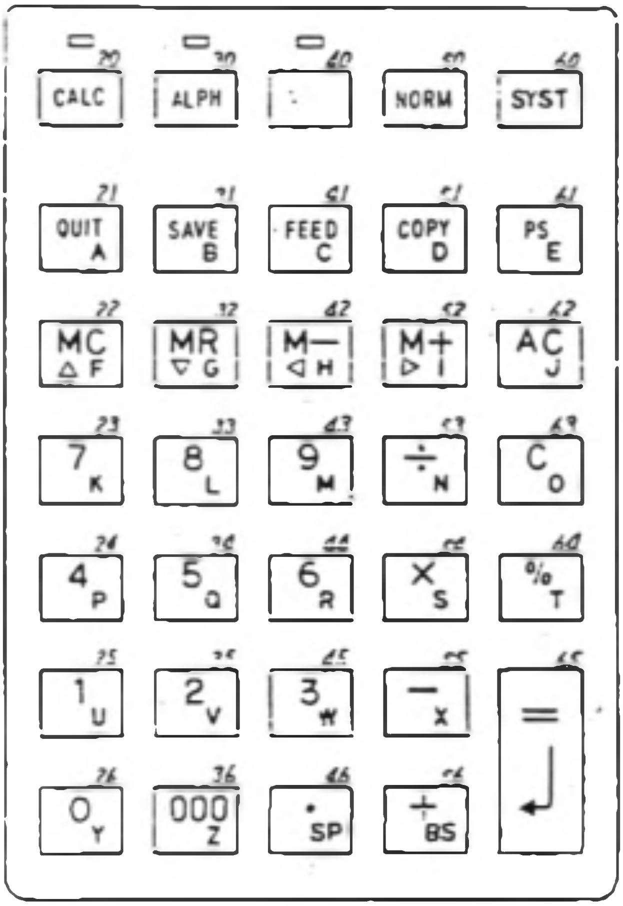


Fig. 4.19 EHT-10/2 keyboard key arrangement

(3) Normal mode

1 Overview

Normal mode is a standard mode in which the user can use the keyboard by defining arbitrary function codes for position codes. (See BIOS PUTPFK.)

2 Setting conditions

The key status is set to normal mode under the following conditions:

(a) Function key F8 is pressed.
(b) Normal mode is specified in BIOS KEYIN.
(c) The application program is initiated immediately after warm boot or restart power on.

3 Functions

Sets the function key tables in normal mode tables, and turns all LEDs off.

4 Function codes

The user can define desired function codes in normal mode. Figure 4.20 lists the system default values.

FE	FD	FC	FB	FA

01	02	F7	F8	FA
04	05	06	07	08
37 (7)	38 (8)	39 (9)	2F (/)	09
34 (4)	35 (5)	36 (6)	2A (*)	25 (%)
31 (1)	32 (2)	33 (3)	2D (-)	0D (Ret
30 (0)	(*1) (000	2E (.)	2B (+)	

F8: COPY
F7: FEED
FA: SYSTEM
FB: NORM
FD: ALPH
FE: CALC

Fig. 4.20 Function codes in normal mode

*1 Value 0 (30H) is returned three times sequentially.
(Setting code is F9H.)

Notes:

1. The values in the figure are represented in hexadecimal notation.

2. The value enclosed in parentheses is an ASCII code corresponding to the function code.

3. Codes FAH to FEH, F7H, and F8H are assigned with system functions. Even if these keys are pressed, the key codes are usually not returned to the user.

(4) Letter mode

1 Overview
Letter mode is a standard input mode supported by the system for inputting letters (alphabetic characters).

2 Setting conditions
The key status is set to letter mode under the following conditions:

(a) Function key FD is pressed.
(b) Letter mode is specified in BIOS KEYIN.

3 Functions
Sets the function key tables in normal mode tables, and turns only the LED for "letter (ALPH)" on.

4 Function codes

The function codes to be used in letter mode can be defined arbitrarily by the user (see BIOS PUTPFK). Figure 4.21 shows system default values.

FE	FD	FC	FB	FA

41 (A)	42 (B)	43 (C)	44 (D)	45 (E)
46 (F)	47 (G)	48 (H)	49 (I)	4A (J)
4B (K)	4C (L)	4D (M)	4E (N)	4F (O)
50 (P)	51 (Q)	52 (R)	53 (S)	54 (T)
55 (U)	56 (V)	57 (W)	58 (X)	0D (Ret
59 (Y)	5A (Z)	20 (SP)	08 (BS)	

F8: COPY
F7: FEED

FA: SYSTEM
FB: NORM

FD: ALPH
FE: CALC

Fig. 4.21 Function codes in letter mode

Notes:

1. The values are represented in hexadecimal notation.
2. The value enclosed in parentheses is an ASCII code corresponding to the function code.
3. Keys FAH to FEH are assigned with system functions. Even if these keys are pressed, the key codes are usually not returned to the user.

(5) Kana mode

1 Overview

Kana mode is a standard mode supported by the system for inputting kana characters. Roman letters can be entered instead of kana characters. The entered Roman characters are converted to kana characters by the system. If overseas mode has been specified, this mode is ineffective.

2 Setting conditions

The key status is set to kana mode under the following conditions:

(a) Function key FC is pressed.
(b) Kana mode is specified in BIOS KEYIN.

3 Functions

Sets function key tables in letter mode tables and turns only the LED for FC (kana) on. After this, kana characters can be entered through Roman character to kana character conversion.

4 Conversion table

Table 4.6 (is omitted in this manual) shows the Roman-character-to-kana-character conversion system.

4.4.1 Overview

The LCD-related hardware configuration for EHT-10 is different from that for EHT-10/2. Accordingly, the OS specifications for displaying characters in EHT-10 are also different from those in EHT-10/2. Sections 4.4.3 and 4.4.4 explain the display specifications in EHT-10 and those in EHT-10/2, respectively.

Sections 4.4.2, 4.4.5, 4.4.6, and 4.4.7 provide integrated explanation of the functions supported by BIOS, character generator, cursor displaying, and details on the display work area for both EHT-10 and EHT-10/2.

4.4.2 Functions supported by BIOS

The following BIOS functions are supported:

(1) CONOUT: Outputs one character to the CON: device.
(2) PUNCH: Outputs one character to the PUN: device.
(3) GRAPHICS: Supports graphic functions.
(4) TOUCH: Sets the indication for setting the key block of the touch-panel keys (only in EHT-10).
(5) KANJI: Displays and orints kanji characters.

See Section 4.2 for details on BIOS functions.

4.4.3 Display specifications in EHT-10

(1) Overview

1 Hardware specifications

EHT-10 uses LCD controller T6963, and exchanges data and commands with the main CPU through parallel communications. The hardware specifications are as follows:

- LCD display panel: 84 x 154 dots (12 characters x 14 lines)
- Controller: T6963
- Driver: X driver T7778 x 4
  Y driver T6961 x 1
- Duty ratio: 1/48
- Frame frequency: 58.6 Hz
- VRAM: 2 Kbytes (housed in the controller)

All display operations including displaying of characters and cursor and screen scrolling are performed in the graphic mode of the LCD controller under control of the operating system.

2 Software specifications

- Character fonts
Number of fonts: 228 + 13 (external characters for system)
Font size: 5 x 7 dots (ordinary characters)
6 x 8 dots (graphic characters)
7 x 11 dots (external characters)

- Display area
Character: 12 columns x 14 lines
Graphic: 84 (horizontal) x 154 (vertical) dots

- Screen
User screen: Scroll and fixed screens are used as virtual screens, and these screens have a character buffer capable of storing 42 lines of data.
System screen: The system screen has a character buffer capable of storing 14 lines of data (contents of one screen).

- Character attributes
Vertical, upper and lower ruler lines, comma, reverse, and secret

- Cursor attributes
Block/underline
Blink/nonblink

- Scroll
The screen can be scrolled only in the vertical direction.

- Character set adjustment to a specific country
Character set registration can be performed for ten countries including Japan.

(2) Screen mode

1 Screen configuration

EHT-10 has two virtual screens: system screen and user screen. These screens are automatically switched from one to the other by the system, and either of them is displayed on the LCD.

Figure 4.11 shows the relationship between the system screen and the user screen.

<System screen>

The system screen is exclusively used by the system, and it has a character buffer capable of storing 12 columns x 14 lines.

Use of the system screen does not affect the status of the user screen.

The system screen is used for the following functions:

System menu (CONFIG, DL/UL, DISK, or TEST)
Calculator mode
Letter/kana mode
BDOS/CCP error screen
Alarm screen
Charge battery screen
Disk Sum Check error screen

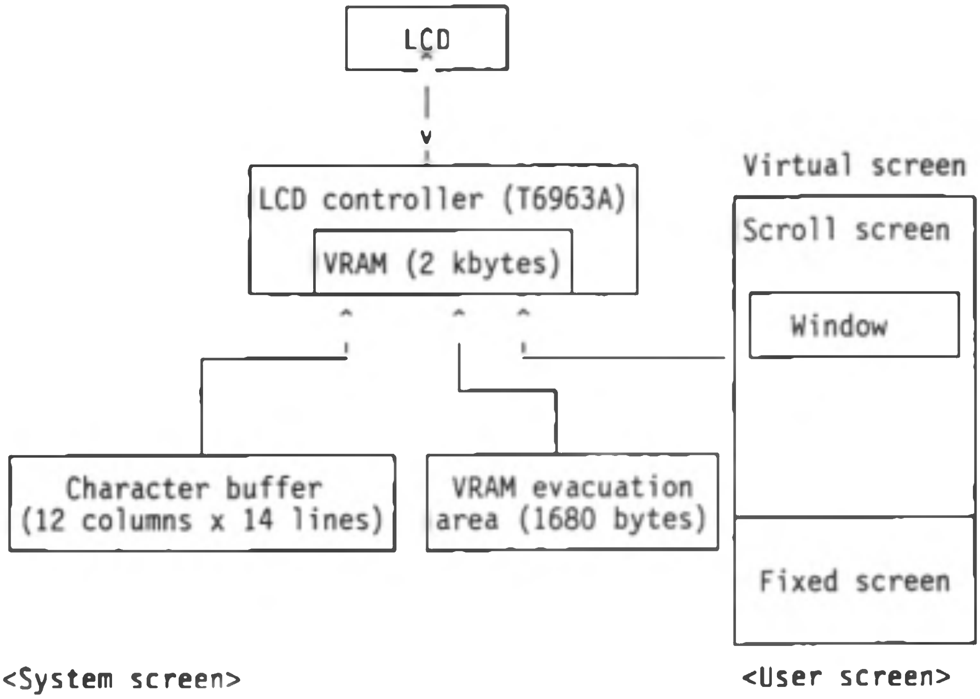


Fig. 4.22 System and user screens

The user screen can be used freely by the user. This screen is already active when the application program is initiated.

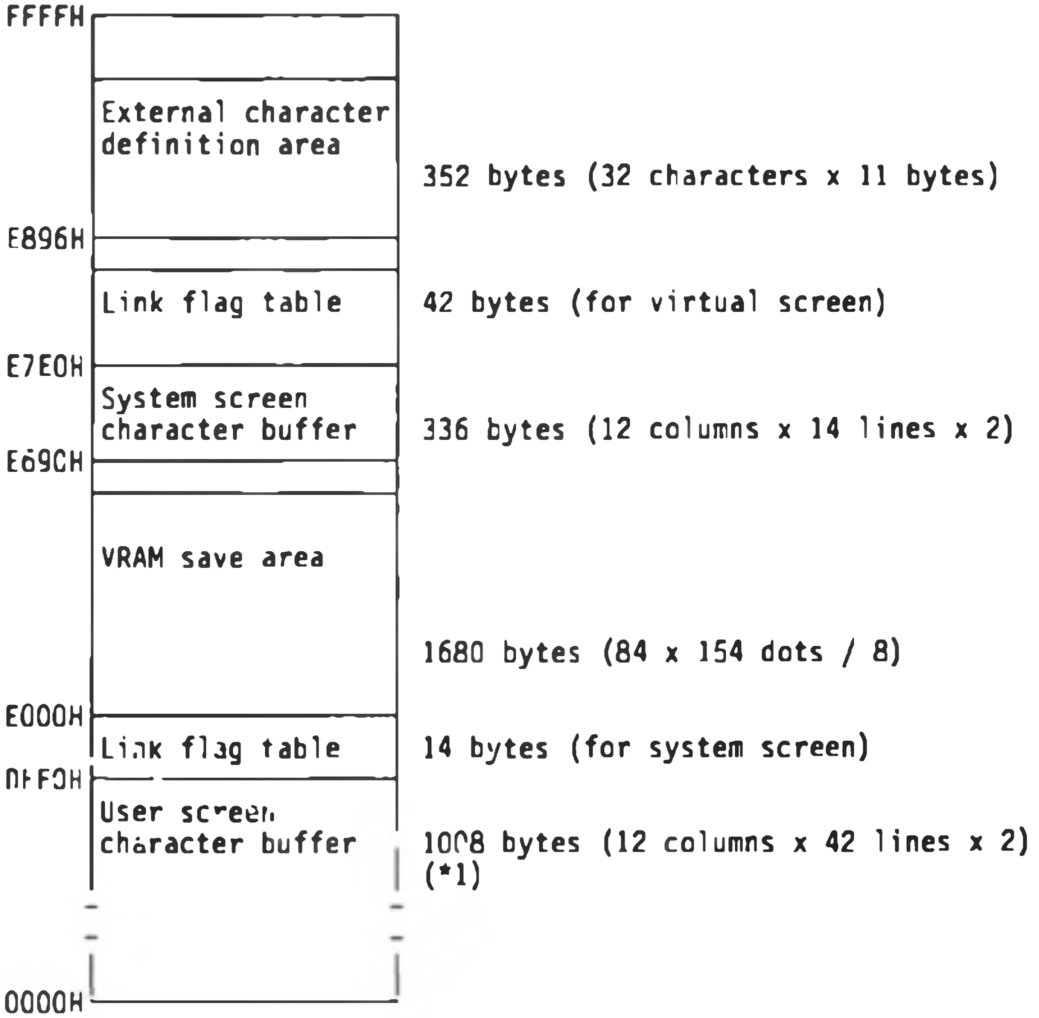
The user screen supports two modes: vertical display mode and horizontal display mode. Either of these modes can be selected in the BIOS CONOUT ESC sequence. The user screen is in vertical display mode immediately after application program initiation.
The user screen has a character buffer consisting of up to 1008 bytes. A logical screen larger than the actual LCD screen size can be configured by assuming the user screen to be a virtual screen. Therefore, only part of the virtual screen contents is actually expanded in VRAM and displayed on the LCD screen. This part of the LCD screen on which the expanded VRAM data is displayed is called the "window".

The maximum virtual screen size is 12 columns x 42 lines in vertical display mode and 25 columns x 20 lines in horizontal display mode.

The virtual screen in vertical display mode consists of the scroll screen and the fixed screen. By making the window size on the scroll screen less than the LCD screen size, both the window and fixed screen can be displayed simultaneously on the LCD screen.
The part of the fixed screen contents overlapped with the window contents cannot be viewed. Scroll and fixed screens can be controlled each individually by BIOS CONOUT.

2 Memory map (display buffer)

Figure 4.23 shows the memory map for the display buffer.



*1 Two bytes (one byte for a character code and one bytes for the character attribute) are used for one display character in the character buffer.

Fig. 4.23 Display buffer memory map

3 Display area configuration

<Vertical display mode>

A total of 168 characters (12 characters x 14 lines) can be displayed in the LCD display area (84 (horizontal) x 154 (vertical) dots).

One display area consists of 7 x 11 dots. A character font and character attribute are written in this area. Each character is displayed on the LCD screen when the corresponding bit image data including the character attribute is written in the VRAM located in the LCD controller.

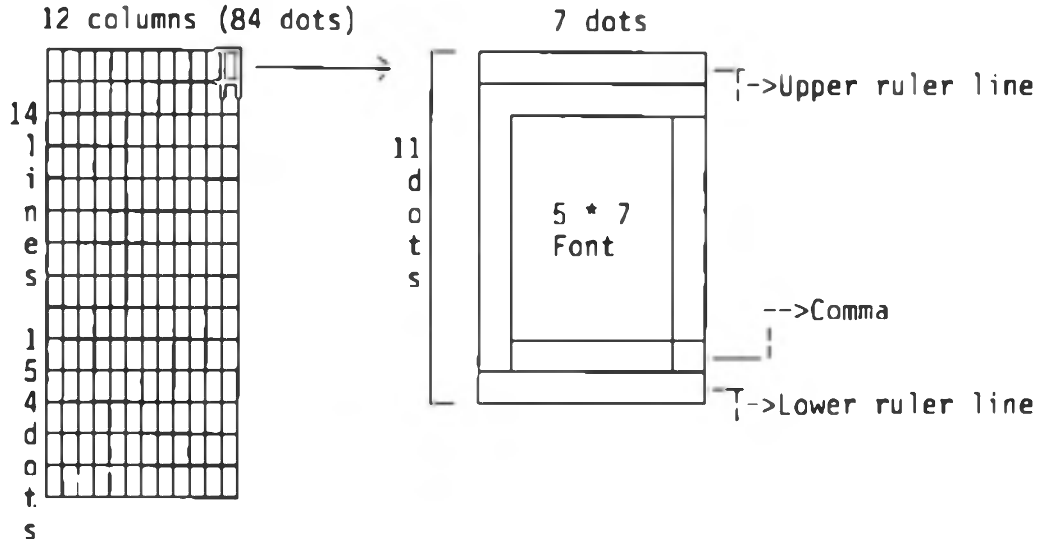


Fig. 4.24 Display area in vertical display mode

<Horizontal display mode>

In horizontal display mode, a total of 250 characters (25 columns x 10 lines) can be displayed by using 80 x 150 dots of the LCD display area (84 (vertical) x 154 (horizontal) dots) on the user screen. One display area consists of 6 x 8 dots. A character font and character attribute are written in this area.

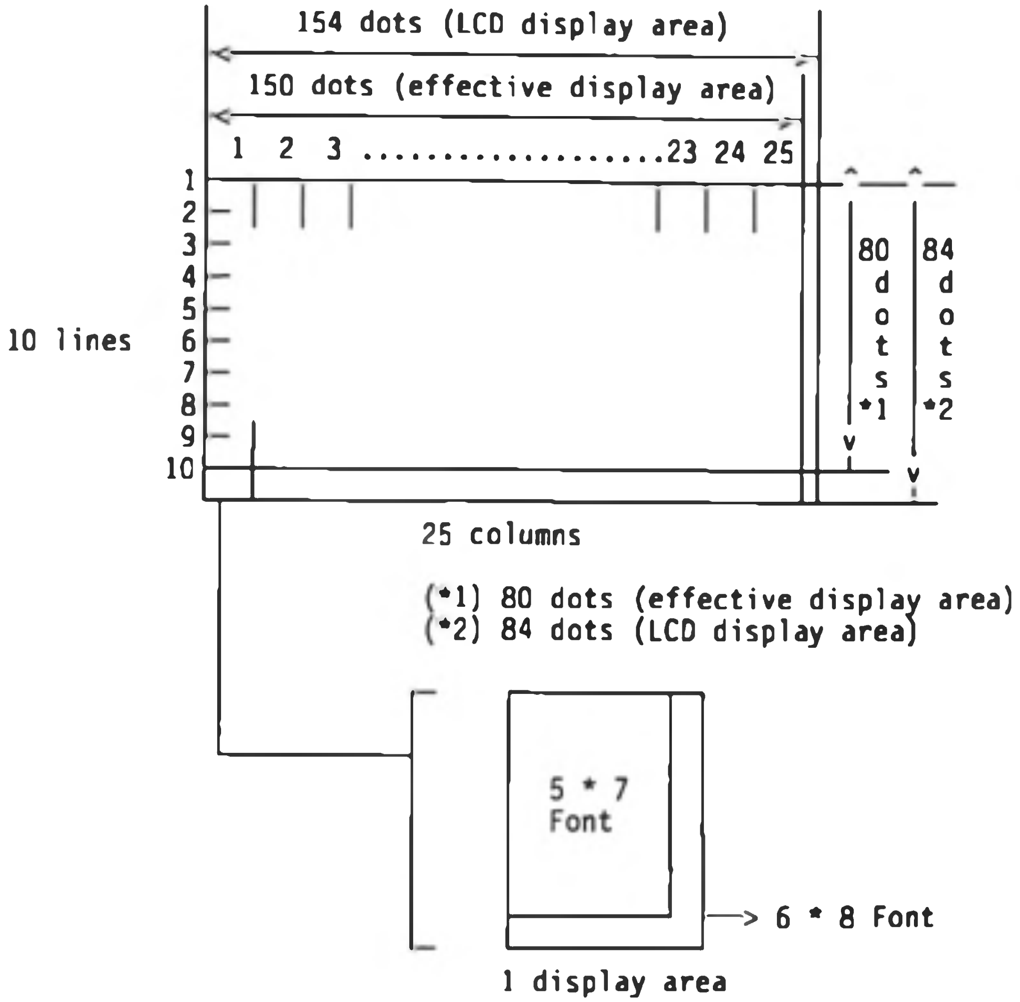


Fig. 4.25 Display area in horizontal display mode

4 VRAM addresses and display positions

In EHT-10, VRAM is located in the LCD controller. The VRAM contents can be accessed directly from the main CPU by using a command.

Figure 4.26 shows the relationship between the VRAM addresses and display positions in the LCD controller.

Notes:
1. Each address is represented in hexadecimal notation, and indicates the relative address from the head of VRAM.

2. The VRAM save area configuration is the same as the VRAM's. Therefore, assume each address in this figure as the relative address from the head of the VRAM save area.

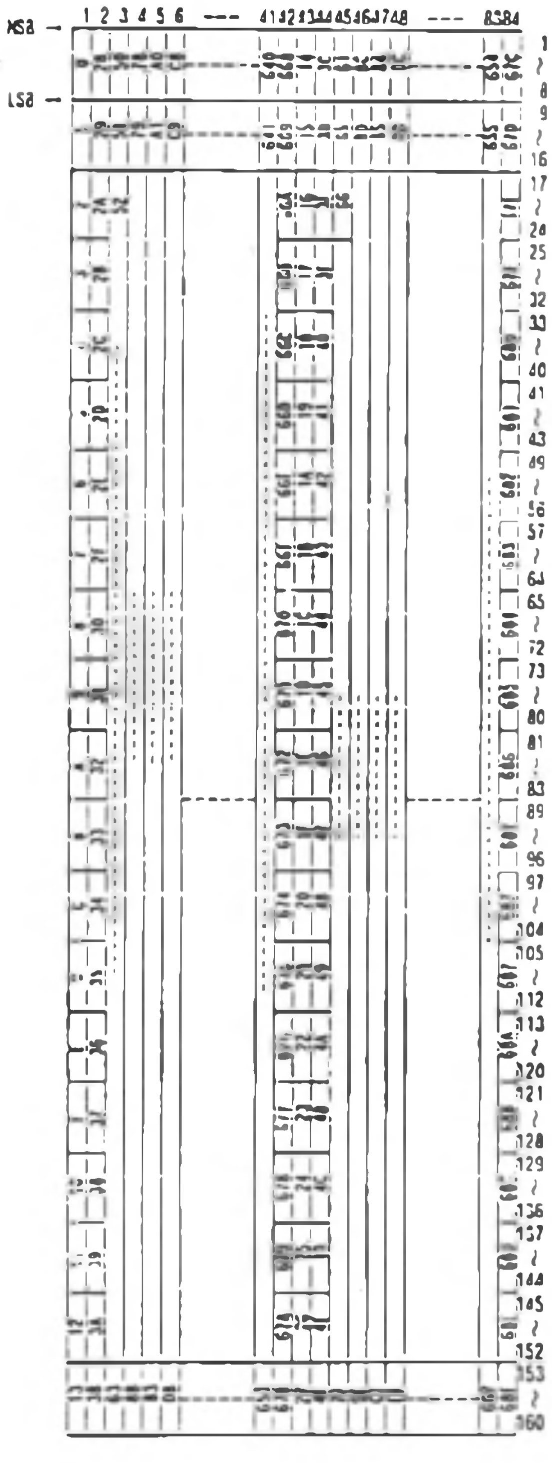
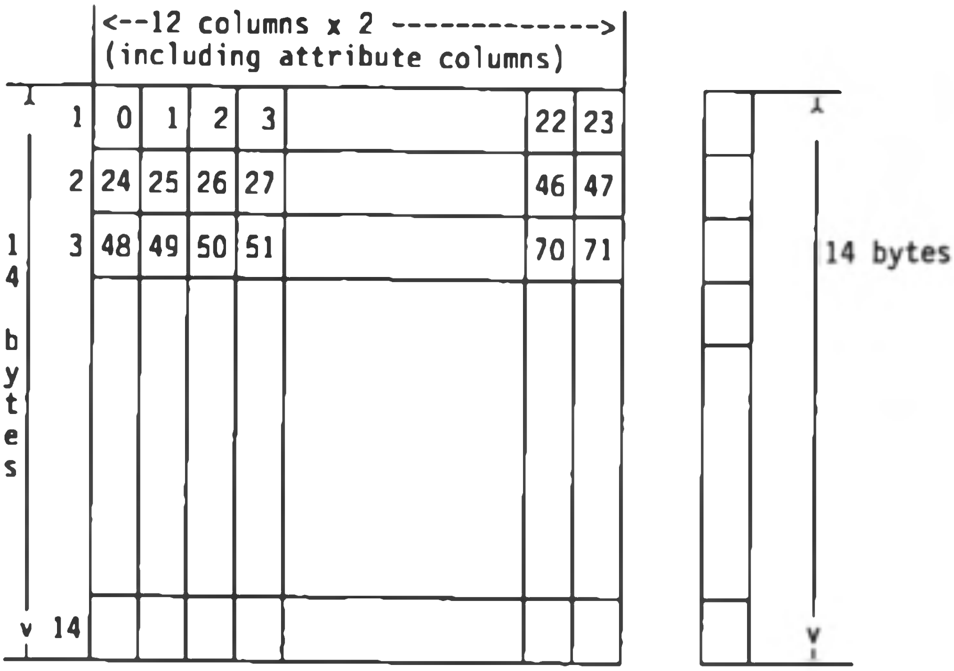


Fig. 4.26 Relationship between VRAM addresses and display positions

(3) System screen

The system screen is used by the system, and it has a character buffer capable of storing the contents of one screen (12 columns x 14 lines) and a link flag table.
Figure 4.27 shows the character buffer structure.



System screen Link flag table

Fig. 4.27 Character buffer structure

Notes:
1. Each value indicates the relative address from the head of the system screen.

2. The even-number-byte data is character (ASCII) data, and the odd-number-byte data is its attribute data.

1 Saving VRAM data

When initiating the system screen, the system saves the contents of VRAM that has been used for user screen processing in the main memory. When system screen processing is terminated, the system restores the saved VRAM data. This enables the system screen to be used without affecting the user screen display data.
The structure of the VRAM data save area is the same as that of VRAM. (See Figure 4.26.)

2 Normal and direct modes

The system screen can be internally used in normal or direct mode. The normal mode is used to write the display data into the character buffer and VRAM during execution of the following functions:

- System menu (CONFIG, DL/UL, DISK, or TEST)
- Calculator mode
- Letter/kana mode
- BDOS error
- CCP error (Bad Load or Command Error)

The direct mode is used to display data during processing such as interrupt processing. In this mode, no data is written into the character buffer but data is directly written into VRAM.

The direct mode is used for the following functions:

- Displaying Alarm
- Displaying Charge Battery
- Displaying Disk Sum Check error

The direct mode can also be initiated in the system screen status. If the direct mode is initiated from the system screen, data other than the graphic data is redisplayed when the direct mode is terminated. (*1)

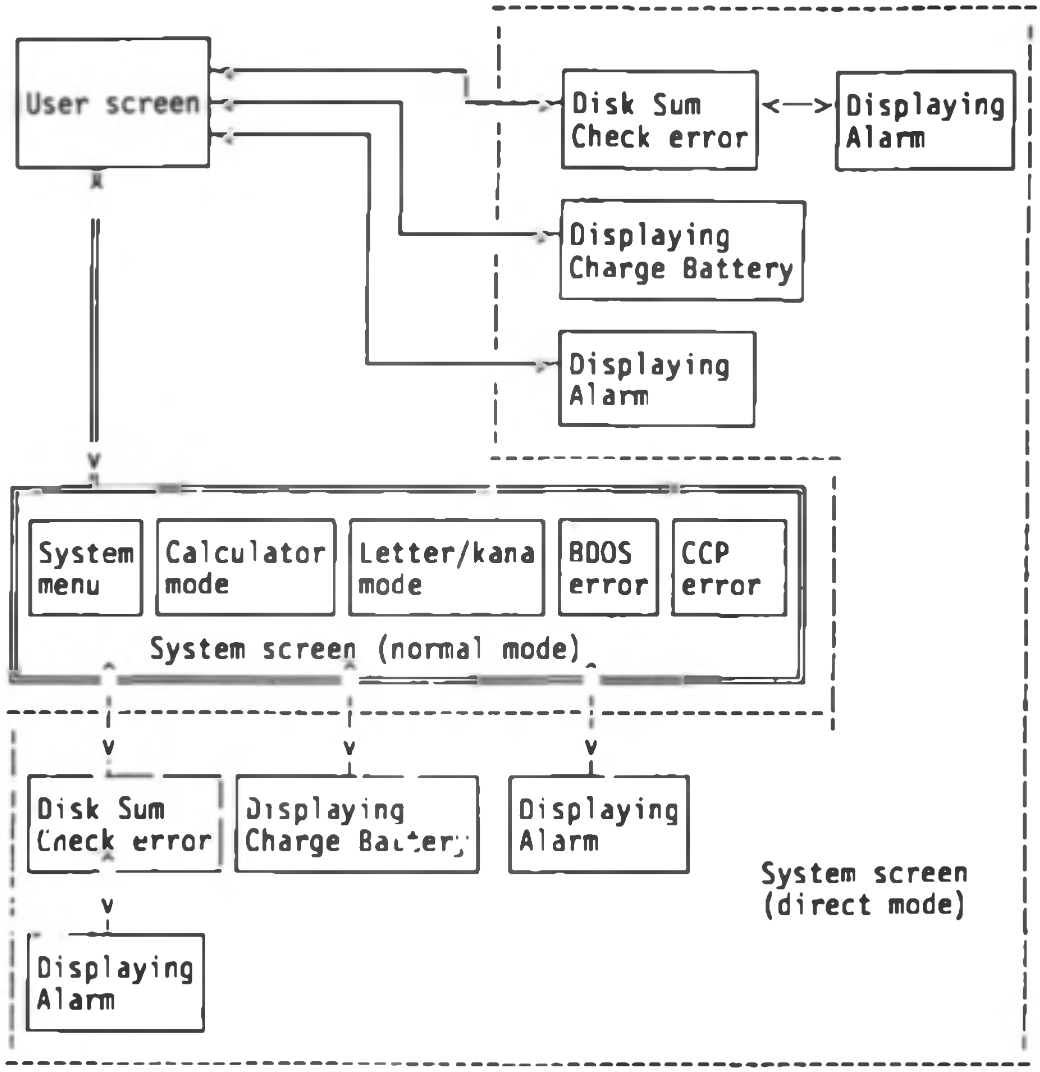
The operations performed at direct mode initiation and termination are as follows:

(a) When initiated while the system screen is in normal mode
At initiation: The display parameters for part of the system screen contents are saved.
At termination: The saved display parameters are restored and the screen buffer contents are redisplayed according to these parameters.

(b) When initiated as the user screen
At initiation: The mode is switched to system screen mode, and VRAM data is saved.
At termination: The saved VRAM data is restored, and the system screen mode is switched to user screen mode.

Figure 4.28 shows the transition of user and system screens (normal and direct modes).

*1 Because graphic data is usually not used for the system screen, the screen contents are completely restored after direct mode termination. If user graphic data (including kanji characters) is written in the input data display area in letter/kana mode, the graphic data is erased after message "Charge Battery" is displayed.



Note:

If the Alarm screen is output when a Disk Sum Check error occurred, the sequence escapes from the Alarm screen, following which the data is redisplayed by the Disk Sum Check error processing routine.

Fig. 4.28 Screen mode transition

(4) Scroll and fixed screens

The user screen of EHT-10 has incorporated the concept of virtual screen. In vertical display mode, the user screen consists of the scroll screen and the fixed screen, and has a character buffer capable of storing up to 12 columns x 42 lines.
The scroll screen is a character screen consisting of up to 12 columns x 42 lines, and part of the screen contents is displayed through the window on the LCD screen.
The fixed screen is a character screen having a fixed size of 12 columns x 14 lines. This screen is output logically behind the window and, when the window is made smaller than the fixed screen, only the part not hidden by the window can be viewed.
The fixed screen is made ineffective when the scroll screen size exceeds 28 lines.

Fixed and scroll screens can be controlled each independently by BIOS CONOUT (*1). Handling one screen does not affect the other screen.

Figures 4.29 and 4.30 shows the fixed screen structure and the relationship between the scroll and fixed screens, respectively.

*1 The screen that can currently be controlled is called an active screen. An active screen can be modified by the BIOS CONOUT ESC sequence.

Command: ESC+01H+n
n=0: Scroll screen
n=1: Fixed screen

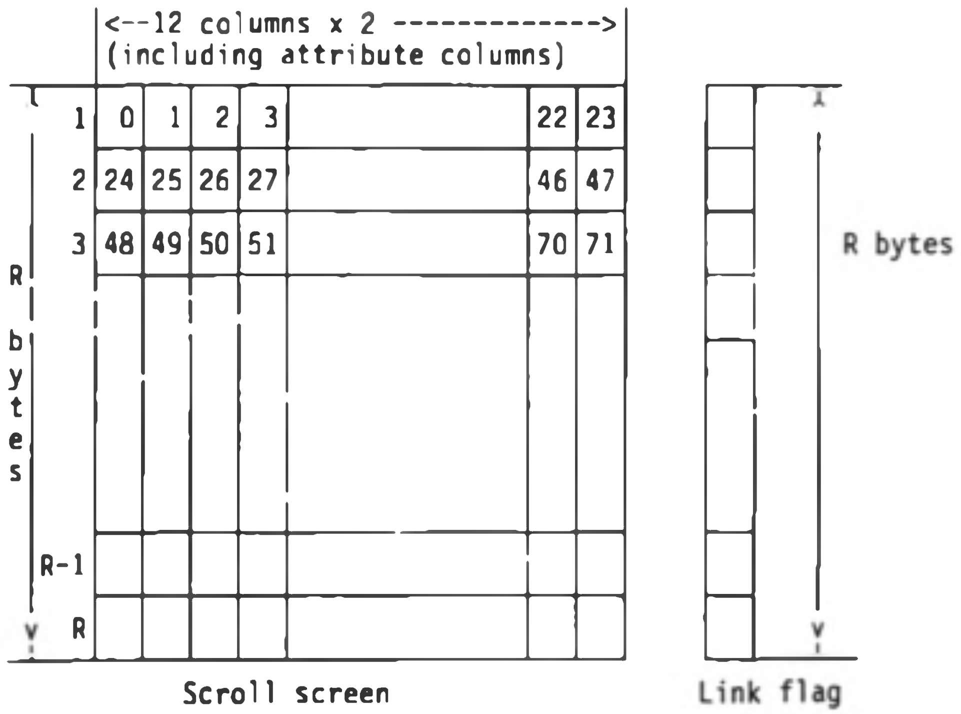


Fig. 4.29 Scroll screen structure ($14 \leq R \leq 42$)

Notes:
1. Each value indicates the relative address from the head address of the virtual screen.
2. The even-number-byte data is character (ASCII) data, and the odd-number-byte data is its attribute data.

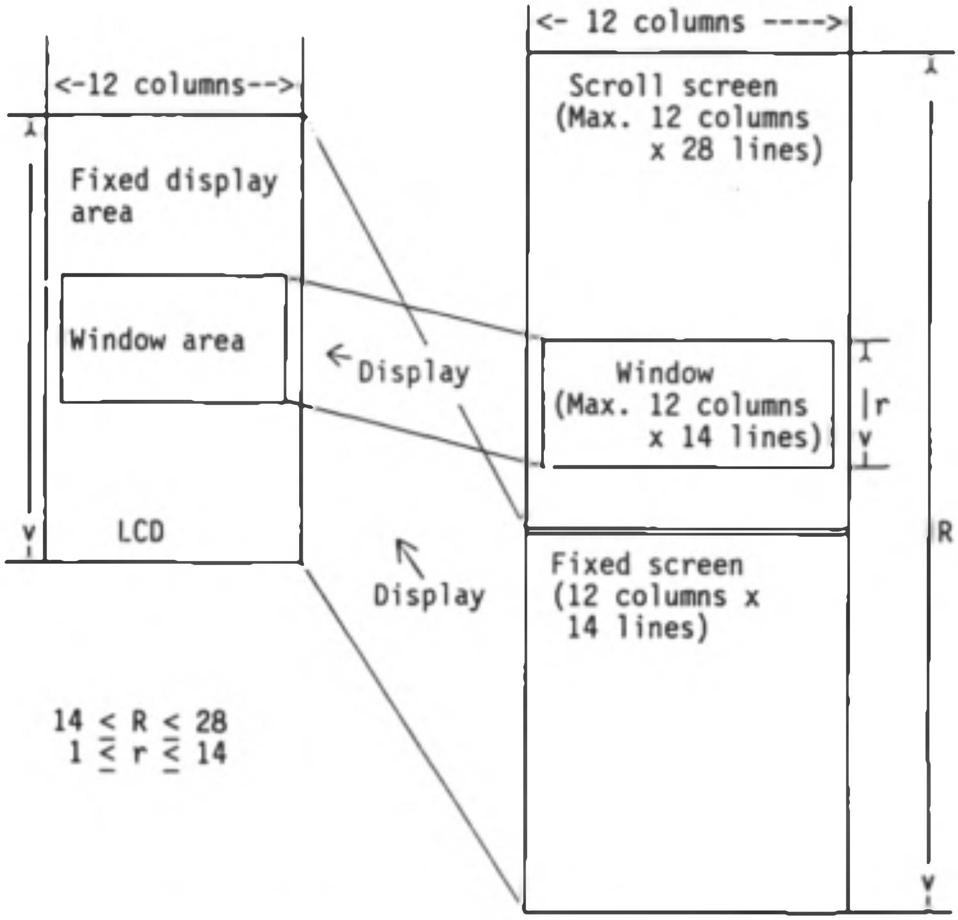


Fig. 4.30 Relationship between fixed and scroll screens

1 Scroll screen and window
The number of the scroll screen columns is fixed to 12 which is the same as the number of LCD display columns. The number of the scroll screen lines can be modified to a maximum of 42 by CONOUT (*1). Part of the scroll screen contents is displayed on the LCD screen by the window (*2).

The window size and the display position on the LCD screen can be specified by CONOUT (*3). The window size is set to 14 lines when the application program is activated.

*1 ESC+D0H+n [Setting the screen size]

*2 The window is usually displayed starting from the home position on the scroll screen, and then scrolled sequentially.

The window can also be moved to an arbitrary position on the scroll screen by using a CONOUT control code.

10H or ESC+96H [Scroll up by one line]
11H or ESC+97H [Scroll down by one line]

*3 ESC+D8H+M+N [Setting the window]

M: Window start line
N: Window end line

2 Window scroll mode

There are two window scroll modes: follow mode and unfollow mode. The scroll mode can be changed by CONOUT (*1).

<Follow mode>

In this mode, the window follows the cursor. If the unfollow mode in which the cursor is positioned outside the window is changed to follow mode, the window is automatically moved to the cursor position.

<Unfollow mode>

In this mode, the cursor does not follow the window. Therefore, the window does not move when data is written for the virtual screen (scroll screen). If the cursor is positioned outside the window when data is written for the virtual screen, the screen is automatically scrolled (*2). However, the display position remains unchanged.

*1 ESC+95H+M [Setting a scroll mode]

M=0: Follow mode (default)
M=1: Unfollow mode

*2 By modifying the system area contents, the mode in which the cursor is always positioned inside the window can be set (LSCROLMD).

3 Fixed screen

The fixed screen is automatically allocated by the system when the scroll screen size does not exceed 28 lines. If the window size is not less than the LCD size, none of the fixed screen contents can be viewed because the fixed screen is output logically at the behind of the window screen of the scroll screen. Because the fixed screen is not scrolled, it is very advantageous when used for displaying fixed data such as touch-panel keys.
The structure of the fixed screen is the same as that of the scroll screen whose size is set to 14 lines.

(5) Horizontal display mode

If horizontal display mode is specified in BIOS CONOUT (*1), 25 columns x 10 lines can be displayed on the LCD screen (*2). Because one display area consists of 6 x 8 dots in horizontal display mode, the external font of 7 x 11 dots for vertical display mode cannot be used (*3).

Only reverse and secret can be specified as the character attributes.

In horizontal display mode, the virtual screen size can be set to a maximum of 25 columns x 20 lines. The fixed screen is not allocated and the window is fixed to 25 lines x 10 lines (LCD size).

*1 ESC+DAH+n [Changing display mode]
n=0: Vertical display mode
n=1: Horizontal display mode

*2 See Item (2) 3 "Display area configuration" for the configuration of the display area on the LCD screen.

*3 In horizontal display mode, external characters of 6 x 8 dots can be registered.

ESC+E0H+n+P(1)+ ... +P(8)
n: External character code (E0H to FFH)
P(i): Font pattern

Notes:
Because BIOS TOUCH does not support horizontal display mode, characters are displayed vertically by the BIOS TOUCH function.

(6) Attributes

Attributes can be specified for each display character. Each display character is displayed according to the specified attributes and font data.
The attribute bit configuration is as follows.

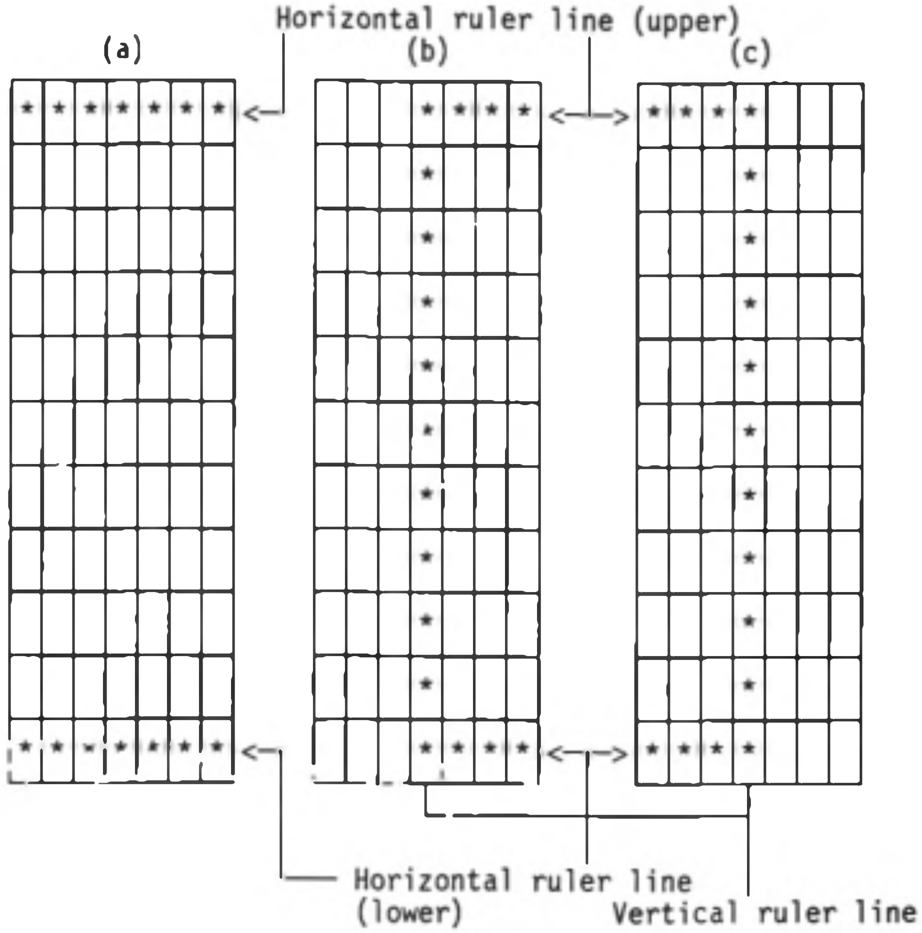
```
  7   6   5   4   3   2   1   0
+---+---+---+---+---+---+---+---+
| 0 | 0 |   |   | * | * | * | * |
+---+---+---+---+---+---+---+---+
          |   |   |   |   |   |
          |   |   |   |   |   └-> Horizontal ruler line (top)
          |   |   |   |   └-> Horizontal ruler line (bottom)
          |   |   |   └-> Vertical ruler line
          |   |   └-> Comma
          |   └-> Reverse
          └-> Secret
               ( 1:Specified  0:Not specified )
```

* : Can be specified only in vertical display mode

1. Horizontal ruler line

(a) When the horizontal ruler line is specified and the vertical ruler line is not specified, the horizontal ruler lines are displayed in full stroke width. (See Figure 4.31 (a).)

(b) When both the horizontal and vertical ruler lines are specified
- When the horizontal ruler line is specified, only the right half of the horizontal ruler line is displayed. (See Figure 4.31 (b).)
- When the horizontal ruler line is specified at the previous character position, only the left half part of the horizontal ruler line is displayed. (See Figure 4.31(c).)



(a) Specifying only the horizontal ruler line
(b) Specifying both horizontal and vertical ruler lines
(c) Specifying the vertical ruler line and specifying the horizontal ruler line for the previous character

Fig. 4.31 Font data and ruler line specification

2 Vertical ruler line

When the vertical ruler line is specified, the vertical ruler line is displayed at the character position.

3 Comma

When a comma is specified, a comma is displayed at position (7,10) of the font data.

4 Reverse

When reverse is specified, the font data of 7 x 11 dots is inverted and displayed. If a comma or ruler line is also specified, the data corresponding to the attribute is also inverted and displayed.

5 Secret

When secret is specified, the corresponding data is displayed as blank. Even if another attribute is specified in this case, the attribute is ignored but written in the attribute data.

(7) Link flag

The link flag indicates whether the line is logically continuous from the previous line.

The state of this flag is referenced by BASIC when keys are pressed, and set or released under the following conditions:

1 Setting

A character is displayed when the cursor is positioned in the last column of the line, and the cursor has moved in the first column of the next line. (Value 01H is set.)

2 Release

(a) When "Clear Screen & Home" is output
The link flag table of the current screen is reset.

(b) When "Erase End of Screen" is output
The link flags of the current cursor line and subsequent lines are reset.

Figure 4.32 shows the link flag table structure.

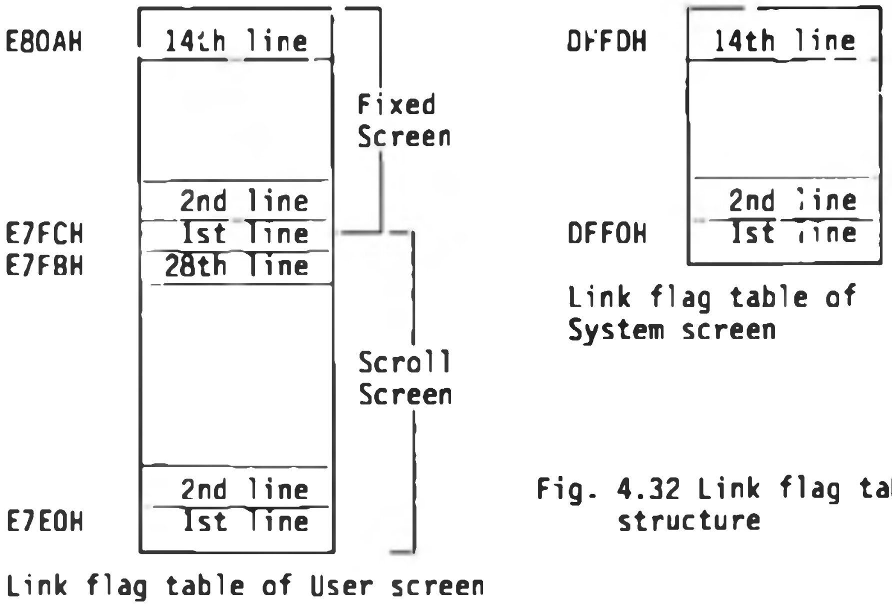


Fig. 4.32 Link flag table structure

Notes:

1. The link flag value is 00H or 01H.

   00H: Continuous
   01H: Incontinuous

2. The link flag table of the user screen is the one when the scroll screen consists of 28 lines. If the scroll screen size exceeds 28 lines, the link flags for the lines that exceeded the 28 scroll screen lines are set in the link flag table area of the fixed screen. In horizontal display mode, the first 20 bytes of the scroll screen are used of the link table.

(8) Graphics

Graphic data is directly written into VRAM. Graphic data can be written and output with character data. If the screen is scrolled and the displayed graphic data overflows from the screen, the overflowed graphic data is lost and, even if the screen is scrolled back, the lost data cannot be redisplayed (*1).

The graphic screen consists of horizontal 84 dots x vertical 154 dots.

GRAPHICS and KANJI are supported as BIOS functions for writing graphic data.

Note:
Graphic data is also lost when the display position or size of the window is modified.

4.4.4 Display specifications in EHT-10/2

(1) Overview

1 Hardware specifications

The LCD display screen in EHT-10/2 uses a uniform matrix of 120 x 32 dots that can be dynamically activated.
The hardware specifications are as follows:

- LCD panel: 120 x 32 dots (20 columns x 4 lines)
- Controller: GAWIS
- Driver
  X driver: SED1180 x 2
  Y driver: SED1190 x 1
- Duty ratio: 1/32
- Frame frequency: Variable
- VRAM: 512 bytes x 3 (in main memory)
- Display mode: Bit image
- Scroll function: Vertical direction dot scroll

- Others
  Display ON/OFF function is supported.

2 Software specifications

- Character fonts
Number of fonts: 228 + 13 (external characters for system)
Font size: 5 x 7 dots (ordinary characters)
           6 x 8 dots (graphic and external characters)
- Display area
Character: 20 columns x 4 lines
Graphic: 120 (horizontal) x 32 (vertical) dots
- Screen
User screen: Two screens (each a virtual screen consisting
             of 20 columns x 25 lines)
System screen: One screen
- Character attributes: None (the system screen has)

- Cursor attributes
  Block/underline
  Blink/nonblink
- Scroll

The screen can be scrolled only in the vertical direction.

- Character set adjustment to a specific country

Character set registration can be performed for ten countries including Japan.

(2) Screen mode

1 Screen configuration

EHT-10/2 has a total of three screens: one system screen and two user screens. Each screen has VRAM and character buffer, and can be controlled completely independently. However, only the two user screens can be used by the user. The user cannot directly control displaying data on the system screen.
Figure 4.33 shows the relationship between the system screen and user screens.

<System screen>

The system screen has a character buffer with the same size as the LCD screen size (20 columns x 4 lines), and used for the following functions:

- System menu (CONFIG, DL/UL, DISK, or TEST)
- Calculator mode
- BDOS/CCP error
- Alarm screen
- Charge Battery screen
- Disk Sum Check error screen

<User screens>

User screens can be used freely by the user. These screens are already active when the application program is initiated.
The configurations of two user screens are the same, and can be switched freely from one to the other by the application program. A user screen has a character buffer consisting of 20 columns x 25 lines. A logical screen larger than the actual LCD screen size can be configured by assuming the user screen to be a virtual screen. Therefore, only part of the virtual screen contents is actually expanded in VRAM and displayed on the LCD screen. This part of the LCD screen on which the expanded VRAM data is displayed is called the "window".

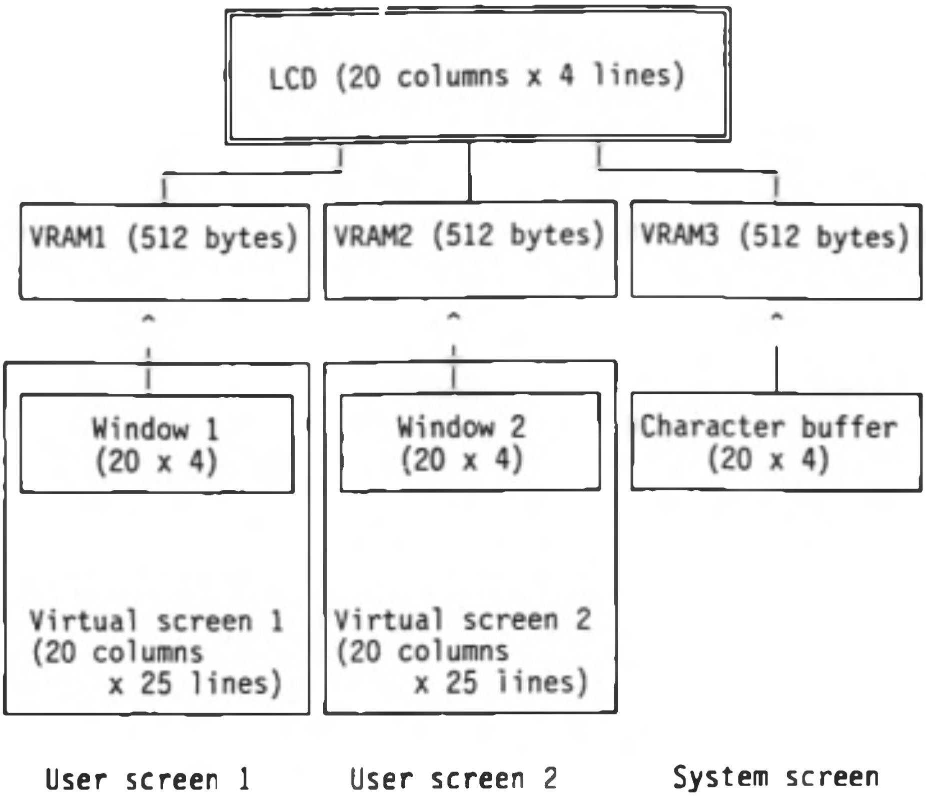


Fig. 4.33 System screen and user screens

2 Memory map (display buffer)

Figure 4.34 shows the memory map for the display buffer.

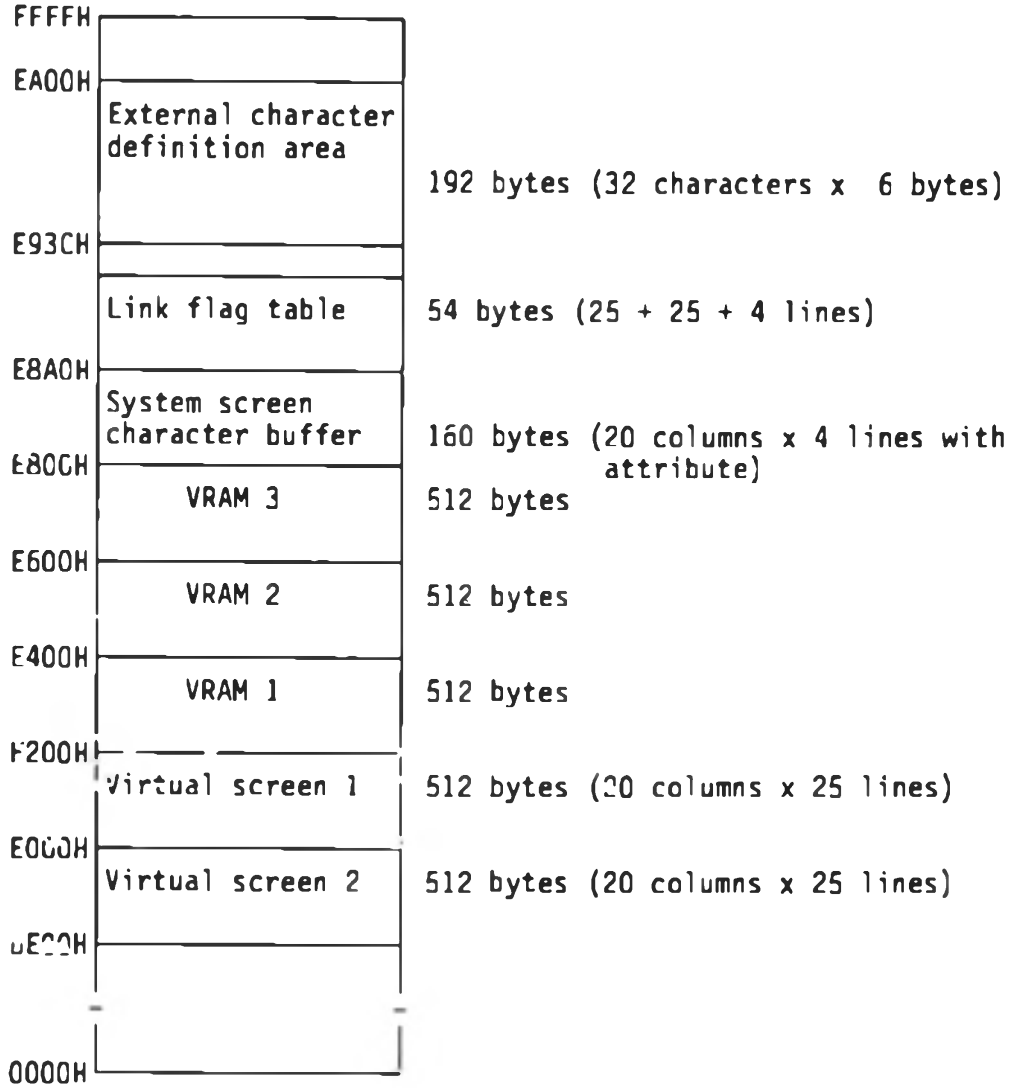
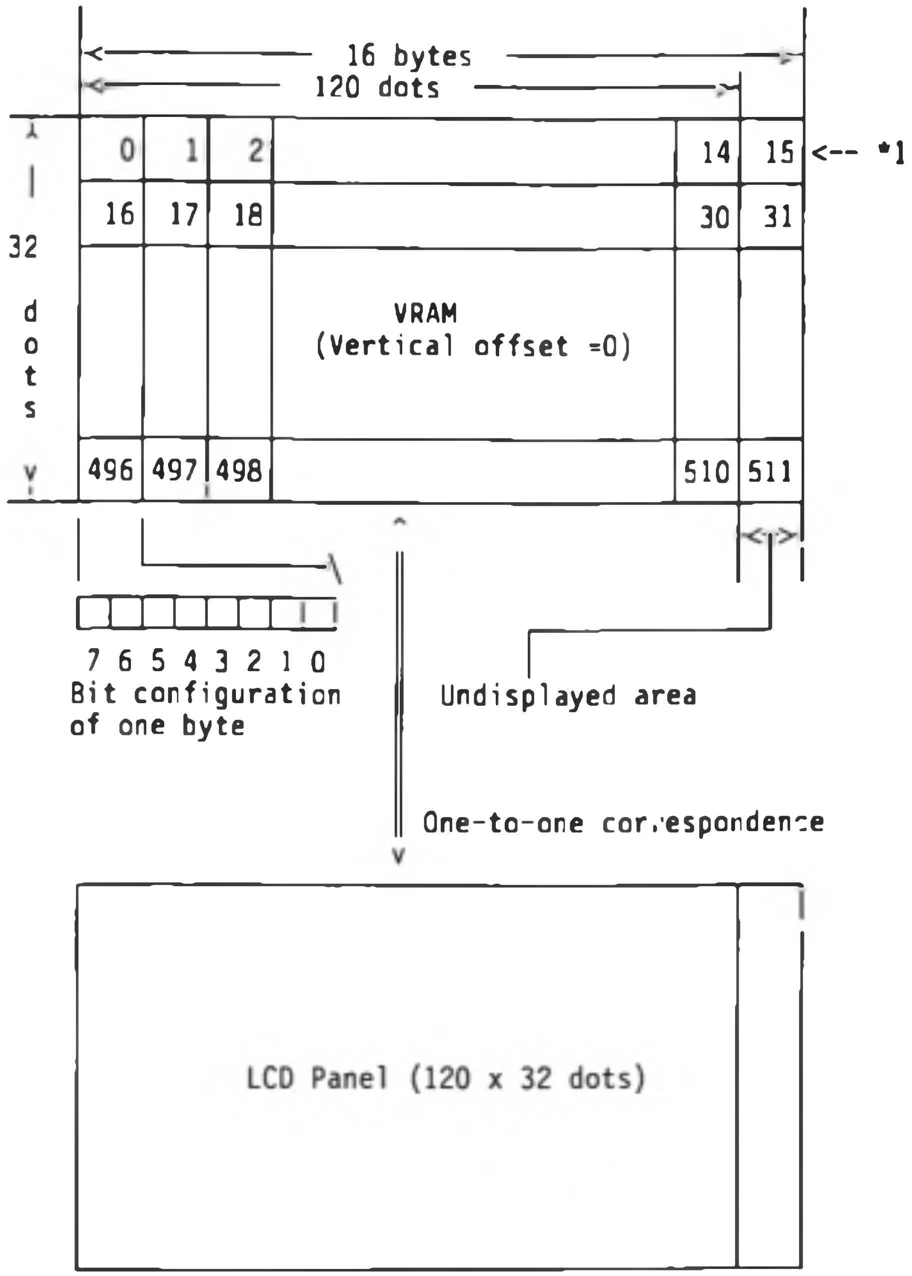


Fig. 4.34 Display buffer memory map

3 VRAM structure and display area

This Item explains the VRAM structure and how to use it. A VRAM relative address corresponds to an LCD panel dot.
Figure 4.36 shows the relationship between VRAM data and LCD panel dots.



*1 : Relative address from the VRAM head

Fig. 4.36 Relationship between VRAM relative addresses and LCD display panel dots

The display start address for displaying data on the LCD panel is determined from the following:

VRAM head address (LSCRVRAM)
Vertical offset value (LVRAMYOF)

The vertical offset value indicates the vertical relationship between VRAM locations and LCD panel dots. The VRAM data is displayed starting from the location indicated by the offset from the bottom end of VRAM and, when processing has reached the VRAM bottom end, VRAM data is displayed starting from the top of VRAM.

The system uses this function when scrolling the screen vertically. Figure 4.35 shows the relationship between VRAM relative byte addresses and LCD panel dots when the offset value is set to 2.

32	33	34		46	47	
			VRAM (Vertical offset =2)			
496	497	498		510	511	
0	1	2		14	15	<-- *1
16	17	18		29	30	

Relative address from the VRAM head

*1 :This line is the first VRAM line.

Fig. 4.35 Relationship between VRAM and LCD panel (vertical offset value = 2)

The configuration of the system area used for displaying VRAM data is as follows.

Address : Variable name (Number of bytes)

F253H : LSCRVRAM (2)
The head address of VRAM1 or VRAM2 whose contents are being displayed on the LCD screen is stored. The value must be E000H or larger, and a 512-byte boundary must be taken into account.

F266H : LVRAMYOF (1)
The vertical VRAM offset value is stored.
$0 \leq \text{LVRAMYOF} \leq 31$
A multiple of 8 is usually stored.

Figure 4.37 shows the display area configuration.

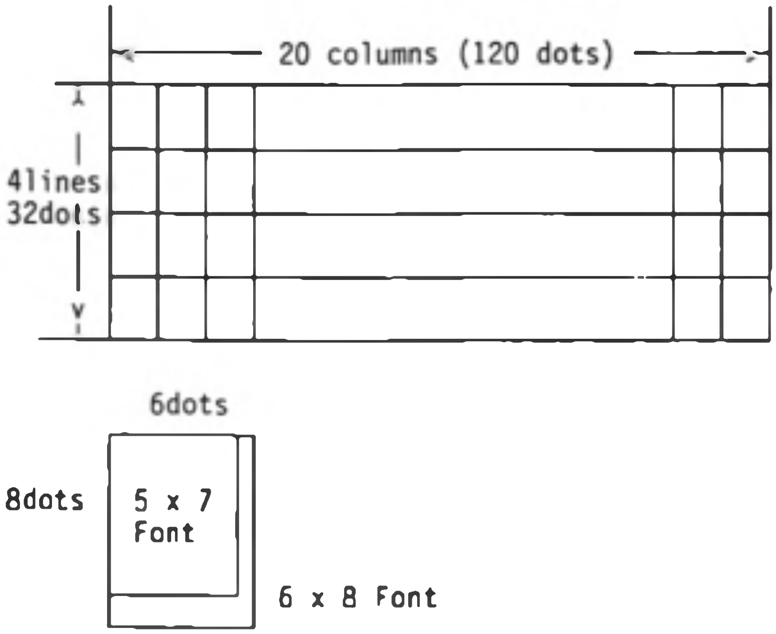


Fig. 4.37 Display area configuration

(3) System screen

The system screen is used by the system, and it has a character buffer capable of storing the contents of one screen (20 columns x 4 lines) and a link flag table.

One display character shares two bytes (one byte for the character code and one byte for the attribute) in the system screen character buffer. The system screen character buffer size is 160 bytes.

Figure 4.38 shows the relationship between the character buffer and link flag table for the system screen.

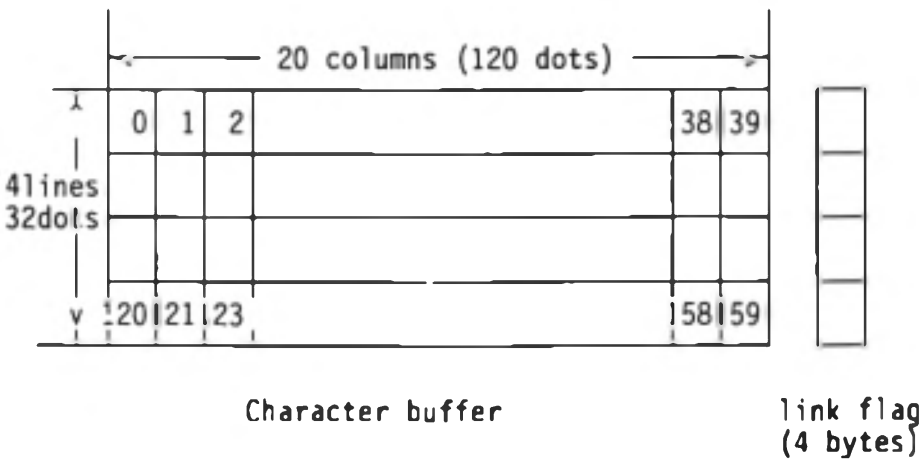


Fig. 4.38 Relationship between character buffer and link flag table

*1 Each value is the relative address from the head of the character buffer.

*2 The even-number-byte data is character data, and the odd-number-byte data is its attribute data.

<Normal and direct modes>

The system screen can be internally used in normal or direct mode.

The normal mode is used to write the display data into the character buffer and VRAM during execution of the following functions:

- System menu (CONFIG, DL/UL, DISK, or TEST)
- Calculator mode
- Letter/kana mode
- BDOS error
- CCP error (Bad Load or Command Error)

The direct mode is used to display data during processing such as interrupt processing. In this mode, no data is written into the character buffer but data is directly written into VRAM.

The direct mode is used for the following functions:

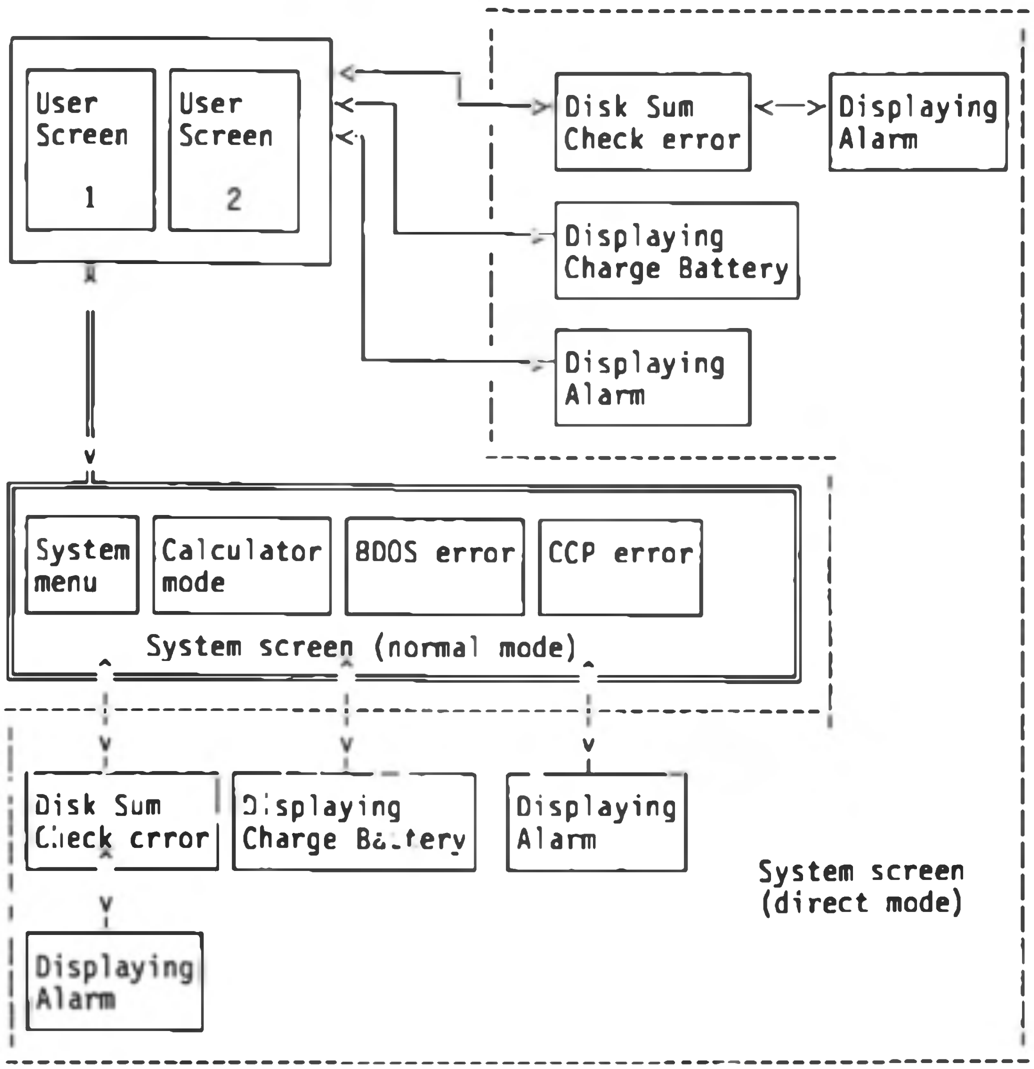
- Displaying an alarm
- Displaying Charge Battery
- Displaying a Disk Sum Check error

The direct mode can also be initiated in the system screen status. If the direct mode is initiated from the system screen, the following operations are performed:

At initiation: After the display parameters for part of the system screen contents are saved, data is written into VRAM (displayed on the screen).
At termination: After the screen buffer contents are rewritten into VRAM, the saved display parameters are restored.

The system screen contents other than graphic data are redisplayed by the above operations. (Because the system usually does not use graphic data, the screen contents are completely restored after use in direct mode.)

Figure 4.39 shows the transition of user and system screens (normal and direct modes).



Note:

If the Alarm screen is output when a Disk Sum Check error occurred, the sequence escapes from the Alarm screen, following which the data is redisplayed by the Disk Sum Check error processing routine.

(4) User screen

1 Virtual screen

Like EHT-10, EHT-10/2 has incorporated the concept of virtual screen which enables the application program to use a screen logically larger than the actual LCD screen (20 columns x 4 lines). The maximum size of the EHT-10/2 virtual screen is 20 columns x 25 lines.

The virtual screen is a character screen, and part of the screen contents is displayed through the window on the LCD screen.

Figure 4.40 shows the virtual screen structure. The attribute data is not added to the virtual screen contents.

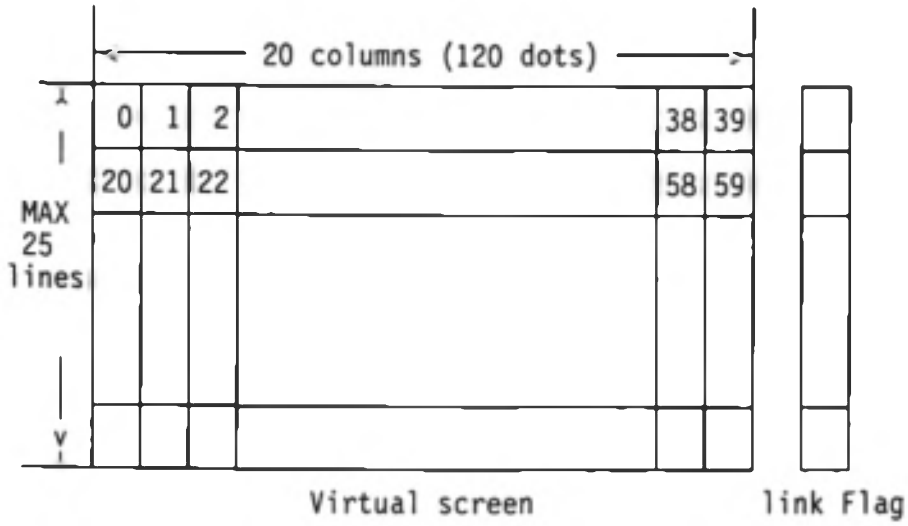


Fig. 4.40 Virtual screen structure

Note:
Each value used in the figure is a relative address from the head of the virtual screen.

2 Switching user screen

There are two types of user screens: user screen 1 and user screen 2. These two screens are identical and can be switched arbitrarily from the one to the other by the user for use.
The user screen can be switched by BIOS CONOUT.

Sequence: ESC+D1H+n
n=0: Screen 1
n=1: Screen 2

3 Window scroll mode

There are two window scroll modes: follow mode and unfollow mode. The scroll mode can be changed by CONOUT.
The window scroll modes for EHT-10/2 are the same as those for EHT-10.
See Section 4.4.3 for details on scroll modes.

(5) Link flag

The link flag indicates whether the line is logically continuous from the previous line.
The state of this flag is referenced by BASIC when keys are pressed, and set or released under the following conditions:

1 Setting

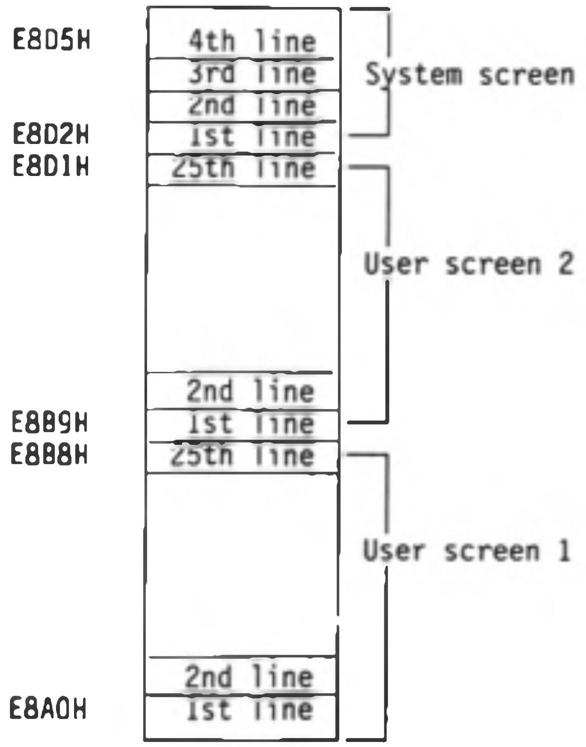
A character is displayed when the cursor is positioned in the last column of the line, and the cursor has moved in the first column of the next line. (Value 01H is set.)

2 Release

(a) When "Clear Screen & Home" is output
The link flag table of the current screen is reset.

(b) When "Erase End of Screen" is output
The link flags of the current cursor line and subsequent lines are reset.

Figure 4.41 shows the link flag structure.



Note:
The link flag value is 00H or 01H.
00H: Continuous
01H: Incontinuous

Fig. 4.41 Link flag table

(8) Graphics

Graphic data is directly written into VRAM. Graphic data can be written and output with character data. If the screen is scrolled and the displayed graphic data overflows from the screen, the overflowed graphic data is lost and, even if the screen is scrolled back, the lost data cannot be redisplayed.

The graphic screen consists of horizontal 84 dots x vertical 154 dots.

GRAPHICS and KANJI are supported as BIOS functions for writing graphic data.

4.4.5 Character generator

(1) Overview

EHT-10 and EHT-10/2 have a character generator in the system ROM. The character generator codes and font sizes are as follows.

00H to 7FH: 5 x 8 dots
80H to 9FH: 6 x 8 dots
A0H to 0FH: 5 x 8 dots

E0H to FFH

EHT-10 (vertical display mode): 7 x 11 dots
EHT-10 (horizontal display mode): 6 x 8 dots
EHT-10/2: 6 x 8 dots

E0H to FFH indicate external characters that can be defined freely by the user (*1).

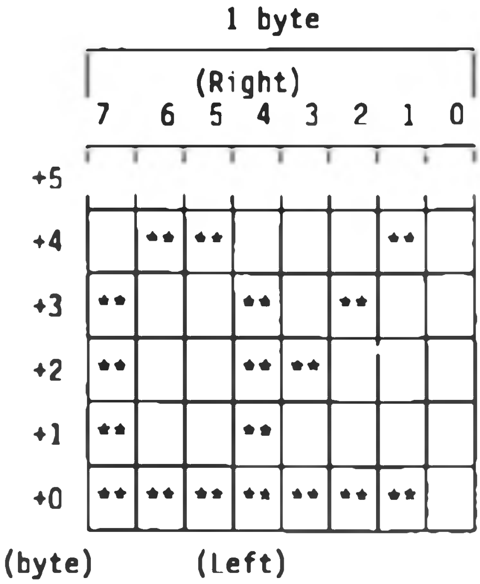
*1 ESC sequence for registering external characters

ESC+E0H+n+p(1)+ ... +p(i)

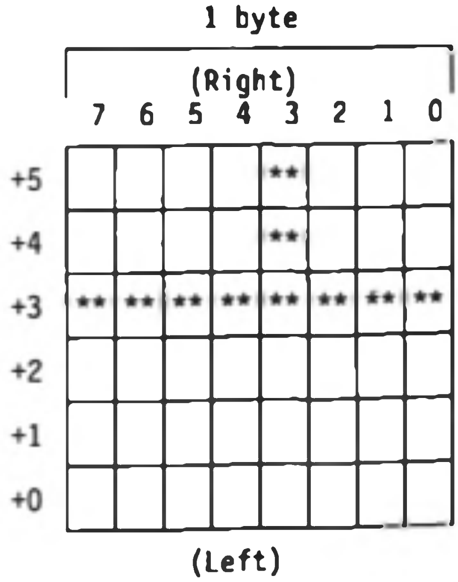
n: External character code (E0H^S < ^Sn^S < ^SFFH)
P: Font pattern (i = 8 or 11)

(2) Font format

Each font consists of five, six, or 11 bytes, and is recorded in the memory.
Each character is recorded in a laid format as shown below. (The ** grid indicates bit=1 and the blank grid indicates bit=0.)



5 x 8 Font



6 x 8 Font

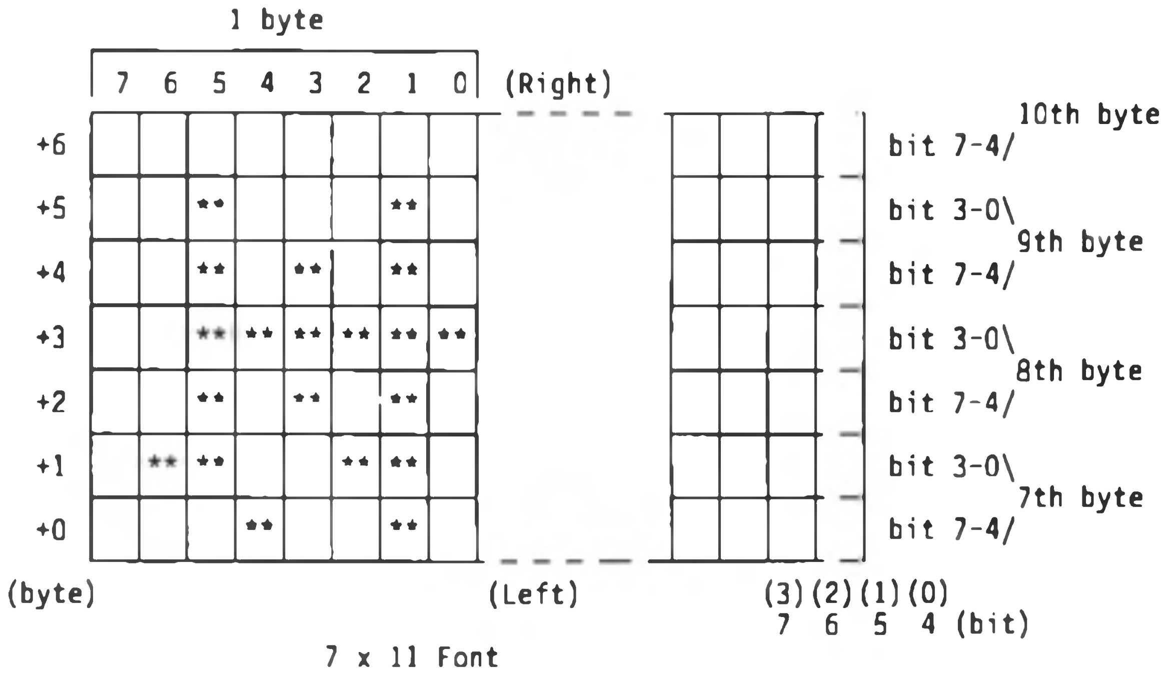


Fig. 4.42 Front format

(3) Character set adjustment to a specific country

EHT-10/2 supports character sets for ten countries including Japan. See Section 2 "CHARACTER CODE TABLE FOR OVERSEAS SPECIFICATIONS" in APPENDIX for the character set used in each country.

The following table lists the character sets to be adjusted to specific countries.

Table 4.7 Characters sets to be adjusted to specific countries

Country name	YLCOUNTRY
U. S. A. (ASCII)	0FH
France	0EH
Germany	0DH
U.K.	0CH
Denmark	0BH
Sweden	0AH
Italy	09H
Spain	08H
Japan	07H
Norway	06H

The character set can be changed by CONOUT (*1) or CONFIG. If the character set is changed, the printer character set is also changed automatically. For Norway, however, the printer character set is set to the ASCII character set.

Address : Variable name (Number of bytes)

F5F3H : YLDFLTC (1)
Default value displayed for each country
This default value is initialized by the system reset and set by CONFIG.

F5F4H : YLCOUNTRY (1)
Current value displayed for each country
These specifications are set by the ESC sequence. YLDFLTC contents are copied during WBOOT processing.

(4) Character generator table configuration

1 Pointer table

EHT-10 and EHT-10/2 have five character generator tables according to the character generator codes. These character generator tables are pointed according to the following pointer table.

Address : Variable name (Number of bytes)

F1B8H : RLCGENX (3)
Pointer data for character generator tables 00H to 1FH

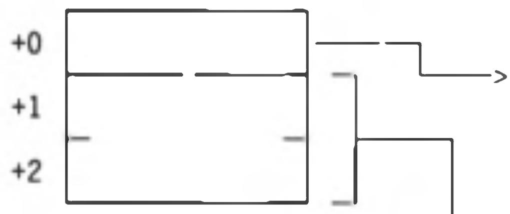
F1BBH : RLCGENN (3)
Pointer data for character generator tables 20H to 7FH

F1BEH : RLCGENG (3)
Pointer data for character generator table containing 80H to 9FH

F1C1H : RLCGENK (3)
Pointer data for character generator table containing A0H to DFH

F246H : RLCGENU (3)
Pointer data for character generator table containing E0H to FFH

The pointer data structure is as follows.



> Font size of one character
EHT-10: 5, 6, or 11 is possible.
EHT-10/2: 5 or 6 is possible

> Character generator table address (8000H or later of the system bank or standard RAM)

2 Character generator table

A character generator table contains font data items in the ascending order of character generator codes.

3 Modifying character generator table contents

By modifying the above pointer table, user-dependent character generator tables can be created.
In this case, tne new character generator tables should be set in address 8000H or later of the standard RAM (subbank 0).
The pointer table value is not initialized until system reset is performed.
The RLCGENU value is set to the default value when an error such as system menu, Alarm, Menu, DLL, or BDOS error occurred.

4.4.6 Displaying cursor

(1) Cursor types

There are four types of cursors, and one of which can be specified by CONOUT (*1).

1 Block cursor that blinks
2 Block cursor that does not blink
3 Underline cursor that blinks
4 Underline cursor that does not blink

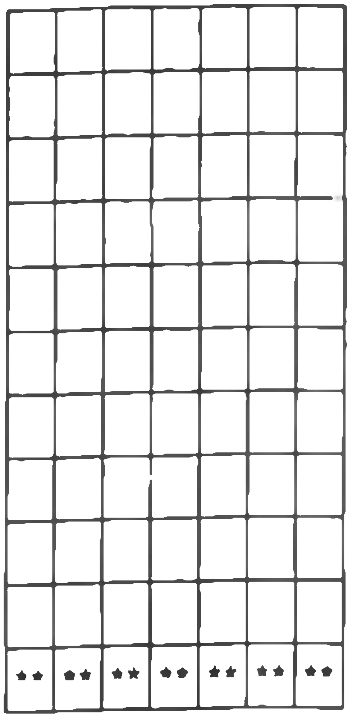
*1 ESC+D6H+n [Setting the cursor type]

(2) Block cursor

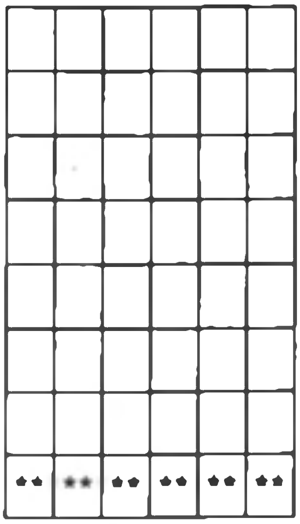
All the contents of one display area are inverted and displayed.
EHT-10 vertical display: 7 (horizontal) x 11 (vertical) dots
EHT-10 horizontal display: 6 (horizontal) x 8 (vertical) dots
EHT-10/2: 6 (horizontal) x 8 (vertical) dots

(3) Underline cursor

The bottom dot line of one display area is inverted and displayed.



EHT-10 vertical display



EHT-10 Horizontal Display
EHT-10/2

Fig. 4.43 Underline cursor

(4) Blinking

Cursor blinking is performed by OVF interruption. The cursor blinks at intervals of approximately 500 milliseconds. The blink interval can be changed by modifying the system area (BLNKTIME) contents.

(5) Cursor-related system work areas

Address : Variable name (Number of bytes)

F075H : BLNKSTAT (1)
Flag which indicates whether the cursor blinks
00H: Blink
01H: Not blink
This flag is referenced by the blink processing routine.

F076H : BLNKCNT (1)
Blink counter
This counter is incremented by 1 wnenever an overflow interrupt is made. When the value of this counter exceeds the BLNKTIME value, the cursor is inverted.

F077H : BLNKRVRS (1)
This indicates the cursor status during blinking.
00H: Cursor ON
FFH: Cursor OFF

F078H : BLNBKTIME (1)
This specifies the blink interval.
The initial value is 04H, and repeats cursor inversion at intervals of approximately 500 milliseconds.

F25FH : LCURSOR (1)
Flag which indicates the cursor status.
Bit 7: Cursor mode
0: Display
1: Not display
Bits 6 to 2: Don't care
Bit 1: Cursor type
0: Block
1: Under line
Cit 0: Specifies blink.
0: Blink
1: Not blink

4.4.7 Details on work areas for displaying

(1) Work areas related to screen modes

Address : Variable name (Number of bytes)

F242H : LDSPMOD (1)
Flag which indicates the current display mode (only for EHT-10)
0: Normal mode
1: Alphanumeric mode
2: Kana mode
3. Calculator mode

F244H : LLCDMOD (1)
Flag holding the current screen mode
0: User screen
1: System screen (normal mode)
2: System screen (direct mode)

F245H : LUWDMOD (1)
Flag which indicates existence or nonexistence of a fixed screen area
0: A fixed screen exists.
1: A fixed screen does not exist.

F24FH : LWORKBFO (47)
Area holding the current screen display status
(See Item (2) for details.)

F27EH : LWORKBF1 (47)
Area used to replace the current screen data with the old screen data when the user screen mode is changed. The configuration of this area is the same as that of LWORKBFO. The contents of this area are replaced by the LWORKBFO contents when the screen is switched between the scroll screen and the fixed screen in EHT-10 or between user screen 1 and user screen 2 in EHT-10/2.

F2ADH : LWORKBF2 (47)
All contents of this area are replaced by the LWORKBF0 contents when the screen is switched between the user screen and the system screen. The configuration of this area is the same as that of LWORKBF0.

F2DCH : LSCMODE (1)
This indicates the current status of the display parameter.
0: System screen (Data is being replaced between LWORKBF0 and LWORKBF2.)
1: EHT-10: Scroll screen
   EHT-10/2: User screen 1
2: EHT-10: Fixed screen
   EHT-10/2: User screen 2
   (Data is being replaced between LWORKBF0 and LWORKBF1.)

F2DDH : LSCMDSV (1)
Area for saving the LSCMODE value when the screen is switched between the user screen and the system screen.

F2DFH : DSPTYPE (1)
Flag which indicates the vertical or horizontal display mode in EHT-10
0: Vertical display mode
1: Horizontal display mode

F2E0H : DSPTPSV (1)
Area for saving the DSPTYPE value when the screen is switched to the system screen in EHT-10

F6A7H : SVCRSR (6)
Area for saving the screen status when the system screen mode is switched from normal to direct

F6B5H : BTRYDSP (2)
Area for saving the old screen status when Battery Fail occurred

(2) Work areas whose contents are saved at screen switching

Address : Variable name (Number of bytes)

F24FH : LSCADDR (2)
Screen buffer head address (address 8000H or later)

F251H : LSCSIZE (2)
Screen buffer size

F253H : LSCRVRAM (2)
EHT-10: VRAM data save area head address
EHT-10/2: VRAM head address
The above address must be saved in address 8000H or later.

F255H : LLNKFLG (2)
Link flag table head address
(Must be saved in address 8000H or later)

F257H : LSCSIZEX (1)
Number of screen buffer columns (fixed value)
EHT-10: 12
EHT-10/2: 20

F258H : LSCSIZEY (1)
Number of screen buffer lines
EHT-10: 14 to 42
EHT-10/2: 4 to 20

F259H : LWDPOSY (1)
Window start line on LCD screen in EHT-10. Ordinate value from 0 to 14

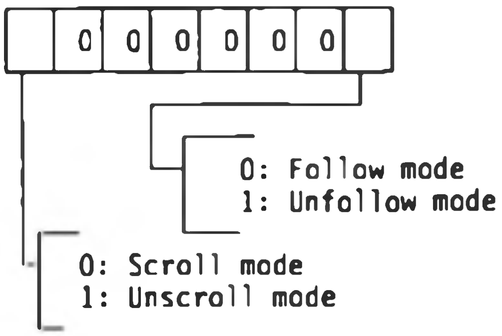
F25AH : LWDSIZEY (1)
Window size
EHT-10: 1 to (14 - LWDSPY)
EHT-10/2: 4 (fixed)

F25BH : LUWDPOXY (1)
Same as LWDPOSY (only for EHT-10)
This area is used by the fixed screen.

F25CH : LUWDSZY (1)
Same as LWDSIZEY (only for EHT-10)
This area is used by the fixed screen.

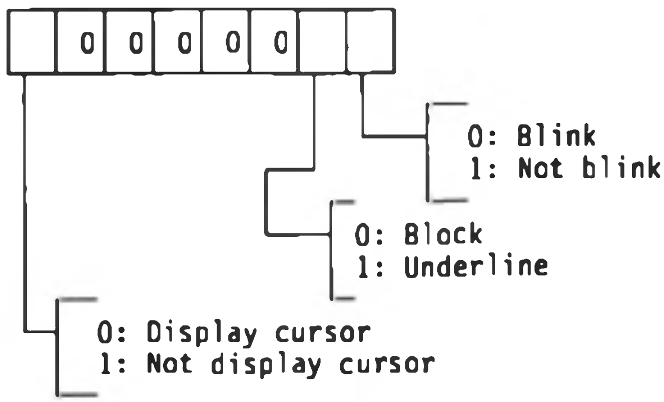
F25DH : LATRCHK (1)
Flag which indicates the existence or nonexistence of attribute data
Bit 7: 0: Existing
1: Not existing

F25EH : LSCROLMD (1)
Flag which indicates the current scroll mode



Specifying follow or unfollow mode is only effective when the scroll mode is specified.
In fixed mode, scrolling by LF cannot be performed.

F25FH : LCURSOR (1)
Flag which indicates the current cursor status



This indicates the horizontal cursor position on the screen buffer.

F260H : LSCCPOSX (1)
This indicates the vertical cursor position in the screen buffer.

F261H : LSCCPOSY (1)
This indicates the horizontal cursor position in the screen buffer.

F262H : LWDXMIN (1)
F263H : LWDYMIN (1)
This indicates the upperleft position of the window in the screen buffer.
LWDXMIN: Horizontal direction (fixed to 1)
LWDYMIN: Vertical direction

F264H : LWDCPOSX (1)
F265H : LWDCPOSY (1)
This indicates the cursor position in the window.
LWDCPOSX: Horizontal direction
LWDCPOSY: Vertical direction

F266H : LVRMYOF : (1)
This indicates the vertical VRAM offset value in EHT-10/2.

F267H : LCURATR (1)
Cursor position attribute data

F268H : LOLDATR (1)
Attribute data for the column immediately before the cursor position

F269H : LCRWAIT (1)
Flag for adjusting the cursor position when CR is specified immediately after one character is displayed in the rightmost column on the screen
0: Being in CR-waiting check status
1: Not being in CR-waiting check status

F26AH : (Reserved)

F26BH : LFKSTAT (1)
F26CH : LFKADDR (2)
The status (LFKSTAT) of the function according to the BIOS CONOUT control code and execution address (LFKADDR)
The status and execution address obtained after retrieval of tables LSCRTB1, LESCTG1, and LESCTB2 are stored.

F26EH : LESCFLG (1)
F26FH : LESCCNT (1)
F270H : LESCPRM (14)
ESC sequence processing work area
LESCFLG: ESC sequence receiving status
- 0: The ESC code has not been accepted.
- 1: Only the ESC code has been accepted.
- 2: The ESC and first parameter (command code) have been accepted.

LESCCNT: This indicates the number of parameters (excluding the command code) accepted by the ESC sequence.
LESCPRM: This is an area for storing the parameters accepted by the ESC sequence.

(3) Work areas used for displaying control

Address : Variable name (Number of bytes)

F074H : DSPFLAG (1)
Display flag for EHT-10/2
00H: LCD display OFF
80H: LCD display ON

F243H : WTONLY (1)
Flag which indicates whether to move the cursor after displaying one character
00H: Move the cursor
01H: Does not move the cursor

F249H : LSCRTB1 (2)
F24BH : LESCTB1 (2)
F24DH : LESCTB2 (2)
Function of the control code used in BIOS CONOUT
Head addresses of the command tables for ESC sequence functions
LSCRTB1: For control codes 00H to 1FH
LESCTB1: For ESC sequences common to EHT-10 and EHT-10/2
LESCTB2: For ESC sequences different between EHT-10 and EHT-10/2

F683H : LVRAMADR (2)
F685H : LVRAMDT (24)
Work area used to read VRAM data
LVRAMADR: Read start address
LVRAMDT: Area to store the read data

F69DH : SCRLFG (1)
The contents of this work area are set to 1 when the screen buffer is scrolled by displaying one character or executing LF.

F69EH : LW_SADDR (2)
F6A0H : LW_DADDR (2)
F6A2H : LW_BPOS (2)
F6A4H : LW_LADDR (2)
Work area used to scroll the screen in EHT-10

4.5.1 Overview

EHT-10 and EHT-10/2 support a printer unit and a printer having an RS-232C interface as LST: devices.
The printer unit is connected to the cartridge interface.
The LST: device can be selected by using CONFIG as well as the I/O byte (see Section 2.6.5).

4.5.2 Functions supported by BIOS

The following BIOS functions are supported:

(1) LIST : Outputs one character to the LST: device.
(2) LISTST : Checks the LST: device status.
(3) SCRNDUMP: Dumps the screen contents (not supported by EHT-10).
(4) KANJI : Prints out kanji characters.

See Section 4.2 for details on the BIOS functions.

4.5.3 Character set adjustment to a specific country

(1) Adjusting a character set to a specific country

In EHT-10 and EHT-10/2, the printer character set is automatically changed according to the display character set by outputting the international character specification sequence. (ESC+'R'+n) to the printer.
The international character specification sequence is output when the first LIST is output under one of the following conditions:

1 After the LST: device contained in the I/O byte is modified
2 After warm boot
3 After power on (*1)
4 After the display character set is changed

Table 4.8 shows the relationship between display character sets and international character specification codes (value n of ESC+'R'+n).

*1 Note that, if the power is turned off while the application program is outputting the ESC sequence, the application program outputs the international character specification sequence when the power is turned on subsequently. Therefore, subsequent data may not be output correctly by the printer. (This symptom occurs only when outputting to the RS-232C interface.)

Table 4.8 Display character sets and international character specification codes

Display character set	Printer unit	RS-232C
ASCII	0	0
France	1	1
Germany	2	2
U.K.	3	3
Denmark	4	4
Sweden	5	5
Italy	6	6
Spain	7	7
Japan	8	8
Norway	0(*2)	9

*2 For a printer unit in Norway, the display character set is set to ASCII. (This is because the printer unit does not support Norway.)

(2) Prohibiting character set adjustment to a specific country

Registering a character set for a specific country can be prohibited by modifying the contents of system parameter PRTINIT (F1CCH).

```
PRTINIT   | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
            |   |
            |   └──────> Printer Unit
            └──────────> RS-232C
                        Bit assignment
                        0: Does not perform character set adjustment
                        1: Perform character set adjustment
```

4.5.4 Printer unit

(1) Overview

The printer unit is a cartridge option, and connected to the cartridge interface when used. The cartridge is used in HS mode (*1), and the device code (*2) is 07H.
The printer unit is micro dot printer Model-180 capable of printing 24 columns and copying two sheets. Because LA-180 is used as the printer controller, the application program can control the printer unit in a variety of ways. Because the printer unit consumes not a small amount of current, the controller is placed in power down mode to save power consumption when the printer is not used for a fixed time.
Data is output to the printer by making ready interrupt, and this enables the system to perform printing concurrently with other processing.

*1 and *2 See Section 7.3 for cartridge mode and device codes.

(2) Functions supported by the system

1 Character set adjustment to a specific country

See Section 4.5.3. If the display character set is set to Norway, the system automatically sets the display character set to ASCII because the printer unit does not support Norway.

2 Screen dump function

This function is supported only by EHT-10/2.

3 Paper feed function

If the paper feed (FEED) key is pressed, the paper is fed one line (8 dots). The paper feed function is enabled only by pressing a desired key to which function key code F8H has been assigned.

4 Concurrent processing during printing

The system has a printer unit output buffer (PRNBUF) consisting of 128 bytes, and from which data items are fetched one by one by the ready interrupts made by the printer. This enables the system to perform printer output processing concurrently with other processing. When the output buffer is full of data, output processing is suspended until the buffer full state is released. The output buffer state can be checked by using LISTST.

Address : Variable name (Number of bytes)

FB2AH : PRNBUF (128)
- Printer unit output buffer
  Put Pointer: PPUTPTR
  Get Pointer: PGETPTR

F4BDH : PPUTPTR (2)
- Pointer to store data in the printer unit output buffer (PRNBUF)
- This pointer is post-incremented so that it always points the head of the empty area.
- The buffer is assumed to be full when PPUTPTR+2=PGETPTR

F4BFH : PGETPTR (2)
- Pointer to fetch data from the printer unit output buffer (PRNBUF)
- This pointer is pre-incremented so that it always points the subsequent data item to be fetched.
- The printer unit output buffer is assumed to be empty when PPUTPTR=PGETPTR.

5 Power-down function

The power-down function saves the power consumption of the printer unit. If the printer is not used for a fixed time (default value is three minutes), the system sets the printer to power-down mode. The power-down time (the time from when the printer unit is used most recently to when the power-down mode is set) can be specified by CONFIG or by directly modifying the PRNPWTM contents (F025H) by using the application program. However, the printer unit is set to power-down

mode approximately five seconds after the printer unit power is first turned on, regardless of the specified power-down time.
The user need not be concerned with power-down mode because the power-down mode is automatically released by the system when a data output request is issued for the printer.

Note:
Output one data line in a batch as far as possible so that the interval between data outputs does not exceed the power-down time. If the power-down time has expired midway through outputting one data line, the printer unit prints the data that it received before the power-down mode was set and then feeds a line. Consequently, subsequent data is printed starting at the head the new line.

[Remarks]
The power-down time is monitored by one-second interruption. When the power-down time has expired, value -1 is set in PRNPWFLG (F3BEH), indicating that the power-down time has expired.
The printer unit power is actually set to power-down mode by PSTBIOS and key input routine (being in key input wait status).

Address : Variable name (Number of bytes)

F025H : PRNPWTM (1)
- Power-down time (initial value)
- Unit: Second
- Default: 80 (seconds)
- A value from 0 to 255 can be specified.
- If value 0 is specified, the power-down mode is not set.

F3BDH : PRNPWCNT (1)
- Counter which indicates the time left up to power-down time expiration. First, the PRNPWTM value is copied into this counter, following which the counter value is decremented by each one-second interruption. When the value of this counter becomes 0, the power-down mode is set.

F3BEH : PRNPWFLG (1)
- Flag which indicates the printer unit power status
  1: Normal mode
  0: Power-down mode
  -1: The power-down time has expired (but the printer has not been set to power-down mode).

6 Power off in continue mode

The printer unit is reset when the mainframe power is turned on, and all states and data that have been set for the printer such as character set adjustment to a specific country and external character definition are initialized. The system output buffer contents are also cleared.
Therefore, note that the specifications made for the printer unit are not held effective. To initialize the printer at power on, use cartridge device HOOK. See Section 10.2 for HOOK.

(3) Control functions

Table 4.9 lists the I/O ports used by the printer unit. The printer unit can execute the following functions through these I/O ports.

1 Switching between offline and online

When $\overline{\text{SLIN}}$ is set to 1, the printer is set to offline mode (*1).

When $\overline{\text{SLIN}}$ is set to 0, the printer is set to online mode.

*1 (a) The printer unit cannot be accessed when the development unit is used.

(b) The printer unit is set to online mode by the following operations:
- Turn the power on.
- Execute DLL.
- Press the PF (paper feed) key.

2 Paper feed function

When the PF key is set to 1 for 50 ms or more in offline mode, paper feeding is started.

3 Buffer-full function

When the same amount of print data (including spaces) as the total number of printer columns (24 columns for character data or 18 bytes for graphic data) is input, the printer prints printer buffer data and feeds paper.

4 Emergency stop function

If one of the abnormalities listed below is detected while the printer is operating, the power that has been supplied to the print head and drive motor is cut, and the printer is set to power-down mode. In this case, reset the controller by turning the mainframe power off and then on to release the emergency stop status.

(a) Motor lock
(b) Malfunctioning of the timing detector (signal ungeneration)
(c) Malfunctioning of the reset detector (signal ungeneration)

Port Address (R/W) : Port name (RAM data address)

P16H (R) : IOSTR
bit 0: (PBUSY): Printer busy signal
1: Busy
0: Ready

P17H (W) : ICCTLR (F00BH)
bit 6:(PRIE): Printer interrupt control
1: Disable
0: Enable

P19H (W) : IOCTLR (FODDH)
bit 0:(PF):Paper feed signal

bit 1:($\overline{\mathrm{SLIN}}$):Printer select-in signal

bit 3:($\overline{\mathrm{PINI}}$)Cartridge reset and power-down mode release signal

P23H (R) : ITSR
bit 1:(IPBUSY): Printer busy signal
Same as IOSTR pbusy

P10H (W) : CHSOR
bit 0 to 7 :Printer output data

P11H (R) : CHSSR
bit 0:(OBF): Out put buffer status
1:Busy
0:Ready

(4) Control commands

Printer unit control commands are listed below.

1 Print (LF: 0AH)

This command prints printer buffer data and feeds paper.

2 Printing enlarged characters (SO: 0EH)

This command prints subsequent print data in double-width enlarged characters. In double-width enlarged character mode, up to 12 characters can be printed in one line. The double-width enlarged character mode is released by control code DC4 (14H), LF (0AH), DC2 (12H), or DC3 (13H).

3 Cancelling input data (CAN: 18H)

This command cancels all data input in the same line.

4 Releasing double-width enlarged character mode (DC4: 14H)

This command releases the double-width enlarged character mode.

5 Power down function 1 (DC2: 12H)

This command sets the controller to power-down mode. The power-down

mode is released by keeping $\overline{\mathrm{PINI}}$ (bit 3 in address 19H of the I/O register) to 1 for five microseconds or more. After the power-down mode is released, the controller is restored to the status set before it was set to power-down mode.
Even if the controller is set to power-down mode by DC2, controller oscillation does not stop.

6 Power-down function 2 (DC3: 13H)

This command sets the controller to power-down mode. The power-down

mode is released by keeping $\overline{\text{PINI}}$ (bit 3 in address 19H of the I/O register) to 1 for one millisecond or more. After the power-down mode is released, the controller is restored to the status set before it was set to power-down mode.
If the controller is set to power-down mode by DC3, controller oscillation stops.

[Remarks]

The amount of power consumption in normal mode and that in power-down mode are as follows.
Normal mode: approx. 5 mA
Power-down mode
    When set by DC2: approx. 1.5 mA
    When set by DC3: approx. 1 microA

Therefore, considerably larger amount of power can be saved when the controller is set to power-down mode by DC3 than by DC2. However, more time is required to return the power-down mode set by DC3 to normal mode.

Notes:
Because the printer unit power is automatically controlled by the system, the user usually need not use control codes DC2 and DC3. Note the following when controlling the printer unit power by using control codes DC2 and DC3 in the user program:

(a) Be sure to set the system power-down time to infinity (PRNPWTM=0).
(b) Release the power-down mode by using the application program. When accessing an I/O port to release the power-down mode, access it in the I/O access procedure explained in Chapter 7.

7 Escape alphabet control commands

Printing can also be controlled by escape alphabet control commands each consisting of ESC (1BH) followed by an alphabetic code and binary data. Value n in the escape alphabet command indicates a one-byte binary data. The plus sign (+) is only given in this document as a separator for clarity, and need not be entered nor output actually.
If an escape alphabet command is input midway in a line, the printer unit prints the data input up to the command entry and then terminates the previous sequence. Therefore, the specification made by the entered escape alphabet command is made effective for the subsequent lines.

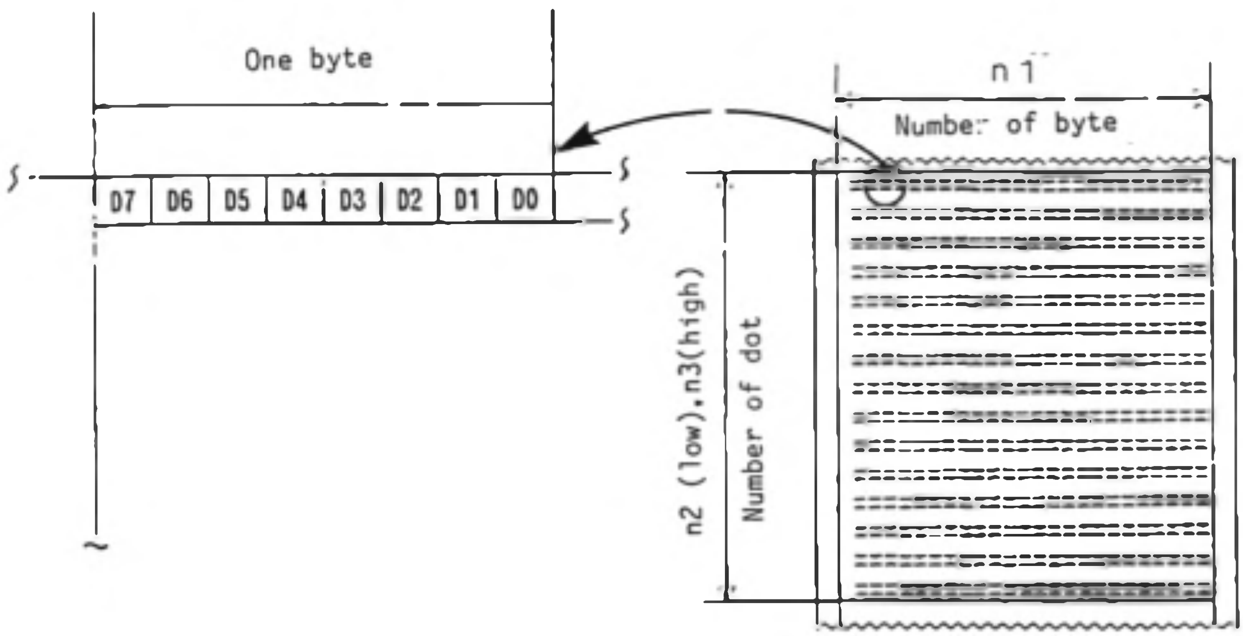
(a) Setting line space (ESC+'A'+n)
This command sets the line space for each dot line. Value n must satisfy the following condition:
$1 \leq n \leq 255$
If continuous printing can be performed, value 0 can be specified as n.

(b) Transferring bit image data (ESC+'K'+n1+n2+n3)
This command processes subsequent data as bit image data. Values n1,n2 and n3 indicate the amount of bit image data to be transferred as follows:

n1: Number of horizontal bytes ($1 \leq n \leq 18$)
n2: The low-order byte for the number of vertical dot lines ($0 \leq n \leq 255$)
n3: The high-order byte for the number of vertical dot lines ($0 \leq n \leq 1$)

After all bit image data is transferred, the printer is automatically returned to text mode.

Note:
In case of bit image data transmission, the printer may operate abnormally if all transmission data is not sent consecutively. The data of one dot line must be sent at transfer rate 18 bytes/15 ms or more.



D0 to D7 indicate dot positions. To print a dot at a dot position, binary 0 must be set for the position. To print a space at a dot position, binary 0 must be set for the position.

Fig. 4.44 Relationship between data and print out

(c) Setting international character set (ESC+'R'+n)
This command sets the character set of the specified country.

Table 4.10 Print character sets and international character specification codes.

n	Country
0	U. S. A.
1	France
2	Germany
3	U.K.
4	Denmark
5	Sweden
6	Italy
7	Spain
8	Japan

Value n greater than 8 is ignored, and the previous n value remains effective.

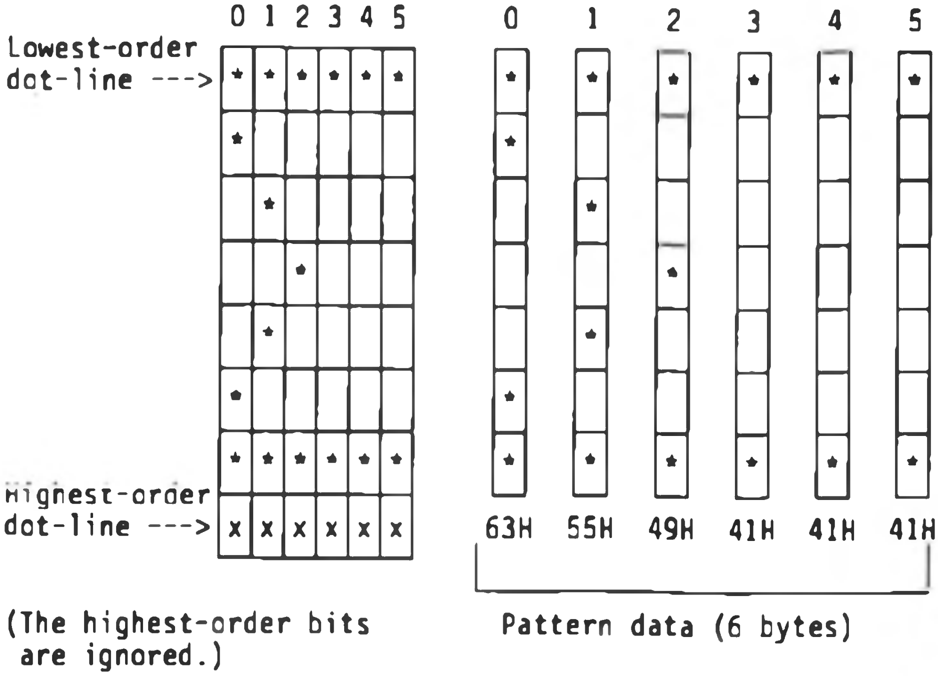
(e) Paper feed command (ESC+'B'+n)
This command feeds the paper n dot-lines. Value n must satisfy the following condition: $1 \leq n \leq 255$

(f) Transferring external character registration data (ESC+'&'+n1+n2)
By entering this command followed by pattern data, an arbitrary pattern consisting of a 6 x 7 dot matrix can be registered in LSI.
Up to eight characters can be registered in desired addresses of the address area (20H to FFH). If a new character pattern is registered in an address in which another character pattern has already been registered, the old pattern is cleared and the new one is made effective. If more than eight character patterns are registered, all external character data that has been registered is cleared.

[Setting addresses]
The address that has been set matches the character code and, after registration, can be accessed like other fixed characters. If a fixed character has been defined in the address that has been set, the fixed character is made ineffective. Values n1 and n2 indicate the registration start and end addresses, respectively. Therefore, the number of characters to be registered is determined by values n1 and n2, and up to eight characters can be registered in addresses from n1 to n2.

[Pattern data configuration]
Each pattern data to be registered consists of 6 x 7 dots that share six bytes. This pattern data is vertically divided into six portions each consisting of one byte, and transferred as 6-byte data as a total.

<Example> Assume transmission of the following pattern data.



[Data transmission system]

Registering one character

Select the address (character code) to be defined from 20H to FFH, and assume the address as A1. When registering one external character, the registration start address (n1) matches the registration end address (n2).

<Example>
Assume that a 6 x 7 dot matrix full-dot pattern is to be registered in address 41H (fixed character code 'A'). (The values are represented in hexadecimal notation.)

```
 1B,  26,  41,  41,  FF,  FF,  FF,  FF,  FF,  FF

ESC  '&'   n1   n2   Pattern data (6 bytes)
```

If character code 41H is specified in subsequent control, data is printed in a 6 x 7 dot matrix full-dot pattern. (Character 'A' cannot be accessed.)

Registering multiple characters

By repeating one character registration, up to eight characters can be registered. To register multiple characters in consecutive addresses (character codes) starting from registration start address n1 and ending with registration end address n2, register (n2-n1+1) characters of pattern data sequentially. In this case, the following condition must be satisfied:
$n1 < n2,\ n2 - n1 \leq 7$

<Example>
Assume that three 6 x 7 dot matrix full-dot patterns are to be registered in addresses E4H to E6H (the digits are represented in hexadecimal notation).

```
 1B,  26,  E4,  E6,  FF....... FF,  FF....... FF, FF....... FF
                     |____________|  |___________| |___________|
ESC  '&'   n1   n2
                    Pattern         Pattern        Pattern
                    registered in   registered in  registered in
                    E4 (6 bytes)    E5 (6 bytes)   E6 (6 bytes)

                            Total 6 x 3 = 18 bytes
```

4.5.5 Outputting data to RS-232C interface

(1) Overview

By specifying RS-232C as printer output in CONFIG or I/O byte, data can be output to an ordinary terminal or portable printer having an RS-232C interface. In EHT-10 and EHT-10/2, the Busy signal of the printer is checked by RS-232C DSR and, when DSR is active, data is output. The system supports BIOS functions LIST, LISTST, SCRNDUMP, and KANJI (kanji print).
SCRNDUMP and KANJI cannot be used in some types of printers because they use printer control codes (see Item (2)). As for kanji character printing by KANJI, a non-kanji printer can also be used because the kanji fonts stored in EHT-10 or EHT-10/2 are output as bit image data.

(2) Connectible printers

Although the terminal printers (*1) and portable printers produced by Seiko Epson can be basically connected to the system, note the following because some control odes and graphic characters are different:

1 Graphic characters

Fonts 80H to 9FH may differ according to the printer. For the printers for which fonts can be down-loaded, use the printers after down-loading the fonts as required.
Fronts E0H to FFH cannot be printed in any printers. Use these fonts after down-loading as required.

2 SCRNDUMP (screen dump)

The screen dump function uses the control codes listed below. The printers that do not have these control codes cannot output screen dump data. Although any specific problems are not caused when the printer does not have control code ESC+'2', subsequent paper feed rate becomes 8/72 inches.

(a) ESC+'K'+n1+n2 (single-density bit image mode)
(b) ESC+'A'+n (paper feed rate is set to n/72)
(c) ESC+'2' (paper feed rate is set to 1/6)

3 KANJI (kanji print)

The kanji print function uses control codes (a) and (b) above. The printers that do not have these control codes cannot output kanji characters.

4 Character set adjustment to a specific country

The following control code is used to adjust the printer character set to the mainframe:
ESC+'R'+n (international character specification)
Character set adjustment to a specific country can be suppressed when it is unrequired or the printer does not have the above control code. (See Section 4.5.3.)

*1 All terminal printers having a parallel interface produced by Seiko Epson can be controlled by the RS-232C after connected with the serial interface board supplied as an optional device.

(3) Setting serial parameters

Like other I/O devices, the system uses the system default values for the serial parameters used for outputting data to the RS-232C interface of the printer.
The system default values can be modified by using CONFIG. The system default values are listed below.

Baud rate: 4800 bps
Data length: 8 bits
Parity bit: None
Stop bit: 2 bits

4.6 *User BIOS*

4.6.1 Overview

In EHT-10 and EHT-10/2, a user BIOS entry is allocated so that the user can extend BIOS and add new entries to it.

The user BIOS area can also be allocated as the area for storing the user BIOS processing section.
The user BIOS area can also be used by machine-language routines (e.g., barcode input routine) used by multiple programs and hook extend processing (e.g., extensions of communication protocols and IC card protocols).

This Section explains the structure of the user BIOS and how to use the BIOS area.

4.6.2 User BIOS

(1) User BIOS expansion procedure

The user BIOS can be extended in the following procedure:

1 Determine the user BIOS extend processing area.
2 Load the user BIOS extend processing routine.
3 Update the user BIOS entry address and sets the new address as the extend processing start address.

(2) User BIOS extend processing area

The user BIOS extend processing area can be reserved in one of the following three areas.

1 User BIOS area

If the user BIOS extend processing area is reserved in the user BIOS area, the user BIOS processing that has been set by one application program can also be used by other application programs.
See Section 4.6.3 for the user BIOS area allocation procedure.
Be sure to pay attention to the notes provided in Section 4.6.3 when using the BIOS area.

2 TPA (user area)

If the user BIOS expand processing area is reserved in the user BIOS area, the extended user BIOS processing is made local and cannot be used by other application programs. In this case, be sure to return the user BIOS entry address to its original address in the procedure explained in Item (6) when application program execution is terminated.

3 Part of the application ROM

Part of the application ROM can also be used as the BIOS extend processing area. In this case, the application program must take note of the bank where the extend processing routine resides when using the user BIOS.

(3) Modifying entry address

EHT-10 and EHT-10/2 each has two BIOS entries RBIOS1 and RBIOS2. To extend the user BIOS, update the RBIOS1 jump table and sets the jump address in the extend processing start address area reserved in Item (2).

WBOOT + 7EH	C3H	Jump instruction code
WBOOT + 7FH		Jump address ---> Extend processing start address
WBOOT + 80H		

The WBOOT address is contained in addresses 0001H and 0002H.

(4) User BIOS call procedure

1 Calling by a load and execute program

To execute the user BIOS after loading it in TPA, obtain the WBOOT entry address from the contents of addresses 1 and 2 and call the (WBOOT + 7EH) address in the same way as that for other BIOSs. In this case, the user can freely use other BIOSs within the user BIOS. When control is returned to the user BIOS, the user stack is used.

2 Calling by a ROM-based program

Like the call procedure 1 , obtain the entry address and call the user BIOS entry address by using BIOS CALLX (*1). Set the bank information to be set by CALLX in bank 0#0 (00H). (*2) After the user BIOS is called by BIOS CALLX, other BIOSs cannot be used within the user BIOS (*3). When control is passed to the user BIOS, the BIOS stack is used.

*1 See Section 4.2 for CALLX.
*2 Bank 0#0 is subbank 0 of bank 0, which is a standard RAM bank.
*3 Use system jump table JSCALLX when calling the user BIOS which uses a BIOS from a ROM-based program. See Section 10.3 for JSCALLX.

(5) Starting user BIOS extend processing

Note the following when performing user BIOS extend processing:

1 The user BIOS is terminated by the RET instruction and returns control to the user program.

2 If the stack was modified during user BIOS execution, return to the stack to its original status when user BIOS execution is terminated.

3 If the user BIOS is called by BIOS CALLX, the user BIOS is in DI status when it received control. Set the user BIOS to EI status as required.

(6) Terminating user BIOS extend processing

If user BIOS extend processing is no more required when the application program is terminated, be sure to disconnect the user BIOS extend processing routine from the user BIOS in the following procedure.

1 Return the user BIOS entry address to its original one as shown below.

1. Jump instruction code
2. RBIOS2 user BIOS entry address

2 When the user BIOS area is used
If the user BIOS area is used as the user BIOS extend processing area, be sure to release the are in the procedure explained in Section 4.6.3.

WBOOT + 7EH	C3H	Jump instruction code
WBOOT + 7FH	81H	Entry address of RBIOS2 USERBIOS
WBOOT + 80H	EBH	

(7) Initializing the user BIOS

Because the operating system initializes the BIOS entry at system reset or system initialization, the user BIOS entry is also initialized. After initialization, the user BIOS does returns only.

4.6.3 User BIOS area

(1) User BIOS area position

The user BIOS area is allocated starting in the address immediately after the RAM disk area and ending in address DBFFH for EHT-10 or DDFFH for EHT-10/2.

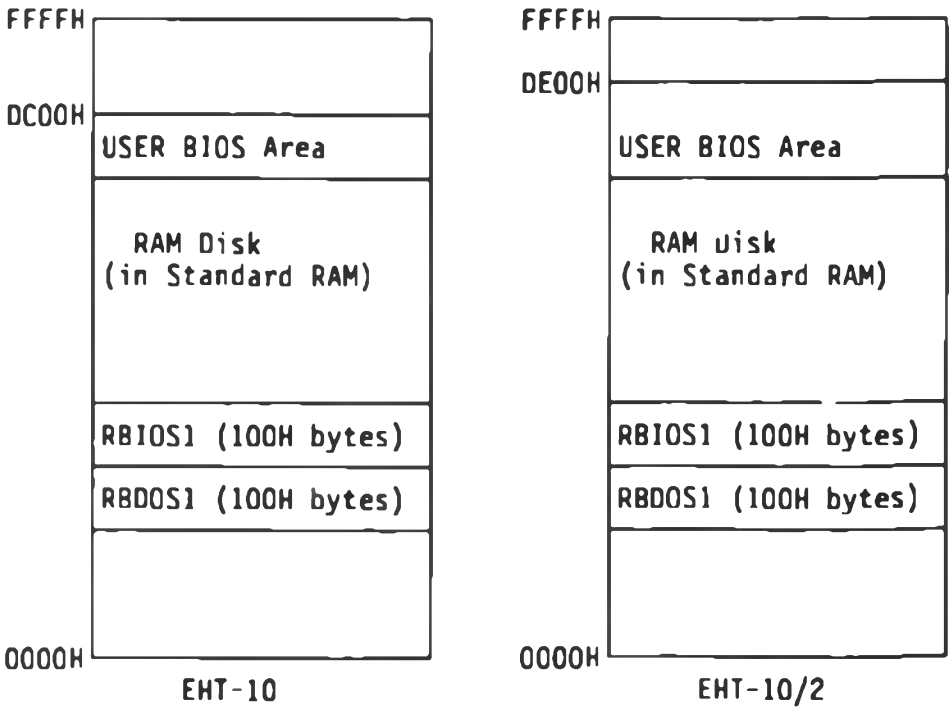


Fig. 4.45 User BIOS area

(2) User BIOS area size

The user BIOS area size can be set in units of pages (256 bytes) by the user. (The default value is 0 page.) The maximum user BIOS area size is determined by the size of the standard RAM area in the RAM disk.

1 EHT-10

User BIOS area size + RAM disk standard RAM area size ≤ 39.5 Kbytes

2 EHT-10/2

User BIOS area size + RAM disk standard RAM area size ≤ 40 Kbytes

(3) Allocating user BIOS area

The user BIOS area can be allocated in one of the following three methods.

1 Allocate by using CONFIG at system initialization.

2 Select CONFIG on the system menu screen, and allocate it there.

3 Allocate by using the user program.

(4) Allocating user BIOS area by using a user program

Figure 4.46 shows the procedure to allocate the user BIOS area by using a user program.

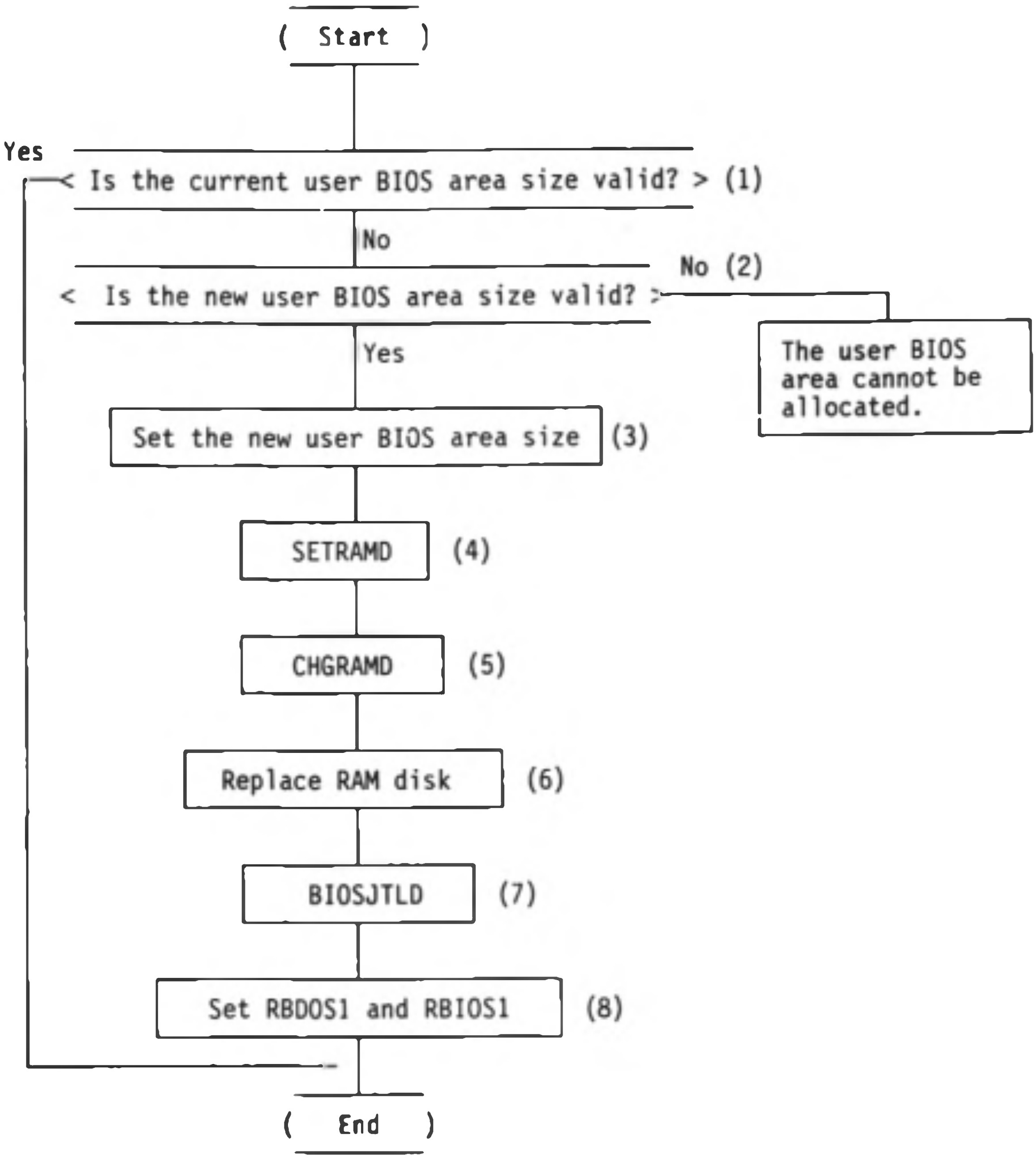


Fig. 4.46 User BIOS area allocation procedure

Step Explanation

(1) Checking current user BIOS area size
- Check whether the current user BIOS area size satisfies the size of the area to be allocated.
- The current user BIOS area size can be determined from the USERBIOS (F00CH) contents. (The unit is 256 bytes.)

(2) Checking new user BIOS area size
- Check whether the sum of the new user BIOS are size and the RAM disk standard RAM area size satisfies the following condition:
EHT-10: $<$ 39.5 Kbytes
EHT-10/2: $\leq$ 40 Kbytes

If the sum exceeds the maximum value, the new BIOS area cannot be allocated.
- The current RAM disk standard RAM area size can be determined from the SIZRAM (F00BH) contents. (The unit is 1 Kbytes.)

(3) Setting new user BIOS area size
- Set the new user BIOS area size in USERBIOS (F00CH) in units of 256 bytes.

(4) SETRAMAD
- Set the CP/M size and RAM system parameters.
- SETRAMAD resides in the jump table allocated in the system bank, and its address is 0016H.
- Call SETRAMAD by using BIOS CALLX.

(5) CHGRAMD
- Change the position of the RAM disk standard RAM area according to the modification of the user BIOS area size.
- CHGRAMD resides in the jump table allocated in the system bank, and its address is 0019H.
- Call CHGRAMD by using BIOS CALLX.
- CHGRAMD entry parameters
  A register = 2 : Position modification
  B register = (CRAMD_SIZE: F068H) : Extended RAM size
  C register = (CSIZRAM: F00BH) : RAM disk standard RAM area size

(6) Replace RAM disk
- Move information in a standard RAM as changing the size of user BIOS

(7) BIOSJTLD
- Load the BIOS jump table in RAM.
- BIOSJTLD resides in the jump table allocated in the system bank, and its address in 0013H.
- Call BIOSJTLD by using BIOS CALLX.

(8) Setting RBDOS1 and RBIOS1
- Set the RBDOS1 and RBIOS1 entry addresses.
- BDSLAD (F005H) + 6 and BDSLAD (F005H) + 7 ---> (0006H and 0007H)
- BI1LAD (F007H) + 3 and BI1LAD (F007H) + 4 ---> (0001H and 0002H)

Notes:

1 RBIOS2 can be used to call the BIOS command for allocating the user BIOS area, and RBIOS1 cannot be used for this purpose. This is because the RBIOS1 area is relocated due to user BIOS area size modification and the RBIOS1 entry address becomes undefined.

2 After the user BIOS area is allocated, do not call 0005H to execute BDOS until WBOOT is executed. This is because, after the user BIOS area size is modified, RBDOS1 is not loaded until WBOOT is executed. Use RBDOS2 BDOS when using BDOS.

Remarks:

The system work areas related to user BIOS area size modification are listed below.

Address : Variable name (Number of bytes)

F00BH : SIZRAM (1)
RAM disk standard RAM area size
EHT-10: 0 ≤ SIZRAM ≤ 39 Kbytes
EHT-10/2: 0 ≤ SIZRAM ≤ 40 Kbytes
The unit is 1 Kbytes.

F068H : RAMD_SIZE (1)
RAM disk extended RAM area size
0 ≤ RAMD_SIZE ≤ 6
The unit is 32 Kbytes.

F00CH : USERBIOS (1)
User BIOS area size
EHT-10: 0 ≤ USERBIOS ≤ 158
EHT-10/2: 0 ≤ USERBIOS ≤ 160
The unit is 256 bytes.

FC05H : BDSLAD (2)
RBDOS1 loading address

F007H : BI1LAD (2)
RBIOS1 loading address

F05CH : TOPRAM (2)
User BIOS area head address

(5) User BIOS area usage procedure

1 Overview

The user BIOS area cannot be used simultaneously by multiple programs. To manage sharing the user BIOS area by multiple programs, the user BIOS area has a header in its end. (This header must be added by the user during user BIOS creation.)
The application program that is to use a program or data stored in the user BIOS area can determine whether the corresponding program or data really resides in the area. This header is also used to determine whether any other program has already used the user BIOS area when loading a program or data in the area.
The user BIOS area contents remain unchanged until the system is initialized or the user BIOS area size is modified.

2 Header structure

(1)	(2)	(3)	(4)	(5)	(6)	(7)
'U' 'B'	routine name	Size	*1	Release address	00H	*2

*1 = Over write flag
*2 = Check sum

The header consists of 16 bytes.
Because the bottom address of the user BIOS area is fixed, the header always resides in the following address area:
EHT-10: DBF0H to DBFFH
EHT-10/2: DDF0H to DDFFH

(a) Header contents

No. : Address of EHT-10, Address of EHT-10/2 : Item (Number of bytes)

(1) : DBF0H, DDF0H : Header ID (2)
- ID which identifies a header
- This ID is fixed to 'UP' (ASCII).

(2) : DBF2H,DDF2H : Routine name (8)
- Name which indicates the name of the routine loaded in the user BIOS area
- A desired name can be set in ASCII codes.

(3) : DBFAH,DDFAH : Size (1)
- Size of the routine loaded in the user BIOS area
- This size can be set in units of 256 bytes in binary notation.

(4) : DBFBH,DDFBH : Over write flag (1)
- Flag which indicates whether a new routine can be overwritten on the routine that has already been loaded.
00H: Prohibits loading of a new routine.
Not 00H: A new routine can be loaded after release processing is performed.

(5) : DBFCH,DDFCH : Release address (2)
- Address of the routine to be executed before loading a routine when the user BIOS area has already been used by another routine
- The release processing can be only performed when the Overwrite flag is not 00H.
- Release addr. must be an address within the user BIOS area. Release processing must terminate with a return instruction.

(6) : DBFEH,DDFEH : Unused (1)
- Fixed to 00H

(7) : DBFFH,DDFFH : Check sum
- The contents of each byte of the 15 bytes from the header head through the field immediately before the Check sum field (CBF0H to CBFEH) are subtracted from 00H, and the results are stored in this field.

(b) Detailed on Overwrite flag

The Overwrite flag must be set to 00H (prohibiting loading a new routine) when the routine must be resident after loaded in the user BIOS area. The routine for which the above flag has been set can be deleted from the user BIOS area only by the program that loaded the routine or after the system is initialized.

The Overwrite flag must be reset to a value other than 00H (a new routine can be loaded after release processing is performed) when the system area can be restored to its original status by performing release processing and, after that, a new routine can be loaded without causing any problems.

(c) Details on release processing

When a routine stored in the user BIOS area modifies the system area contents, the system area contents must be saved in the user BIOS area before the start of modification.
Release processing (processing executed by the routine indicated by Release addr.) is performed to restore the saved system area contents in the system area, thus restoring the system to the status set before the user BIOS routine was loaded. After that, the release processing routine initializes the contents of all header fields to 00H.
The header contents must be cleared even when the system area contents need not be restored to the original. Release processing must be performed using the last 256 bytes of the user BIOS area.
Release processing must terminate with a RETURN instruction.

3. Use procedure

The application program that loads a routine in the user BIOS area performs processing (see Figure 4.47) to check whether the user BIOS area can be used.

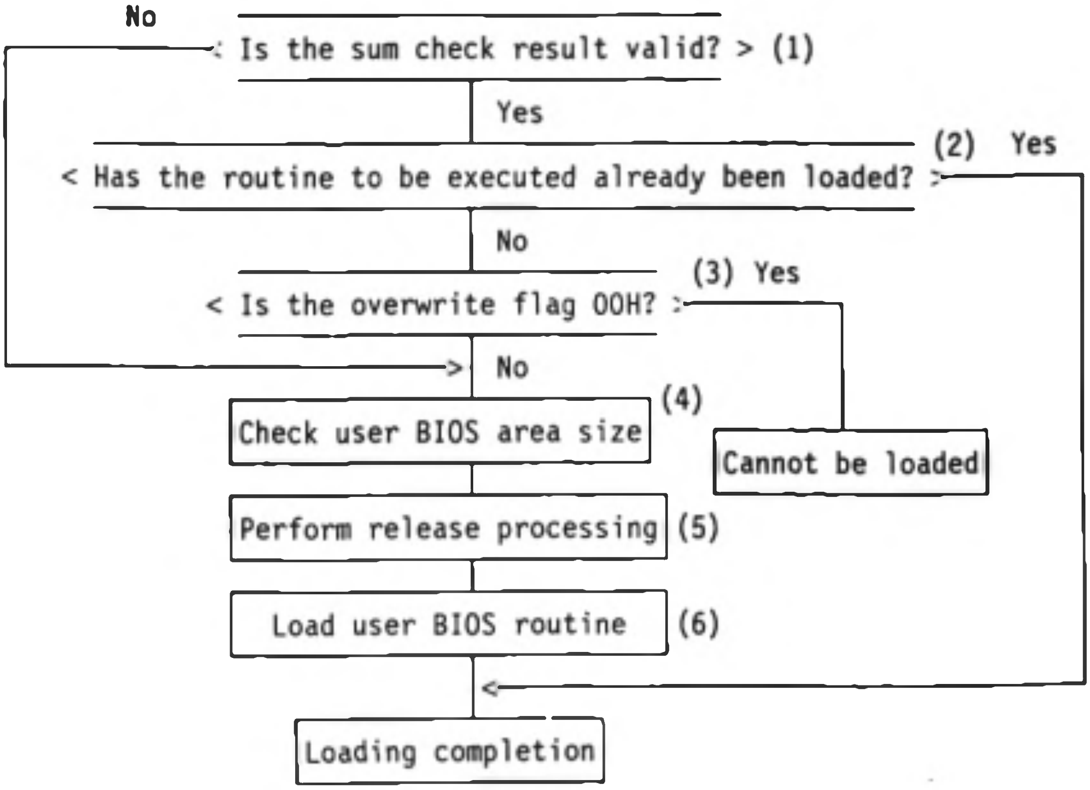


Fig. 4.47 User BIOS header check

Step : Explanation

(1) Checking header
- Sums the contents of all header bytes and checks whether the result becomes 00H. (Sum check)
- If the sum check result is valid, check whether the first two header bytes contain 'UB'.

(2) Checking whether the routine has been loaded
- Check the routine-name field of the header area to determine whether the routine to be executed has already been loaded.
- This check step can be omitted.

(3) Checking overwrite flag
- Check the Overwrite flag of the header area. If overwriting cannot be performed, the routine is not loaded.

(4) Checking user BIOS area size
- Allocate the user BIOS area in the procedure explained in Section 4.1.3 (2).

(5) Release processing
- Call the routine indicated by Release addr of the header area.

(6) Loading user BIOS routine
- Load the new routine in the user BIOS area and create a new header area.

CHAPTER 5 BDOS PROCESSING

5.1 Overview

The operating systems used in EHT-10 and EHT-10/2 are enhanced versions of CP/M Version 2.2.
In EHT-10 and EHT-10/2, two BDOS reside in different locations so that:

(1) The upperlimit of the usable RAM memory space (TPA) where an ordinary CP/M application program as is can operate can be indicated. (RBDOS1)

(2) The ROM-based program can call BDOS without taking note of bank switching. (RBDOS2)

In EHT-10 and EHT-10/2, extended devices have some restrictions on use.
This chapter mainly explains the specifications unique to EHT-10. See APPENDIX 7 for BDOS functions.

5.2 BDOS Processing Flow

When BDOS is called by an application program in EHT-10 or EHT-10/2, the control is stored in BDOS allocated in RAM, following which the bank is switched and the actual BDOS stored in the OS ROM allocated in the system bank is called.
When processing is terminated, control and return information are returned to the bank first called and then returned to the application program.
The BIOS being used by the BDOS stored in ROM directly calls the BIOS stored in the OS ROM.

The application program calls BDOS in the following procedure.

(1) The load-and-execute program calls JMP RBDOS1 stored in address 0005H.

(2) The ROM-based program calls JMP RBDOS2 stored in address FF90H.

The procedure to use BDOS called by a load-and-execute program is the same as the procedure to use BDOS called by a ROM-based program.
Figure 5.1 shows the BDOS processing flow from when BDOS is called by an application program till when control is returned to the application program.

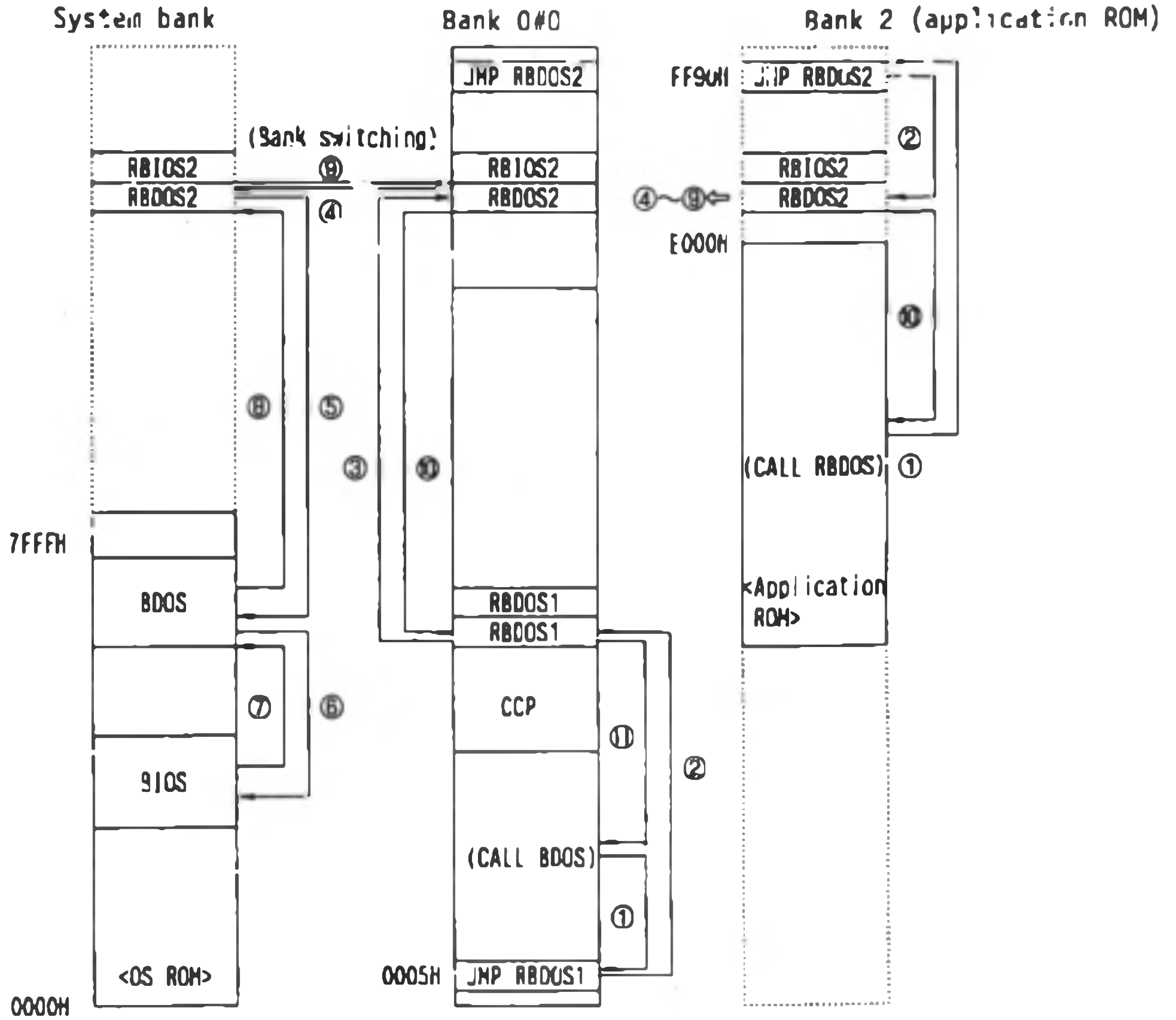


Fig. 5.1 BDOS processing flow

5.3 BDOS Error

There are following types of BDOS errors.

(1) BAD SECTOR ERROR

1 Cause

Data cannot be input to or output from the disk normally.

2 Response

Pressing the 'Abort' or 'Return' key suspends processing. After processing is suspended, the system is warm-booted and application program execution is terminated. Pressing the 'Ignore' key or a key other than 'Return' ignores the defective sector and continues processing.

(2) SELECT ERROR

1 Cause

An unexisting drive name was specified.

2 Response

Pressing any key including 'Press' sets drive A to the logged drive.

(3) R/O ERROR

1 Cause

An attempt was made to write data on a read-only disk.

2 Response

Pressing any key including 'Press' warm-boots the system.

The states including the following are referred to as "the drive is not ready":

1 The ROM socket has not been mounted.
2 The IC card has not been mounted.
3 The IC card protocol is defective.
4 The floppy disk drive power is off.
5 The floppy disk drive cable has not been connected.
6 The floppy disk drive diskette has not been set.

Table 5.1 shows the relationship between disk devices and BDOS errors.

Table 5.1 Disk device and BDOS error

Cause	RAM disk	ROM socket	IC card	FDD	BDOS processing
Check sum error	0	-	-	-	Bad Sector
Directory full	0	-	0	0	Return with A=FFH
Disk full	0	-	0	0	Return with A=FFH
Write processing	-	0	-	-	R/O
Write processing in write protect mode	0	-	0	0	R/O
The file is not in the directory	0	0	0	0	Return with A=FFH
The drive is not ready.	-	0	0	0	Select

0:Occurs
-:Does not occur

As shown above, BDOS displays four types of errors. Because these errors are handled by BDOS independently, a message or input request unrelated to the application program may be output or the system may be warm-booted by a key pressed by the user after the error is displayed.

To avoid them, the application program must notify and recover the errors by itself. Only the following two procedures are explained here.

(1) The procedure to receive a BDOS error as a return code
(2) The procedure to update the BDOS error processing jump vector and perform discrete error processing

(1) The procedure to receive a BDOS error as a return code

1 Modification procedure

Modify so that the application program can call OS ROM (system bank) address 000AH. This enables the application program to store the BDOS error information in a register . (SETERR)
Also, modify so that the application program can call OS ROM address 000DH. This enables BDOS to notify errors in an ordinary way.
BIOS CALLX (WBOOT + 66H) is used by the application program to call a routine that resides in OS ROM. (See APPENDIX 9 SAMPLE26)

2 Return codes

The following return codes are returned after SETERR execution.

Error	Register A	Register H	Explanation
BAD SECTOR BAD SELECT R/O Disk R/O File	FFH FFH FFH FFH	01H 02H 03H 04H	CP/M Standard BDOS Error
		00H	Not an error

When the H register contains 00H, a return code corresponding to the CP/M return information has been set in the A register.

When BAD SECTOR ERROR occurred, more detailed error information are set in system area BIOSERROR (F417H).

Address : Variable Name (Number of bytes)

F417H : BIOSERROR (1)
Return code for BIOS disk read/write operation
=00H: Normal termination
=01H: Read Error
=02H: Write Error
=03H: Write Protect Error
=04H: Time Over Error
=05H: Seek Error
=06H: Break Error
=07H: Power Off Error
=08H: Mount Error
=FEH: Other Error

3 Notes

(a) Once SETERR is executed, BDOS only returns the error status to a register and does not perform error processing until RSTERR or WBOOT is executed.

(b) If the application program does not determine the error and recovers it by itself after SETERR execution, normal subsequent processing is not guaranteed.

(2) Procedure to update the BDOS error processing jump vector

1 Modification procedure

The BDOS error processing jump vector is allocated at the head of BDOS in RAM. By updating the contents of this jump vector, the application program can perform its discrete error processing.

The jump vector configuration is shown below. (The RBDOS1 address can be obtained from the contents of addresses 0006H and 0007H.) See Appendix 9. Sample 26.

Address	Data	Contents
RBDOS1+03H	DW PERERR	Permanent error processing address (BAD SECTOR)
RBDOS1+05H	DW SELERR	Select error processing address (BAD SELECT)
RBDOS1+07H	DW RODERR	R/O disk error processing address (R/O DISK)
RBDOS1+09H	DW ROFERR	R/O file error processing address (R/O FILE)

2 Notes

(a) Because the BDOS stack that belongs to the system is used, the BDOS stack must be switched to the application stack when returning control directly to the application program.

(b) Because the bank is in all-RAM (bank 0#0) state, be careful when updating this jump vector with a ROM-based program.

(c) Take the bank into consideration when updating the jump vector contents with a ROM-based program because the jump vector may be positioned behind the application ROM where the ROM-based program resides.

(d) Keep the following two items when returning control to the system (BDOS) after performing error processing by the application program.

1 Do not call BDOS during error processing.
2 Return control to the system after switching the current bank to the system bank.

CHAPTER 6 DISK SYSTEM

6.1 Overview

In EHT-10 and EHT-10/2, the I/O devices related to the memory are assigned to disk drives so that they can be handled easier. The relationship between I/O devices and disk drives is as follows.

```
Drive A: RAM disk (internal and extended RAMs)
      B: ROM socket
      C: IC card
      D: External disk (floppy disk)
      E: External disk (floppy disk)
```

6.2 Logical and Physical Drives

(1) In EHT-10 and EHT-10/2, the correspondence between logical and physical drives can be modified arbitrarily. The default values are as given in Section 6.1.
The correspondence table for logical and physical drives is allocated in system area DSKTBL. By updating the contents of this table, the correspondence can be modified.
The contents of this table are held until the system is reset (BOOT).
The DISKTBL contents are shown below.
The correspondence is modified by changing the physical drive codes assigned to the logical drives in DISKTBL.

Example: To change drive B to a floppy disk drive, replace 01H, which is the contents of address DISKTBL + 1 (F0E5H), with 03H.

Address : Variable name (Number of bytes)

F0E4H : DISKTBL (5)
Logical/physical drive correspondence table

Address	Initial value	Corresponding logical drive name
F0E4H	00H	Logical drive A:
F0E5H	01H	Logical drive B:
F0E6H	02H	Logical drive C:
F0E7H	03H	Logical drive D:
F0E8H	04H	Logical drive E:

Physical drive codes

00H: RAM disk
01H: ROM socket
02H: IC card
03H: Floppy disk
04H: Floppy disk

(2) Notes on changing correspondence

Note the following when changing the correspondence between logical and physical drives.

1 If a physical drive code between 05H and FFH is specified, the corresponding drive cannot be accessed.

2 Two or more identical physical drive codes must not be specified in a five-byte area of DISKTBL.

3 If the DISKTBL contents are modified, the corresponding DISKROV (F0E9H) vector contents that indicate whether the drive is used for R/O or R/W must also be modified. However, DISKROV modification is only made effective after warm BOOT is executed.

4 If the DISLTBL contents are modified, the contents of the area which indicates the ROM socket position in the table must also be modified.

Address : Variable name (Number of bytes)

FOE9H : DISKROV (2)

Disk R/O vector

	7	6	5	4	3	2	1	0
+00H	0	0	0	E	D	C	B	A

State of each bit
0: R/W
1: R/O

	7	6	5	4	3	2	1	0
+01H	0	0	0	0	0	0	0	0

Above bits correspond to logical drives A: to E:.
The initial value for drive B: is 1 (R/O).

6.3 RAM Disk

6.3.1 Overview

The extended RAMs and part of the standard RAM are allocated as the RAM disk. Standard RAM and extended RAM are handled as a logically consecutive drive disk.
Check sum is performed for the RAM disk during read or write processing to determine whether data was not destroyed.

Data can be written into and read from the RAM disk at higher speeds than for a floppy disk, and which largely increases the performance of EHT-10 and EHT-10/2.
A maximum of 231 Kbytes for EHT-10 and 232 Kbytes for EHT-10/2 can be allocated as the RAM disk when all extended RAMs are mounted.

6.3.2 Drive name and capacity

(1) Drive name

The drive name is A:.

(2) Capacity

1 When no extended RAM is used: 0 to 40 Kbytes (*1)

2 When one or more extended RAMs are used: Total extended RAM size + size of the disk area allocated in the main RAM

3 The RAM disk capacity can be modified by CONFIG on the system menu screen. However, only the RAM disk area size allocated in the main RAM can be modified by the user because all extended RAMs are automatically assumed as part of the RAM disk area by the system.

4 Fig 6.3 lists the specifications of the RAM disk for each extended RAM size.

*1 Although a maximum of 40 Kbytes can be allocated in EHT-10/2, a maximum of 39 Kbytes can be allocated in EHT-10. This is because the system area size in EHT-10 is different from that in EHT-10/2.

Table 6.1 RAM disk specifications for each extended RAM size

Extended RAM	0 KB	64 KB	128 KB	192 KB
Disk size (*1)	0 to 40KB	64 to 104KB	128to 168KB	192to 232KB
Sector/track	128 bytes			
Track/sector	64 sectors			
Track	0 to 4	0 to 12	0 to 20	0 to 28
Sector	0 to 63	0 to 63	0 to 63	0 to 63
Number of directories	16(512B)	32(1 KB)	64 (2 KB)	
Check sum area	512B	1 KB	2 KB	

*1 The maximum disk sizes listed in this table are those in EHT-10/2, and corresponding maximum disk sizes in EHT-10 are one Kbyte less than these values (that is, 39 Kbytes, 103 Kbytes, 167 Kbytes, and 231 Kbytes).

6.3.3 Format

Figure 6.1 shows the RAM disk format.

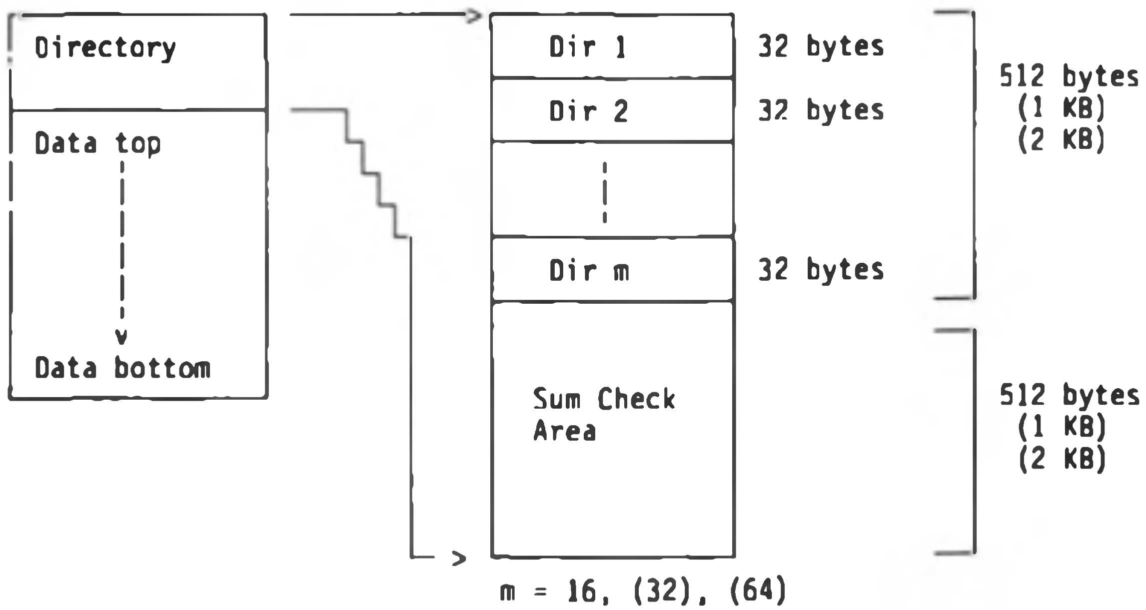


Fig. 6.1 RAM disk format

(1) Directory area

1 Each directory consists of 32 bytes.

2 The number of directories depends on the extended RAM capacity. They are 16, 32, and 64.

(2) Sum check area

1 The such check area size also depends on the extended RAM capacity. They area 512 bytes, 1 Kbytes, and 2 Kbytes. Check sum is performed for each 128-byte data, and the result is stored in a one-byte area. The check sum results are updated when data is written, and check sum is performed when data is read or the power is turned on.

2 Figure 6.1 shows the logical format of the RAM disk, and actual directory positions differ according to the extended RAM size. Figure 6.2 shows the positional relationship between extended RAM sizes and directory positions.

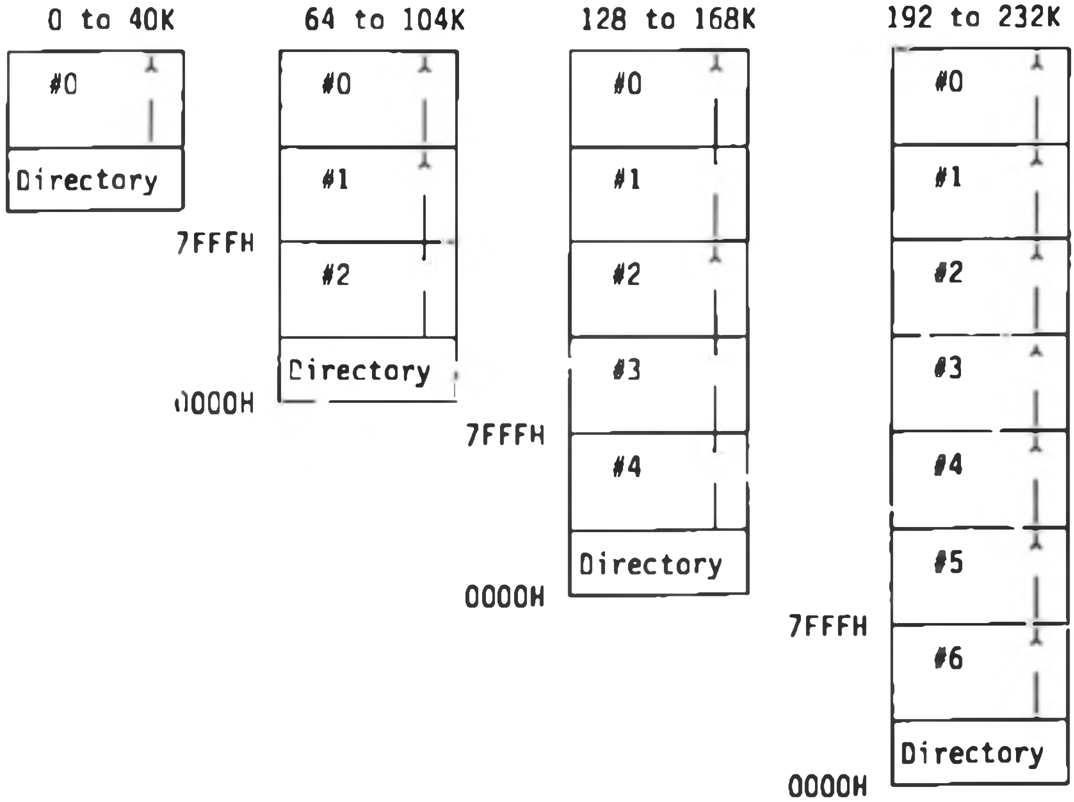


Fig. 6.2 Positional relationship between extended RAM sizes and directory positions

Notes:

1. Values #0 to #6 indicate subbank numbers.

2. The arrow indicates from data top toward data bottom.

3. A value between 0 and 40 (or 39) Kbytes can be specified as the capacity of the RAM disk to be used as the subbank #0 disk.

6.3.4 Miscellaneous

(1) Because the RAM disk area is usually allocated on the back of the application ROM (subbanks #0 to #6 of the bank), the bank must be switched in order to access the RAM disk data. However, the user need not be aware of it because switching is performed by the operating system.

(2) The operating system checks subbanks, starting from subbank #0, for whether the extended RAM has been mounted. When a subbank for which the extended RAM has not been mounted is detected, the operating system does not check subsequent subbanks.

(3) Setting RAM disk capacity with CONFIG.

Note the following when setting the standard RAM disk size.

1 If the new size is less than the current size, format all the RAM disk.

2 If the new size is greater than the current size, format only the enlarged area part. The contents of the old area part are remain unchanged.

3 If the new size is equal to the current size, the contents of the current area remain unchanged.

4 Although the RAM disk contents remain unchanged when the user BIOS size is enlarged, the RAM disk contents are destroyed when the user BIOS size is made smaller.

5 When ROM has been mounted in the ROM socket, the user BIOS size cannot be changed.

6.4 ROM Socket

6.4.1 Overview

The EHT-10/EHT-10/2 mainframe is equipped with one ROM socket in which ROM can be mounted. ROM is mapped in the EHT-10/EHT-10/2 memory space, and can be accessed by switching the bank.
A CMOS masked ROM or CMOS EPROM with a capacity of 256 Kbits (e.g., 27C256), 512 Kbits, or 1 Mbits (e.g., HN62301) can be used as the ROM. (*1)
The programs and data stored in the ROM socket can be loaded in the RAM and executed and processed by the EHT-10/EHT-10/2 application programs.

The system supports the following ROM program formats as standard formats:

M format: Format used for load-and-execute programs in HX-40, PX-4 and PX-8.
P format: Format which enables PX-4, HX-40 ROM contents execution.

Note:

When using a 256-Kbit EPROM or masked ROM, the ROM contents switching jumper must be opened. When using a 512-Kbit or 1-Mbit EPROM or masked ROM, the switching jumper must be closed. Refer to EHT-10 Operation Manual.

6.4.2 Drive name and capacity

(1) Drive name

The drive name is B:.

(2) Capacity

Table 6.2 shows the ROM socket disk capacity.

Table 6.2 ROM capacity

Capacity	256 Kbits 32 KB	512 Kbits 64 KB	1 Mbits 128 KB
Track	0 to 3	0 to 7	0 to 15
Sector	0 to 63	0 to 63	0 to 63
Maximum number of directories	31	31	31

*1 The values used in this table are those when the number of directories is 28 to 31. If the number of directories is less than 28, the number of logical tracks increases.

Example: When the ROM capacity is 256 Kbits and the number of directories is 1 to 3, logical tracks 0 to 4 and sectors 0 to 6 can be used. This is because the directory area is allocated in units of 128 bytes (one sector), and the area less than one block (1 Kbytes) is used as a data area. Consequently, the data area size increases by a maximum of seven sectors.

6.4.2 Format

(1) Overview

There are two types of application programs executed under control of the EHT-10/EHT-10/2 CP/M.

1 Load-and-execute application programs

These application programs are loaded in TPA and executed like an ordinary CP/M.

(a) Files can be loaded from the disk.
(b) Files can be loaded through a communication line.

2 ROM-based application programs

In addition to the load-and-execute programs, the programs stored in the ROM socket ROM can be directly executed in ROM. For this purpose, the programs contained in the ROM mounted in the ROM socket have the following two formats.

(a) M format
This ROM program format is the same as that of the PX-8 ROM socket and PX-4/HX-40 ROM cartridge, and used when the programs are loaded in TPA and executed.

(b) P format
This ROM program format is the same as that of the HX-40,PX-4 ROM socket (ROM-based), and used when the programs are directly executed in the ROM.

Since format type of the ROM mounted in the ROM socket is automatically determined and format addresses are also automatically calculated by the operating system, the user need not be aware of the difference in the ROM program format.

(2) M format

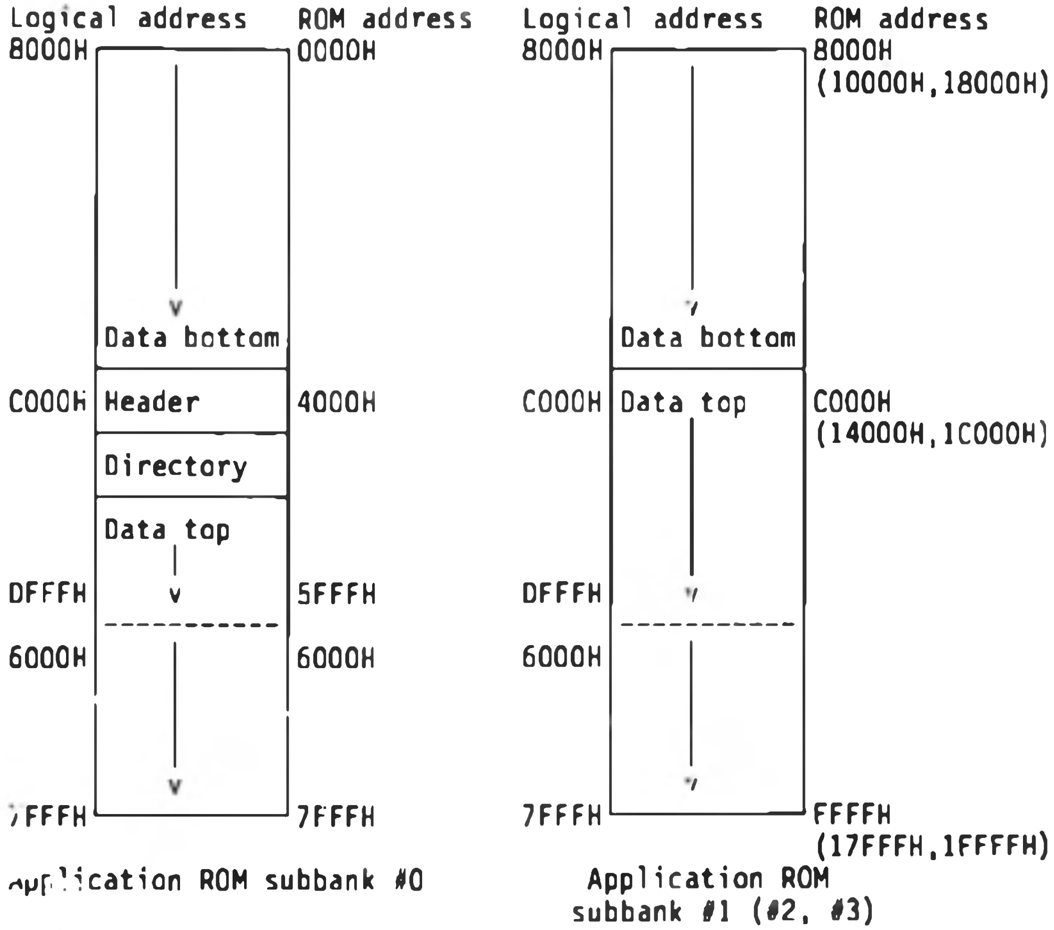


Fig. 6.7 Relationship between ROM addresses and data addresses (M format)

Note:

For a 256-Kbit ROM, only subbank #0 is assumed to be a consecutive ROM area. For a 512-Kbit ROM, subbanks #0 and #1 are assumed to be a consecutive ROM area. For a 1-Mbit ROM, subbanks #0, #1, #2, and #3 are assumed to be a consecutive ROM area.

1 Header area (32 bytes)

The header area consists of 32 bytes, and used to store data such as the format, size, and the number of directories of the ROM. Figure 6.4 shows the header structure.

00H	E5H *	
01H	37H *	--> Value 37H indicates M format. (This value becomes 50H for P format.)
02H	ROM capacity *	--> 256Kbits ROM : 20H 512Kbits ROM : 40H 1 Mbits ROM : 80H
03H	Check sum	
04H		
05H	System name	
06H		
07H		
08H	ROM name (14 bytes)	
09H		
15H		
16H	Number of directories + 1 *	The result of (the number of directories + 1) is stored. However, the result must be rounded up to a multiple of 4. (The result becomes 4 when the number of directories is 2.)
17H	'V'	
18H	Version No.	
19H		
1AH	Date (6 bytes)	
1BH		
1FH		

Note: All values for the P format are the same as those for the M format except the following: Value 37H in byte 2 becomes 50H for the P format.

The items with an asterisk (*) must be set.

Fig 6-8 ROM header structure

2 Directory area

One directory consists of 32 bytes. Up to 31 directories can be registered. Refer to the corresponding CP/M-related manual for directory configuration. The header directory can be automatically created by using EHT-10/EHT-10/2 development tool WPROMFRM.

3 Data area

The top address of the data area changes according to the number of directories.
The data area top address is calculated as follows:

Header top address + 20H x m

(Value m is (n + 1) rounded to a multiple of 4 where n is the number of directories.)

The data top address is fixed to track 0 and sector 8 regardless of the number of directories. If an attempt is made to read the data on an unexisting sector (m/4 to m/7), return code E5H is always returned.

(3) P format

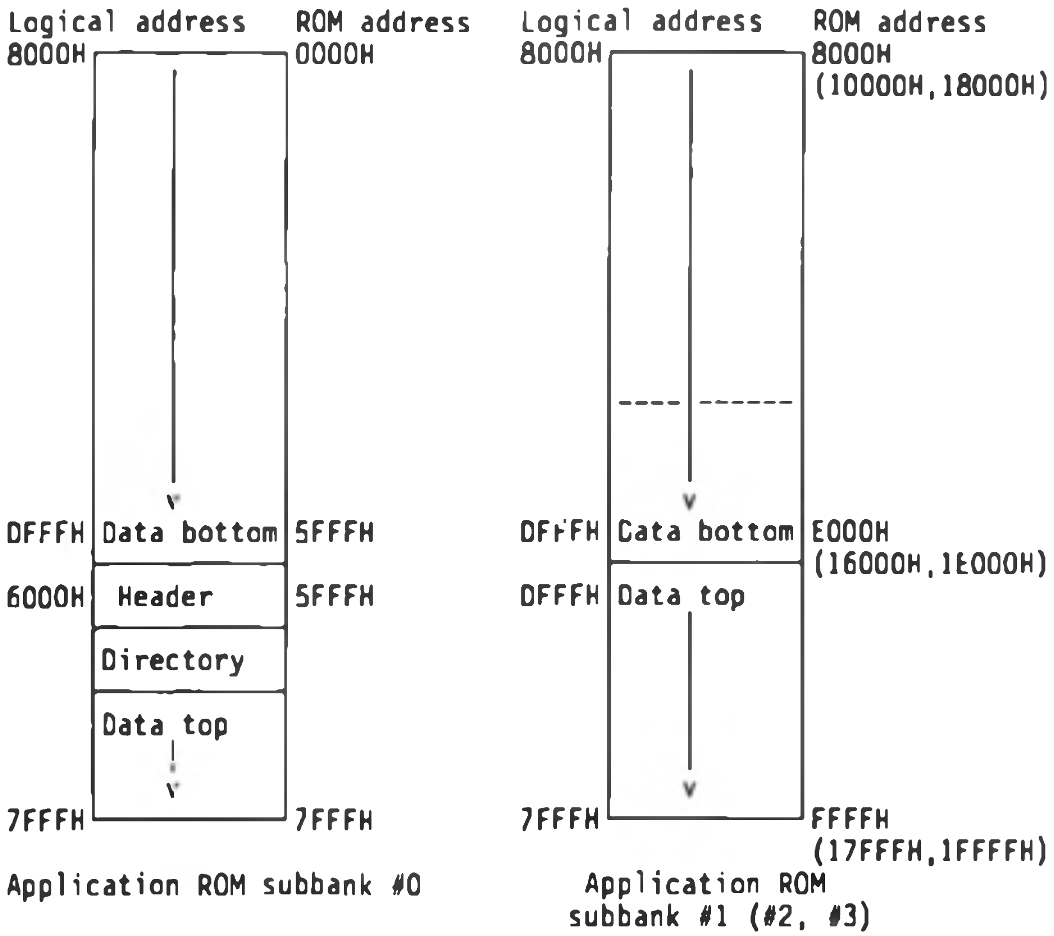


Fig 6.5 Relationship between ROM addresses and data addresses (P format)

Note:

For a 256-Kbit ROM, only subbank #0 is assumed to be a consecutive ROM area. For a 512-Kbit ROM, subbanks #0 and #1 are assumed to be a consecutive ROM area. For a 1-Mbit ROM, subbanks #0, #1, #2, and #3 are assumed to be a consecutive ROM area.

1 Header area (32 bytes)

If value 50H instead of 37H is set in byte 2 of the M-format header, the header becomes a P-format header. Other values are the same as those of the P-format header. (See Figure 6.4.)

2 Directory area

Same as the M-format directory area

3 Data area

The top address of the data area is the same as that of the M-format data area. The following five-byte data must be added to the head of a ROM- based application program (see Item 3.4):

DDH,DEH,00H,00H,00H
("DDH,DEH" is an ID which indicates that the program is a ROM-based program.)

The application program is initiated starting from the byte immediately after the above five bytes.

(3) Miscellaneous

1 The above five-byte data must not be added to the head of a program other than a ROM-based program.

2 There are two types of P-format programs: load-and-execute programs without an ID area and ROM-based programs with an ID area. These two types of programs can simultaneously reside in the same ROM.

6.4.4 Executing ROM socket programs

(1) Load-and-execute programs

The M- or P-format program having no ID at the head is loaded in TPA and executed starting from address 100H.

(2) ROM-based programs

If a P-format program having an ID area at the head is specified as an execution file, the BIOS ROM handler reads one sector and checks the ID area to determine whether the format of the specified file is P format (ROM-based program).
If the program is determined as a ROM-based program, control is passed to the ROM address next to the (file top address + 5) address.

6.4.5 Miscellaneous

(1) The relationship between logical addresses and ROM addresses in an M-format program is different from that in a P-format program. This should be taken note of when creating a ROM-based program.

(2) When executing a ROM-based program by using a 512-Kbit or 1-Mbit ROM, sufficient address management must be performed and bank switching must be reflected in the program.

(3) When creating a ROM-based program, determine the program execution start address by taking the directory area capacity beforehand because the data top address changes according to the number of directories.

6.5 IC Card

6.5.1 Overview

EHT-10/EHT-10/2 supports IC card I/F as a standard feature that conforms to ISO standards DP7816/1 and DP7816/2.(*1) (However, part of the power specifications is different.)
Commands and data are exchanged through serial communications between EHT-10/EHT-10/2 and the IC card. In this case, data is transferred according to an specified protocol.
The IC cards that conform to the Toppan printing protocol used in EHT-10/EHT-10/2 OS version 2.0 are supported as disk drives. For other IC cards, hooks are prepared. The user can use these IC cards as disks by creating hook processing programs.
This chapter explains IC cards supported only as disks.

(*1) ISO (International Standardization Organization) standards stipulate items on the IC card, including the size, physical characteristics, and contact part.

6.5.2. Drive name and capacity

(1) Drive name

The drive name is C:.

(2) Capacity

The capacity of the disk is as follow.

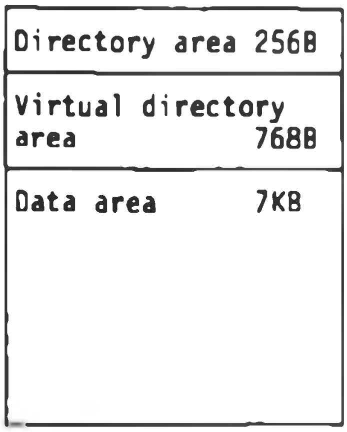
Capacity	8KB(*1)
Track	0
Sector	0-63
Number of directories	8(*2)

(*1) The capacity and the number of directories can be changed by using IDSK function. Please see 6.5.4
(*2) If the number of directories is 8, directory area is actually from 0 track 0 sector to 0 track 1 sector. However, OS specifies from 0 track 0 sector to 0 track 7 sector as directory area. Therefore, the data area always starts from 0 track 8 sector. In such a case user cannot access from 0 track 2 sector to 0 track 7 sector.

6.5.3 Format

(1) Format

Fig. 6.6 shows the IC card format.



---- 32B x 8 = 256B (0 track 0 sector to 1 sector.)

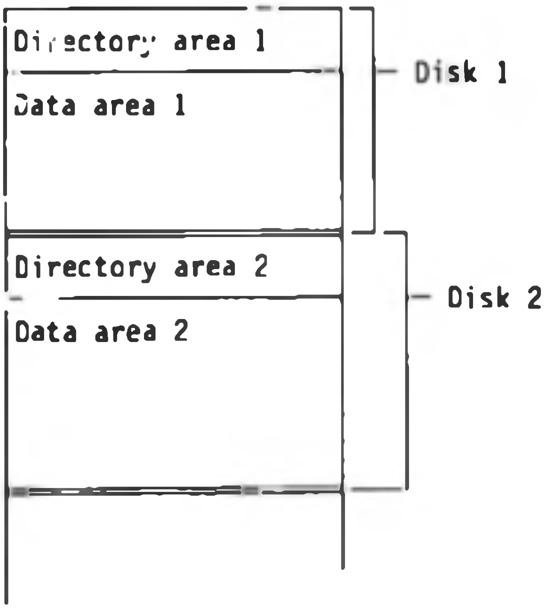
---- User cannot access. (0 track 2 sector to 7 sector.)

---- 128B x 56 record = 7KB (0 track 8 sector - 63 sector)

* Number of directories and records can be changed by using IDSK function. (Numbers should be even.)

Fig. 6.5 IC card format (1)

By using ISET, IDSK function, user can make plural disks on one IC card. Fig. 6.7 shows the such a example.



*Disk n forms one CP/M disk. In this case, File name (*1), System PSW, and User PIN should be set for each disk. And user cannot access 2 disks at the same time.

(*1) For the IC card, one disk area is regarded as a file. Therefore, this "File name" means the name of a disk area.

(*2) Fig. 6.6 and 6.7 are both logical architecture. Actually the area for file control or check sum is included.

Fig. 6.7 IC card format (2)

(2) Directory area

Fig. 6.8 shows the architecture of directory area.

Directory 1 (32B)
Directory 2 (32B)
Directory 3 (32B)
Directory 4 (32B)
Directory 8 (32B)

*Each directory has 32 bytes. Physically, the data for check sum is added each 1 sector(64B:*1) automatically.

(*1) This 1 sector means one unit for data control for IC card. (Different from 1 sector of CP/M)

(*2) The number of directories can be changed by IDSK function.

Fig. 6.8 Directory area of IC card

(3) Data area

Fig. 6.9 shows data area of the IC card.

Record 1 (128B)
Record 2 (128B)
Record 3 (128B)
Record 56 (128B)

*Each record has 128 bytes. Physically the data for check sum is added each 1 sector(64B:*1) automatically.

(*1) This 1 sector means one unit for data control of IC card. (Different from 1 sector of CP/M).

(*2) The number of records can be changed by IDSK function.

Fig. 6.9 Data area of IC card

6.5.4 Protocol

(1) Overview

The specification of communications between EHT-10/EHT-10/2 and an IC card are as follow.

Transfer rate:	9600 bps
Data length:	8 bits
Stop bit:	1 bit
Parity bit:	Even parity
Protocol:	Toppan printing
Control line:	Unused

(2) IC Card interface

There are following 2 methods for accessing the IC card for the EHT-10/EHT-10/2.

A. Command Through mode
B. Disk mode

A. Command through mode
IC card has a CPU and EEPROM so that we can regard the IC card as a kind of computers. Command through mode is that the EHT-10/EHT-10/2 send the data to (and receive the data form) the IC card directly .It is just like as to communicate with another computer via RS-232C port. In case of using command through mode, application program should send the command block to the IC card according to the IC card protocol. Also, application should control the answer from the IC card (response block). If you want to make an application by using the command through mode, therefore, you should know the IC card protocol which you want to use.
For the EHT-10/EHT-10/2, the following 2 ways are supported for using command through mode.
(a)BIOS ICCARD
BIOS ICCARD can send/receive 1 byte in each time.
(b) ICMD function
If you use the Toppan printing IC card, you can send/receive 1 command block in each time by using ICMD function.

B. Disk mode
Disk mode is the mode that to use the IC card just like as FDD or RAM disk. Drive "C:" is assigned for the IC card in the EHT-10/EHT-10/2. Therefore, OS should be able to control the IC card as other disk devices. For example, to control the data as each record or be able to format the disk. OS should be extended according to the IC card protocol.
For the OS version 2.0, Toppan printing IC card (64Kbit type) can be used as drive C:. If you want to use other manufacturer's IC card, OS can be extended by using HOOKs.
Toppan printing IC card has such functions as password, ID and so on. OS Vers. 2.0 supports 8 function interface commands for using these functions.

(3) Function Interface commands

EHT-10/EHT-10/2 IC card function interface commands have the following characteristics.

(a) For the purpose of optimum efficiency in memory use, the length of one block (the minimum unit used in file management) of a CP/M is 256 bytes; 512- or 1024-byte block units can also be selected as required.

(b) "n" number of files (with CP/M, independent disks can exist on the IC Card) can be created on a single IC Card. Each file can convert System PSW or User PIN and it is possible to prevent unintentional access from the external world.

(c) The size of one file (with CP/M, the size of a message of one disk) of the IC Card can be specified. In addition, the quantity of directories in each disk can be specified according to the amount of data therein.

(d) The following 8 functions are provided: ISTS, ISET, IDSK, IFMT, EJCT, IOPN, ICLS, and ICMD. These support the exclusive functions of the IC Card.

1. ISTS: Checks the loading status of the Expansion Program, the installation status of the IC Card R/W Unit, and the insertion status of the IC Card.
2. ISET: Sets the file name and the contents of System PSW and User PIN.
3. IDSK: Sets the disk size and the quantity of directories.
4. EJCT: Eject the IC card from IC card reader/writer unit.
5. IFMT: Formats the IC Card based on the values specified by ISET and IDSK, enabling use of the IC Card as a disk.
6. IOPN: Starts the COMMAND THROUGH Mode and sets the IC Card to OPEN status.
7. ICLS: Terminates the COMMAND THROUGH Mode and sets the IC Card to CLOSED status.
8. ICMD: Transmits text to the IC Card, then stores and returns the response to that transmission.

The remainder of this section describes each of the above functions separately.

1. ISTS (N%) (FFB7H)

Function: Checks the status of the IC Card.

Format:
```
ISTS = &HFFB7
N%   = -1
CALL   ISTS (N%)
```

Processing: Checks and reports OS version is 2.0 and the insertion status of the IC Card.

N% = -1: OS version is 1.0 (IC card cannot be used.)
= 0: IC Card power is ON. Or during the auto power OFF.
= 1: IC Card power is OFF.
= 2: Excess current goes through IC card

Caution: This function can be used at any point during execution. It merely checks the status of the IC Card without accessing it.

Remarks: This function should be executed at the beginning of the BASIC program. After this function is executed, other 7 function can be used.

2. ISET (FILE$, PSW$, PIN$) (EAA4H)

Function: Sets the file name and password onto RAM.

```
Format:   ISET   = &HEAA4
          FILE$  = "File name"(8 characters) + CHR$(N)
          PSW$   = "System PSW"(16 characters) + CHR$(R)
          PIN$   = "User PIN"(16 characters) + CHR$(R)
          CALL     ISET (FILE$, PSW$, PIN$)
```

Processing: Memorizes the file name, system password, user password(PIN), and the number of retries onto RAM. After execution of this function, the file (*1)specified therein can be accessed.

Caution: In case a disk will be used, this function must be used to set the values for that disk in advance. The default value of FILE$, PSW$ and PIN$ are all 00H.

Remarks: Within the escape code CHR$, "N" represents the file number, and "R" represents the quantity of retry errors permitted. The "N" value must be sequentially incremented from "00". The "R" value (00H to 0FH) is referred during the execution of IFMT.

Errors: An I/O error will occur unless the length of FILE$ is nine bytes, and that of PSW$ and PIN$ is 17 bytes each.

(*1) For the IC card, one disk area is regarded as a file. Therefore, this "File name" means the name of a disk area.

Function: Specifies the size of the IC Card as a disk.

Format:
```
IDSK = &HEAA8
N%   = (Size of the data area)
M%   = (Quantity of directories)
L%   = (Length of one data block)
CALL   IDSK (N%, M%, L%)
```

Processing: Sets the parameters required for using the IC Card as a disk. Default value is N%=56, M%=8, L%=256.

Caution: In case a disk will be used, this function must be used to set the values for that disk in advance. After executing this function, the RESET command must always be executed.

Remarks: The "N%" value specifies the size of the data area in 128-byte units up to a maximum of 56 records using multiples of two. The "M%" value specifies the quantity of directories up to a maximum of 16 directories using multiplies of two. The "L%" value specifies the length of the data block to either 256, 512 or 1024 bytes.
The default values are as follows: N% = 56, M% = 8, and L% = 256.
If you want to make plural disks on an IC card, you should consider the size of the IC card and specify the each parameter.If you make only one disk on an IC card, you can specify the each parameter according to the following expression.

N% x 2 + M% / 2 ≤ 76H (118)

Errors: No error checking is performed.

Function: Formats the IC Card as a disk.

Format:
```
IFMT = &HEAAC
CALL   IFMT
```

Processing: This functions creates a file on the IC Card based on the file name(*1) and the PSW/PIN contents that were specified by the ISET function, allocates the area corresponding to the size specified by the IDSK function, then formats the file to enable its use as a disk.

Caution: This function can only be used in DISK Mode. It takes about 1 or 2 minutes to finish the formatting. If you push the reset switch or remove the IC card during the formatting, IC card may be broken. Don't push the reset switch or remove the IC card.

Error: Any error that occurs will be regarded as an I/O error and its occurrence will terminate any processing that is currently being executed.

(*1) For the IC card, one disk area is regarded as a file.
Therefore, this "File name" means the name of a disk area.

Function: Eject the IC card from IC card R/W unit.

Format: EJCT = &HEAB0
CALL = EJCT

Processing: After closing the IC card, eject the IC card from IC card R/W unit. This function is for the Ic card R/W unit. Except the ejection, the process is the same as ICLS function.

Function: Supplies power to the IC Card and sets it to COMMAND THROUGH Mode.

Format:
```
IOPN = &HEAB4
CALL  IOPN
```

Processing: This function switches OFF the power supply to the IC Card, then switches it back ON again.

Caution: After this function is executed, the IC Card will be in COMMAND THROUGH Mode till ICLS or EJCT function is executed.

Remarks: When the power to the IC Card is switched ON, the Reset response from the IC Card is automatically sent to RAM.

Error: In case an error occurs because the IC Card has not been inserted, such error will be regarded as an I/O error.

Function: Discontinues the power supply to the IC Card and sets it to DISK Mode.

Format:
```
ICLS = &HEAB8
CALL   ICLS
```

Processing: This function switches OFF the power supply to the IC Card.

Caution: After this function is executed, the IC Card will be in DISK Mode.

Function: Transmits text to the IC Card and receives the response to that transmission.

Format:
```
ICMD  = &HEABC
TEXT$ = "Command data"
ANS$  = SPACE$(N)
CALL    ICMD (TEXT$, ANS$)
```

Processing: This function transmits the command specified by TEXT$ to the IC Card, then stores the response from the IC Card in ANS$.

Caution: This function can only be executed in COMMAND THROUGH Mode.

Remarks: Since the Start code, Length, and Check will be automatically appended, only the Command, Reference, and Data are required for a CALL operation. (The same format is also applicable for the response data.) The "N" value specifies the size of the response data.

Error: In case an error occurs due to, for example, a NC response from the IC Card, it will regarded as an I/O error.

6.5.5 Disk Mode Interface

(1) Beginning Use

When the IC Card is accessed in DISK Mode, power is automatically supplied to the IC Card, collation of ID, PSW, PIN, etc. is performed according to your command specification, and the IC Card assumes a status wherein Read/Write operations can be performed.
Before accessing the IC card as a disk, therefore, the disk status must be specified in advance at the application program, using the ISET and IDSK functions. And Before 1st disk access, you should check the IC card status by using ISTS function.

(2) READ/WRITE Operations

Read/Write operations can be executed just as the other disks.
Since the block length of a data area is determined by the "L%" parameter of the IDSK function, the data boundaries can be in units of 256, 512 or 1024 bytes, depending on the "L" value. (For example, when L%=256, 500 bytes of data will occupy a data area of two blocks (512 bytes)).

(3) Terminating Use

Use of the IC Card in DISK Mode is terminated by the CLOSE command of BASIC. At this time, however, the power supply to the IC Card will remain ON.
To terminate the power supply, execute the ICLS or EJCT function.

(4) Miscellaneous

a. Power Switch OFF Status or Low Battery Status
In case the POWER SW is set OFF or the battery runs low during access of the IC Card, processing is performed to the end of the current command then the power supply is stopped. When the power is next switched back ON, it is possible to continue the previous processing. (There is no need for special consideration of the Power OFF status within application programs.)

b. Auto Power OFF Function
To reduce power dissipation when using the IC Card in DISK Mode, the power supply of the IC Card can automatically be switched OFF. In case the IC Card has not been accessed during a fixed time period, the Auto Power OFF function automatically stops the power supply to the IC Card. This function can be inhibited by the application program.

c. Use of Multiple Disks
A single IC Card can be used as multiple disks, but such use requires that the following points be strictly observed:
(i) Before changing to another disk (that is, accessing a file of the IC Card which has a different file name, PSW, and PIN), the ICLS function must first be used to close the currently open file. (If you wish to perform disk access using BASIC, be sure to first execute the CLOSE command.)
(ii) The RESET command will initialize the disk status of BDOS.
(iii) Use of the ISET and IDSK functions sets the status of the IC Card as a new disk.

(iv) Performing Steps (i) to (iii) completes the preparation for accessing a new disk, so the OPEN command can be used to start disk access.

6.5.6 Command Through Mode Interface

(1) Beginning Use

The IOPN functions sets the COMMAND THROUGH Mode. At this time, power is supplied to the IC Card and a Reset response is received.

(2) Access

Access is performed using the ICMD function. In case a communication error, data error, No response error or other such errors occur, they will be regarded as I/O errors. Even if the status of the response data from the IC Card consists of an error code, however, it will not be regarded as an error and the corresponding response data will be returned.

(3) Terminating Use

The ICLS functions terminates the COMMAND THROUGH Mode and activates the DISK Mode. At this time, the power supply to the IC Card is also stopped.

(4) Miscellaneous

a. Power Switch OFF Status or Low Battery Status
In case the POWER SW is set OFF or the battery runs low during access of the IC Card, processing is performed to the end of the current command then the power supply is stopped. Special attention should be paid to the fact that, in this case, when the power is next switched back ON, the previous processing cannot be continued.

b. Disk Access
When a disk access operation is executed in COMMAND THROUGH Mode, a Bad Select or Bad Sector error will occur.

6.5.7 Machine Language Interface

If you want to use the IC card interface functions (such as ISET, IDSK) in the machine language application, do as the following.

(1) Check whether the OS version is 2.0.

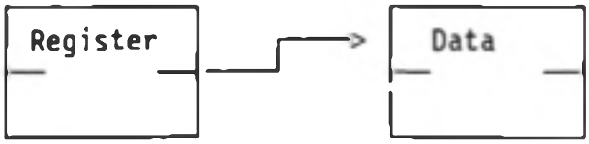
(2) Call the ICBASCMD(bank:1#1,Address:6031H) using the BIOS CALLX. At that time, specified parameters should be set.

(3) Return parameter is the following.
CY=0: The function is terminated normally.
=1: The function is terminated abnormally.

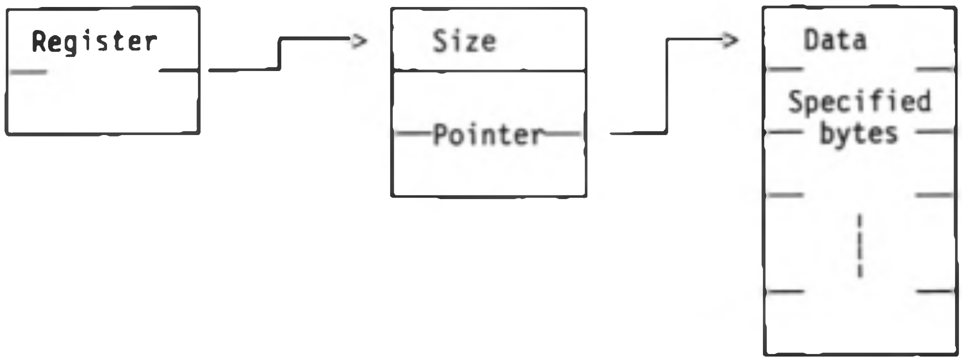
The followings are the parameters of each function. For detailed information about the each function, please see "6.5.4. Protocol"

(Note) The architecture of the parameter depends on the parameter type.

<Integer type>



<Character type>



a. ISTS (N%)
    A reg. = 00H
    HL reg.= Integer type (The status of the IC card)

b. ISET (FILE$,PSW$,PIN$)
    A reg. = 01H
    HL reg.= Character type (File name)
    DE reg.= Character type (System PSW)
    BC reg.= Character type (User PIN)

c. IDSK (N%,M%,L%)
    A reg. = 02H
    HL reg.= Integer type (Size of the data area)
    DE reg.= Integer type (Number of the directories)
    BC reg.= Integer type (Size of the data block)

d. IFMT
    A reg. = 03H

e. EJCT
    A reg. = 04H

f. IOPN
    A reg. = 05H

g. ICLS
    A reg. = 06H

h. ICMD (TEXT$,ANS$)
    A reg. = 07H
    HL reg.= Character type (Send text data)
    DE reg.= Character type (Area for the received data)

<Example>

```
        ISTS      EQU  00H
        DISBNK    EQU  F41BH
        CALLX          EQU  EB69H
        ICBASCMD  EQU  6031H

                  LD   A,11H
                  LD   (DISBNK),A          ; SET THE BANK

                  LD   HL,-1
                  LD   (0200H),HL          ; PARAMETER FOR ISTS

                  LD   HL,0200H
                  LD   A,ISTS         ; PARAMETER FOR ISTS
                  LD   IX,ICBASCMD
                  CALL CALLX               ; BIOS CALLX
```

6.5.8 Miscellaneous

(1) Relationship with BIOS

If BIOS ICCARD and an IC card as a disk are used simultaneously, the protocol may not be satisfied because BIOS ICCARD also access the IC card.
To solve this problem, the operating system prohibits simultaneous use of an IC card and BIOS ICCARD by returning an error code when an attempt is made to use either of them while the other is being used. (For example, an error is assumed if an attempt is made to open BIOS ICCARD while an IC card is open as a disk.)

(2) Relationship between an IC card and serial communication

An IC card is connected to EHT-10/EHT-10/2 through an IC card interface. Although three types of I/Fs (RS-232C, IC card and Cartridge SIO) are available for serial communications with EHT-10/EHT-10/2, none of these I/Fs can be used simultaneously with an IC card because only one port is used internally by way of switching.
Supporting automatic serial port switching by one operating system enables an IC card to be used while, for example, a floppy disk (RS-232C) is being used.

(3) Registration of the card ID

When you start to access the IC card, ID check is required. Please do the following to register the card ID.

a. If the card ID is not set, register the ID by command through mode.

b. Set the ID (which is registered to the IC card) onto RAM by using the ICDIDPNT (F643H:*1).

c. After a. and b., you can use the IC card. First, you should format the IC card by using ISET, IDSK and IFMT functions.

(*1) The architecture of the ICDIDPNT is the following.

F643H: ICDIDPNT (4)
Represents the storage area for ID data during registration of the card ID. (Valid only in DISK Mode)

```
+0  | Pointer |  ---->  | Counter      |  --- Specifies the amount
+1  |         |         | Card Reg.(1) |      of data in the
+2  | Bank    |         | Card Reg.(2) |      registered ID.
                        | Card Reg.(3) |      Default value = 00H
                        |      :       |
                        | Card Reg(12) |  --- Specifies 12 bytes of
                        | Length(1)    |      the registration no.
                        | ID 1         |      of the card.
                        | Length(2)    |      Default value = All 0s
                        | ID 2         |
                        |              |  --- "ID" can be specified
                                              by a maximum of 64
                                              characters.  "Length"
                                              specifies the ID
                                              length.
                                              The number of ID data
                                              which is specified by
                                              counter is required.
                                              If the Counter=0, no ID
                                              data is required
```

"Bank" specifies the bank indicated by the pointer. Default value is EFEAH and points RAM bank. With a RAM bank, the default data area is 22 bytes.

(4) Erasure of files

Basically, it is not possible to erase a file (one disk on CP/M) of the IC Card. (That is, once a file has been created by the IFMT command, that file cannot be erased.) Consequently, erasure of a file must be performed by directly erasing the file in COMMAND THROUGH Mode.

(5) BASIC DSKF command

Execution of the DSKF command will report the size of the remaining memory area of the specified disk. In case of the IC Card (Disk C), however, the size will be reported in the units specified by the "L%" parameter of the IDSK function.

(6) Reset during the accessing

Do not the reset the EHT-10/EHT-10/2 during accessing to the IC card (Especially during the writing). IC card may be broken.

(7) Default value related to the IC card

If you want to use the IC card without executing the ISET or IDSK function, you should set the status of IC card registration to the default value of the IC card. The default values are the following.

```
ID registration --- Number of the registered ID data = 0
Password ---------- System password = All 0s
                    User password (PIN) = All 0s
Disk -------------- Number of directories = 8
                    Block size = 256 bytes
                    Data area = 56 records
```

If you register ID only, passwords are automatically registered by executing IFMT function or format of the CONFIG utility.
All the values are initialized by reset.

(8) Formatting IC card

The IC card can be formatted by the DISK utility on the system menu screen.

(9) IC card power-off time

To save the power consumed by the IC card, the operating system stops supplying the power (power off) to the IC card when the IC card is not accessed for a fixed time (one minute as the default value).
Therefore, the next access is started from reset response.
When BIOS ICCARD is used, the power-off time for the IC card becomes infinite, and power is supplied until the IC card is closed.

6.6 Floppy Disks

6.6.1 Overview

An external 5.25-inch or 3.5-inch floppy disk drive can be connected to EHT-10/EHT-10/2 through an RS-232C interface. Read and write operations are performed for the connected external floppy disk drive in units of KB. The connectible floppy disks are as follows:

TF-15 (single and dual)
PF-10

6.6.2 Drive name and capacity

(1) Drive name

The drive names are D: and E:.

The relationship between drive names and floppy disks is as follows:

TF-15 (single drive)	D:
TF-15 (dual drives)	D: and E:
PF-10	D:

(2) Capacity

Capacity per drive: 320 Kbytes
Number of tracks per drive: 80
Number of sectors per track: 16
Storage capacity per sector: 256 bytes
Number of directories: 64
User capacity: 278 Kbytes

6.6.3 Format

Figure 6.10 shows the disk format on the media.
Although tracks 0 to 3 are used by the system, the user can perform read and write operations for these tracks by using BIOS.
Sectors 1 to 16 of track 4 are used as the directory area.

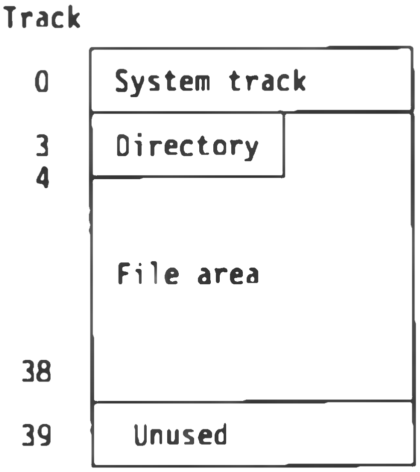


Fig 6.10 Disk format

6.6.4 Protocol

(1) Overview

The specifications of communications made between EHT-10/EHT-10/2 and a disk drive are as follows:

Transfer rate: 38400 bps
Data length: 8 bits
Stop bit: 1 bit
Parity bit: none
Protocol: Used
Control line: Unused

(2) EPSP (EPSON Serial Communication Protocol)

Communications is made between a disk drive and EHT-10/EHT-10/2 according to the EPSP (EPSON Serial Communication Protocol).
The general EPSP format is shown below.

FMT
DID
SID
FNC
SIZ
Text data

FMT: Head block format
 00H: Indicates data sent from EHT-10/EHT-10/2
 01H: Indicates data sent from the FDD

DID: ID of the destination device
SID: ID of the source device

Device IDs are as follows:

EHT-10/EHT-10/2: 23H
FDD (D: or E:): 31H

Therefore, the following is assumed:
When data is sent from EHT-10/EHT-10/2 to the FDD
 DID=31H and SID=23H

When data is sent from the FDD to EHT-10/EHT-10/2
 DID=23H and SID=31H

FNC: Command issued for the FDD
SIZ: Text data length, which is the actual text data length minus 1

The data sent from the floppy disk drive unit (FDD) has a return code at the end of the text data.

Tables 6.3 and 6.4 lists EPSP functions and return codes, respectively.

Item No.	Command	FNC	Function
1	RESET	00H	Resets the disk drive.
2	READ	77H	Directly reads data from the disk.
3	WRITE	78H	Directly writes data to the disk.
4	WRITEHST	79H	Forcibly writes data to the disk.
5	COPY	7AH	Copies the disk contents to a volume.
6	FORMAT	7CH	Formats the disk.

Table 6.3 List of EPSP functions codes

Return code	Contents	
00H	Normal termination	
FAH	BDOS error	Read error
FBH		Write error
FCH		Drive select error
FDH		Write protection
FFH		Other error

Table 6.4 List of EPSP return codes

<Function>

The RESET command resets the disk drive.

<Send data>

+00H	00H	(FMT)
01H	31H	(DID)
02H	23H	(SID)
03H	0DH	(FNC)
04H	00H	(SIZ)
05H	00H	

<Receive data>

+00H	01H	(FMT)
01H	23H	(DID)
02H	31H	(SID)
03H	0DH	(FNC)
04H	00H	(SIZ)
05H	Return code	

<Explanation>

- Upon receiving the RESET command, the FDD initializes itself and enters receive wait state.

- Return code 00H is sent as response to the mainframe.

2. READ

<Function>

The READ command reads the data stored on the specified sector of the disk.

<Send data>

+00H	00H	(FMT)
01H	31H	(DID)
02H	23H	(SID)
03H	77H	(FNC)
04H	02H	(SIZ)
05H	Drive code	
06H	Track No.	
07H	Sector No.	

Drive code = 1 or 2
Track No. = 0 to 39
Sector No. = 1 to 64

<Receive data>

+00H	01H	(FMT)
01H	23H	(DID)
02H	31H	(SID)
03H	77H	(FNC)
04H	80H	(SIZ)
05H	Read data	128 bytes
06H	Read data	
⋮	⋮	
84H	Read data	
85H	Return code	

<Explanation>

- Upon receiving the READ command, the FDD transfers data (128 bytes) corresponding to the specified logical track and sector numbers and a return code.

<Function>

The WRITE command writes data onto the specified sector of the disk.

<Send data>

+00H	00H	(FMT)
01H	31H	(DID)
02H	23H	(SID)
03H	78H	(FNC)
04H	83H	(SIZ)
05H	Drive code	
06H	Track No.	
07H	Sector No.	
08H	Write type	
09H	Write data	128 bytes
0AH	Write data	
⋮	⋮	
88H	Write data	

<Receive data>

+00H	01H	(FMT)
01H	23H	(DID)
02H	31H	(SID)
03H	78H	(FNC)
04H	00H	(SIZ)
05H	Return code	

Drive code = 1 or 2
Track No. = 0 to 39
Sector No. = 1 to 64
Write type = 0 to 2

<Explanation>

- Upon receiving the WRITE command, the FDD writes the specified data (128 bytes) onto the disk area indicated by the specified logical track and sector numbers.

- Write type = 00H: Standard writing (The FDD performs blocking.)
  01H: Forcible writing (The FDD does not perform blocking and immediately writes data to the disk.)
  02H: Sequential file writing (The FDD performs blocking and writes data at a high speed.)

<Functions>

The WRITEHST command forcibly writes data to the disk.

<Send data>

+00H	00H	(FMT)
01H	31H	(DID)
02H	23H	(SID)
03H	79H	(FNC)
04H	00H	(SIZ)
05H	00H	

<Receive data>

+00H	01H	(FMT)
01H	23H	(DID)
02H	31H	(SID)
03H	79H	(FNC)
04H	00H	(SIZ)
05H	Return code	

<Explanation>

- Upon receiving the WRITEHST command, the FDD forcibly writes the contents of the 1-Kbyte host buffer containing the data sent by the WRITE command to the disk.

<Function>

The COPY command copies the disk contents into a volume.

<Send data>

+00H	00H	(FMT)
01H	31H	(DID)
02H	23H	(SID)
03H	7AH	(FNC)
04H	00H	(SIZ)
05H	Drive code	

<Receive data>

+00H	01H	(FMT)
01H	23H	(DID)
02H	31H	(SID)
03H	7AH	(FNC)
04H	00H	(SIZ)
05H	(*1)	
06H	(*2)	
07H	Return code	

Drive code = 1 or 2
Number of the track being copied = 0 to 39
(*1) Number of the track being copied (low-order)
(*2) Number of the track being copied (high-order)

<Explanation>

- Upon receiving the COPY command, the FDD copies the specified disk contents to another disk in the same FDD.

- This command cannot be used by an FDD having only a single drive.

- Refer to Appendix 9. sample 34

<Function>

The FORMAT command formats the disk.

<Send data>

+00H	00H	(FMT)
01H	31H	(DID)
02H	23H	(SID)
03H	7CH	(FNC)
04H	00H	(SIZ)
05H	Drive code	

<Receive data>

+00H	01H	(FMT)
01H	23H	(DID)
02H	31H	(SID)
03H	7CH	(FNC)
04H	00H	(SIZ)
05H	(*1)	
06H	(*2)	
07H	Return code	

Drive code = 1 or 2
Number of the track being formatted = 0 to 39 (FFFFH = End)
(*1) Number of the track being formatted (low-order)
(*2) Number of the track being formatted (high-order)

<Explanation>

- Upon receiving the FORMAT command, the FDD formats two tracks and returns the corresponding logical track number and return code to the system. The FDD repeats this operation.

- When formatting is completed, the logical track number of the receive data becomes FFFFH.

(3) Using system utilities

Although the operating systems for EHT-10/EHT-10/2 support read/write processing for the FDD by using BIOS, the application program can directly exchange data with the FDD.
The application program uses the following system utilities when directly exchanging data with the FDD:

EPSPSND (EPSP data send utility)
EPSPRCV (EPSP data receive utility)

See Section 10.3 for how to use these system utilities.

6.6.5 Miscellaneous

(1) Forcible writing

When floppy disk TF-15 or PF-10 is used, blocking and deblocking of data are performed to upgrade the data read/write efficiency.
Therefore, when data is written to this floppy disk by using EHT-10/EHT-10/2, data may not yet be actually written to the disk even when the processing by EHT-10/EHT-10II is completed.
To solve this problem, the operating system of EHT-10/EHT-10/2 sends a forcible write command to the disk drive unit at warm BOOT or power off execution to prevent write data from being lost.

(2) Relationship between a disk and serial communications

A floppy disk is connected to EHT-10/EHT-10/2 through an RS-232C Interface. Although three types of I/Fs (RS-232C, IC card, and cartridge SIO) are available for serial communications with EHT-10/EHT-10/2, none of these I/Fs can be used simultaneously with a floppy disk because only one port is used internally by way of switching.
Supporting automatic serial port switching by the operating system enables a floppy disk to be used while, for example, an IC card (BIOS ICCARD) is being used.

Address : Variable name (Number of bytes)

F00BH : SIZRAM (1)
RAM disk standard RAM area size (unit: Kbyte)

F05FH : QT_ROM_CP1 (1)
ROM disk capacity
0: None
20H: 32 Kbytes
40H: 64 Kbytes
80H: 128 Kbytes

F060H : QT_RAM_IN (1)
RAM disk capacity (unit: Kbyte)

F061H : DR_ROM_CP1 (1)
Number of RAM disk directories

F062H : DR_RAM_IN (1)
RAM disk directory area size (unit: 128 bytes)

F063H : AD_ROM_CP1 (2)
ROM disk start address
P format: 6000H
M format: C000H

F065H : AD_RAM_IN (2)
Bottom address of the RAM disk standard RAM area

F067H : CS_RAM_IN (1)
RAM disk check sum area size (unit: 128 bytes)

F068H : RAMD_SIZE
Size of the extended RAM used as the RAM disk (unit: 32Kbytes)

F0E4H : DISKTBL (5)
Logical/physical drive set table (See Section 6.2.)

F0E9H : DISKROV (2)
Disk R/O vector (See Section 6.2.)

F0EBH : FTSTAB (10)
Initial disk select jump vector
Jump vector when the first disk select is specified by BIOS SELDSK

F0F5H : READTAB (10)
Jump vector for disk reading
This vector is used for jumping to each read process during BIOS READ execution.

F0FFH : WRTTAB (10)
Jump vector for disk reading
This vector is used for jumping to each write process during BIOS WRITE execution.

F109H : DPBASE
Refer to the CP/M-related manual for the disk parameter header configuration.
DPE0 (16): RAM disk DPH
DPE1 (16): ROM disk DPH
DPE2 (16): IC card DPH
DPE3 (16): Floppy disk drive 1 DPH
DPE4 (16): Floppy disk drive 2 DPH

F159H : DPB0 (15)
RAM disk DPB (Disk Parameter Block)

D168H : DPB1 (15)
ROM disk DPB

D177H : DPB2 (15)
IC card DPB

D186H : DPB3 (15)
Disk drive DPB

F4D7H : DIRBUF (128)
Directory access buffer

F557H : ALV0 (29)
RAM disk allocation area

F574H : CSV0 (0)
RAM disk check sum area (Defined label name only)

F574H : ALV1 (16)
ROM disk allocation area

F584H : CSV1 (0)
ROM disk check sum area (Defined label name only)

F584H : ALV2 (8)
IC card allocation area

F58CH : CSV2 (8)
IC card check sum area

F59CH : ALV3 (18)
Floppy Disk drive 1 allocation area

F5AEH : CSV3 (16)
Floppy Disk drive 1 check sum area

F5BEH : ALV4 (18)
Floppy Disk drive 2 allocation area

F5D0H : CSV4 (16)
Floppy Disk drive 2 check sum area

F82DH : SYSFCB (36)
System FCB area

F851H : SYSDMA (128)
System DMA buffer

CHAPTER 7 I/O INTERFACE OVERVIEW

7.1 Overview

This chapter explains the EHT-10/EHT-10/2 I/O interface.

(1) I/O registers

For EHT-10/EHT-10/2, all I/O operations are controlled through I/O registers in the gate array. The I/O registers in the gate array can be directly controlled by I/O commands from the main CPU.

EHT-10/EHT-10/2 uses the 39 I/O ports listed in the table below among 256 I/O ports.

I/O port	Explanation
P00H to P0FH	Used by the system.
P10H to P13H	Used by the system. However, this address space is used to extend a cartridge.
P14H to P26H	Used by the system.
P27H to PFFH	Unused

Table 7.1 I/O Ports

Refer to "PART 3 HARDWARE" for details on I/O structure.

(2) Accessing I/O registers

The EHT-10/EHT-10/2 OS stores the current output contents of an I/O register as the I/O register exit in the system area. An application program can modify I/O register contents as follows:

		(Example)
The value stored in the system area is fetched.	(1)	LD A, (RZIOCTLR)
Only the bit to be modified is manipulated.	(2)	SET 6, A
The modified value is reset in the system area.	(3)	LD (RZIOCTLR), A
The value set in step 3 is output to an I/O port.	(4)	OUT (19H), A

Fig 7.1 Accessing an I/O Register

Table 7.2 shows the I/O registers that require the operations shown in Figure 7-2.

I/O address : Register name (RAM data address :RAM variable name)

P00H : CTLR1 (F0D6H : RZCTLR1)

P02H : CTLR2 (F0D8H : RZCTLR2)

P04H : IER (F42EH : RZIER)
Interrupts are suppressed while this register is being rewritten.

P05H : BANKR (F42CH : RZBANKR)
Interrupts are suppressed while this register is being rewritten.

P08H : VADR (F254H : LSCRVRAM +1)

P09H : YOFF (F266H : LVRAMOF) Only bits 6 to 0 are stored.
(F074H : DSPFLAG) Only bit 7 is stored.

P15H : ARTMR (F0D9H : RZARTMR)

P16H : ARTCR (F0DAH : RZARTCR)

P17H : ICCTLR (F0DBH : RZICCTLR)
Interrupts are suppressed while this register is being rewritten.

P18H : SWR (F0DCH : RZSWR)

P19H : IOCTLR (F0DDH : RZIOCTLR)

P22H : SBKR (F42DH : RZSBBANKR)
Interrupts are suppressed while this register is being rewritten.

P23H : CTLR3 (F0DEH : RZCTLR3)
Interrupts are suppressed while this register is being rewritten.

Table 7.2 I/O Registers of Which System Area Must be Rewritten

7.2 Serial Interface

7.2.1 Overview

EHT-10/EHT-10/2 has RS-232C, cartridge SIO, and IC card as the serial communication devices. However, EHT-10/EHT-10/2 has only one serial I/O port and the output destination is switched by the internal serial switch. Because of this, these serial devices cannot be used at the same time as a rule.
The user is not required to switch the serial switch since the OS automatically switches the serial mode.

7.2.2 Setting a serial device

Table 7.3 shows the I/O registers used by the serial I/F. Refer to "PART 3 HARDWARE" for details on each I/O register.

(1) Switching the serial port

The three serial ports for RS-232C, IC card, and cartridge SIO are switched by SWR (P18H).

R/W	I/O Address	Register name	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Notes
W	P18H	SWR					SSW1	SSW0			(*1)

(*1) Bits other than bits 3 and 2 are used for others.

Serial mode	SSW1	SSW0	RXD/TXD	Example of corresponding device
0	0	0	Cartridge SIO	
1	0	1	IC card	IC card
2	1	0	RS-232C	FDD, printer, or coupler

Table 7.3 Serial Port Switching

(Note) Both SSW1 and SSW0 cannot be turned to 1 at the same time.

(2) Serial parameters and data control line

Table 7.4 shows the I/O registers related to setting the serial parameters, reading/writing data, and setting/reading the control line.

R/W	I/O Address	Register name	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Notes
R	P14H	ARTDIR	7 or 8 bits data								
R	P15H	ARTSR	RDSR		FE	OE	PE	TX Emp	RX RDY	TX RDY	
R	P16H	IOSTR			* RCTS	* RCD	RXD				(*1)
W	P00H	CTLR1	BRG3	BRG2	BRG1	BRG0					(*2)
W	P14H	ARTDOR	7 or 8 bits data								
W	P15H	ARTMR	STOP		EVEN	PEN			DATA Lng.		
W	P16H	ARTCR			* RRTS	ER	SBRK	RXE	* RDTR	TXE	

(*1) Bits other than bits 5 to 3 are used for others.
(*2) Bits other than bits 3 to 0 are used for others.
(Note) A bit indicated by an asterisk (*) is effective only when the serial switch is set to RS-232C.

Table 7.4 I/O Registers Related to Control Line

7.2.3 Serial ports supported by OS

(1) Overview

The EHT-10/EHT-10/2 OS supports the following methods to effectively use the three serial ports:

1 By the user using BIOS RSIOX
2 By the user using BIOS ICCARD
3 As a system I/O device (LST:, PUN:, RDR:, or CON:)
4 As a terminal floppy disk (TF)
5 As an IC card disk

OS classifies 1 , 2 , and 3 as one mode among the above five and automatically switches the serial switch so that each of the following three modes can operate independently from each other:

1 Used by the user (BIOS RSIOX, TCAM, ICCARD, and I/O device)
2 As TF
3 As IC card disk

Table 7.5 and Figure 7.2 show the relationship between the modes and devices.

Module to be newly used / Current use mode		User			System	
		RSIOX TCAM	ICCARD	I/O device	FDD	IC card (disk)
User	RSIOX,TCAM	-	x	x	o(*1)	o(*1)
	ICCARD	x	-	x	o(*1)	x
	I/O device	x	x	-	o(*2)	o(*1)
System	FDD	o	o	o(*2)	-	o
	IC card (disk)	o	x	o	o	-

Table 7.5 Relationship between Devices and Modes

(*1) Receive data may be ignored if FDD or IC card is used while the user is using the serial port. To prevent this, use a protocol that does not accept receive data while FDD or IC card is being used.

(*2) This is meaningless because the same device (RS-232C I/F) is actually used.

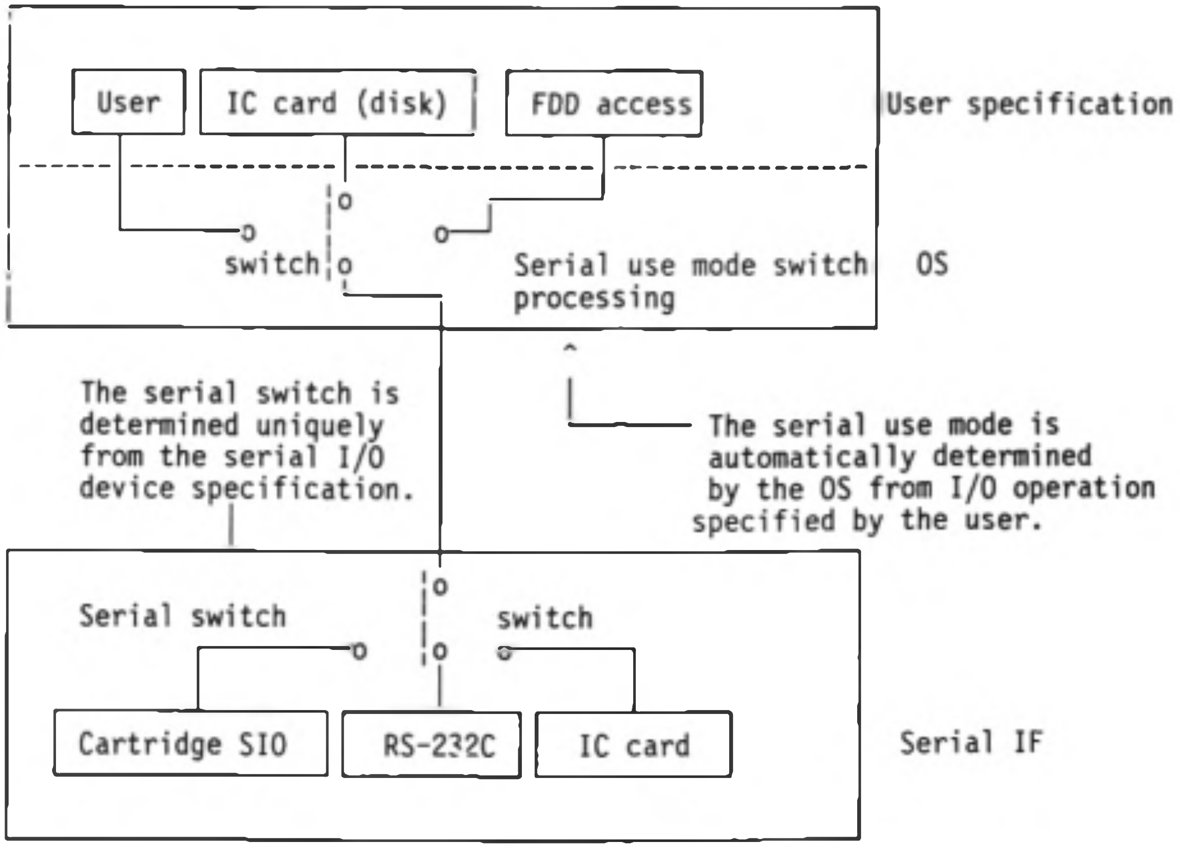


Fig. 7.2 Concept of Using Serial Port

Serial port switch processing is performed as follows:
The current serial use mode is stored in SRMODE (F241H). Processing is executed without changing the current serial use mode when the use mode is equal to the current one.
Processing is executed after the new serial use mode is set when the use mode is not equal to the current one.
The parameters used to change the serial use mode are stored in the 15-byte area starting from SRTABL (F196H).

7.2.4 Serial communication by the user

(1) Overview

The serial interface is generally controlled by BIOS RSIOX and TCAM and not by the user directly.
However, the user must directly control the serial interface to extend serial communication or the IC card protocol explained in Chapter 11. This section explains the procedure used by the user to directly control the serial I/F.

(2) Open processing

The serial I/F is opened in the following procedure:

Open check
Whether the serial I/F is being used is checked,

```
RSODEV (F623H) =00H:  RS-232C
               =01H:  IC card (BIOS ICCARD)
               =03H:  Cartridge SIO
               =FFH:  The serial I/F is not in open status.
```

An error occurs if the serial I/F is in open status (see note 1).

The code of the device to be used is set if the serial I/F is not in open status.

(Note 1) The serial I/F has been opened for RS-232C if RS-232C is being used as the I/O device (LST:, PUN:, RDR:, or CON:). In this case, the serial I/F can be closed by using BIOS RSIOX.

2 Setting the serial parameters
A 15-byte area starting from SRSTABL (F196H) contains three 5-byte packets for IC card (used as a disk), user, and FDD in this order. The serial parameters are set in one of these packets.
The serial parameters are set in the IC card packet for an IC card (used as a disk) or in the user packet in any other cases (see the following).

Address : Variable name (Byte length)

F241H : SRMODE (1)

- This indicates the current serial use mode as follows:
  - =FFH: Unused
  - =00H: IC card (as a disk) is being used.
  - =01H: The user is using the serial I/F.
  - =02H: FDD is being used.
- The initial value is FFH and this parameter is initialized when warm boot is executed.

F196H : SRTABL (15)

- This table is used to switch the serial use mode.

Name	Byte length	Contents
SYSCTLR1	1	IC card (disk) CTLR1
SYSARTMR	1	IC card (disk) ARTMR
SYSSWR	1	IC card (disk) SWR
SYSARTCR	1	IC card (disk) ARTCR
SYSSOUT	1	unused
RS2CTLR1	1	User CTLR1
RS2ARTMR	1	User ARTMR
RS2SWR	1	User SWR
RS2ARTCR	1	User ARTCR
RS2SOUT	1	unused
HSCTLR1	1	FDD CTLR1
HSARTMR	1	FDD ARTMR
HSSWR	1	FDD SWR
HSARTCR	1	FDD ARTCR
HSSOUT	1	unused

- The table is separated in the units of 5 bytes and data corresponds to I/O registers CTLR1, ARTMR, SWR, ARTCR, or IOCTLR.
- The bits not related to serial I/F must be left as 0
- See "Part 2 Hardware chapter 2 I/O register" for details.

3 Switching the serial switch

According to the parameters explained in 2 , the serial switch is changed and the serial parameters are set for the serial controller. The serial switch is changed by using FSELSER shown in APPENDIX 9 SAMPLE20.
In this case, the entry parameter is set in register C as follows:
Register C=00H: IC card as a disk
=01H: Other than above

4 Allowing receive interrupts

Receive interrupts must be enabled during open processing if receive interrupts are required for serial communication (including IC card) extension processing (see "CHAPTER 8 ART INTERRUPTS).

(3) Sending and receiving data and controlling the control line

The serial mode must be switched by SELSER before send/receive operation every time data is sent or received (see note 1).
The entry parameters explained in Section (2) for FSELSER is used for SELSER.

The serial mode must also be switched to control the control line or to recover from an error.

(Note 1) When data is received or sent during extension processing or when FDD and IC card are used at the same time, the current mode is not necessarily be the one set at open processing because OS automatically changes the serial switch.

(4) Close processing

1 FFH is set in RSODEV (F623H) to indicate that there is not a device in open status.

2 FFH is set in SRMODE (F241H) to indicate that a serial device is not being used.

3 Receive interrupts must be disabled if they were enabled.

7.3.1 Overview

The EHT-10/EHT-10/2 functions can be extended by connecting an option cartridge to the cartridge interface.
The printer unit is supported in standard as a cartridge option. However, the user can create a cartridge device by using the universal cartridge.
The cartridge I/F has HS, IO, DB, and OT modes. One of these mode can be selected according to the characteristics of the cartridge device.
See the following sections for the cartridge interface:

1 PART 3 HARDWARE Section 4.1 Cartridge Interface
2 Section 11.4 Extending a Cartridge Device

7.3.2 Modes and how to set a mode

(1) Modes

EHT-10/EHT-10/2 cartridge I/F has four modes (HS, IO, DB, and OT). One of these modes can be selected according to the characteristics of the cartridge device.

1 Hand shake (HS) mode
This is CPU-to-CPU interface mode used for the device that has CPU at the option side. Data is sent or received through the input or output buffer. Data transmission is controlled according to the flags (IBF and OBF).

2 Input output port (IO) mode
The interface is in the form of 4-bit input and output ports.

3 Data bus (DB) mode
In this mode, an option looks as a general I/O device in the view of the main frame. The cartridge interface simply connects the data bus of the main frame to the cartridge data bus.

4 Output port (OT) mode
The interface is in the form of 8-bit output port.

See "PART 3 HARDWARE Section 4.11 Cartridge Interface" for details on each mode.

(2) Setting mode

The cartridge interface mode can be selected by using the cartridge switches CSW1 and CSW0 (bits 1 and 0) in SWR (P18H).
CSW1 and CSW0 are set by the initialization routine in the gate array as follows:
CSW1=0 and CSW0=0 (HS mode)

CSW1	CSW0	Mode
0	0	HS mode
0	1	IO mode
1	0	DB mode
1	1	OT mode

(Note 1) OS automatically sets a mode so that a user program is not required to set it. See the next section for the mode setting procedure done by OS.
(Note 2) HS mode is set after the power on of the EHT-10/EHT-10/2 main frame. However, OS automatically resets the specified mode at power on processing so that the user is not required to consider the mode.

7.3.3 Determining the mode

(1) Overview

Since the EHT-10/EHT-10/2 OS automatically sets the cartridge mode, user programs need not set the mode.
Mode setting is performed in the following timing:

1 EHT-10/EHT-10/2 main frame power on
2 Reset processing
3 System initialization

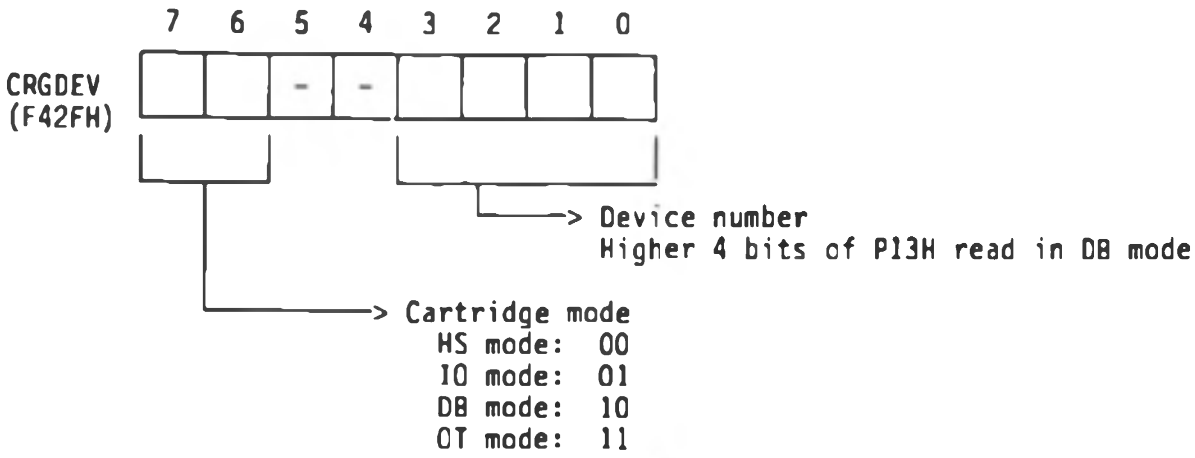
The device number of cartridge device and the CSEL signal must be set for a cartridge option created by the user because the mode is determined from the device number and CSEL signal (see "PART 2 HARDWARE Section 4.1 Cartridge Interface" for details).

(2) CSEL signal and device number

1 CSEL signal
The CSEL signal is the signal line used to check whether the cartridge option is in HS mode. The CSEL signal is in bit 6 of I/O register P16H.
CSEL=1: HS mode
=0: Other mode (IO, DB, or OT mode)

2 Device number
The device number indicates the type of cartridge option. The device number is in the higher 4 bits of I/O register P13H read in DB mode. A device number is indicated by a code and is set according to the operation mode of the cartridge option (see Table 7.6).

3 Device management by OS
OS manages the cartridge device according to the device code consisting of the cartridge mode and device number and stored in CRGDEV (F42FH).



Device number	CSEL=1		CSEL=0	
0H	No option			
1H 2H 3H	Printer unit	HS mode		For DB mode extension
4H 5H 6H 7H	(M160 Printer)	For HS mode extension		IO mode
8H 9H AH BH				DB mode
CH DH EH FH				OT mode

Table 7.6 CSEL and Device Number

(3) Mode determination procedure

Figure 7.3 shows the mode determination procedure. See "Section 11.4 Extending Cartridge Device" for CRGHOOK in the figure.

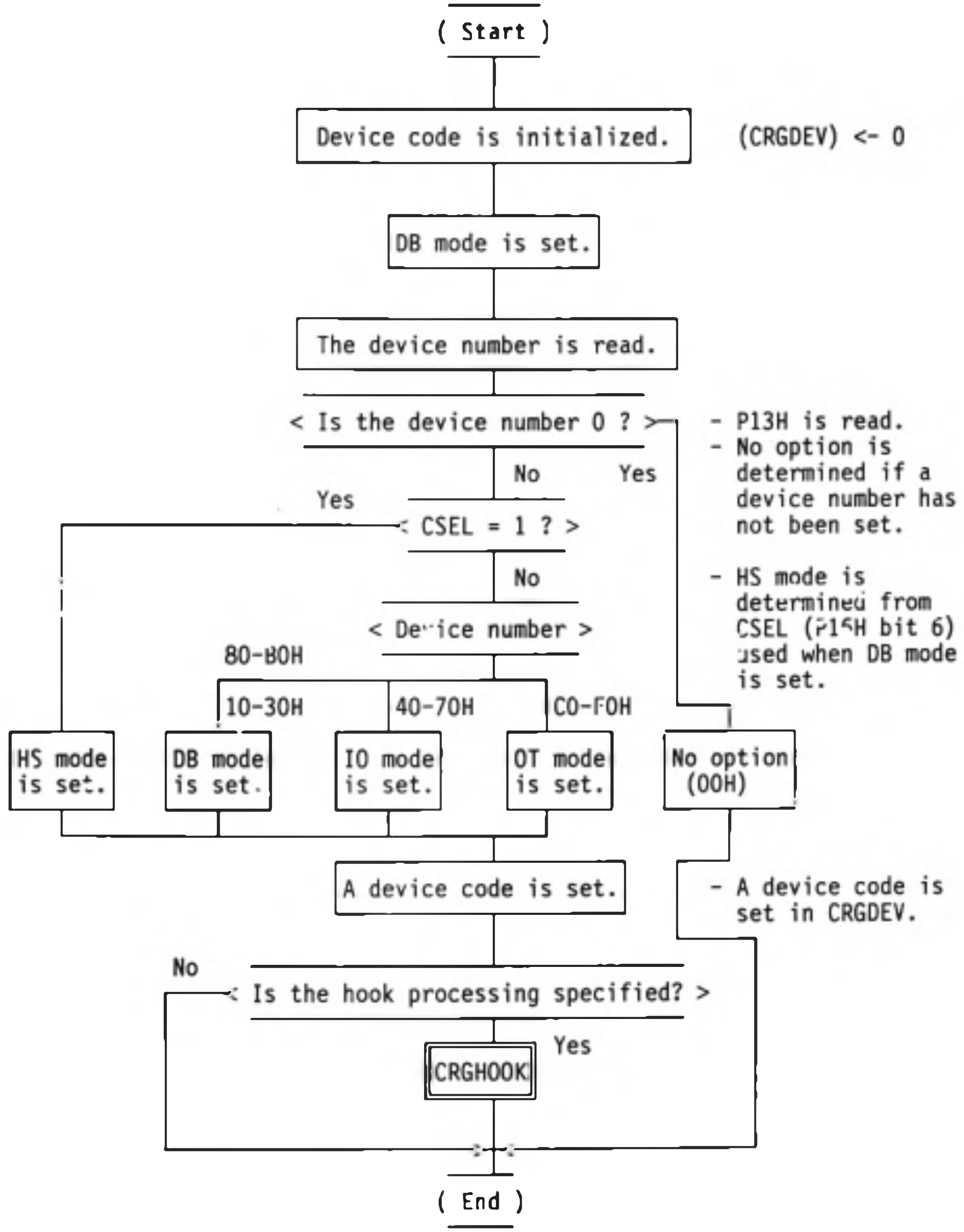


Fig 7.3 Cartridge Mode Determination

7.3.4 I/O registers used for cartridge I/F

Table 7.7 shows the I/O registers used to control the cartridge devices. The contents of I/O registers P10H to P13H differ depending on the cartridge mode. See "PART 2 HARDWARE Section 4.1 Cartridge Interface" for details.

See the beginning of this chapter for accessing I/O registers.

[Note]

The following I/O registers among the ones related to cartridge I/F must not be accessed (modified) by the user:

1 DCTG (P17H bit 4)
This bit is controlled at debugger operation and is usually set to 0.

2 CSW0 and CSW1 (P18H bits 0 and 1)
They are the cartridge mode switch and are usually set in device management processing executed by OS.

3 PINTDS (P23H bit 3)
This masks IC card and cartridge interrupts. Generally, interrupt allowed status is set. IC card and cartridge interrupts are masked independently in ICCTLR (P17H).

Table 7.7 I/O Registers Related to Cartridge Devices

```
Port Address (R/W) : Port name (RAM data address)

P10H  to P13H (R) : General input register
                    The contents differ depending on the cartridoe mode (see
                    "PART 3 HARDWARE" for details).

P10H  to P13H (W) : General output register
                    The contents differ depending on the cartridge mode (see
                    "PART 3 HARDWARE" for details).

P16H (R) : IOSTR
     bit 0 : (PBUSY)  Interrupt status
                    0:  Interrupt requested
                    1:  No interrupt requested
     bit 6 : (CSEL)  Cartridge option determination signal
                    0:  Mode other than HS mode
                    1:  HS mode

P17H (W) : ICCTLR (F0DBH)
     bit 4 : (DCTG)  Development cartridge switching signal
                    0:  Normal mode
                    1:  Development cartridge mode
                    (The user must not modify this bit.)
     bit 6 : (PRIE)  Interrupt mask
                    0:  Interrupt allowed
                    1:  Interrupt suppressed

P18H (W) : SWR (F0DCH)
     bit 0 to 1 : (CSW0 and CSW1)  Cartridge mode switch
                    (The user must not modify these bits.)
```

P19H (W) : IOCTLR (F0DDH)

bit 0 : (PF) Outputting to general output line (CCTL0)

bit 1 : ($\overline{\text{SLIN}}$) Outputting to general output line (CCTL1)

bit 3 : ($\overline{\text{PINI}}$) Cartridge reset

0: Reset on

1: Reset off

P23H (R) : ITSR

bit 1 : (IPBUSY) Status of interrupt from IC card or cartridge

0: Interrupt requested

1: No interrupt requested

P23H (W) : CTLR3 (F0DEH)

bit 3 : (PINTDIS) Masking an interrupt sent from IC card or cartridge

0: Interrupts allowed

1: Interrupts suppressed

(0 is generally set.)

7.4.1 Overview

EHT-10/EHT-10/2 has the IC card interface and OS supports BIOS ICCARD and an IC card as a disk. See "CHAPTER 4 BIOS OVERVIEW", "Section 6.5 IC card" and "Section 11.3 Extending IC Card Protocol" for details.

This section explains the I/O registers used to control IC card I/F and power control related to the IC card I/F.

7.4.2 I/O registers used for IC card I/F

The I/O address space related to the IC card I/F is structured as shown below.

R/W	I/O Address	Register name	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Notes
R	P16H	IOSTR						IC CLS			(*1)
W	P17H	ICCTLR			IC ITE			IC DIR	ICC	ICO SC	
W	P19H	IOCTLR						ICR			(*1)

(*1) Bits other than bit 2 are used for others.

Table 7.8 I/O Registers Related to IC Card I/F

The serial communication I/O port is used in addition to the I/O ports shown above because an IC card is accessed through serial communication. Other bits must be saved when data is written in a write register (ICCTLR or IOCTLR). Write data saved in memory is fetched, only the bit to be modified is manipulated, and then the value is written in memory and the register. The memory addresses of data corresponding to registers are listed below.

I/O		PAM	
I/O address	I/O name	Address	Name
P17H	ICCTLR	FODBH	RZICCTLR
P19H	IOCTLR	FODDH	RZIOCTLR

Register name

IOSTR (IO Status Register)

ICCLS: Indicates the lid status of IC card reader.

=1: Closed

=0: Opened

ICCTLR (IC card Control Register)
ICITE: Interrupt control according to the lid status of IC card reader
=0: Disabled
=1: Enabled
ICDIR: IC card reader read/write switch
=0: Write
=1: Read
ICC: IC card reader power control
=0: Vcc not supplied
=1: Vcc supplied
ICOSC: IC card reader clock oscillation control
=0: Not oscillated
=1: Oscillated

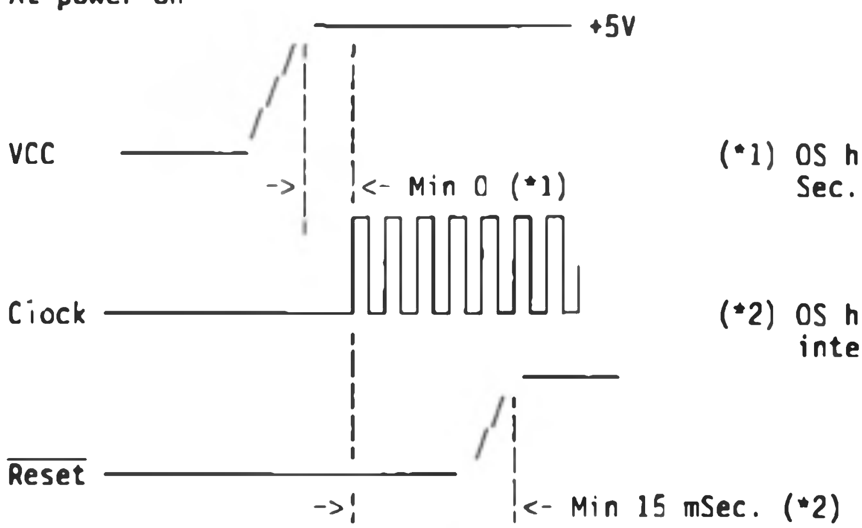
IOCTLR (IO Control Register)
ICR: IC card reader reset signal control
=0: Reset off
=1: Reset on

7.4.3 Controlling IC card reader power supply

Power (+5 V), clock, and reset signals are supplied to the IC card reader. The power and clock require a fixed time after turned on to become stable.

This section explains the power, clock, and reset control timings at power on and off.

(1) At power on

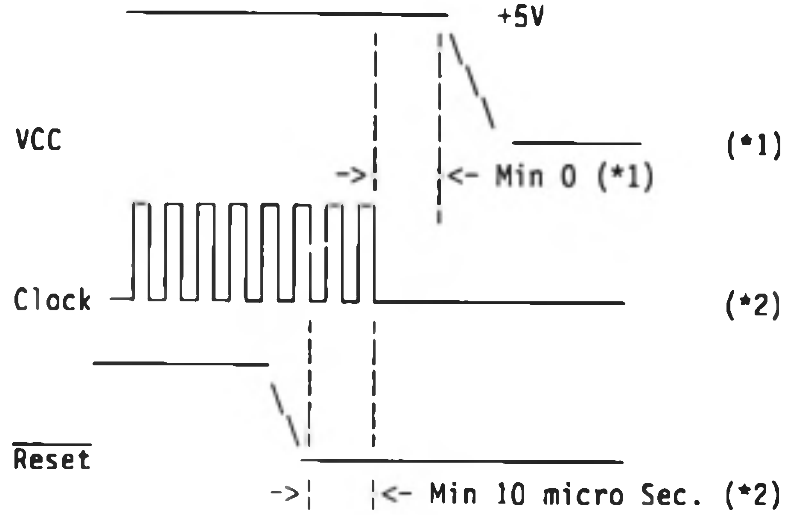


(*1) OS has about 10 micro Sec. interval.

(*2) OS has about 50 mSec. interval.

Fig 7.4 At IC Card Reader Power On

(2) At power off



(*1) OS has about 10 micro Sec. interval.

(*2) OS has about 50 mSec. interval.

Fig 7.5 At IC Card Reader Power Off

7.5 Barcode Reader Interface

7.5.1 Overview

EHT-10/EHT-10/2 has the barcode interface and OS supports the barcode interface with BIOS BARCODE. BIOS BARCODE supported by OS is explained in detail in "CHAPTER 4 BIOS OVERVIEW" and "Section 11.5 Adding Barcode Decoder".
This section explains the I/O registers used for barcode I/F and the controlling barcode reader power supply.

7.5.2 I/O registers for barcode I/F

The EHT-10/EHT-10/2 barcode interface consists of +5 V logic power supply, +5 V LED power supply, and barcode input signals.
The following figure shows the I/O address space related to the barcode I/F:

R/W	I/O Address	Register name	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Notes
R	P00H	ICRL.C	Lower 8 bits of the current FRC.								
R	P01H	ICRH.C	Upper 8 bits of the Current FRC.								
R	P02H	ICRL.B	Lower 8 bits of FRC held when the barcode input signal is changed.								
R	P03H	ICRH.B	Upper 8 bits of FRC held when the barcode input signal is changed.								
R	P04H	ISR						ICF			(*1)
R	P05H	STR							BCRD		(*2)
W	P00H	CTLR1					SW BCR	BCR1	BCR0		(*3)
W	P04H	IER						ICF			(*1)
W	P17H	ICCTLR					BCRP				(*4)

(*1) Used by others except for bit 2
(*2) Used by others except for bit 1
(*3) Used by others except for bit 3 to 0
(*4) Used by others except for bit 3

Other bits must be saved when data is written in a write register (CTLR1, IER, or ICCTLR). Write data saved in memory is fetched, only the bit to be modified is manipulated, and then the value is written in memory and the register. The memory addresses of data corresponding to registers are listed below.

I/O		RAM	
I/O address	I/O name	Address	Name
P00H	CTLR1	F0C6H	RZCTLR1
P04H	IER	F42EH	RZIER
P17H	ICCTLR	F0DBH	RZICCTLR

Register name

ICRL.C (Input Capture Register Low Command Trigger)
    Lower 8 bits of the current free running counter (FRC)

ICRH.C (Input Capture Register High Command Trigger)
    Higher 8 bits of the current FRC. ICRL.C must be read first if it is required to be read.

ICRL.B (Input Capture Register Low Barcode Trigger)
    Lower 8 bits of FRC held when the barcode input signal is changed.

ICRH.B (Input Capture Register High Barcode Trigger)
    Higher 8 bits of FRC held when the barcode input signal is changed. The interrupt occurred due to barcode input signal change is reset when this register is read.

ISR (Interrupt Status Register)
    ICF: Interrupt that occurs when FRC is latched to ICR due to barcode input signal change

STR (Status Register)
    BCRD: Barcode input signal

CTLR1 (Control Register 1)
    SWBCR: +5 V logic power switch for barcode reader
        0 = Off
        1 = On
    BCR1 or 0: Latch trigger polarity selection
        00 = Trigger suppressed
        01 = Fall trigger
        10 = Rise trigger
        11 = Rise and fall triggers

IER (Interrupt Enable Register)
    ICF: ICF interrupt control

ICCTLR (IC card Control Register)
    BCRP: Barcode reader LED power switch

7.5.3 Barcode reader power supply control

Logic power (+5V) and LED power (+5V) are supplied to the barcode reader. Logic power and LED power require a fixed time after turned on to become stable.

Fig 7.6 shows the logic and LED power timings.

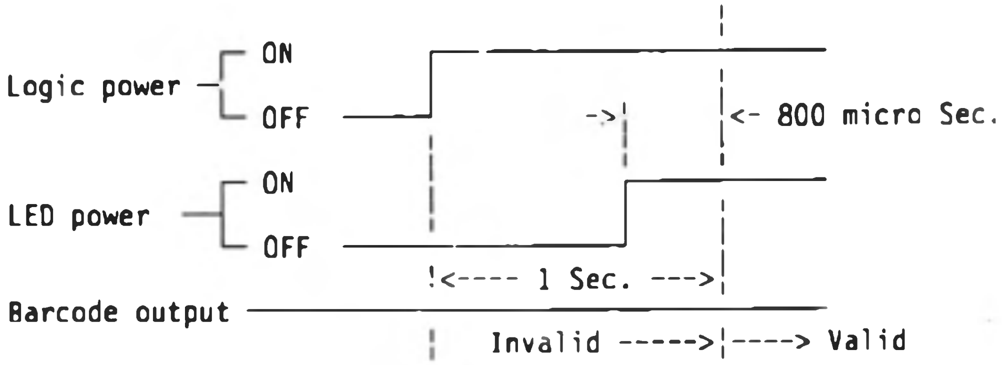


Fig 7.6 Barcode Reader Power Supply Control Timing

Barcode output is not stable for 1 second after the logic power is turned on. Also, barcode output is not stable for 800 micro seconds after the LED power is turned on.

7.6 7508 Commands

This section explains the function overview of slave CPU 7508, data transmission to and from 7508, and various commands.
See "Section 4.3 Key Input" and "Section 8.4 7508 Interrupts" for reference.

7.6.1 7508 functions

7508 has the following functions:

(1) Keyboard scan and control
(2) Main-CPU power on/off
(3) Reset switch control
(4) Battery voltage management and switching
(5) Alarm function
(6) 1-second timer
(7) Power switch control
(8) Calendar clock
(9) D-RAM refresh signal control
(10) Serial data transmission to and from main CPU

Commands and data are transmitted to or from Z-80 through serial data line in handshake mode.

For functions (1) to (7), an interrupt is sent to Z-80 and the Z-80 side can find out the cause by reading the 7508 status.

7.6.2 7508 interface

(1) Input and output ports

EHT-10/EHT-10/2 provides the following I/O ports to transmit commands and data to or from 7508:

1 For serial data transmission

P06H (R) [SIOR]

7	6	5	4	3	2	1	0
8 bits data							

Data sent from 7508.

P06H (W) [SIOR]

7	6	5	4	3	2	1	0
8 bits data							

Command and data sent to 7508

P05H (R) [STR]

7	6	5	4	3	2	1	0
x	x	x	x	RDYSIO	x	x	x

RDYSIO=0: Access prohibited
1: Access allowed

P01H (W) [CMDR]

7	6	5	4	3	2	1	0
x	x	x	x	x	x		x

└> RESET RDYSIO: Controls RDYSIO signal.
=0: No operation is performed.
=1: The RDYSIO signal is reset.

2 Interrupts

P04H (R) [ISR]

7	6	5	4	3	2	1	0
x	x	x	x	x	x	x	

——>INT0

This indicates that an interrupt is sent from 7508.
INT0=0: No interrupt
1: An interrupt is sent.

P04H (W) [IER]

7	6	5	4	3	2	1	0
x	x	x	x	x	x	x	

——>IER0

This indicates that interrupts sent from 7508 are suppressed or allowed.
IER0=0: Interrupts suppressed
1: Interrupts allowed

x indicates that this bit is not related to 7508.

(2) 7508 and transmission procedure

This section explains the procedure to transmit data to or from 7508 and notes on transmission operations.

1 Sending data to 7508
Figure 7.7 shows the procedure to send a command or data to 7508.

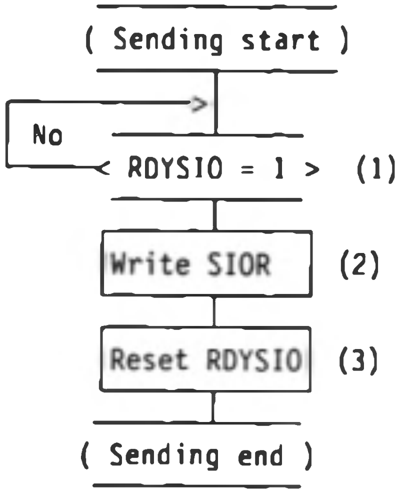


Fig 7.7 Command (Data) Send Procedure

To send data after each command, the above operation is repeated as much as the number of commands and data items.

Step : Explanation

(1) RDYSIO?
Whether 7508 is ready to receive a command (data) is checked. P05H is read, processing goes to step (2) if bit 3 in P05H is 1, and step (1) is repeated if bit 3 is 0.

(2) Write SIOR
A command (data) is sent to 7508.
Data or command to be sent is written in P06H.

(3) Reset RDYSIO
7508 RDYSIO signal is reset. 02H is written in P01H.

2 Receiving data from 7508
Figure 7.8 shows the procedure to receive data from 7508.

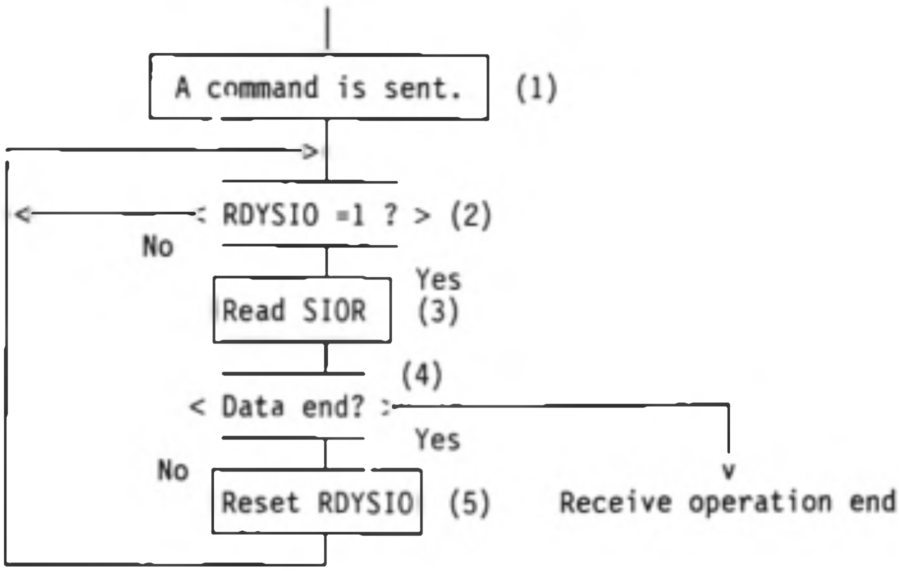


Fig 7.8 Data Receive Procedure

Step : Explanation

(1) Command sending
A command is sent according to the send procedure shown in the above figure.

(2) RDYSIO?
Whether data sent from 7508 can be received is checked.
P05H is read, processing goes to step (3) if bit 3 in P05H is 1, and step (2) is repeated if bit 3 in P05H is 1.

(3) Read SIOR
Data received from 7508 is read.
P06H is read to fetch data sent from 7508.

(4) Data end?
Whether data items as much as the number of sent commands are received is checked. Processing goes to step (5) if there are more data items to be received.

(5) Reset RDYSIO
7508 RDYSIO signal is reset. 02H is written in P01H.

(3) Notes

This section explains the notes on transmitting commands and data to or from 7508.

1 Interrupts sent from 7508 must be suppressed while a command or data is transmitted to or from 7508.
Interrupts can be suppressed in the following three ways:

(a) DI instruction
(b) Rewriting IER (P04H)
(c) BIOS MASKI

Interrupts must be suppressed while processing shown in Figures 7.7 or 7.8 is being executed.

2 Send or receive data sequence for 7508 commands must be completed. In other words, subsequent operations are not guaranteed unless required number of data items are sent or received.

7.6.3 7508 commands

This section explains the 7508 commands. Table 7.10 shows the list of 7508 commands.

Number : Function (Code)

1 : Power OFF (01H)
2 : Read Status (02H)
3 : KB Reset (03H)
4 : KB Repeat Timer 1 Set (04H)
5 : KB Repeat Timer 2 Set (14H)
6 : KB Repeat Timer 1 Read (24H)
7 : KB Repeat Timer 2 Read (34H)
8 : KB Repeat OFF (05H)
9 : KB Repeat ON (15H)
10 : KB Interrupt OFF (06H)
11 : KB Interrupt ON (16H)
12 : Clock Read (07H)
13 : Clock Write (17H)
14 : Power Switch Read (08H)
15 : Alarm Read (09H)
16 : Alarm Set (19H)
17 : Alarm OFF (29H)
18 : Alarm ON (39H)
19 : DIP Switch Read (0AH)
20 : 7 Characters Buffer (0CH)
21 : 1 Character Buffer (1CH)
22 : 1 Sec. Interrupt OFF (0DH)
23 : 1 Sec. Interrupt ON (1DH)
24 : KB Clear (0EH)
25 : System Reset (0FH)

Table 7.10 7508 Commands

Command or data can be identified from MSB. MSB is 0 for a command and 1 for data.

[Function]
This command turns off the main-CPU power.

[Sequence]

7	6	5	4	3	2	1	0	
0	0	0	0	0	0	0	1	(Write)

<Explanation>

(1) The main-CPU (Z-80) power is turned off.

(2) This command cannot be used in an application program. BIOS POWER OFF is used in an application program in order to turn the power off.

[Function]

This command reads the 7508 status or key code.

[Sequence]

	7	6	5	4	3	2	1	0	
	0	0	0	0	0	0	1	0	(Write)
									(Read) bits 7 to 0 : Status or key code

<Explanation>

(1) This command is used to read the 7508 status when an interrupt is sent from 7508 or after reset is cleared.

(2) The status is mainly classified as follows:

1 Key code
2 Status

The rest of this section explains the key code and status.

1 Key code (scan code)
When the read 7508 status is BEH or less, data is a key scan code. Figure 7.9 shows the scan codes of EHT-10/2 keys. 30H is indicated as the status when a touch-panel key on the EHT-10/2 is pressed. Actually, touch-panel key is scanned during OS interrupt processing.

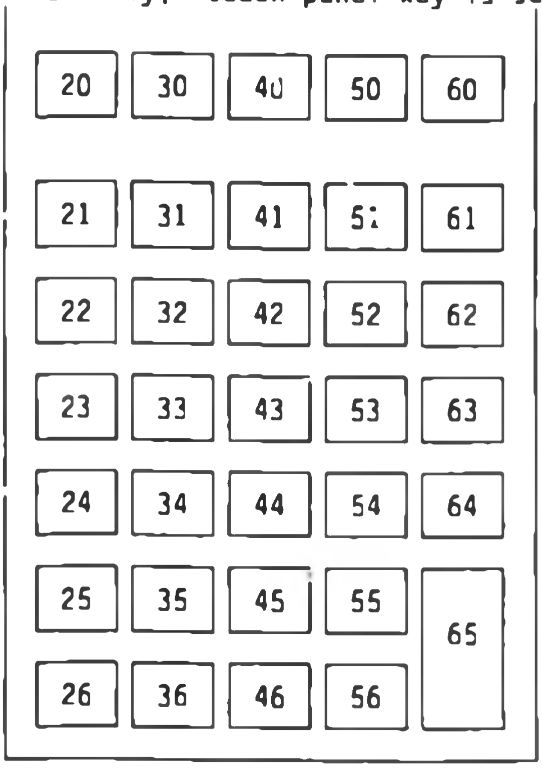


Fig 7.9 Scan Codes (EHT-10/2)

(Note) The values in the left figure are represented in hexadecimal notation .

2 Status

When the read 7508 status is C0H or more, the status has been changed due to interrupt occurrence or reset clearance. BFH is indicated when the status has not been changed.

The cause of status change is indicated by a bit as shown below. If there are two or more causes of status change, two or more bits are turned to 1.

7	6	5	4	3	2	1	0
1	1						

- -> Power switch change
- -> Alarm interrupt (Occurrence is indicated by 1.)
- -> Power failure interrupt (Occurrence is indicated by 1.)
- -> 7508 reset (Occurrence is indicated by 1.)
- -> Reset (Occurrence is indicated by 1.)
- -> 1-second interrupt (Occurrence is indicated by 1.)

To determine the subsequent operations, the EHT-10/EHT-10/2 OS reads the 7508 status during interrupt processing when an interrupt occurs, or during 0-address start processing (system initialization, reset, power on, or alarm/wake processing) when reset is cleared.

[Function]
This command initializes the keyboard status.

[Sequence]

7	6	5	4	3	2	1	0	
0	0	0	0	0	0	1	1	(Write)

<Explanation>

(1) The keyboard status is initialized.
1 656 ms is set as the keyboard repeat start time.
2 70 ms is set as the keyboard repeat interval.
3 The key buffer is cleared.
4 Interrupts caused by key input operation are allowed.

(2) The keyboard is scanned and the information of the key that has been pressed is stored in the buffer.

(3) OS uses this command during system initialization and reset processing.

[Function]

This command sets the keyboard repeat start time.

[Sequence]

7	6	5	4	3	2	1	0	
0	0	0	0	0	1	0	0	(Write)
1								(Read) bits 6 to 0 : Status or key code

<Explanation>

(1) When a general key other than switches and special keys is kept pressed, the input is repeated. This command sets the time after a key is pressed until repeat operation is started.

(2) A value up to 2 seconds can be specified in the unit of 1/64 seconds (about 15 ms) as send data. 1 must be set as MSB of data.

(3) The initial value is 656 ms.

[Function]
This command sets the keyboard repeat interval.

[Sequence]

7	6	5	4	3	2	1	0	
0	0	0	1	0	1	0	0	(Write)
1								(Read) bits 7 to 0 : Repeat interval

<Explanation>

(1) This command sets the repeat interval to be used when a key is kept pressed.

(2) A value up to 0.5 seconds can be specified in the unit of 1/256 seconds (about 3.9 ms) as send data. 1 must be set as MSB of data.

(3) The initial value is about 70 ms.

6. KB Repeat Timer 1 Read — 7508 command

[Function]

This command reads the keyboard repeat start time.

[Sequence]

7	6	5	4	3	2	1	0	
0	0	1	0	0	1	0	0	(Write)
1								(Read) bits 6 to 0 : Repeat Start time

<Explanation>

(1) The current keyboard repeat start time is indicated.

(2) As the command explained in 4, a value is indicated in the units of 1/64 seconds (about 15 ms) as receive data. MSB of receive data is 1 but MSB of data indicating the repeat start time is 0.

7. KB Repeat Timer 2 Read — 7508 command

[Function]

This command reads the keyboard repeat interval.

[Sequence]

7	6	5	4	3	2	1	0	
0	0	1	1	0	1	0	0	(Write)
1								(Read) bits 6 to 0 : Repeat interval

<Explanation>

(1) The current keyboard repeat interval is indicated.

(2) As the command explained in 5, a value is indicated in the units of 1/256 seconds (3.9 ms) as receive data. MSB of receive data is 1 but MSB of data indicating the repeat interval is 0.

[Function]

This command disables the keyboard repeat function.

[Sequence]

7	6	5	4	3	2	1	0	
0	0	0	0	0	1	0	1	(Write)

<Explanation>

(1) This command disables the automatic keyboard repeat function. Only one key code is sent even if a key is kept pressed.

(2) OS specifies to disable the automatic keyboard repeat function as the default for 7508.

(3) OS supports BIOS CONOUT (ESC+F0H) to switch the automatic keyboard repeat function on/off.

[Function]
This command enables the keyboard repeat function.

[Sequence]

7	6	5	4	3	2	1	0	
0	0	0	1	0	1	0	1	(Write)

<Explanation>

(1) This command enables the automatic keyboard repeat function. When a key is kept pressed, a key code is repeated every specified repeat interval after the specified repeat start time elapses.

(2) The automatic keyboard repeat function is disabled as the default.

(3) OS supports BIOS CONOUT (ESC+F0H) to switch the automatic keyboard repeat function on/off.

[Function]

This command disables all interrupts caused by key input operations.

[Sequence]

7	6	5	4	3	2	1	0	
0	0	0	0	0	1	1	0	(Write)

<Explanation>

(1) Interrupts caused by key input operations and sent to Z-80 are disabled. When a key is pressed, only a code is input to the 7508 key buffer but an interrupt is not sent to Z-80. When the command "KB Interrupt ON" explained in Section 11 is executed after a key is pressed, interrupts caused by the pressed keys are sent to Z-80 unless the key buffer is empty.

(2) Key codes occurred after the key buffer becomes full are discarded.

(3) OS supports BIOS MASKI for this operation.

[Function]
This command enables all interrupts caused by key input operations.

[Sequence]

7	6	5	4	3	2	1	0	
0	0	0	1	0	1	1	0	(Write)

<Explanation>

(1) After this command is executed, interrupts are sent to the main CPU.

(2) Key interrupts are enabled as the default.

(3) OS supports BIOS MASKI for this operation.

[Function]

This command reads the current 7508 time.

[Sequence]

7	6	5	4	3	2	1	0		
0	0	0	0	0	1	1	1	(Write)	
0	0	0	0					(Read)	Tens digit of year
0	0	0	0					(Read)	Unit digit of year
								(Read)	Month
								(Read)	Day
								(Read)	Hour
								(Read)	Minutes
								(Read)	Second
0	0	0	0					(Read)	Day of week

Month, Day, Hour, Minutes, Second — Tens digit in higher 4 bits (bits 7 to 4) and unit digit in lower 4 bits (bits 3 to 0)

<Explanation>

(1) This command reads the 7508 calendar clock.

(2) The receive data indicates year, month, day, hour, minute, second, and day of week in this order. Each item is represented in BCD code.

(3) The hour is indicated by 24-hour clock. As the day of week, 1 indicates Sun, 2 indicates Mon,...6 indicates Sat. If a logically incorrect calendar clock is set, the contents of read data is not guaranteed because 7508 does not check the set data.

(4) OS supports BIOS TIMDAT for this operation.

[Function]

This command sets the 7508 calendar clock.

[Sequence]

7	6	5	4	3	2	1	0		
0	0	0	1	0	1	1	1	(Write)	
1	0	0	0					(Read)	Tens digit of year
1	0	0	0					(Read)	Unit digit of year
1								(Read)	Month
1								(Read)	Day
1								(Read)	Hour
1								(Read)	Minutes
1								(Read)	Second
1	0	0	0					(Read)	Day of week

Month to Second: Tens digit in higher 4 bits (bits 7 to 4) and unit digit in lower 4 bits (bits 3 to 0)

<Explanation>

(1) Send data specifies year, month, day, hour, minute, second, and day of week in this order. Each item is represented in BCD code. 1 must be set as MSB of the data.

(2) The hour is indicated by a 24-hour clock. The day of week is automatically updated between 0 to 6. If a logically incorrect data is set, the contents of the calendar clock are not guaranteed because 7508 does not check the set data.

(3) To set only one or more item, set 1 to all the bits of other items that are not to be set and send the entire data.

(4) OS supports BIOS TIMCAT for this operation.

14. Power Switch Read   7508 command

[Function]
This command reads the status of current power switch.

[Sequence]

7	6	5	4	3	2	1	0	
0	0	0	0	1	0	0	0	(Write)
0	0	0	0	0	0	0		(Read) Power Switch status Bit 0 = 0 : SW OFF = 1 : SW ON

<Explanation>

(1) This command reads the status of the power switch placed at the side of the EHT-10/EHT-10/2.

(2) OS supports BIOS READSW for this operation.

[Function]

This command reads the current alarm time.

[Sequence]

7	6	5	4	3	2	1	0		
0	0	0	0	1	0	0	1	(Write)	
								(Read)	Month
								(Read)	Day
								(Read)	Hour
								(Read)	Minutes
0	0	0	0					(Read)	Tens digit of second
0	0	0	0					(Read)	Day of week

Month, Day, Hour, Minutes — Tens digit in higher 4 bits (bits 7 to 4) and unit digit in lower 4 bits (bits 3 to 0)

<Explanation>

(1) This command reads the current alarm time in the order of month, day, hour, minute, second, and day of week. Each item is represented in BCD code.

(2) OS supports BIOS TIMDAT for this function.

[Function]

This command sets the alarm time.

[Sequence]

7	6	5	4	3	2	1	0		
0	0	0	1	1	0	0	1	(Write)	
1								(Read)	Month
1								(Read)	Day
1								(Read)	Hour
1								(Read)	Minutes
1	0	0	0					(Read)	Tens digit of second
1	0	0	0					(Read)	Day of week

Month, Day, Hour, Minutes: Tens digit in higher 4 bits (bits 7 to 4) and unit digit in lower 4 bits (bits 3 to 0)

<Explanation>

(1) This command sets the alarm time in the order of month, day, hour, minute, second, and day or week. Each item is represented in BCD code. 1 must be set as MSB of the data.

(2) The item is ignored when 1 is set to all bits of an item (for example, every minute is assumed when 1 is set to the entire bits indicating minute).

(3) The year and the unit digit of second cannot be set.

(4) If the command "Alarm ON" explained in Section 18 is executed after this command is executed, an alarm is generated at the specified time.

(5) OS supports BIOS TIMDAT for this operation.

[Function]
This command disables the alarm function.

[Sequence]

7	6	5	4	3	2	1	0	
0	0	1	0	1	0	0	1	(Write)

<Explanation>

(1) This command disables interrupts caused by the alarm function and sent to the main CPU. The alarms occurred during alarm off status are ignored (the alarm interrupts occurred in this status will not be generated even after the status is changed to alarm on).

(2) OS supports BIOS TIMDAT and MASKI for this operation.

[Function]
This command enables the alarm function.

[Sequence]

7	6	5	4	3	2	1	0	
0	0	1	1	1	0	0	1	(Write)

<Explanation>

(1) This command enables alarm interrupts sent to the main CPU. An alarm interrupt occurs when the set time comes after this command is executed.

(2) OS supports BIOS TIMDAT and MASKI for this operation.

[Function]

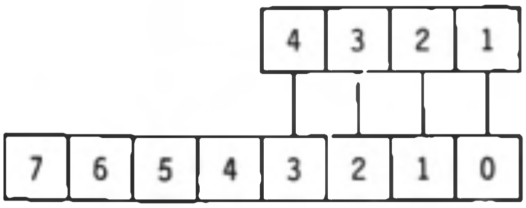
This command reads the status of DIP switches placed at the back of the main frame.

[Sequence]

7	6	5	4	3	2	1	0	
0	0	0	0	1	0	0	1	(Write)
								(Read) bit 3 to 0: Dip switch status

<Explanation>

(1) The DIP switches correspond to the read data bits as follows:



(2) Each receive data bit indicates on when it is 1 and off when it is 0. However, MSB (bit 7) for EHT-10 differs from the one for EHT-10/2 as follows:

MSB=1: EHT-10/2
=0: EHT-10

(3) OS uses the dip switch 1 (bit 0 of the received data) to change the setting of the EHT-10/EHT-10/2.

DIP switch 1 = 0: Overseas
= 1: Japan

The reading of the dip switch setting is supported by READSW of the BIOS.

[Function]

This command specifies 7 characters as the length of key code buffer.

[Sequence]

7	6	5	4	3	2	1	0	
0	0	0	0	1	1	0	0	(Write)

<Explanation>

(1) This command reserves a 7-character buffer at the 7508 side and stores the key code when a key is pressed and the key code is not fetched.

(2) The key and switch codes of keys and switches pressed after the buffer becomes full are ignored.

(3) A 7-character buffer is used as the default.

[Function]
This command specifies 1 character as the length of key code buffer.

[Sequence]

7	6	5	4	3	2	1	0	
0	0	0	1	1	1	0	0	(Write)

<Explanation>

(1) This command sets 1 character as the length of key buffer at the 7508 side.

(2) The key buffer function is same as the one explained in 20. " 7 Character Buffer".

(3) OS also has a key buffer and this buffer can store up to 32 characters.

[Function]
This command disables interrupts.

[Sequence]

7	6	5	4	3	2	1	0	
0	0	0	0	1	1	0	1	(Write)

<Explanation>

(1) This command disables interrupts sent every second from 7508. 1-second interrupts are no longer sent to the main CPU after this command is executed.

(2) OS supports BIOS MASKI for this operation.

(3) OS uses 1-second interrupts for the following operations:
1 Automatic power off time monitoring
2 Alarm screen display time monitoring

The above operations can no longer be executed when 1-second interrupts are disabled.

[Function]
This command enables 1-second interrupts sent from 7508.

[Sequence]

7	6	5	4	3	2	1	0	
0	0	0	1	1	1	0	1	(Write)

<Explanation>

(1) This command enables interrupts sent every second from 7508. 1-second interrupts are sent to the main CPU every second after this command is executed.

(2) OS supports BIOS MASKI for this operation.

(3) OS sends the "1 Sec. Interrupt ON" command during reset or system initialization.

[Function]

This command resets the keyboard section.

[Sequence]

7	6	5	4	3	2	1	0	
0	0	0	0	1	1	1	0	(Write)

<Explanation>

(1) This command clears the 7508 key buffer, scans the keyboard, and stores the information of the current key.

(2) Differing from the "KB Reset", the "KB Clear" command does not initialize the repeat start time, etc.

[Function]
This command resets 7508.

[Sequence]

7	6	5	4	3	2	1	0	
0	0	0	0	1	1	1	1	(Write)

<Explanation>

(1) This command initializes entire 7508.

(2) An application program cannot use this command. System initialization is executed for EHT-10/EHT-10/2 when the "System Reset" command is sent.

7.7 Other I/O

7.7.1 Overview

This section explains controlling the buzzer, timer and backlight.

7.7.2 Buzzer

EHT-10/EHT-10/2 has piezo-buzzer that is controlled by software. OS supports BIOS BEEP for controlling this buzzer (see "CHAPTER 4 BIOS OVERVIEW").
Figure 7.10 shows the overview of the circuit.
Software can specify 512 Hz, 1024 Hz, or 2048 Hz for the frequency.
Further, an optional frequency can be output by outputting the frequency generated by the software timer to the SP terminal.

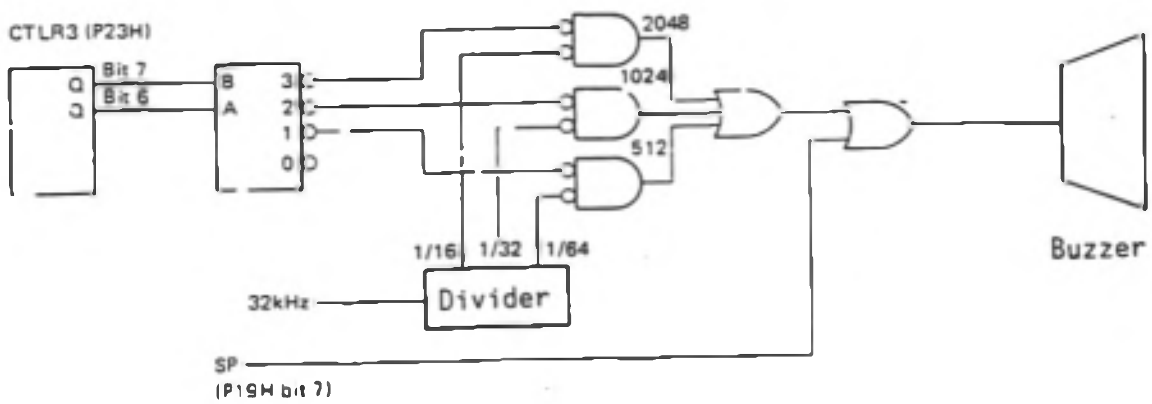


Fig 7.10 Buzzer Interface

(Note 1) 0 is generally set to the SP terminal (P19H, bit 7) to enable outputs from CTLR3.

7.7.3 Timer

Figure 7.11 shows the EHT-10/EHT-10/2 timer circuit.
FRC is a 16-bit free running counter and the input clock is 614.4 KHz (1.6276 sec). Therefore, FRC overflows every 0.106667 sec determined as follows:

$$1.6276 \times 2^{16} = 0.106667 \text{ sec}$$

The value in FRC can be found out by reading ICRL.C (P00H) and ICRH.C (P01H). (Note 1) ICRL.C and ICRH.C must be read in this order because the FRC value is latched in the input capture register(ICR) when ICRL.C is read. When FRC overflows, an interrupt (OVF interrupt) is sent to the main CPU and the status is set in ISR (P04H) bit 3 (INT3). The OVF status is reset when 1 is set to CMDR (P01H) bit 2 (Res OVF) and write operation is executed. To use the timer in a user program, disable OVF interrupts and structure the timer by using ICRL.C, ICRH.C, and ISR bit 3 .

(Note 1) The FRC value is also latched in ICR when the signal sent from the barcode reader changes. In this case, the FRC value is read from ICRL.B (P02H) and ICRH.B(P03H).

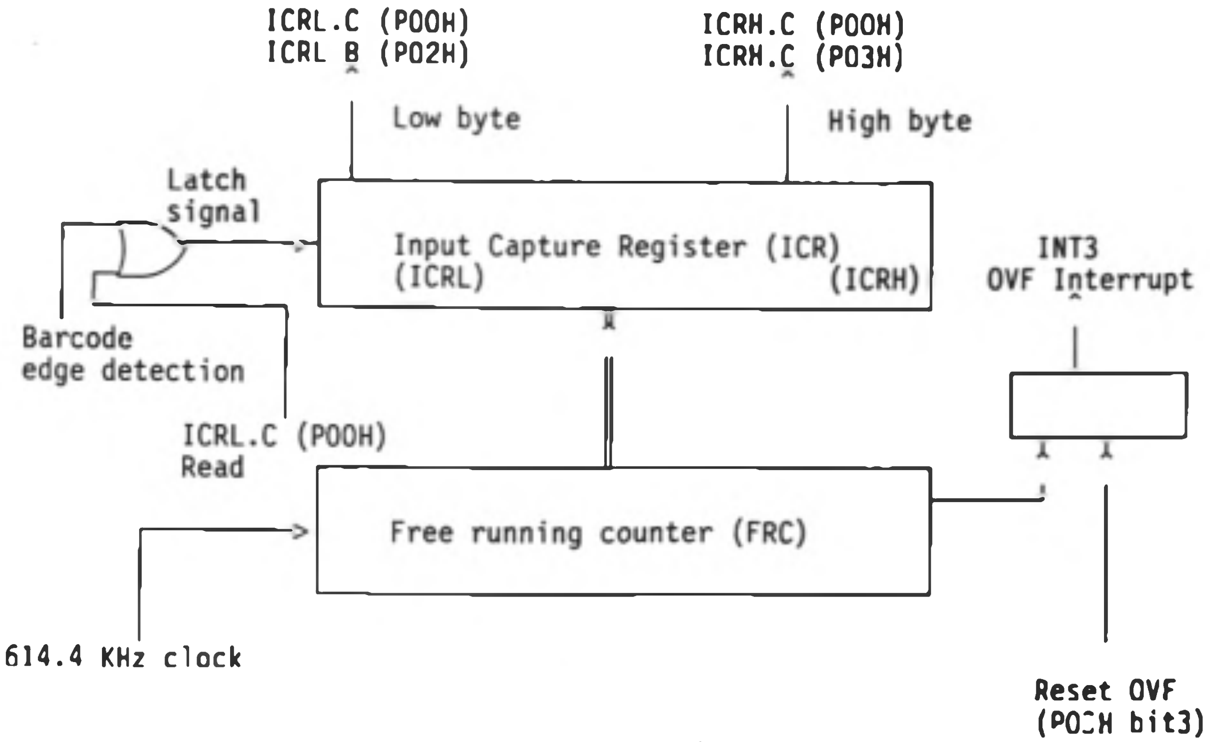


Fig 7.11 Timer Clock

7.7.4 EL backlight (EHT-10/2B)

EHT-10/2B model that has the EL backlight is provided so that it can be used in a dark room.
The EL backlight can be turned on or off by the backlight switch or controlled by software.
OS supports BIOS BACK LIGHT and automatic backlight off function at key input for this operation (see "CHAPTER 4 BIOS OVERVIEW" and "Section 2.3.5 Sleep function and automatic power off function").
Controlling the EL backlight by software is effective only when the backlight switch is on.

CHAPTER 8 INTERRUPT PROCESSING

8.1 Overview

EHT-10/EHT-10/2 supports five types of interrupts. When an interrupt occurs, the vector corresponding to the interrupt vector is set and control is passed to the interrupt processing routine.
EHT-10/EHT-10/2 supports the following five types of interrupts:

1 7508 (4-bit CPU)
2 ART (RXRDY)
3 ICF (Input capture)
4 OVF (FRC overflow)
5 EXT (timer, cartridge, and IC card)

7508 (4-bit CPU) interrupts are caused by the following events:

1 Keyboard
2 Power switch on/off
3 Alarm time
4 Power failure
5 1-second interrupt

EXT interrupts are caused by the following events:

1 1-msec/8-msec timer interrupt
2 Interrupt sent from cartridge I/F
3 Interrupt sent from IC card (back panel or overcurrent)

Table 8.1 shows the causes and reset conditions of interrupts.

Interrupt	Cause	Reset conditions
7508	- Keyboard input - 1-second interval - Alarm time - Power switch on or off - Power voltage failure	- A response is sent to 7508
ART (RXRDY)	- Serial data receive flag RXRDY in the ART section (gate array) is set.	- Receive data register ARTDIR (P14H) is read.
ICF	- The input signal sent from the barcode reader is changed.	- The input capture register (P03H) is read.
OVF	- Free running counter (FRC) overflows. The cycle is about 106.7 msec because FRC is 16 bits and the input clock is 614.4 kHz.	- "Reset OVF" command (that writes 1 to bit 2 of command register CMDR (P02H))
EXT	- 1-msec or 8-msec interval.	- 1 is written in P23H bit 1 to reset the interrupt.
	- An interrupt signal is received from the cartridge I/F.	- A response is sent to the cartridge device (in the case of a printer unit, the printer unit is made busy by data output because the interrupt is a ready interrupt).
	- The back panel of IC card is opened. - Overcurrent is flowed in IC card.	- This cannot be reset by software.

Table 8.1 Causes and Reset Conditions of Interrupts

EHT-10/EHT-10/2 supports five types of interrupts. The interrupt vector table is stored in FFF0H to FFFFH.
Interrupts requested are accepted in the priority order.

Priority order	Interrupt cause	Vector	Address in RAM
1 (Highest priority)	7508 (4bit CPU)	F0H	FFF0H,FFF1H
2	ART (RXRDY)	F2H	FFF2H,FFF3H
3	ICF (Input Capture)	F4H	FFF4H,FFF5H
4	OVF (Overflow of FRC)	F6H	FFF6H,FFF7H
5	EXT (timer, cartridge, and IC card)	F8H	FFF8H,FFF9H

Table 8.2 Interrupt Vector Table

EHT-10/EHT-10/2 controls interrupts as follows:

(1) Setting interrupt mode and interrupt vector

The following operations are executed when processing is started from address 0000H (system initialization, reset, or power on start):
1 mode-2 interrupt is specified as the interrupt mode.
2 FFH (indicating to use FFF0H to FFFFH as the interrupt vector table) is set in register I.

(2) Loading the interrupt vector table

The interrupt vector data stored in OS ROM is loaded to FFF0H to FFFFH at reset or system initialization.

8.3.1 Controlling interrupts by the system

(1) Setting by the system

The EHT-10/EHT-10/2 OS sets the interrupt initial status at the timing listed in the table below.

	7508	ART	ICF	OVF	EXT	7508			EXT		
						Key	1Sec	Alarm	1/8mSec	Cartridge	IC card
System initiali-zation	0	X	X	0	0	0	0	X	X	X	X
Reset	0	X	X	0	0	0	0	-	X	X	X
Warm boot	-	X	X	-	-	-	-	-	-	-	-
Restart power on	0	X	X	0	0	-	-	-	-	-	-
Continue power on	-	-	-	-	-	-	-	-	-	-	-

0: Enabled X: Disabled -: No change

Table 8.3 Setting Interrupt Mask Status

(2) Modules in which interrupts are disabled

While data is read from or written to FDD or an IC card as a disk, DI status is set so that any interrupts are disabled. OVF interrupts are disabled during BEEP operation.

(3) Interrupts occurred during interrupt processing

Interrupt processing is generally executed in DI status so that multiple interrupt processing is not executed. However, interrupts other than 7508 interrupts are enabled during 7508-interrupt processing.
Also, any interrupts are enabled during 7508-alarm interrupt processing.

8.3.2 Disabling and enabling interrupts

(1) Overview

Interrupts are disabled or enabled in the following two ways:
1 By BIOS MASKI
2 Interrupt enable register IER (P04H) is directly rewritten.

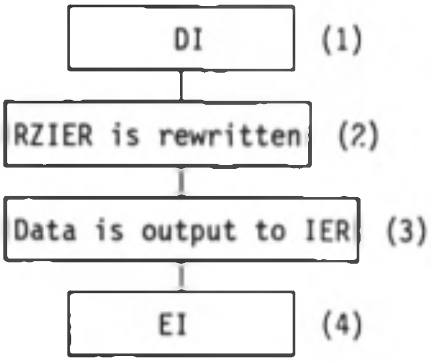
See "CHAPTER 4 BIOS OVERVIEW" for using BIOS MASKI.

Further, the following interrupts can be enabled or disabled individually by using a command or rewriting the I/O register:

- 7508 interrupts
- Key interrupt
- Alarm interrupt
- 1-second interrupt
- EXT interrupts
- 1-msec/8-msec interrupt
- Cartridge interrupt
- IC card interrupt

(2) Directly rewriting IER

Figure 8.1 shows how to directly rewrite IER.



RZIER (F42EH) is the interrupt enable or disable status storage area.

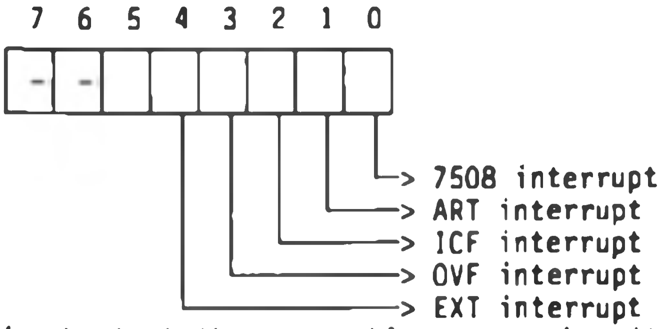
(Example)
External interrupts are enabled.

```
DI
LD   A,(RZIER)
OR   10H
LD   (RZIER),A
OUT  (IER),A
EI
```

Fig. 8.1 Rewriting IER

Address : Variable name (Byte length)

F42EH : RZIER (1)
- Interrupt enable/disable status storage area



(Each bit indicates enable status when it is 1 and disable status when it is 0.)
- The RZIER bit structure is same as IER (P04H) bit structure.

(3) Controlling 7508 interrupts

See "Section 7.6 7508 Commands" for controlling 7508 interrupts.

(4) Controlling EXT interrupts

Generally, 1 (interrupt enabled) is set to the IER EXT interrupt bit (bit 4) for EXT interrupts so that each interrupt can be disabled or enabled.
Table 8.4 shows the I/O registers used to control each EXT interrupt. The contents of each I/O register are output by rewriting the data stored in memory and outputting the rewritten data to the I/O port in the same way as rewriting IER explained in (2).

```
Port Address (R/W) : Port name (RAM data address)

P17H (W) : ICCTLR (F0DBH)
     bit 6 :(PRIE) cartridge interrupt
          =0:  Enabled
          =1:  Disabled
     bit 5 :(ICITE) IC card interrupt
          =0:  Disabled
          =1:  Enabled

P23H (W) : CTLR3 (F0DEH)
     bit 3 :(PINTDIS) cartridge and IC card interrupt
          =0:  Enabled
          =1:  Disabled
     bit 4 :(OVFITV) 1-msec/8-msec interrupt switch
          =0:  8 msec
          =1:  1 msec
     bit 5 :(OVFEN) 1-msec/8-msec interrupt
          =0:  Disabled
          =1:  Enabled
```

Table 8.4 EXT Interrupt Mask I/O Registers

8.3.3 Interrupt processing time

Table 8.5 shows the interrupt processing time required for EHT-10/EHT-10/2.

Interrupt		Number of states	Time (micro Sec)	Notes
7508	Key input	4720	1282	For EHT-10 (touch panel)
		2384	647	For EHT-10/2 (keyboard)
	1 Sec	1913	519	
	Alarm	2089	568	Alarm display disabled
	Power switch on	1642	446	Power off processing disabled
	Power switch off	1701	462	Power off processing disabled
	Power failure	1713	465	Power failure processing disabled
ART		1716	466	No buffer control specified
ICF		-	-	Barcode reading is performed in ICF interrupt processing.
OVF		1263	343	No cursor blinking
EXT	Timer	1572	427	
	Cartridge	1603	436	For printer unit
	IC card	1757	477	

Table 8.5 Interrupt Processing Time

(Note 1) The above table lists the standard processing time for each interrupt and the time required for actual operation may slightly differ.

(Note 2) The time required for an 7508 interrupt is determined by using the communication time to 7508 CPU as 0 micro Sec. For actual operations, about 3000 to 6000 states (800 to 1600 micro Sec) must be added for communication.

8.3.4 Processing executed when an interrupt occurs

(1) Interrupt processing

EHT-10/EHT-10/2 has four types of banks and processing may be in progress in any bank when an interrupt occurs. Because of this, the entry section of interrupt processing is in the resident section (E000H to FFFFH), the bank is switched to the appropriate one in this entry section, and actual interrupt processing is executed in the OS ROM that stores the main section of interrupt processing.

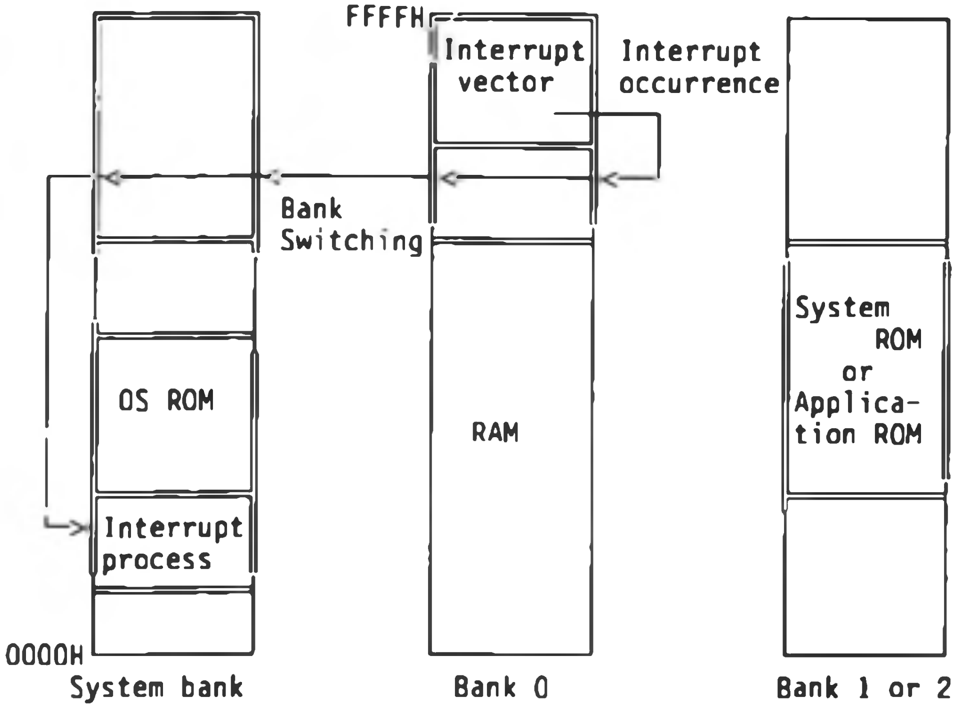


Fig 8.2 Processing Flow at Interrupt Occurrence

(2) Relationship between interrupt processing and BIOS
If an interrupt occurs during BIOS and interrupt processing is executed immediately, the current program may not be continued or power off processing in continue mode cannot be done after control returns. Therefore, BIOS turns on the BIOS execution flag (PREBIOS) at the beginning of processing and the interrupt flag is set to indicate interrupt occurrence if such an interrupt occurs.
At the end of BIOS processing, the interrupt flag is checked and actual interrupt processing is executed if the flag indicates interrupt occurrence (PSTBIOS).

In BIOS processing that forms a loop (CONIN, RSIOX, TCAM, or ICCARD), the interrupt flag is checked as explained above and interrupt processing is executed if the flag indicates interrupt occurrence.

See "CHAPTER 4 BIOS OVERVIEW" for PREBIOS and PSTBIOS.

8.4.1 Overview

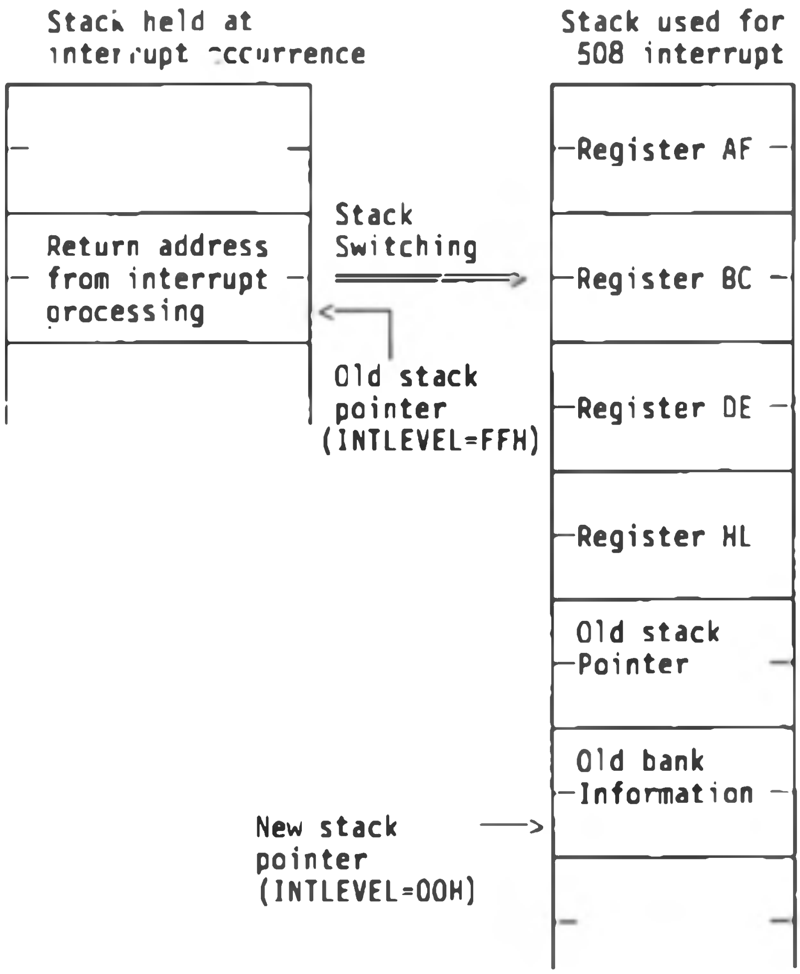
When an interrupt is sent from 7508, interrupt processing performs serial communication with 7508 to read the 7508 status. One of the following processing is executed according to this status:

- Key interrupt
- 1-second interrupt
- Power failure interrupt
- Alarm interrupt
- Power switch interrupt

7508 interrupt processing can perform multiple 7508 interrupt processing so that key input operations are enabled during alarm screen display.

8.4.2 Multiple interrupt processing

7508 interrupt processing can perform multiple interrupt processing by taking into account of another interrupt occurrence during interrupt processing.
Multiple interrupt processing is performed by using interrupt level INTLEVEL (F092H) and stack. Figure 8.3 shows the structure of multiple interrupt processing.



INTLEVEL (F092H): Indicates the number of 7508 interrupts

The left figure shows the status in which the first 7508 interrupt occurs. When the second 7508 interrupt occurs in this status, the memory contents starting from register AF to the old bank information are saved without switching the stack and the contents of INTLEVEL is increased by 1.

Fig. 8.3 Multiple 7508 Interrupt Processing

8.4.3 7508 interrupts

(1) Overview

There are five types of 7508 interrupts and the type can be identified by reading 7508 status.
When an interrupt is sent from 7508, a Status Read command (02H) is sent to 7508 to read the 7508 status. The status indicates a hardware code for key interrupt when the read status is less than BFH. A power switch, 1-second, alarm, or power failure interrupt is indicated when the status is C0H or more.

This status is stored in STS7508 (F3C3H). Power switch interrupts are further classified into power switch on and power switch off interrupts.

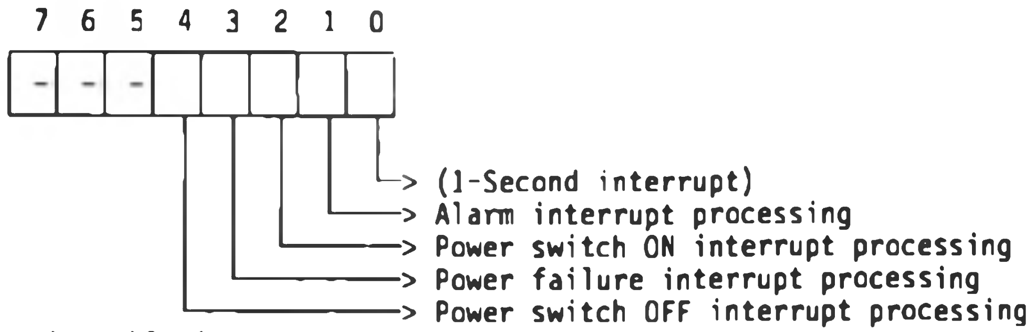
Interrupts processing is executed after the status indicating power switch on/off, i-second, alarm, or power failure interrupt is set in INTFG (F3C2H). The operations to be executed in interrupt processing is determined from the status according to the TBL7508 (F0A3H).

The execution status of 1-second interrupt processing is set in FG7508 (F3C4H).

Address : Variable name (Byte length)

F0A3H : TBL7508 (16)

- This table is used to determine 7508 interrupt processing.
Each byte is structured with the following bits:
(Processing is executed when a bit is 1.)



- The table is structured as follows:

Address	7508 status	Power switch status	Initial value
F0A3H	C0H OR E0H	OFF	00H
F0A4H		ON	10H
F0A5H	C1H OR E1H	OFF	04H
F0A6H		ON	00H
F0A7H	C2H OR E2H	OFF	02H
F0A8H		ON	12H
F0A9H	C3H OR E3H	OFF	04H
F0AAH		ON	02H
F0ABH	C4H OR E4H	OFF	08H
F0ACH		ON	10H
F0ADH	C5H OR E5H	OFF	08H
F0AEH		ON	08H
F0AFH	C6H OR E6H	OFF	08H
F0B0H		ON	10H

F0B1H	C7H OR E7H	OFF	08H
F0B2H		ON	08H

- Whether 1-second interrupt processing should be executed is determined by software after conversion.

F3C2H : INTFG (1)
- This indicates the type of 7508 interrupt processing executed by OS.
- 00H is set in INTFG when key interrupt processing is executed.

7	6	5	4	3	2	1	0
-	-	-					

- Bit 0 —> 1-Second interrupt
- Bit 1 —> Alarm interrupt
- Bit 2 —> Power switch ON interrupt
- Bit 3 —> Power failure interrupt
- Bit 4 —> Power switch OFF interrupt

(Interrupt occurrence is indicated when a bit is 1.)

F3C3H : STS7508 (1)
- This indicates the status read from 7508 by OS when a 7508 interrupt occurs.
- When 00H ≤ STS7508 ≤ BFH, this indicates hardware key code for a key interrupt.

7	6	5	4	3	2	1	0
1	1		-	-			

- Bit 0 —> Power switch status(0:OFF, 1:ON)
- Bit 1 —> Alarm interrupt
- Bit 2 —> Power failure interrupt
- Bit 5 —> 1-Second interrupt

(Interrupt occurrence is indicated when a bit is 1.)

F3C4H : FG7508 (1)
- 7508 interrupt processing flag

7	6	5	4	3	2	1	0
-		-	-		-	-	-

- Bit 3 —> First alarm interrupt
- Bit 6 —> 1-second interrupt

(Interrupt occurrence is indicated when a bit is 1.)

(2) Key interrupt processing

When an interrupt is sent from 7508, the 7508 status is read through serial communication from 7508. If this status is less than BFH, the interrupt is handled as a key interrupt.

For EHT-10, the 7508 status indicates 80H when an interrupt is sent from the touch panel.
When 7508 status indicates 80H during interrupt processing, the touch panel is scanned to find out the pressed touch-panel key.
See "HARDWARE Section 4.5 Touch Panel Interface" for scanning the touch panel.

For EHT-10/2, the read 7508 status indicates a position code (hardware code) in the keyboard matrix. See "Section 7.6.2 7508 interface" for the relationship between keys and position codes.
During key interrupt processing, the key hardware code is stored in the key buffer (KBUF). The key code is discarded if the key buffer is full.

(3) 1-second interrupt processing

1-second interrupt processing performs the following operations:
(a) 16-bit timer TIMER0 (F02DH) is increased by 1.
(b) 16-bit timer TIMER1 (F02FH) is decreased by 1.
(c) Timer function TMFUNC (F205H) processing
The following operations are performed when TMFUNC (F025H) is not equal to 00H:
TMSEC (F3C6H) is decreased by 1.
If TMSEC becomes 00H as a result of decrease operation, 00H is set in TMFUNC and FFH is set in TMFLAG (F206H).
(d) The power down mode time for printer unit is counted down.
(e) The power off time of IC card is counted down.
(f) The alarm repeat count is counted down.

(a), (b), and (c) are the 1-second counters for application programs.
(d), (e), and (f) are the counters used for system control.
TIMER0 is also used by the system to monitor automatic-power-off time.

1 Using TIMER0 and TIMER1
TIMER0 and TIMER1 are the 16-bit 1-second counters and are increased or decreased by 1 every time a 1-second interrupt occurs. These counters can be used as reference counters without changing the contents to measure processing time.

2 Using TMFUNC
TMFUNC is a user function used to monitor a fixed time. Figure 8.4 shows how to use TMFUNC.

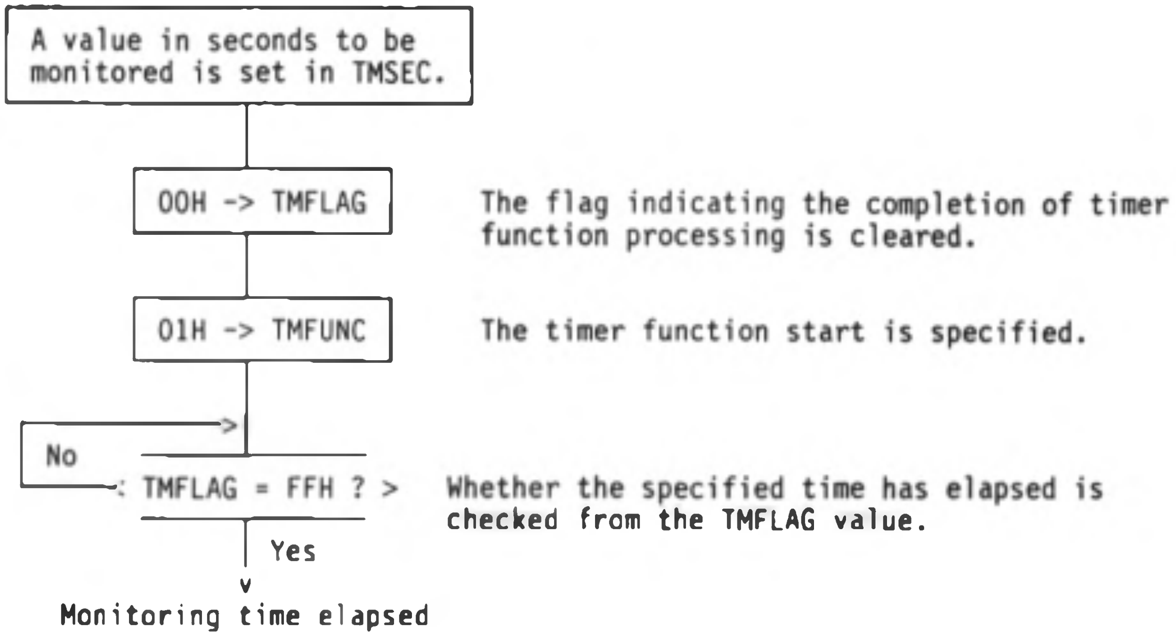


Fig 8.4 Timer Function

Address : Variable name (Byte length)

F020H : TIMERC (2)
- 16-bit timer
- This is increased by 1 when a 1-second interrupt occurs.

F022H : TIMER1 (2)
- 16-bit timer
- This is decreased by 1 when a 1-second interrupt occurs.

F205H : TMFUNC (1)
- Timer function
  - =00H: Function not specified
  - ≠00H: Function specified

F206H : TMFLAG (1)
- Timer flag
- TMSEC is decreased by 1 for every 1-second interrupt occurrence when TMFUNC is not equal to 00H. FFH is set in TMFLAG when TMSEC becomes equal to 0000H.

F3C6H : TMSEC (2)
- Timer counter
- TMSEC is decreased by 1 for every 1-second interrupt occurrence when TMFUNC is equal to 00H. When TMSEC becomes 0000H, FFH is set in TMFLAG and 00H is set in TMFUNC.

(4) Alarm interrupt processing

An alarm interrupt occurs when the time specified by "BIOS TIMDAT Set Alarm/Wake" comes.
Alarm and wake differ only for software and there is no wake interrupt.
An alarm time can be specified by month, day, day of week, hour, minute, and second. The minimum unit is the tens digit of seconds.
This is because 7508 uses the time down to tens unit of seconds for

the alarm time determination. When the specified alarm time comes, an alarm interrupt is generated up to 10 times.
OS assumes only one alarm interrupt among the generated alarm interrupts as the actual alarm interrupt. A hook is set in alarm interrupt processing so that processing can be extended by application programs. See "CHAPTER 10 SYSTEM HOOK AND JUMP TABLE" for details on hook.
Using YALRMDS (F0B6H), alarm screen display can be suppressed by software during alarm interrupt processing.

(5) Power switch on interrupt processing

This interrupt is generated when the power switch is turned on. Power switch on interrupt processing sets power switch flag PWSWONFG (F3C8H).

Address : Variable name (Byte length)

F3C8H : PWSWONFG (1)
- Power switch on flag
  - =00H: Power switch on interrupt not occurred
  - =FFH: Power switch on interrupt occurred
- 00H is set at power on start when the main power is off.

(6) Power failure interrupt processing

A power failure interrupt occurs when the voltage of the main battery is lowered. Power failure occurs when the voltage of NiCd battery is lowered to about 4.7 V or less.
When a power failure interrupt occurs, power failure interrupt processing sets power failure interrupt flag BTRYFG (F0B3H) and displays the power failure screen to warn the user.
A power failure interrupt is sent every second after occurrence. 7508 forcibly turns off the main power supply if the main power supply is not turned off within 50 seconds after interrupt occurrence.

Address : Variable name (Byte length)

F0B3H : BTRYFG (1)
- Power failure interrupt flag
  - =00H: Power failure interrupt not occurred
  - =FFH: Power failure interrupt occurred
- 00H is set at power on start.

(7) Power switch off interrupt processing

A power switch off interrupt occurs when the power switch is turned off. Power switch off interrupt processing sets the power switch status to power switch off flag PWSWOFFG (F3C9H) and turns the power off.

Address : Variable name (Byte length)

F3C9H : PWSWOFFG
- Power switch off flag
  - =00H: Power switch off interrupt not occurred
  - =01H: Power switch off interrupt occurred

8.4.4 Suppressing alarm or power off (failure) interrupt processing

(1) Overview

For an alarm, power off, or power failure interrupt, interrupt processing can suppress the interrupt and only indicates the interrupt occurrence so that the screen is not displayed and actual power off processing is not executed.
Suppressing interrupts is used when processing should not be interrupted or terminated during serial communication or I/O processing for cartridge I/F until group of consecutive operations are completed.
OS performs this suppress processing by using PREBIOS and PSTBIOS.
During BIOS processing, the alarm screen is not displayed or the power is not turned off (interrupt processing is executed if an interrupt occurs when processing is waiting for key input operation or RSIOX receive or send operation).

(2) Method of suppressing interrupts

The system provides YPOFDS (F0B4H) and YALMDS (F0B6H) as the power off processing and alarm/wake processing suppress specification flags.
Each flag is structured as follows:

YPOFDS (F0B4H) (Power off processing suppress flag)

Bit	
0	Reserved
1	Reserved
2	Suppress bit for calculator/system menu
3	Reserved suppress bit
4	Suppress bit for DLL/MENU
5	Suppress bit for alarm screen display
6	Suppress bit for user
7	Suppress bit for BIOS

YALMDS (F0B6H) (Alarm screen display processing suppress flag)

Bit	
0	Forced suppress bit (See the note)
1	Reserved
2	Suppress bit for calculator/system menu
3	Reserved
4	Suppress bit for DLL/MENU
5	Suppress bit for alarm screen display
6	Suppress bit for user
7	Suppress bit for BIOS

When a bit is 1, power off is not executed or the alarm screen is not displayed.

(Note 1) Generally, alarm processing is executed at key input operation regardless of YALMDS bits. However, if YALMDS bit 0 is 1, alarm processing is not executed even in key input operation.

Interrupt processing copies the YPOFDS and YALMDS values to YPOFST (F0B5H) and YALMST (F0B7H) and executes power off or alarm screen display processing if these values are 00H. If these values are not 00H, no operation is executed and processing is terminated.

(4) Procedure to suppress interrupts

Figures 8.5 and 8.6 show how to suppress interrupts. When a general application program suppresses an interrupt, use the bit (bit 6) for application programs and other bits should not be manipulated.

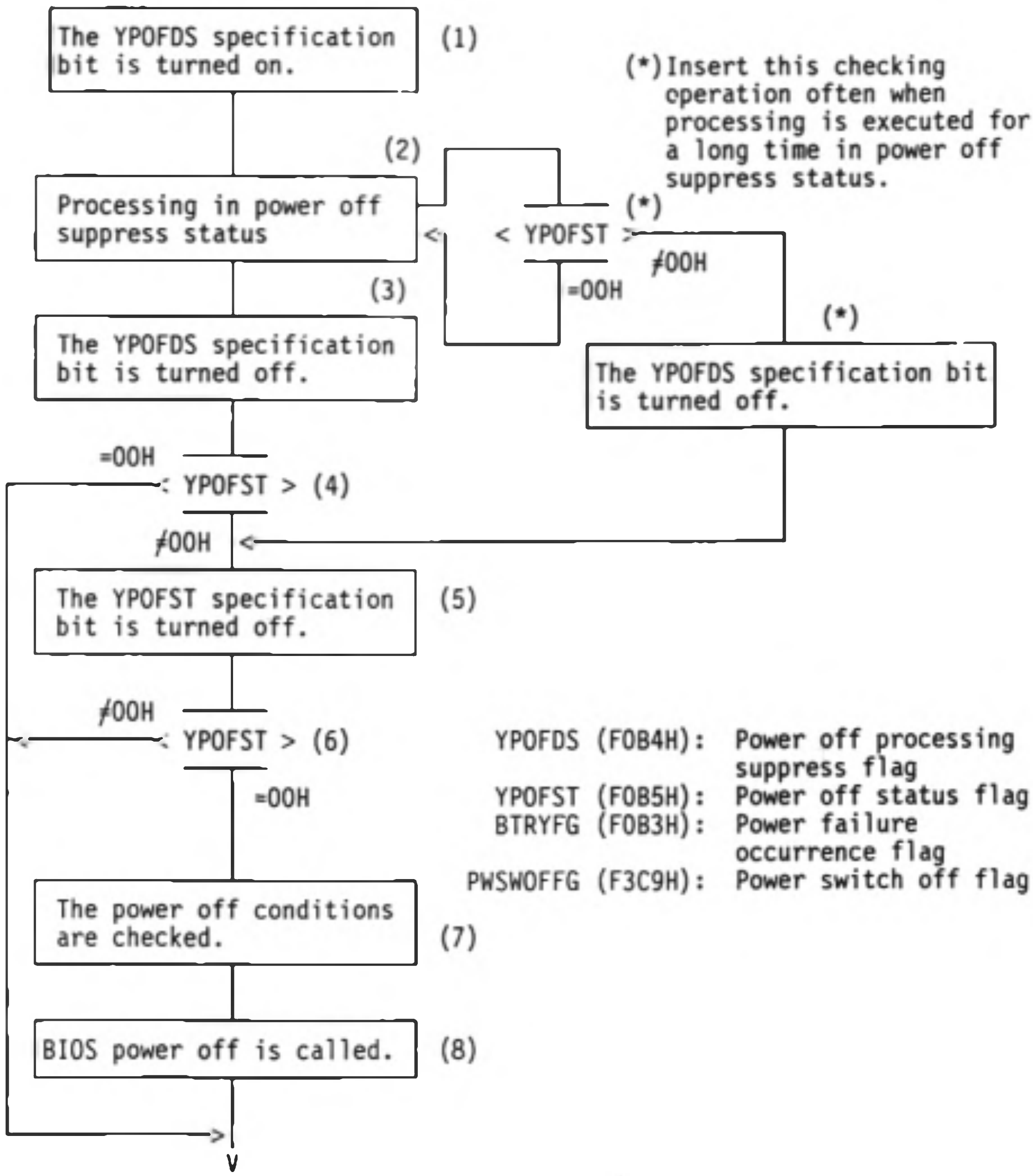


Fig 8.5 Suppressing Power off Interrupts

Step : Explanation

(1) The YPOFDS specification bit is turned on.
- The YPOFDS (F0B4H) specification bit is turned to 1.
- Bit 0 is used by an application program.

(2) Processing in power off suppress status
- Processing is executed in power off suppress status.
- Power off occurrence must be checked often if processing is executed for a long time in power off suppress status.

(3) The YPOFDS specification bit is turned off.
- The bit turned to 1 in step (1) is turned to 0.

(4) Checking YPOFST
- The YPOFST (F0B5H) value is checked.
- If this value is not equal to 00H, it means that a power off interrupt occurred while power off interrupts are suppressed.

(5) The YPOFST specification bit is turned off.
- The YPOFST (F0B5H) bit turned on in step 1 is turned to 0.

(6) Checking YPOFST
- The YPOFST value held after step (5) execution is checked.
- If this value is not equal to 00H, it means that another module is suppressing the power off interrupts.

(7) Checking power off conditions
- The power off status is checked.
- If BTRYFG (F0B3H) is not equal to 00H, the power is turned off in continue mode. If BTRYFG is equal to 00H, PWSWOFFG (F3C8H) is checked. The power is turned off in restart mode if PWSWOFFG is not equal to 02H. In any other cases, the power is turned off in continue mode.

(8) BIOS POWER OFF is called.
- BIOS POWER OFF is called according to the power off status obtained in step (7).

(note In step (7) and (8), you may call JPSTBIOS by turning bit 7 on.)

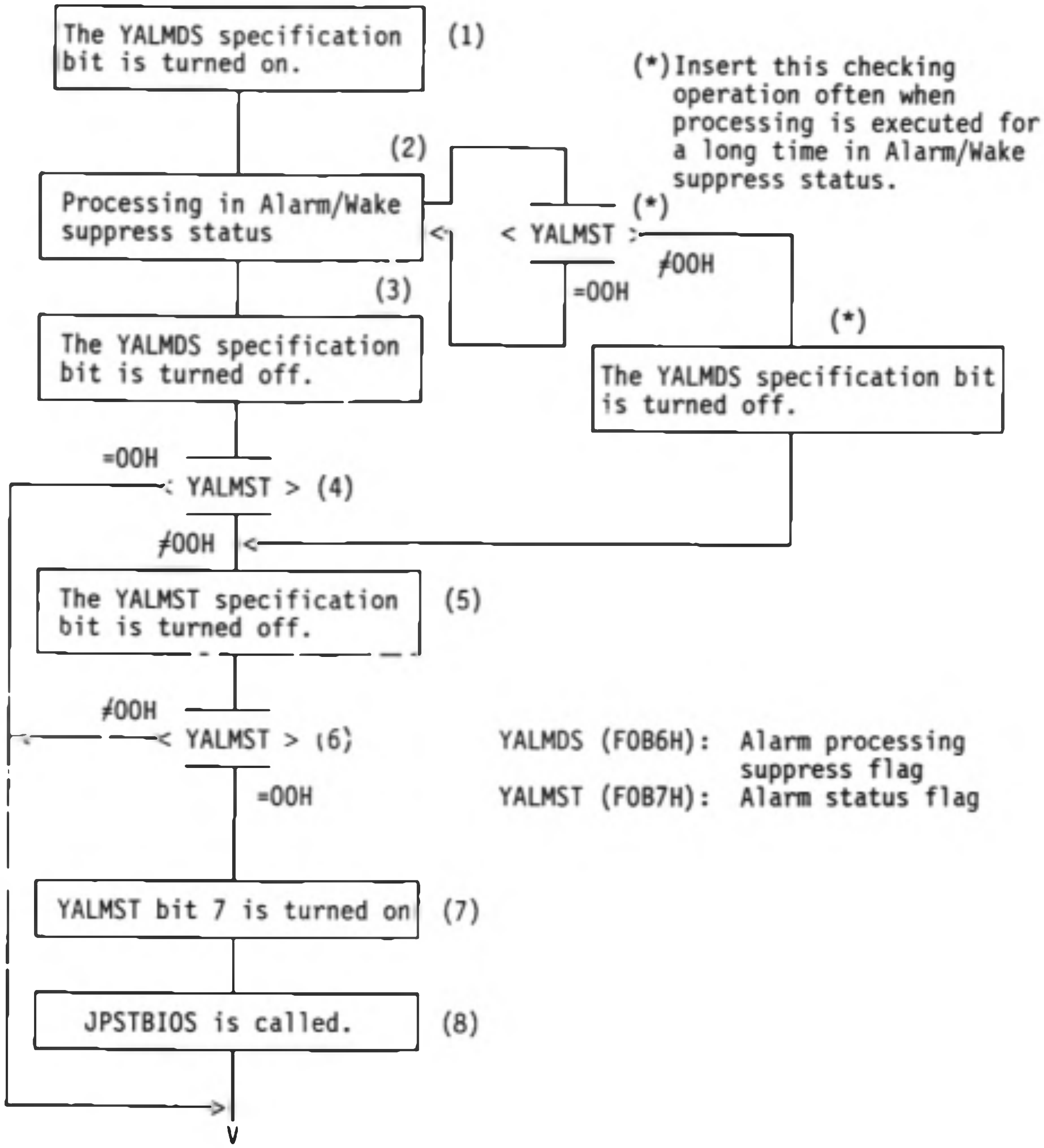


Fig 8.6 Suppressing to Display Alarm Screen

Step : Explanation

(1) The YALMDS specification bit is turned on.
- The YALMDS (F0B6H) specification bit is turned to 1.
- An application program uses bit 0.

(2) Processing in alarm/wake suppress status
- Processing is executed in alarm/wake suppress status.
- Alarm/wake interrupt occurrence must be checked often if processing is executed for a long time in alarm/wake suppress status.

(3) The YALMDS specification bit is turned off.
- The bit turned to 1 in step (1) is turned to 0.

(4) Checking YALMST
- The YALMST (F0B7H) value is checked.
- If this value is not equal to 00H, it means that an alarm interrupt occurred while alarm interrupts are suppressed.

(5) The YALMST specification bit is turned off.
- The YALMST (F0B7H) bit turned on in step 1 is turned to 0.

(6) Checking YALMST
- The YALMST value held after step 5 execution is checked.
- If this value is not equal to 00H, it means that another module is suppressing the alarm interrupts.

(7) YALMST bit 7 is turned on.
- YALMST (F0B7H) bit 7 is turned on.
- After bit 7 is turned on, the alarm/wake screen is displayed by PSTBIOS executed in step (8).

(8) RSPSTBIOS is called.
- JPSTBIOS (FF96H) is called to display the alarm/wake screen.

(4) Notes

When processing is made to wait for key input operation by BIOS CONIN, power switch off, power failure, or alarm/wake interrupt processing is executed unconditionally even if interrupts are suppressed.
Interrupt processing is also executed unconditionally when an interrupt occurs during BIOS RSIOX waiting for send or receive operation or BIOS waiting for printer to become ready (other than screen dump).

Whether a power off or alarm/wake interrupt occurred is checked often if processing is executed for a long time in power off or alarm/wake suppress status.

8.5 ART Interrupts

An ART interrupt occurs when serial data receive flag RXRDY in the ART section is set.

The operations listed below are executed when an ART interrupt occurs. See the sections explaining BIOS RSIOX and ICCARD "Section 11.2 Extending Communication Protocol" and "APPENDIX9 SAMPLE 29" for ART interrupt processing.

(1) No operation is executed if RSIOX Open has not been executed.

(2) The error status is checked (framing error, receive overrun error, parity error, or receive buffer overflow).

(3) Receive data is read and stored in the receive buffer.

(4) The byte length of receive data stored in the receive buffer is checked if buffer control (XON/XOFF, DTR/DSR, or RTS/CTS) is specified. An XOFF code is sent if data is stored in 3/4 or more of the receive buffer.

8.6 ICF Interrupts

An ICF interrupt occurs when the input signal sent from the barcode reader is changed. Generally, ICF interrupts are masked by IER [P04H] and this masking is cleared when BIOS BARCODE performs open operation.

See "Section 11.5 Adding Barcode Decoder", "Section 7.5 Barcode Reader Interface", and "Chapter 4 BIOS OVERVIEW" for details on ICF interrupts and barcode reader.

8.7 OVF Interrupts

An OVF interrupt occurs when the free running counter (FRC) overflows. An OVF interrupt is generated every 106.7 msec because FRC is a 16-bit counter and input clock 614.4 KHz is used.

The system uses an OVF interrupt for cursor blinking. The cursor blinks every 500 msec. This blink interval can be changed by rewriting BLNKTIME (F078H).

Address : Variable name (Byte length)

F078H : BLNKTIME (1)

- Cursor blink interval
- The unit is 100 msec and the initial value is 04H.

8.8.1 OVerview

An EXT interrupt occurs when one of the three events listed below occurs. The interrupt cause can be determined by reading the I/O register status (see Table 8.6).

1 1-msec/8-msec timer interrupt
2 Interrupt sent from the cartridge I/F
3 The back panel of IC card is opened or overcurrent flows.

I/O address (R/W) : Register name

P16H (R) : IOSTR
bit0 :(PBUSY) status of interrupt sent from cartridge I/F
=0: No interrupt
=1: Interrupt occurred
bit2 :(ICCLS) Interrupt caused by opened IC card back panel or by overcurrent
=0: Interrupt occurred
=1: No interrupt

P23H (P) : ITSR
bit0 :(OVFINT) 1-msec/8-msec timer interrupt status
=0: No interrupt
=1: Interrupt occurred
bit1 :(IPBUSY) Status of interrupt sent from IC card or cartridge I/F
=0: Interrupt occurred
=1: No interrupt

Table 8.6 EXT Interrupt Status

8.8.2 1-msec/8-msec timer interrupt

OS uses a 8-msec timer interrupt for key input sound, BIOS BEEP, or barcode reader LED blinking.
The user can use 1-msec/8-msec timer interrupts as explained below.

(1) Enabling 1-msec/8-msec interrupts

BIOS CALLX is used to call system jump table ENINTVL (002EH). The following entry parameters are used:
Register B: 1 msec/8 msec specification
B=00H: 8 msec is specified.
B=10H: 1 msec is specified (see note 1).
Register C: Specification of module that uses the timer interrupt.
The user must specify 40H.

(Note 1) If 1 msec is specified and the barcode reader is used, LED blinking is quickened.

(2) Disabling 1-msec/8-msec interrupts

BIOS CALLX is used to call system jump table DISINTVL (0031H) (see note 2). The following entry parameters are used:
Register C: Code of the used module
The user must specify 40H.

If 1-msec interrupt is used, RZCTLR3 (F0DEH) bit 4 must be turned to 0 after DISINTVL is called.

(Note 2) Even if DISINTVL is called while the key input sound is generated or the barcode reader is used, the interrupts are not actually disabled and the system only checks that the user operation is completed.

(3) 1-msec/8-msec timer

2-byte timer TIMER1M (F032H to F033H) is provided for 1-msec/8-msec interrupt processing and this timer can be referenced freely by the user.
However, TIMER1M cannot be rewritten by the user because it is also used by the system.
TIMER1M is 1-msec timer and can count up to 65,536 msec.

Address : Variable name (Byte length)

F032H : TIMER1M (2)
1-msec/8-msec counter. Unit is 1 msec.

F098H : INTVLFG (1)
Use status of 1-msec/8-msec interrupts

```
 7   6   5   4   3   2   1   0
+---+---+---+---+---+---+---+---+
| - | - | - | - | - |   |   |   |
+---+---+---+---+---+---+---+---+
                      |   |   |
                      |   |   +--> Used for key input operation
                      |   +------> Used for barcode reader
                      +----------> Used by the user
```

(Note "APPENDIX 9 SAMPLE 30" shows the way to extend the interrupts of EXT.)

8.8.3 Interrupt sent from the cartridge

See the following Sections for details on interrupts sent from the cartridge:
1 4.3 Printer
2 7.3 Cartridge Interface
3 11.4 Extending Cartridge Device

It is needed to prohibit the interrupts from the cartridge I/O in the standard execution of the system. See "11.4.3" for details.

8.8.4 Interrupt sent from IC card

An interrupt is sent from the IC card when one of the following events occurs:

1 The back panel of IC card is opened (see note 1).
2 Overcurrent flowed to IC card.

The cause of an interrupt sent from IC card cannot be determined by software. Also, an interrupt sent from IC card cannot be reset by software.

OS forcibly closes the IC card when an interrupt is sent from IC card. Check the following when an interrupt is sent from IC card:

1 Close the back panel.
2 Check that the IC card is not broken.

8.9.1 Overview

The following three ways are provided to extend interrupt processing:

1 The interrupt hook is used.
2 The interrupt vector is rewritten.
3 Interrupt occurrence status is checked.

8.9.2 Extension by using interrupt hook

The following four interrupt processing hooks are provided:

1 HK8251 (ART interrupt hook)
2 ICFHOOK (ICF interrupt hook)
3 OVFHOOK (OVF interrupt hook)
4 EXTHOOK (EXT interrupt hook)

See "Section 10.2 System Hook" for details on each hook.

8.9.3 Extension by rewriting interrupt vector

Interrupt processing created by the user can be executed by rewriting the contents of interrupt jump vectors (FFF0H to FFFFH). Note the following for rewrite operation:

1 Rewrite operation must be done in interrupt suppress status. (DI status).
2 The new interrupt processing must be set in the resident section (E000H to FFFFH) (in other words, the contents of rewritten jump vector must indicate E000H to FFFFH). If required at the jump destination, the bank is switched and processing routine is called again.
3 Interrupt processing must reserve its stack area.
4 Interrupt processing cannot use BDOS or BIOS.
5 The register contents held at interrupt processing start must be recovered after interrupt processing is ended (in other words, all registers must be saved).

8.9.4 Checking the interrupt occurrence status

(1) Overview

In this method, interrupt processing is not directly executed but whether interrupt has occurred is checked often and extended processing is executed if an interrupt has occurred.
Whether an interrupt has occurred is determined in the following two ways:

1 Interrupt occurrence flag INTTYPE (F093H) is checked.
2 Interrupt status register [ISR: P04H] is checked.

7508 and EXT interrupts have two or more interrupt causes. Whether a 7508 or EXT interrupt has occurred can also be determined.

(2) Determination by INTTYPE

The user program must set 00H in INTTYPE in advance to check the interrupt status by using INTTYPE (F093H).

Each INTTYPE bit has the following meaning:

```
           7   6   5   4   3   2   1   0
         +---+---+---+---+---+---+---+---+
INTTYPE  |   |   | - |   |   |   | - | - |
(F093H)  +---+---+---+---+---+---+---+---+
           |   |       |   |   |
           |   |       |   |   +-----------> EXT interrupt occurred
           |   |       |   +---------------> OVF interrupt occurred
           |   |       +-------------------> ICF interrupt occurred
           |   +---------------------------> ART interrupt occurred
           +-------------------------------> 7508 interrupt occurred
```

The corresponding bit is turned to 1 when an interrupt occurs.

(3) Determination by ISR

See "PART 3 HARDWARE" for the meaning of each ISR [P04H] bit.
To check the interrupt status from ISR, interrupts must be suppressed by using IER [P04H] because each ISR bits are reset during interrupt processing.

(4) Checking the cause of 7508 interrupt

The cause of 7508 interrupt can be found out from INTFG (F3C2H). The user must set 00H in INTFG in advance. The INTFG contents set for the first interrupt is erased if two different interrupts (for example, 1-second interrupt and key interrupt) occurs consecutively.

```
  7   6   5   4   3   2   1   0
+---+---+---+---+---+---+---+---+
| - | - | - |   |   |   |   |   |
+---+---+---+---+---+---+---+---+
              |   |   |   |   |
              |   |   |   |   +--> 1-Second interrupt
              |   |   |   +------> Alarm interrupt
              |   |   +----------> Power switch ON interrupt
              |   +--------------> Power failure interrupt
              +------------------> Power switch OFF interrupt
```
(Interrupt occurrence is indicated when a bit is 1.)

The causes of alarm, power off, and power failure interrupts can be also found out by another way. See "Section 8.4.4 Suppressing alarm and power off (power failure) interrupt processing".

(5) Checking the cause of EXT interrupt

The cause of EXT interrupt is checked as follows:
1 Interrupts to be checked are suppressed.
2 The interrupt status is checked.
3 Processing is executed if the checked interrupt status indicates interrupt occurrence.

See Table 8.4 and Table 8.6 for suppressing interrupts and checking interrupt status.

CHAPTER 9 UTILIZING SYSTEM FUNCTIONS

9.1 Overview

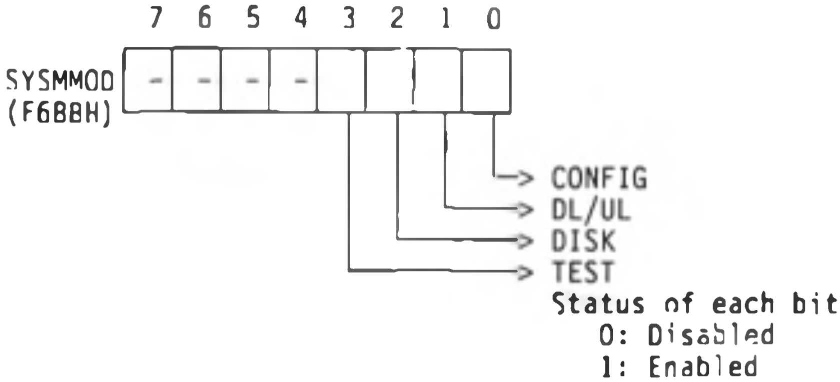
This chapter explains how to use such systems functions as MENU, system menu, and power-on in application programs.

This chapter does not explain how to operate the above system functions. Refer to EHT-10/EHT-10/2 Operation Manual for how to operate these system functions.

9.2.1 Controlling system menu

(1) Enabling/disabling system menu functions

Although DL/UL, DISK, and TEST functions are disabled during application program execution, these functions can be enabled by modifying the SYSMMOD (F6BBH) contents.
However, special attention must be paid because normal operation may not be guaranteed when a specific system function is used under a specific condition. (*1)

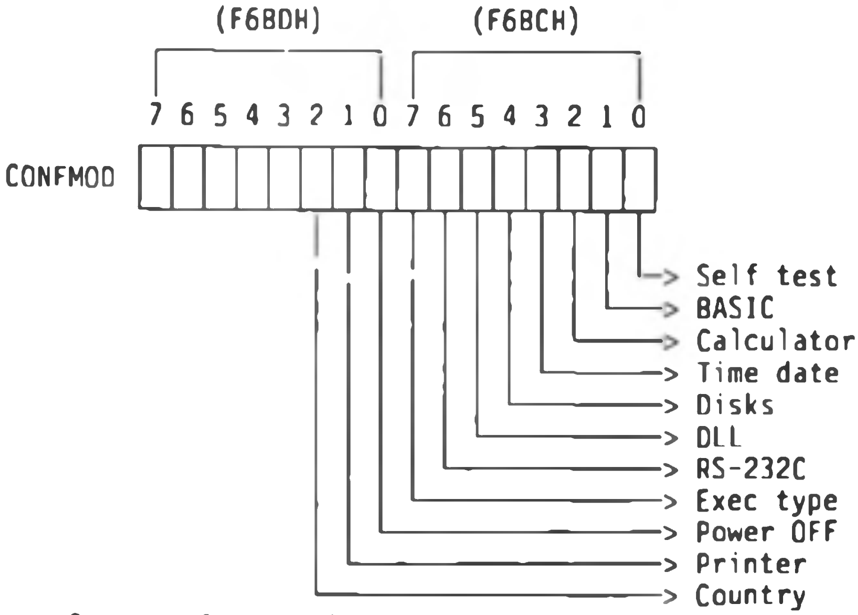


The SYSMMOD contents are reset when WBOOT or restart power-on is executed.

*1 If DISK is used while the RAM disk is open, for example, subsequent processing becomes abnormal.

(2) Enabling/disabling CONFIG processing

Each CONFIG parameter function can be enabled or disabled by modifying the corresponding CONFMOD (F6BCH and F6BDH) contents.



Status of each bit
0: Disabled, 1: Enabled

The CONFMOD contents are reset when WBOOT or restart power-on is executed.

9.2.2 Setting CONFIG parameters

(1) Overview

The system parameters set by the CONFIG functions can also be set by the user program.
This chapter explains how to set each system parameter.

(2) Self test ON/OFF

Self test ON/OFF can be set by updating the contents of the following system areas:

1 RAM sum check: RAMTFLG (F07BH)
2 RAM disk sum check: DSKTFLG (F07AH)

When 01H is set in both of the above areas, self test ON is set. When 00H is set in both of the above areas, self test OFF is set.

(3) BASIC parameters

1 Number of files that can be opened: BASICF (F07CH)
2 Number of random file records: BASICR (F07DH and F07EH)
3 Receive buffer size: BASICB (F07FH and F080H)

The values greater than the values that can be set by CONFIG can also be set in these parameters as long as the BASIC program can be initiated actually.

Note:
Although each BASIC parameter can also be set by the BASIC program, each parameter is made effective after the BASIC program is reinitiated.

(4) Calculator display mode

RNDFIG (F1E0H) is used to set the decimal place up to which the decimal values are displayed. If FFH is set in RNDFIG, values are displayed in a floating-point representation.

(5) Setting date and time

Set the date and time in BIOS TIMDAT. (See Chapter 4.)

(6) Setting RAM disk size and user BIOS area

See Item "CHGRAMD" in Section 10.3 and Section 4.6.

(7) Setting DLL parameter

The DLL parameters can be set by modifying the contents of the seven bytes starting from TCAMPRM (F1CDH).

Address	Offset	Contents
TCAMPRM (F1CDH)	+0	Type
	+1	Line
	+2	Bit rate
	+3	Character length
	+4	Parity check
	+5	Stop bit
	+6	Special parameter

1 Type
   00H: Specifies No protocol Direct-C.
   01H: Specifies No protocol Direct-B.
   02H: Specifies Filink protocol.
   03H: Specifies extended protocol.

2 Line
   00H: RS-232C
   03H: Cartridge SIO

All the above parameters except the type and line parameters are the same as those when open processing was performed by BIOS RSIOX. (See Chapter 4.)

(8) Setting RS-232C parameters (for the system)

The RS-232C parameters for the system can be set by modifying the contents of five bytes starting from SRSPAK (F014H).
Each parameter is the same as that of BIOS RSIOX. (See Chapter 4.)

Address	Offset	Contents
	-4	Receive buffer address
	-2	Receive buffer size
SRSPAK (F014H)	+0	Bit rate
	+1	Character length
	+2	Parity check
	+3	Stop bit
	+4	Special parameter

The low-order four bytes starting from SRSPAK contain the default values of the address and size of the receive buffer. The address and size of the receive buffer can be modified by updating these default values.
If the default receive buffer address is not modified, the maximum buffer size that can be specified is 256 bytes.

(9) Specifying execution type

The execution type can be set by modifying the contents of the following system area:

Address : Variable name (Byte length)

F019H : EXECTYPE (1)
00H: Automatic determination
01H: Forcible selection
02H: Forcible loading

When forcible selection is specified, the execution file name and parameters must be set in the area staring from EXESTRNG (F30DH).

	Character string (Max 33 bytes)

└─> Character string length (one-byte area)

(10) Setting power-off time

1 Automatic power-off time

ATSHUTOFF (F020H): This value must be set in units of minutes.
ATSOTIME (F021H and F022H): This value must be set in units of seconds.

Be sure to set both ATSHUTOFF and ATSOTIME.
If 00H is specified for ATSHUTOFF, automatic power-off is not performed.

2 Setting printer power save mode

Set a value in PRNPWTM (F025H) in units of seconds. If 00H is set, the power save mode is not enabled.

3 Setting automatic backlight-off time (only for EHT-10/2B)

Set a value in ELOFTIME (F023H and F024H) in units of seconds. If 00H is set, the automatic backlight-off function is not enabled.

(11) Setting printer I/F

Set an I/O device code ir RIOBYTE (F415H) and bits 6 and 7 of address 3. See Section 2.6.5 for details.

(12) Specifying a character set for a specific country

Set the code corresponding to the specific country in YLDFLTC(F5F3H) and YLCOUNTRY (F5F4H).

Table 9.1 shows the correspondence between country names and codes.

Country name	Code
U. S. A. (ASCII)	0FH
France	0EH
Germany	0DH
U.K.	0CH
Denmark	0BH
Sweden	0AH
Italy	09H
Spain	08H
Japan	07H
Norway	06H

Table 9.1 Country names and codes for character set adjustment

9.2.3 Specifying DL/UL drives

Although drive A (RAM disk) has been specified in DL/UL as the drive for which down-load or up-load operation is to be performed, another drive (disk) can be specified by modifying the DSKNUM (F1DFH) contents.
The relationship between the values and drive names is as follows.

00H: Drive A (RAM disk)
01H: drive B (ROM socket)
02H: Drive C (IC card)
03H: drive D (Floppy disk)
04H: drive E (Floppy disk)

The value that has been set remains valid until the system is reset.
Because the DSKNUM contents are also referenced by DLL and system menu DISK, the corresponding drive for DLL and DISK is also changed. (The DISK format is not modified.)

9.3 Extending MENU processing

9.3.1 Changing/adding display drives

The current menu display drives can be changed or new menu display drives can be added by modifying the contents of five bytes starting from MDRV (F0B9H).

MDRV +0H (F0B9H)	01H	---> Drive A (RAM disk)
+1H	02H	---> Drive B (ROM disk)
+2H	03H	---> Drive C (IC card)
+3H	FFH	---> Terminator (This and subsequent ones are not disks.)
+4H	FFH	

The following values can be specified:

- 01H: Drive A (RAM disk)
- 02H: drive B (ROM socket)
- 03H: Drive C (IC card)
- 04H: Drive D (floppy disk)
- 05H: Drive E (floppy disk)

Up to four drives can be specified, and FFH must be written at the end. The specified values remain valid until they are reset. If the MDRV contents are updated, the menu drives for which forcible selection has been specified in the CONFIG EXECTYPE parameter are also changed.

9.3.2 Modifying/adding file types

Although the types of files that can be displayed by the MENU function are COM and BAS, these file types can be changed or new file types can be added by updating the contents of 10 bytes starting from MFTYP (F0BEH). However, the files of types other than COM and BAS cannot be processed. (If the file type is not specified (space), COM file type is assumed to be specified.)

MFTYP +0H (F0BEH)	'C'	File type 1
+1H	'O'	
+2H	'M'	
+3H	'B'	File type 2
+4H	'A'	
+5H	'S'	
+6H	FFH	File type 3
+7H	FFH	
+8H	FFH	
+9H	FFH	

Up to three file types can be specified, and FFH must be written at the end.
The specified values remain valid until the system is reset.

If the MFTYP contents are updated, the menu drives for which forcible selection has been specified in the CONFIG EXECTYPE parameter are also changed.

9.4 Extending Power-on Processing

In EHT-10/EHT-10/2, continue mode is always set during application program execution, and the application program need not take power-off into account.

If a device such as the printer unit that has been used since it was set at power off/on need be initialized, power-on processing must be performed by the application program.

There are the following two power-on processing procedures.

(1) Power-on processing by using the cartridge mount processing hook (CRGHOOK)

(2) Power-on processing through checking the power-on flag (SRSTMODE: F389H)

For procedure (1), see Section 11.4. CRGHOOK can perform power- on processing required for the application program regardless of the cartridge device type.
SRSTMODE (F389H) used in procedure (2) is a flag which is set to 01H at power on. Procedure (2) is as follows.

1 00H is set in SRSTMODE at the beginning of the program.
2 The SRSTMODE contents are always checked and, if they are 01H, Step 3 is performed.
3 00H is set in SRSTMODE.
4 The power-on processing routine created by the user is called.

CHAPTER 10 SYSTEM HOOKS AND JUMP TABLES

10.1 Overview

The EHT-10 and EHT-10/2 systems have system hooks and jump tables so that the application program can extend system functions or use specific functions that the system has.
There are 27 system hooks. These hooks are used in interrupt processing, extension and modification of BIOS functions, and other operations.

There are the following three jump tables, each used to jump to a specific system processing routine.

1. Resident area jump table
2. OS ROM jump table (I)
3. OS ROM jump table (II)

There are 31 system processing routines.

This chapter explains the structure and usage procedure of each jump table and the system hook control flow.

10.2 System Hooks

10.2.1 Overview

The EHT-10 and EHT-10/2 systems have 27 system hooks so that the application program can easily perform interrupt processing, extension of BIOS and serial communication functions, and other operations.

The user can enable the above processing simply by replacing the jump address of the hook with the address of the extend processing routine.
This section explains the structure and usage procedure of each hook table and the system hook control flow.
The functions of some hooks are also explained in detail in relevant chapters. Refer to these chapters also.

10.2.2 Structure and usage procedure of each hook table

(1) Hook table structure and processing flow

A total of 27 system hooks are configured into two tables shown in Figures 10.1 and 10.2.
Figures 10.3 and 10.4 show control flow through each hook table.

Fig. 10.1 Structure of System hook table (I)

Address		
FFC0H	ALMHK0	
FFC3H	ALMHK1	
FFC6H	ALMHK2	Alarm interrupt hooks
FFC9H	ALMHK3	
FFCCH	ALMHK4	
FFCFH	Reserved	
FFD2H	HK8251	ART interrupt hook
FFD5H	ICFHOOK	ICF interrupt hook
FFD8H	OVFHOOK	OVF interrupt hook
FFDBH	EXTHOOK	EXT interrupt hook
FFDEH	TMDT83	BIOS TIMDAT extension hooks
FFE1H	TMDT85	
FFE4H	TMDT86	
FFE7H	BIOSHK	BIOS hook
FFEAH	BASHOOK	BASIC hook
FFEDH	Reserved	

Hook table (I)

C3H (JP instruction)
Jump address

Structure of each hook

Address		
FF60H	TEXHK1	BIOS TCAM extension hook
FF63H	DLLHK1	DLL extension hook
FF66H	DLHK1	DL extension hook
FF69H	ULHK1	UL extension hook
FF6CH	CRGHOOK	Cartridge mount hook
FF6FH	ICDHK1	IC card hook
FF72H	ICDHK2	
FF75H	ICDHK3	
FF78H	ICDHK4	
FF7BH	ICDHK5	
FF7EH	ICDHK6	
FF81H	ICDHK7	
FF84H	ICDHK8	
FF87H	Reserved	
FF8AH	Reserved	
FF8DH	Reserved	

Hook table (II)

Bank information
Jump address

Structure of each hook

Note:

The current bank is automatically switched according to the bank information, and the corresponding processing routine is called through system hook table (II).

Fig 10.2 Structure of system hook table (II)

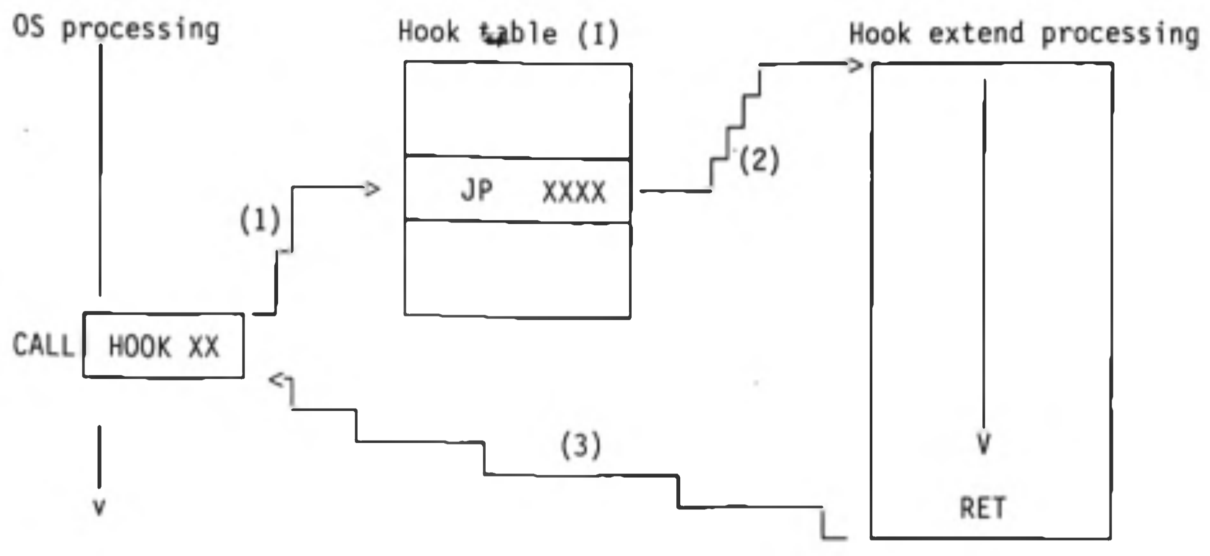


Fig 10.3 Control flow through hook table (I)

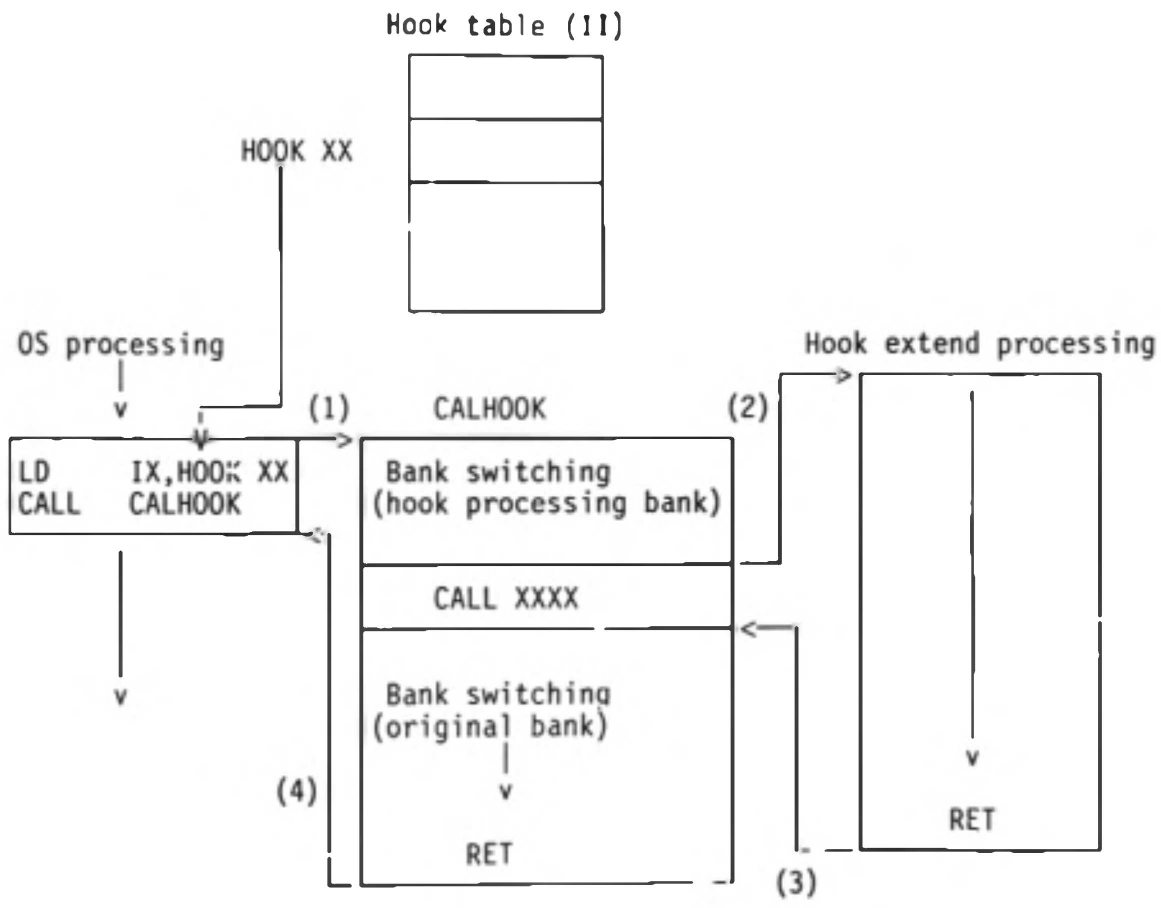


Fig 10.4 Control flow through hook table (II)

(2) Hook types

System hooks can be classified to the following three types according to the usage procedure.

1 System hooks that jump to the hook processing routines under the system bank condition.
The 10 hooks listed below belong to this system hook group. The hook processing routines for these hooks must reside in RAM address 8000H or later.

ALMHK0, ALMHK1, ALMHK2, ALMHK3, ALMHK4, HK8251, TMDT83, TMDT85, TMDT86, and BIOSHK

2 System hooks that jump to the hook processing routines after switching the current bank to all-RAM bank (bank 0#0)
The four hooks listed below belong to this system hook group. The hook processing routines for these hooks must reside in RAM.

ICFHOOK, OVFHOOK, EXTHOOK, and BASHOOK

3 System hooks that jump to the hook processing routines after switching the current bank to the user-specified bank
All hooks in hook table (II) belong to this system hook group.
The user can specify the bank and address of the hook processing routine. The operating system switches the current bank to the bank specified by the user and then jumps to the specified address.

(3) Updating hooks

A hook can be updated after being set to DI status.
Also, the hook processing routine must be loaded before the hook is updated.

1 To update a hook registered in hook table (I), set the execution start address of the corresponding hook processing routine in the two-byte area starting from the address obtained by adding 1 to the entry address of the hook.

C3H	JP instruction
Hook processing address	This address must be set by the user.

2 To update a hook registered in hook table (II), set bank information and hook processing address (low-order byte and high-order byte) sequentially in this order starting from the entry address of the hook.

Bank information	These information must be set by the user.
Hook processing address	

(4) Notes on creating hook processing routine

Note the following when creating a hook processing routine.

1 Saving registers

Because only the minimum number of required registers are saved for each hook processing, all registers to be used must be saved. Especially, if all registers to be used are not saved before an interrupt hook is executed, the program may go into an abnormal loop when control is returned from the interrupt routine.

2 Saving interrupt status

Some interrupt hook routines are performing processing in DI status, and multi-interrupt processing is not taken into account for some of these interrupt hook routines.
Therefore, the interrupt disabled status for each hook must be saved.
The interrupt disabled status for each hook is as follows.

ALMHK0: DI status
ALMHK1: DI status
ALMHK2: EI status, interrupt by 7508 CPU disabled
ALMHK3: EI status, interrupt by 7508 CPU disabled
ALMHK4: EI status
HK8251: DI status
ICFHOOK: DI status
OVFHOOK: DI status
EXTHOOK: DI status

3 Hook processing is usually terminated by an RET instruction.

(5) Miscellaneous

1 Hook processing routines can be placed in the user BIOS area. In this case, the routine must conform to the user BIOS area usage procedure.
See Section 4.6 for the user BIOS area usage procedure.

2 Hook processing can be invalidated by the following:

- For a hook in hook table (I): Set jump address F000H. (*1)
- For a hook in hook table (II): Set bank information FFH (system bank) and jump address F000H.

The hook table is initialized when the system is reset or initialized.

*1 Because an RET instruction has been stored in address F000H, the hook returns control without performing any processing when it is called.

10.2.3 System hooks registered in system hook table (I)

This chapter explains how to use 14 system hooks registered in system hook table (1).

(1) Alarm hooks

EHT-10 and EHT-10/2 each has five hooks for alarm processing. These hooks are used for such operations as automatic updating of alarm/wake data.
These hooks can be called at the following point in time:

ALMHK0: Immediately before the alarm screen is displayed during the processing after an alarm occurred in the power-off status

ALMHK1: Immediately after the alarm screen is erased during the processing after an alarm occurred in the power-off status

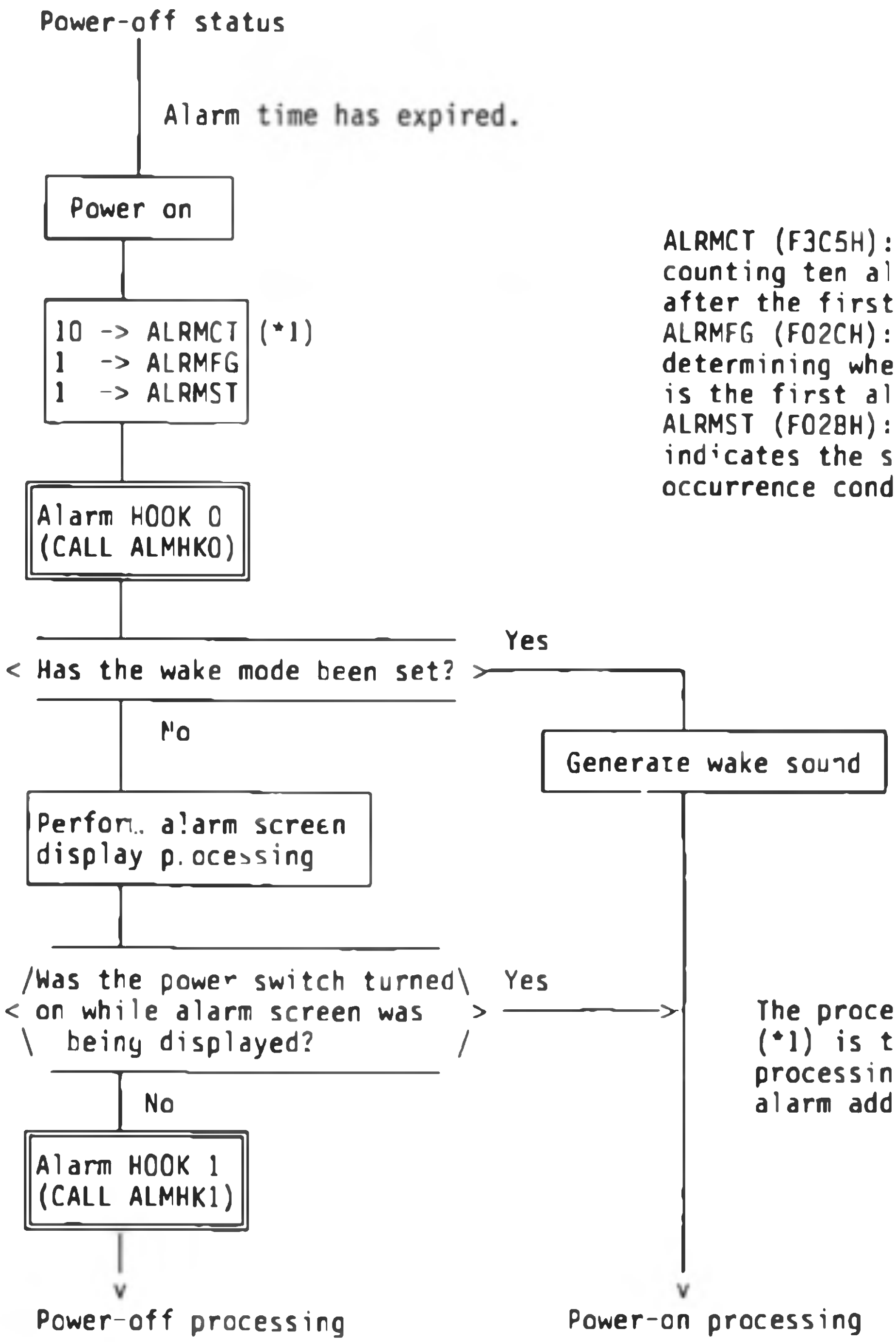
ALMHK2: Immediately before the alarm screen is displayed during the processing after an alarm occurred in the power-on status

ALMHK3: Immediately after the alarm screen is erased during the processing after an alarm occurred in the power-on status

ALMHK4: Immediately before the sequence escapes from the alarm screen during the processing after an alarm occurred

Figures 10.5 to 10.7 show control flow among these hooks.

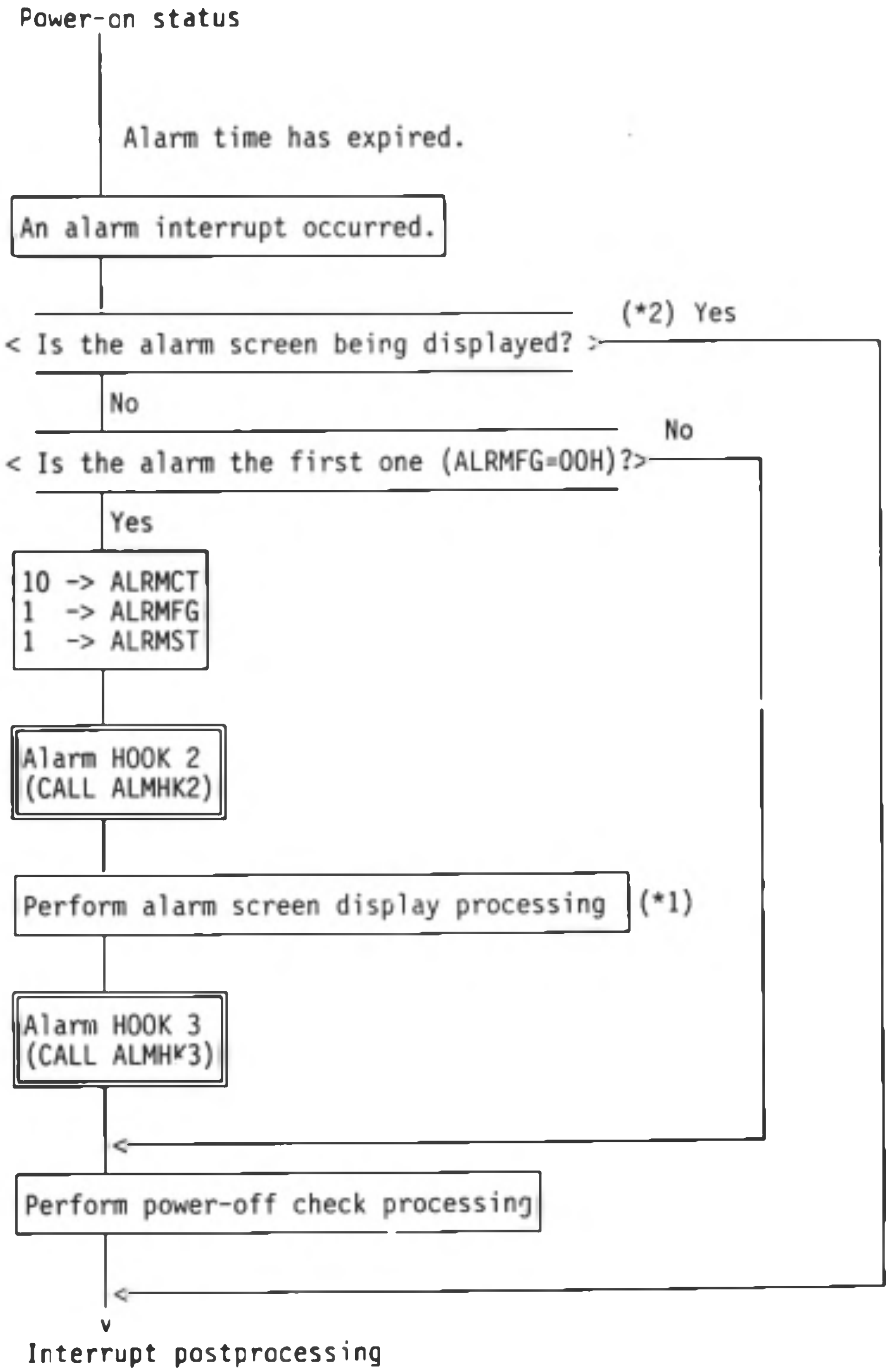
For example, if an alarm or wake occurred when ALMHK0 and ALMHK2 are used during updating alarm/wake data, the next alarm or wake time is set by using these hooks.



ALRMCT (F3C5H): Counter for counting ten alarm interrupts after the first alarm occurred
ALRMFG (F02CH): Flag for determining whether the alarm is the first alarm
ALRMST (F028H): Flag which indicates the specified alarm occurrence condition

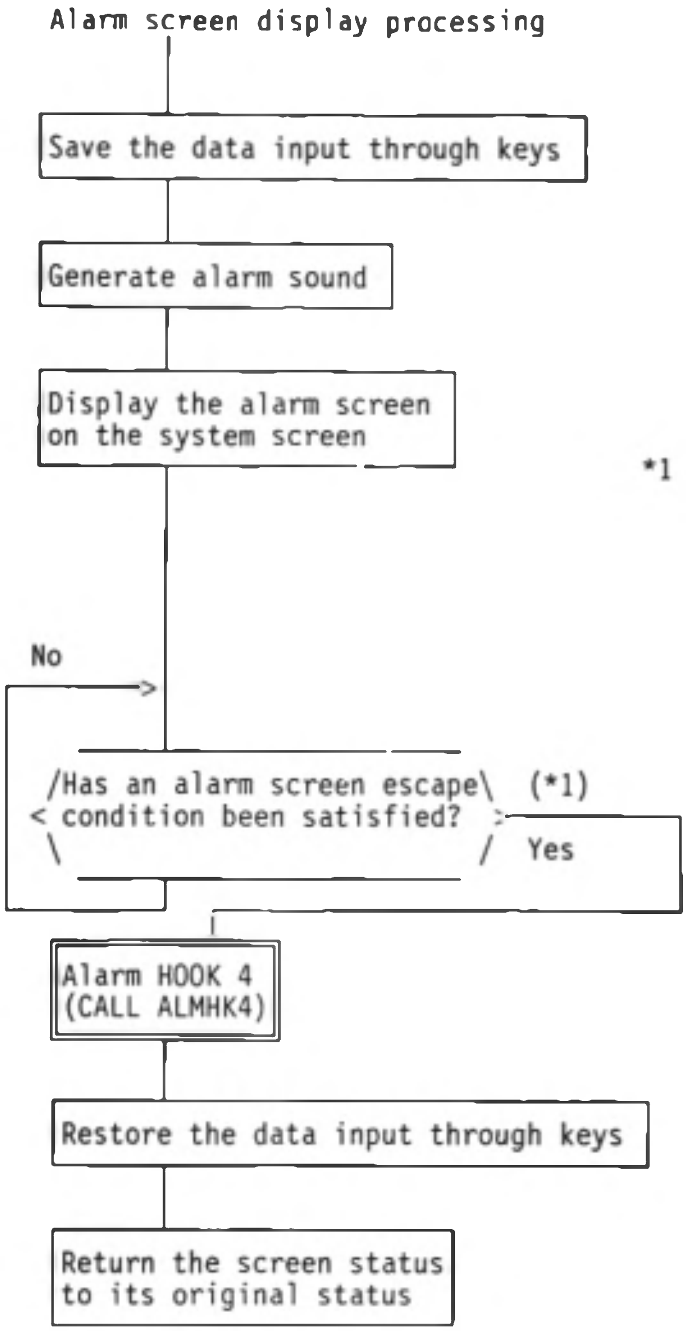
The processing after (*1) is the OS start processing at alarm address 0

Fig. 10.5 Alarm processing in power-off status



*1 If the displaying of the alarm screen has been disabled, alarm screen display processing is skipped.
The processing after (*2) are OS alarm interrupt processing

Fig. 10.6 Alarm processing in power-on status



*1 The sequence can escape from the alarm screen when one of the conditions including the following is satisfied:

1 Five seconds have elapsed.
2 The power switch is turned off.
3 Charge battery

Fig 10.7 Alarm screen display processing

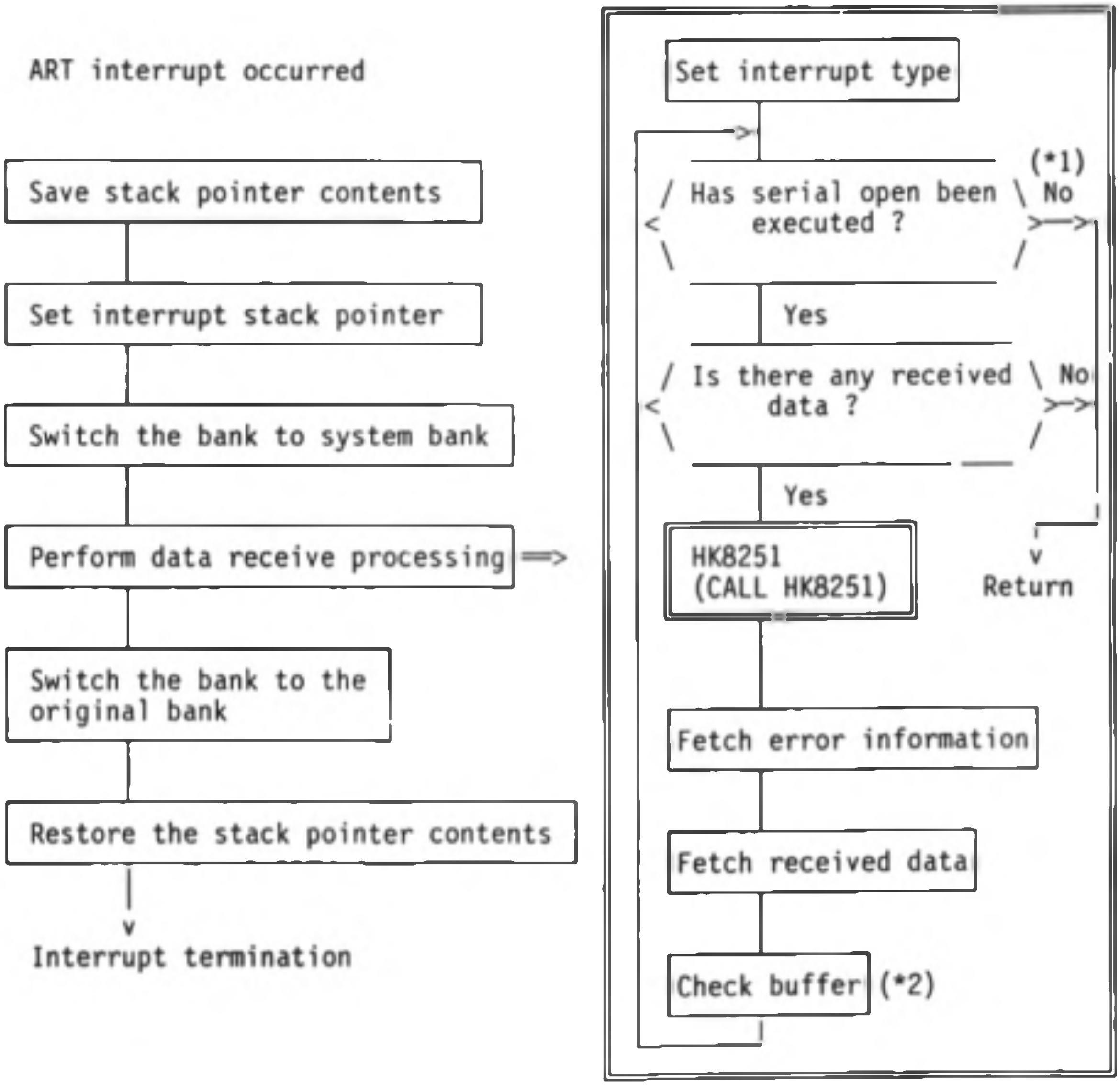
(2) Interrupt hooks

EHT-10 and EHT-10/2 each has four hooks for interrupt processing. These hooks can be called at the following point in time.

HK8251: Before data is received during ART interrupt processing
ICFHOOK: During ICF interrupt processing
OVFHOOK: Immediately before blink processing during OVF interrupt processing
EXTHOOK: During EXT interrupt processing

1 HK8251

HK8251 is stored in OS ROM so that it can extend ART interrupt processing.
Figure 10.8 shows where HK8251 is used in ART interrupt processing. As indicated in the figure, no processing is performed and control is returned even if an interrupt occurs before BIOS serial OPEN is executed. This should be noted. See "APPENDIX 9 SAMPLE 29".



(Processing in resident area) (Processing in OS ROM)

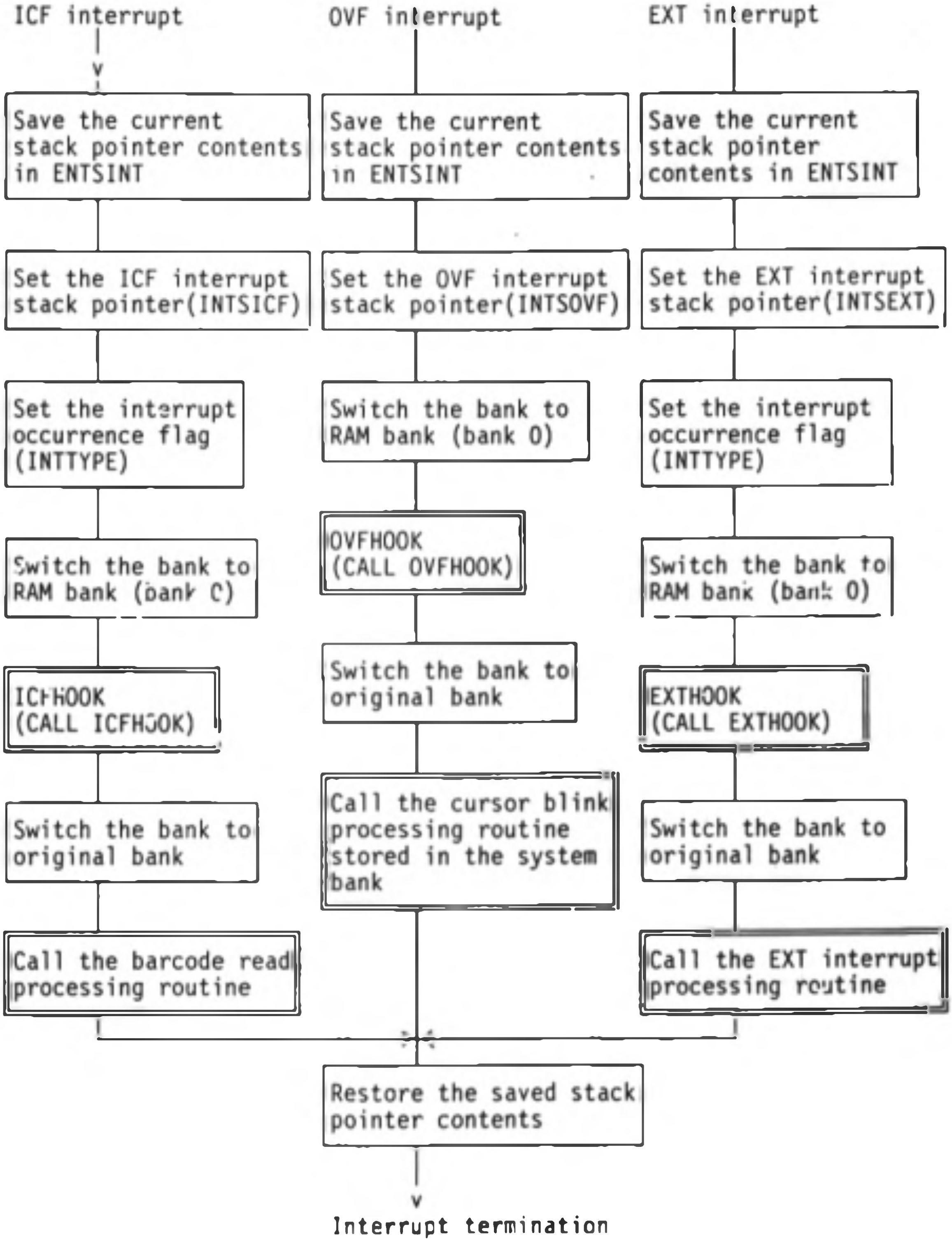
Fig 10.8 ART interrupt processing

(*1) This step is performed to check whether serial open processing has been done by BIOS RSIOX, ICCARD, or TCAM CONNECT.

(*2) This step is performed to determine whether a send suspend request is to be issued for the send destination when a buffer control such as XON/XOFF, RTS/CTS, or DTR/DSR has been specified.

2 ICF, OVF, and EXT interrupt hooks

Before an ICF, OVF, or EXT interrupt hook is executed, the current bank is switched to bank 0 (RAM) and, when hook execution is terminated, the bank is switched again to the original bank. Figure 10.9 shows when each hook is called in interrupt processing.
Because the size of the stack area allocated for each interrupt processing is restricted to the minimum, a new stack area size must be set when extending the interrupt processing.
There are three causes for EXT interrupts: 1 ms or 8 ms timer, cartridge I/F, and IC card I/F. EXTHOOK must check which of the above three is the cause of the current interrupt, and perform extend processing only for the determined interrupt cause. See Section 8.8 for details.



ENTSINT (F3C0H): Stack pointer save area
INTSICF (F09DH): ICF interrupt stack pointer set address
INTSOVF (F09FH): OVF interrupt stack pointer set address
INTSEXT (F0A1H): EXT interrupt stack pointer set address
INTTYPE (F093H): Interrupt occurrence flag

Fig 10.9 ICF, OVF, and EXT interrupt processing

(3) BIOS TIMDAT hooks

EHT-10 and EHT-10/2 each has the following three hooks for BIOS TIMDAT extend processing.

TMDT83: For BIOS TIMDAT when C=83H
TMDT85: For BIOS TIMDAT when C=85H
TMDT86: For BIOS TIMDAT when C=86H

Only the DE register contents are saved when BIOS TIMDAT is called.

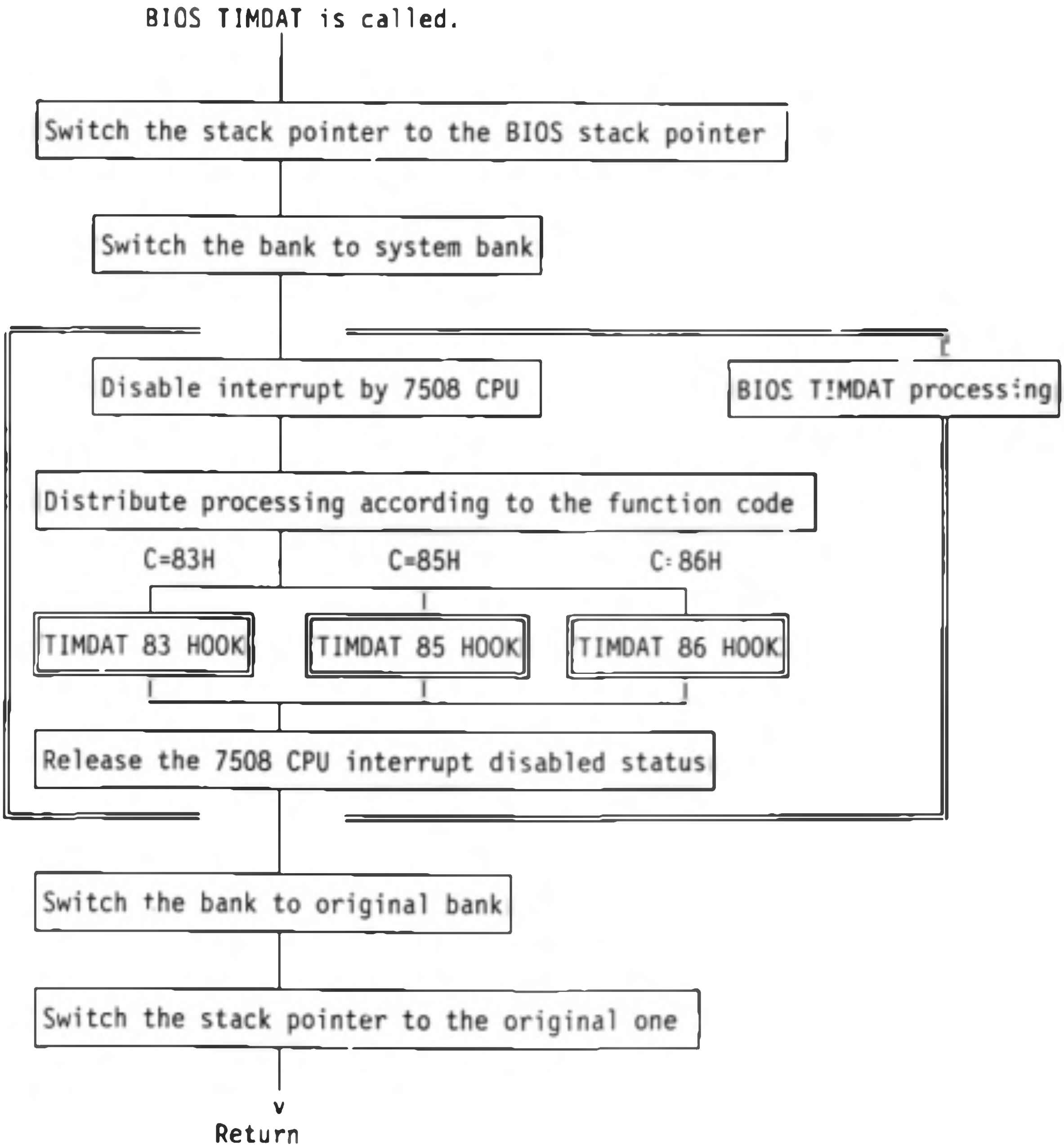


Fig 10.10 TIMDAT processing

(4) In addition to user BIOS functions, EHT-10 and EHT-10/2 each has a BIOS hook for extending each BIOS function.
This BIOS hook enables conventional BIOS processing to be modified arbitrarily. How to use this BIOS hook is explained below.
See "APPENDIX 9 SAMPLE31".

1 BIOS hook calling point in time

The BIOS hook is called through the hook table allocated in OS ROM in the system bank immediately before the sequence jumps to a specific BIOS function.
Figure 10.11 is a system processing flowchart in which the BIOS hook is executed.

If the BIOS function is called by BDOS, only Steps (3) to (8) are executed as BIOS processing.

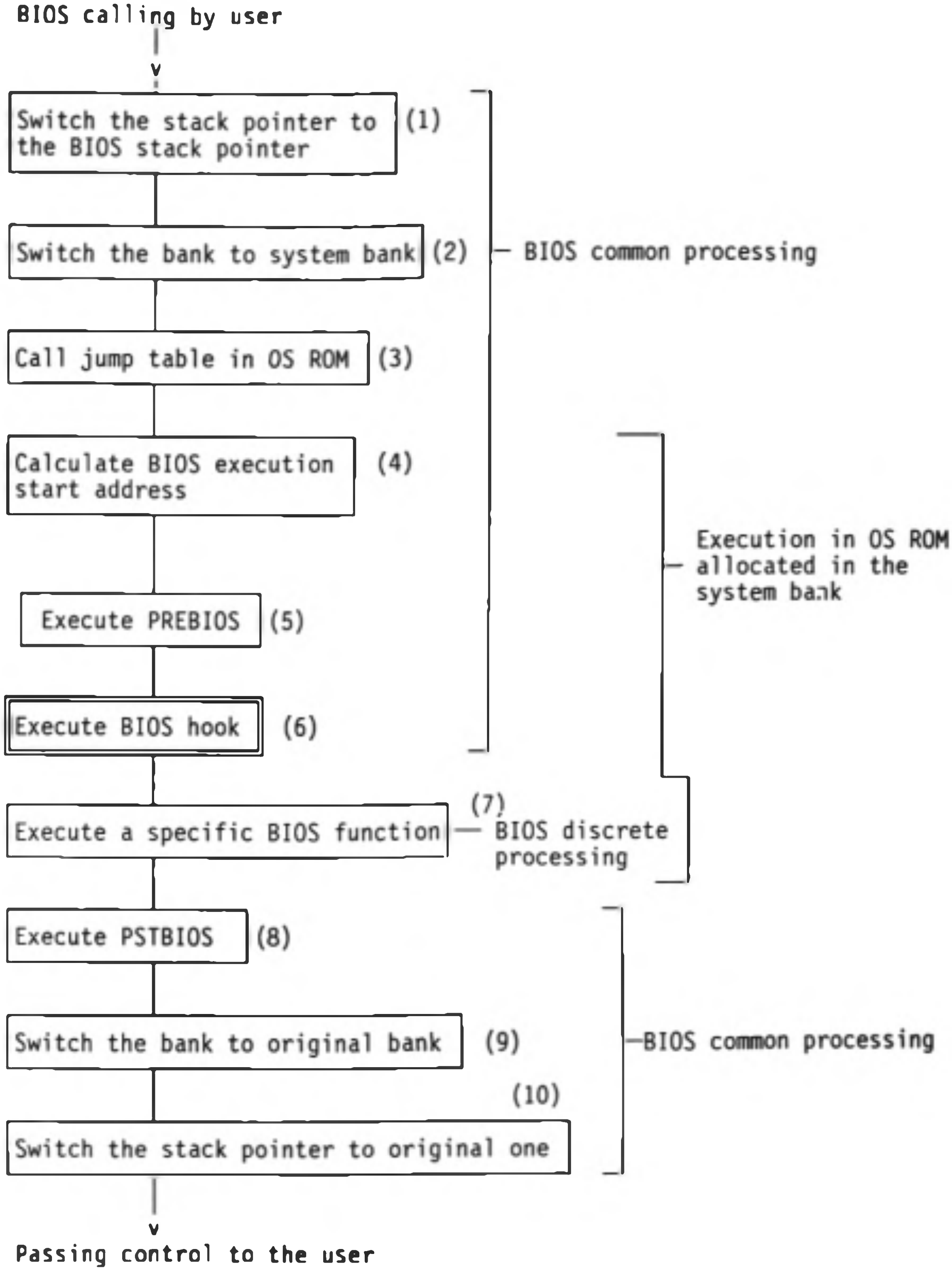


Fig 10.11 BIOS processing flowchart

Step : Explanation

(1) Switching the current stack pointer to the BIOS stack pointer
The current stack pointer value is saved in USRSBI. The current stack pointer is switched to the BIOS stack pointer.

(2) Switching the current bank to the system bank
The BIOS function number is checked. The DMA address is translated to the system DMA address. The current IO byte is checked and its contents are set in RIOBYTE.
The current bank is switched to the system bank and the original bank information is saved.

(3) Calling jump table in OS ROM
The jump table stored in OS ROM is called according to the BIOS function number.

(4) Calculating the BIOS execution start address
The BIOS execution start address is calculated according to the address of the called jump table.

(5) Executing PREBIOS
PREBIOS (see Section 4.1.2) is executed.

(6) Executing BIOS hook
BIOS hook (FFE7H) is called. The register contents remain unchanged since BIOS was called.

(7) Executing a specific BIOS function
A BIOS processing routine is called according to the BIOS execution start address calculated in Step (4) .

(8) Executing PSTBIOS
PSTBIOS (see Section 4.1.2) is executed.

(9) Switching the current bank to the original one
The current bank is switched to the original bank according to the bank information saved in Step (2) . For a read function, the DMA data is copied.

(10) Switching the current stack pointer to the original one
The stack pointer value saved in Step 1 is restored.

The following table lists the system areas used in BIOS common processing.

Address : Variable name (Number of bytes)

F415H : RIOBYTE (1)
IO byte storage area.
See Section 2.6.5 for the configuration of this area.

F418H : OLDBNK (1)
Bank information save area
FFH: System bank
00H: Bank 0
01H: Bank 1
02H: Bank 2

F424H : USRSBI (2)
BIOS user stack pointer save area

F426H : BIOSFN (1)
BIOS function number storage area
00H: BOOT
03H: WBOOT
.
.
.
A5H: INFORM

F430H : SAVEIX (2)
IX register save area

F432H : SAVEIY (2)
IY register save area

2 Logic of the hook processing routine

The BIOS hook is always called when any BIOS function is called. When the BIOS hook is called, the user must determine whether the called BIOS function is the one to be extended.

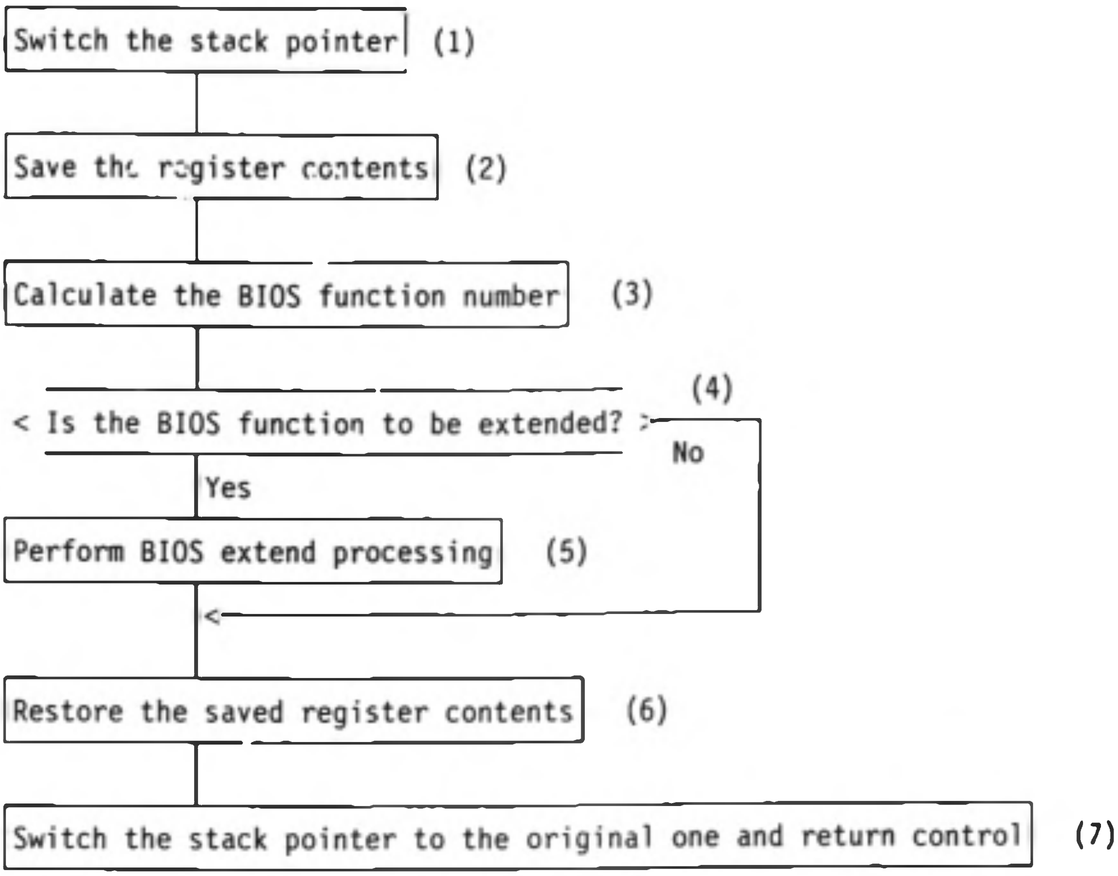


Fig 10.12 Example of BIOS hook processing

Step : Explanation

(1) Switching the stack pointer
Because the system BIOS stack pointer is currently used, a new stack area must be allocated when the processing is to use much stack area is to be used in the processing.

(2) Saving register contents
Because the parameters corresponding to each BIOS function have been set in the current registers, the current register contents must be saved when the registers are to be used for a new BIOS function.

(3) Calculating BIOS function number
Calculate the BIOS function number from the BIOS execution start address and the BIOS jump table top address.

```
Stack pointer for Step  (1)-> +-------------------------+
                              |  BIOS hook              | (L)
                              |- return address        -|
                              |                         | (H)
                              +-------------------------+      --+
                              |  BIOS execution         | (L)    |
                              |- start address         -|        |  (A)
                              |                         | (H)    |
                              +-------------------------+      --+
                              |  BIOS postprocessing    | (L)
                              |- execution address     -|
                              |                         | (H)
                              +-------------------------+
                              |                         |


                       0007H  +-------------------------+      --+
                              |  BIOS jump table        | (L)    |
                              |- top address           -|        |  (B)
                       0008H  |                         | (H)    |
                              +-------------------------+      --+
```

The difference of (A) - (B) is the offset value from BOOT(00H, 03H, ...).

(4) Determining whether the BIOS function is to be extended
The function number of the called BIOS function is compared with the function number obtained in Step (3) to determine whether the called BIOS function is the one to be extended.
If the following condition is satisfied when the function number of the called BIOS function is n, it is determined that the object BIOS function has been called:

(A) - (B) = 3 x n

```
n = 00H ... BOOT
  = 01H ... WBOOT
  = 02H ... CONST
    .
    .
    .
  = 37H ... INFORM
```

(5) BIOS extend processing
BIOS extend processing is performed. The user inserts BIOS extend processing here.

(6) Restoring saved register contents
The register contents saved in Step (2) are restored.

(7) Switching the stack pointer to the original one and returning control
The stack pointer value saved in Step (1) is restored.
Control is returned, following which subsequent BIOS processing is performed by OS. To suppress BIOS processing by OS, control is returned after poped two times (4 bytes). In this case, the sequence skips Step (7) and directly goes to Step (8) in the BIOS processing flowchart shown in Figure 10.11.

3 Updating BIOS hook jump address

The jump address of the BIOS hook can be updated.

FFE8H	BIOS hook	(L)
FFE9H	jump address	(H)

The initial value is F000H. RET has been stored in address F000H.
By replacing the contents of addresses FFE8H and FFE9H with the top address of the hook processing routine, the hook processing routine is executed after the BIOS function is called.

4 Notes on using BIOS hook

(a) Because the system bank is used when control is passed to the BIOS hook, the hook processing routine must reside in address 8000H or later. Reserve a user BIOS area and process the hook processing routine in the area.

(b) The sequence of each BIOS function used in BDOS also goes through this BIOS hook.

(c) The hook processing routine cannot call BIOS and BDOS (RBIOS1, RIBIS2, RBDOS1, and RBDOS2) that reside in RAM. To use BIOS, BIOS that reside in OS ROM must be directly called. (BDOS cannot be used.)

(d) To pass return information to IX and IY, the IX and IY values must be set in SAVEIX (F430H) and SAVEIY (F432H), respectively.

(5) BASIC activation hook

BASHOOK has been provided as a hook for BASIC activation. See "PART 3 BASIC" for details.

10.2.4 System hooks registered in system hook table (II)

The 13 system hooks registered in system hook table (II) are explained in detail in relevant sections as shown below.

```
TEXHK1 ─┐
DLLHK1  ├──> Section 11.2  Communication Protocol Expansion
DLHK1   │
ULHK1  ─┘

CRGHK1 ────> Section 11.4  Cartridge Device Expansion

ICDHK1 ─┐
ICDHK2  │
ICDHK3  │
ICDHK4  │
ICDHK5  ├──> Section 11.3  IC card Protocol Expansion
ICDHK6  │
ICDHK7  │
ICDHK8 ─┘
```

10.3 System Jump Tables

10.3.1 Overview

EHT-10 and EHT-10/2 each has the following two types of system jump tables.

1 Resident area jump table

This jump table is used to jump to system routines related to the banks relocated in the resident area.

2 OS ROM jump table

This jump table is used to jump to system routines that reside in OS ROM. The OS ROM jump table is further classified to the following two types:

1 System bank jump table
2 Bank 2 (subbank 2#1) jump table

10.3.2 Jump table configuration

(1) Configuration of resident area jump table

The resident area jump table resides in addresses FF90H to FFBFH in the resident RAM, and can be accessed from any bank through a CALL instruction.
Figure 10.13 shows the configuration of the resident area jump table. The detailed function of each system routine called through this jump table is explained in Section 10.3.3.

Address		
FF90H	JRBDOS	Entry point of RBDOS2
FF93H	JPREBIOS	PREBIOS processing
FF96H	JPSTBIOS	PSTBIOS processing
FF99H	JSCALLX	Almost same as BIOS CALLX
FF9CH	JSELBNK	Bank switch routine
FF9FH	JLDAXX	Almost same as BIOS LOADX
FFA2H	JSTAXX	Almost same as BIOS STORX
FFA5H	JLDIRXX	Almost same as BIOS LDIRX
FFA8H	JJUMPXX	Almost same as BIOS JUMPX
FFABH	JCALLXX	Almost same as BIOS CALLX
FFAEH	JINTCPR	Almost same as BIOS MASKI
FFB1H	JSDBGIOX	Debugger control routine (for the system)
FFB4H	JUDBGIOX	Debugger control routine (for the user)
	Reserved	
FFC0H		

Fig 10.13 Resident area jump table configuration

(2) Configuration of OS ROM jump table (I)

OS ROM jump table (I) is allocated staring from the head of OS ROM (system bank address 0000H). Figure 10.14 shows its structure. The current bank must be switched to the system bank when calling a routine through OS ROM jump table (I).

1 BIOS CALLX is used when the application program calls a system routine.

2 JCALLX or JSCALLX of the resident area jump table is used when calling a system routine during interrupt extend or BIOS extend processing.

The detailed function of each system routine called through this jump table is explained in Section 10.3.4.

Address		
0000H	(NOP)	
0001H	(STARTER)	Address 0 start
0004H	BDOSTABL	BDOS function table address
0007H	BIOSTABL	BIOS function table address
000AH	SETERR	Setting BDOS special error code
000DH	RSTERR	Setting BDOS ordinary error code
0010H	GOBACK	BDOS error recovery processing
0013H	BIOSJTLD	Relocate processing of RBIOS1, RBIOS2
0016H	SETRAMAD	Setting load address of RBDOS1, RBIOS1
0019H	CRGRAMD	Modifying RAM disk size
001CH	FMTRAMD	Formatting RAM disk
001FH	EPSPSND	Sending under EPSP protocol
0022H	EPSPRCV	Receiving under EPSP protocol
0025H	MELODY	Melody
0028H	WRT7508	Outputting a command to 7508 CPU (only for a non response command)
002BH	CMD7508	Outputting a command to 7508 CPU (only for a response command)
002EH	EINTVL	Enabling 1 mS / 8 mS timer interrupt
0031H	DISINTVL	Disabling 1 mS /8 mS timer interrupt

Fig 10.14 Configuration of OS ROM jump table (I)

(3) Configuration of OS ROM jump table (II)

OS ROM jump table (II) is allocated in an OS ROM area starting from address 8000H (address 6000H of bank 201). Figure 10.15 shows its structure.
The current bank must be switched to bank 201 when calling a routine through OS ROM jump table (II).

1 BIOS CALLX is used when the application program calls a system routine.

2 JCALLX or JSCALLX of the resident area jump table is used when calling a system routine during interrupt extend or BIOS extend processing.

The detailed function of each system routine called through this jump table is explained in Section 10.3.5.

6000H	NOP
6001H	Checking IC card mounting
6004H	Formatting IC card

Fig 10.15 Configuration of OS ROM jump table (II)

10.3.3 Resident area jump table

13 system routines have been registered in the resident area jump table. This section explains the function and parameters of each system routine called through the resident area jump table.

JSELBNK, JKDAXX, JSTAXX, JJUMPXX, JCALLXX, JSCALLX, and JINTOPR are used to control banks and interruption without using BIOS such as in interrupt extend or BIOS extend processing.

[Function]
JRBDOS indicates the entry address of RBDOS2.

[Entry parameters]
Same as those of each BDOS function

[Return parameters]
Same as those of each BDOS function

[Explanation]

(1) In EHT-10 and EHT-10/2, there are two BDOS entries at different locations in RAM. This is because:

1 The upperlimit of the usable RAM memory range can be indicated so that an ordinary CP/M application program can run without being modified.

2 ROM-based program can use BDOS without taking bank switching into account.

(2) JRBDOS indicates the entry address of RBDOS2 that resides in the system common area.

(3) A ROM-based program calls JRBDOS (FF90H) instead of calling address 0005H that must be called when using conventional BDOS.

(4) See APPENDIX 7 and Chapter 5 for details on each BDOS function.

[Function]

JPREBIOS sets the BIOS execution flag on, disabling subsequent operations such as alarm and power-off operations.
JPSTBIOS performs operations that occurred after JPREBIOS execution such as alarm and power-off operations, and sets the BIOS execution flag off.

[Entry parameters]

None

[Return parameters]

None

[Explanation]

(1) The contents of all registers are saved.

(2) PREBIOS is used to disable interruption to be caused by such an event as power-off or alarm during BIOS processing by OS. PSTBIOS is used to release the interrupt disabled status.

(3) JPREBIOS and JPSTBIOS are routines enhanced from PREBIOS and PSTBIOS, respectively, so that they can be used by application programs.

(4) The application program uses JPREBIOS and JPSTBIOS under the following conditions:

1 The power must not be turned off during program execution.
2 The alarm screen must not be displayed during program execution.
3 Power failure must be suppressed during program execution.
4 BIOS functions that reside in OS ROM must be directly used.

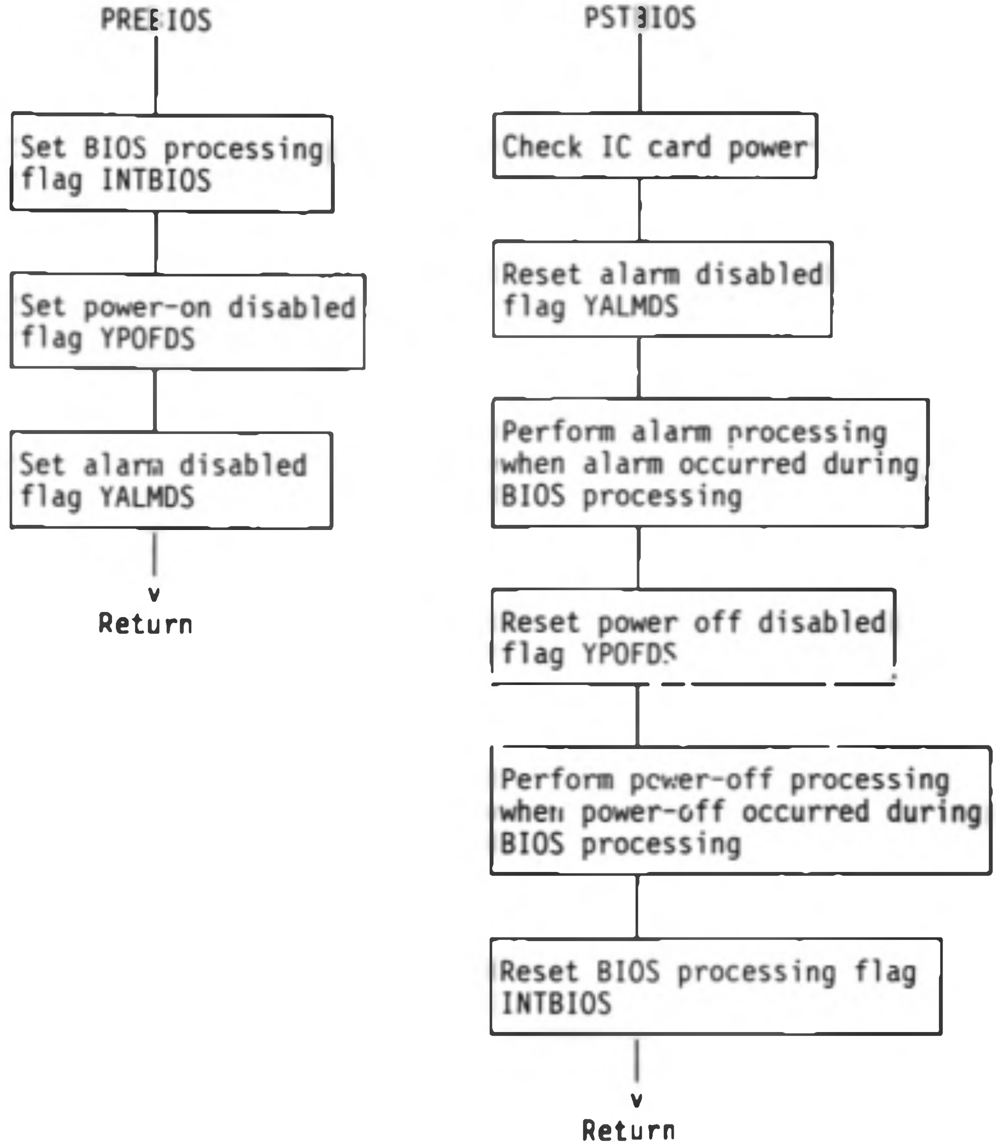
(5) Because JPREBIOS and JPSTBIOS performs processing after internally switching the current stack pointer to the BIOS stack pointer, these routines cannot be used when the BIOS stack pointer has already been used.

(6) If JPREBIOS is executed, JPSTBIOS must be executed later. If JPSTBIOS is not executed after JPREBIOS, power-off and alarm operations are not performed.

(7) If a BIOS function is used after JPREBIOS execution, PSTBIOS is automatically executed at the end of the BIOS processing. After that, alarm and power-off are immediately processed when they occurred. To use a BIOS function after JPREBIOS execution, the BIOS function that resides in OS ROM must be directly called.

(8) If the processing time between JPREBIOS and JPSTBIOS becomes too long, the processing is separated. This is because, if power-off operation is not performed within 50 seconds after a power failure occurred, the mainframe power is forcibly turned off.

Figure 10.16 shows processing flow of PREBIOS and PSTBIOS.



INTBIOS (F091H): BIOS processing flag
  00H: BIOS processing is not being performed.
  FFH: BIOS processing is being performed.
YPOFDS (F0B4H): Power-off disabled flag
  Bit 5 is used during BIOS execution.
YALMDS (F0B6H): Alarm disabled flag
  Bit 5 is used during BIOS execution.

Fig 10.16 Processing flow of PREBIOS and PSTBIOS

Figure 10.17 shows the processing flow of JPREBIOS and JPSTBIOS.

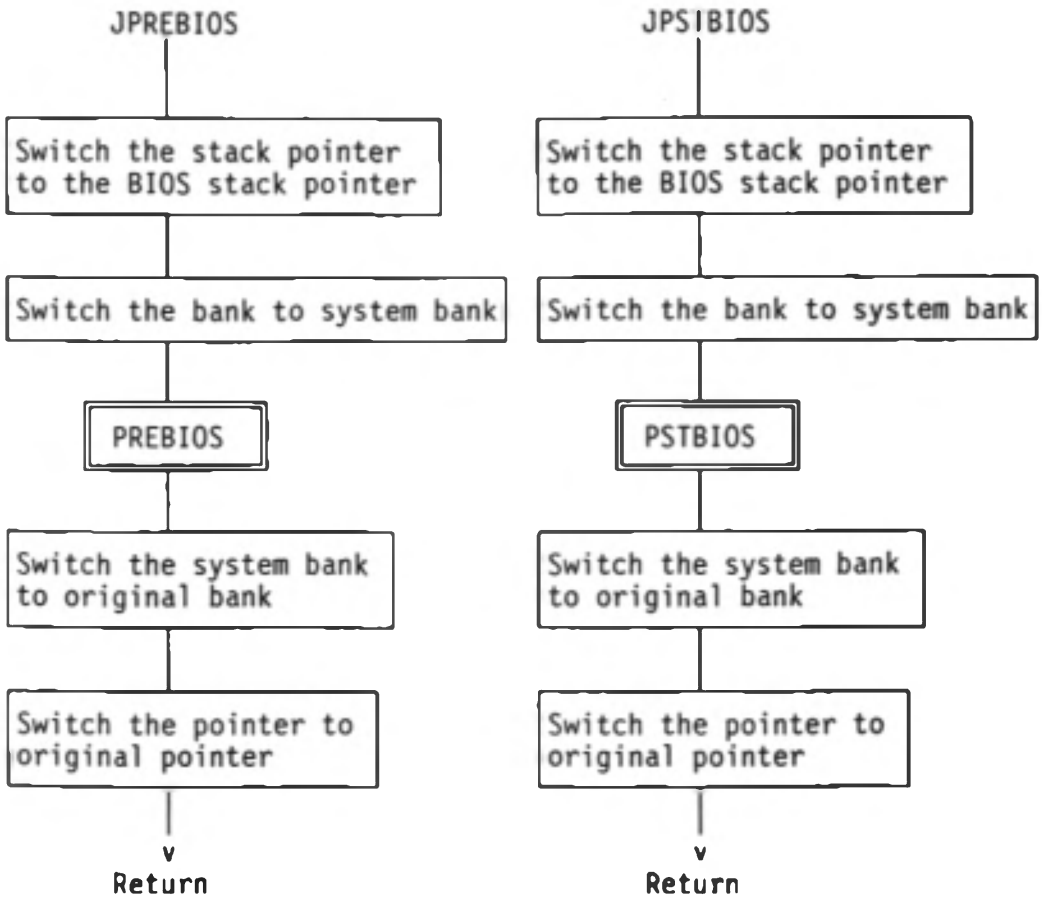


Fig 10.17 Processing flow of JPREBIOS and JPSTBIOS

[Function]
JSCALLX calls a subroutine that resides in the specified bank.

[Entry parameters]
DISBNK (F41BH) = Call destination bank information
IX = Call destination address

[Return parameters]
None. (This depends on the return parameters of the called subroutine.)

[Explanation]

(1) This routine has the same function as that of BIOS CALLX except the following points:

1 SCALLX can also be used by the routine called by BIOS CALLX or JCALLXX.

2 SCALLX can be executed without switching the stack pointer.

(2) The contents of all parameters set by the user are saved when this routine is called and restored when control is returned to the calling program.

(3) Because the stack pointer area that has been used when this routine is called is used as it is, the stack pointer area must be allocated in a location that can be referenced even if the current bank is switched to the bank where the calling routine resides.

(4) This routine enters EI status after returning control.

[Function]
JSELBNK switches the current bank to the specified bank.

[Entry parameters]
C = Information about the bank to replace

[Return parameters]
C = Information about the bank to be replaced

[Explanation]

(1) The contents of all registers except for C are saved.

(2) The bank information is the same as that of BIOS MEMORY.

(3) Parameter errors are not checked.

(4) Because the stack pointer area that has been used when this routine is called is used as it is, the stack pointer area must be allocated in a location that can be referenced even if the current bank is switched to another bank.

(5) This routine enters EI status after returning control.

[Function]
JLDAXX reads one byte from the specified address in the specified bank.

[Entry parameters]
C = Information about the bank containing the data to be read
HL = Address of the data to be read

[Return parameters]
A = Read data

[Explanation]

(1) The contents of all registers except for A are saved.

(2) The function and parameters of this routine are the same as those of BIOS LOADX except that this routine can be executed without switching the stack pointer.

(3) Parameter errors are not checked.

(4) Because the stack pointer area that has been used when this routine is called is used as it is, the stack pointer area must be allocated in a location that can be referenced even if the current bank is switched to the bank from which data is to be read.

(5) This routine enters EI status after returning control.

[Function]
JSTAXX writes one-byte data into the specified bank.

[Entry parameters]
A = Data to be written
C = Information about the bank into which data is to be written
HL = Write address

[Return parameters]
None

[Explanation]

(1) The contents of all registers are saved.

(2) This routine has the same function as that of BIOS STORX except that this routine can be executed without switching the stack pointer.

(3) Because the stack pointer area that has been used when this routine is called is used as it is, the stack pointer area must be allocated in a location that can be referenced even if the current bank is switched to the bank into which data is to be written.

(4) This routine enters EI status after returning control.

[Function]

JLDIRXX transfers the specified bank data to another bank.

[Entry parameters]

SRCBNK (F41DH) = Transfer source bank information
DISBNK (F41BH) = Transfer destination bank information
HL = Top address of the transfer data
DE = Transfer data storage start address
BC = Number of transfer data bytes

[Return parameters]

HL = HL + BC
DE = DE + BC
BC = 0000H
(Same as LDIR instruction)

[Explanation]

(1) This routine has the same function as that of BIOS LDIRX except that this routine can be executed without switching the stack pointer.

(2) Parameter errors are not checked.

(3) Because the stack pointer area that has been used when this routine is called is used as it is, the stack pointer area must be allocated in a location that can be referenced even if the current bank is switched to the bank from which data is to be copied.

(4) This routine enters EI status after returning control.

[Function]
JJUMPXX jumps to the specified bank.

[Entry parameters]
DISBNK (F41BH) = Jump destination bank information
IX = Jump destination address

[Return parameter]
None

[Explanation]

(1) This routine has the same function as that of BIOS JUMPX except that this routine can be executed without switching the stack pointer.

(2) Control is passed to the jump destination routine with all register contents saved.

(3) Error check is not performed.

(4) Because the stack pointer area that has been used when this routine is called is used as it is, the stack pointer area must be allocated in a location that can be referenced even if the current bank is switched to the bank to which this routine is to jump.

(5) This routine enters EI status after returning control.

[Function]
JCALLXX calls a subroutine that resides in the specified bank.

[Entry parameters]
DISBNK (F41BH) = Call destination bank information
IX = Call destination address

[Return parameters]
None. (This depends on the return parameters of the called subroutine.)

[Explanation]

(1) This routine has the same function as that of BIOS CALLX except that this routine can be executed without switching the stack pointer.

(2) The contents of all parameters set by the user are saved when this routine is called and restored when control is returned to the calling program.

(3) Because the stack pointer area that has been used when this routine is called is used as it is, the stack pointer area must be allocated in a location that can be referenced even if the current bank is switched to the bank from which a subroutine is to be called.

(4) This routine enters EI status after returning control.

[Function]
JINTOPR sets or resets the interrupt mask.

[Entry parameters]
B = Interrupt mask data (IER)
C = 7508 CPU or EXT interrupt mask data

[Return parameters]
BC = Old interrupt mask status

[Explanation]

(1) This routine has the same function as that of BIOS MASKI except that this routine can be executed without switching the stack pointer.

(2) The stack pointer area must be allocated in address 8000H or later.

[Function]

JSDBGIOX and JRDBGIOX each supports communications with the development cartridge. JSDBGIOX and JRDBGIOX each has the following 10 functions according to the B register contents.

B = 00H: Opens the development cartridge (OPEN).
= 01H: Checks the receive buffer status (INSTS).
= 02H: Checks whether sending is possible (OUTST).
= 03H: Receives one-byte data from the receive buffer (GET).
= 04H: Sends one-byte data (PUT).
= 05H: Switch the cartridge I/F status (SWITCH).
= 06H: Resets the development cartridge (DRSINT).
= 08H: Checks the development cartridge mount status (RDSTS).
= 09H: Changes the user cartridge mode (MODECHG).
= 0AH: Initializes the receive buffer (INITBUF).

The details on each function are explained later.

[Explanation]

(1) JSDBGIOX and JRDBGIOX each controls the RS-232C port of the development cartridge.

(2) The difference between the functions of JSDBGIOX and those of JRDBGIOX lies in that JSDBGIOX switches the current stack pointer to the BIOS stack pointer but JRDBGIOX uses the current stack pointer without switching it. Therefore, the stack pointer area must be allocated in RAM address 8000H or later when calling JRDBGIOX. (The stack pointer area must be allocated in address E000H or later when calling JSDBGIOX from the application ROM.)

(3) Because neither JSDBGIOX nor JRDBGIOX has a close function, perform close processing in the following procedure.

1 Set the current mode to ordinary mode by using the SWITCH function.
2 Initialize the receive buffer by using the INITBUF function.

JSDBGIOX FFB1H

(OPEN)

JRDBGIOX FFB4H

[Function]

JSDBGIOX and JRDBGIOX each opens the development cartridge.

[Entry parameters]

B = 00H (function number)
HL = Receive buffer top address

[Return parameters]

A = 00H: Normal termination
  = 01H: Another development cartridge has already been mounted.
  = FFH: No development cartridge has been mounted.

[Explanation]

(1) This function performs the following operations:

1 Sets the receive buffer.

2 Changes the cartridge mode to debugger mode.

3 Resets the debugger.

4 Enables interruption from the cartridge I/F.

(2) If the cartridge has already been opened, this function does not reset the debugger. However, the receive buffer is set again.

(3) If the development cartridge has not been mounted, this function performs no operation.

(4) The receive buffer must be allocated in RAM address 8000H or later. The receive buffer size is automatically set to 256 bytes.

(5) This function sets the following flag according to the development cartridge mount status:

DBGFLG (F203H): 00H: Not mounted
                01H: Mounted

[Function]

This function checks the receive buffer status.

[Entry parameters]

B = 01H: Function number

[Return parameters]

A = 00H: The receive buffer contains no data.

= FFH: The receive buffer contains data.

[Explanation]

(1) This function checks whether any data has been received in the receive buffer and sets the information in the A register.

(2) Note that this function does not change the cartridge mode.

[Function]
This function checks whether data sending is possible.

[Entry parameters]
B = 02H: Function number

[Return parameters]
A = 00H: Sending possible
= FFH: Sending impossible

[Explanation]

(1) This function checks the send buffer status and set the status in the A register.

(2) If the cartridge mode is not debugger mode, this function forcibly changes the mode to debugger mode and checks the send buffer status.

JSDBGIOX (GET) **FFB1H**
JRDBGIOX **FFB4H**

[Function]
This function fetches one-byte data from the receive buffer.

[Entry parameters]
B = 03H: Function number

[Return parameters]
A = Received data

[Explanation]

(1) If the receive buffer contains no data, the routine waits until data is received.

(2) Because this function does not change the cartridge mode, be sure to check the current mode and, if the current mode is not debugger mode, change it to debugger mode. (See Item "SWITCH".)

[Function]
This function sends one-byte data.

[Entry parameters]
B = 04H: Function number

[Return parameters]
None

[Explanation]

(1) If sending is not possible, this routine waits until sending is made possible.

(2) If the cartridge mode is not debugger mode, this function changes the current mode to debugger mode and sends data.

JSDBGIOX (SWITCH) FFB1H

JRDBGIOX FFB4H

[Function]

This function switches the cartridge I/F status.

[Entry parameters]

B = 05H: Function number

A = 00H: Changes the current mode to debugger mode.

= 01H: Close debugger

= FFH: Changes the current mode to ordinary mode.

[Return parameters]

None

[Explanation]

This function changes the current mode to debugger or ordinary mode.

To access the cartridge operating in ordinary mode after OPEN, OUTST, or PUT execution, set FFH in the A register and execute this function.

JSDBGIOX FFB1H
(DRSINT)
JRDBGIOX FFB4H

[Function]
This function resets the development cartridge.

[Entry parameters]
B = 06H: Function number

[Return parameters]
None

[Explanation]

(1) This function resets the development cartridge.

(2) This function must be executed after a communication error occurred.

(3) Check that the current mode is debugger mode before executing this function.

[Function]
This function checks the development cartridge mount status.

[Entry parameters]
B = 08H: Function number

[Return parameters]
CY = 1: Development cartridge mounted
= 0: Development cartridge not mounted

[Explanation]

(1) This function checks the development cartridge mount status.

(2) This function is also effective even if the OPEN function has not been executed.

[Function]
This function changes the user cartridge mode.

[Entry parameters]
B = 09H: Function number
A = Cartridge mode

[Return parameters]
None

[Explanation]

(1) This function changes the mode of the cartridge device connected to the development cartridge.

(2) This function is also effective even if the development cartridge has not been mounted.

(3) The mode values that can be specified in the A register are as follows.

A register = 00H: HS mode
= 01H: IO mode
= 02H: DB mode
= 03H: OT mode

(4) After this function is executed, the mode is changed to ordinary mode.

(5) This function can be executed even if the cartridge has not been opened.

JSDBGIOX (INITBUF) FFB1H

JRDBGIOX FFB4H

[Function]
This function initializes the receive buffer.

[Entry parameters]
B = 0AH: Function number

[Return parameters]
None

[Explanation]

This function actually resets the buffer pointer to the status set when the development cartridge was opened.

10.3.4 OS ROM jump table (I)

16 system routines have been registered in OS ROM jump table (I).

This section explains the specifications of each system routine registered in OS ROM jump table (I).

[Function]

BDOSTABL indicates the top address of BDOS function on vector that resides in OS ROM.

[Explanation]

(1) This jump vector is used to directly reference a BDOS function that resides in OS ROM.

(2) Because each BDOS function that resides in OS ROM neither changes the stack pointer nor performs error processing, special attention must be paid when using it.

```
BDOSTABL -> +-----------+
            | DW FUNC0  |
            +-----------+
     +02H   | DW FUNC1  |
            +-----------+
     +04H   | DW FUNC2  |
            +-----------+
     +06H   | DW FUNC3  |
            +-----------+
       |    |     |     |
       |    |     |     |
       |    |     V     |
            +-----------+
     +50H   | DW FUNC40 |
            +-----------+
```

[Function]

BIOSTABL indicates the top address of the BIOS jump table that resides in OS ROM.

[Explanation]

(1) This jump address is used to directly use a BIOS function that resides in OS ROM.

(2) Because each BIOS function that resides in OS ROM neither changes the stack pointer nor executes PREBIOS and PSTBIOS, special attention must be paid when using it.

```
BIOSTABL -> | JP BOOT    |
            |            |
      +03H  | JP WBOOT   |
            |            |
      +06H  | JP CONST   |
            |            |
      +09H  | JP CONIN   |
            |            |
        |   |     |      |
        |   |     |      |
        |   |     V      |
      +A5H  | JP INFORM  |
            |            |
```

[Function]
SETERR changes the BDOS error processing vector to a special vector.

[Entry parameters]
None

[Return parameters]
None

[Explanation]

(1) If this routine is called, this routine updates the BDOS error processing vector so that, even if a disk access error occurred, no error message will be displayed and return parameters will be passed.

(2) This routine is used when a file access error is processed by an application program.

(3) There are following types of BDOS errors.

1 Bad Sector
Data input/output for the disk is abnormal.
2 Bad Select
An unexisting drive or a drive being in unready state has been selected.
3 R/O disk
An attempt was made to write data to a read-only disk.
4 R/O file
An attempt was made to write data to a read-only file.

(4) After SETERR execution, an error code corresponding to the BDOS error is set in registers A and H.

Register name / Error type	A	H
BDOS processing normal termination	BDOS return code	00H
Bad Sector	FFH	01H
Bad select	FFH	02H
R/O Disk	FFH	03H
R/O File	FFH	04H

(5) When 00H is set in the H register, a value corresponding to the CP/M return information is set in the A register. See Section 5.3 for the SETERR usage procedure.

[Function]
RSTERR initializes the BDOS error processing vector.

[Entry parameters]
None

[Return parameters]
None

[Explanation]

(1) If this routine is called, this routine initializes the BDOS error processing vector and, if a disk error occurred after that, outputs a corresponding error message.

(2) If WBOOT is executed, the BDOS error processing vector is also initialized.

[Function]
GOBACK performs BDOS termination processing.

[Entry parameters]
None

[Return parameters]
None

[Explanation]

(1) GOBACK performs BDOS termination processing.

(2) After a BDOS function is executed, this routine performs the following operations as termination processing:

1 Restores the current disk contents.
2 Sets the return information.

[Function]
BIOSJTLD generates the RBIOS1 jump table.

[Entry parameters]
None

[Return parameters]
None

[Explanation]

(1) BIOSJTLD is used to regenerate the RBIOS1 jump table after the RAM disk or user BIOS area size is changed.

(2) According to BI1LAD (F007H), this routine generates the jump table to be linked to RBIOS2 (resident area).

[Function]
SETRAMAD sets the top addresses of the user BIOS area, RBDOS1, and RBIOS1 according to the sizes of the RAM disk and user BIOS area.

[Entry parameters]
SIZRAM (F00BH) = RAM disk standard RAM area size
USERBIOS (F00CH) = User BIOS area size

[Return parameters]
TOPRAM (F05CH) = user BIOS area top address
BI1LAD (F007H) = RBIOS1 top address
BDSLAD (F005H) = RBDOS1 top address

[Explanation]

SETRAMAD is used to set the top addresses of RBDOS1, RBIOS1, and user BIOS area after the RAM disk or user BIOS area size is changed.

[Related routines]

BIOSJTLD (0013H)
CHGRAMD (0019H)

[Function]

CHGRAMD modifies the RAM disk size.

[Entry parameters]

A = function
  = 0: Modifies only the size.
  = 1: Modifies the size and formats the disk.
  = 2: Increase the size and relocate the disk contents.
B = extended RAM size (in units of 32 Kbytes)
C = new RAM disk size (in units of Kbytes)

[Return parameters]

CY = error information
   = 0: Normal termination
   = 1: Error occurred

[Explanation]

(1) This function is used to modify the size of the RAM disk or user BIOS area.

(2) Although an arbitrary extended RAM size can be specified in units of 32 Kbytes in the B register, the size to be specified must not exceed the total size of the actually-mounted extended RAM.
If the value specified in the B register is less than the actually-mounted extended RAM size, no access is made by OS for the extended RAM area that exceeds the B register value. Therefore, this area can be arbitrarily used by the user.
The user can know the size of the actually-mounted extended RAM by referencing the contents of RAM_SET (F069H) in the system area.

(3) Relocating RAM disk contents
When the RAM disk size is increased, the RAM disk contents can be relocated (without being destroyed) by calling CHGRAMD with A register = 2, thus the sizes of the areas allocated in the RAM disk can be modified. However, the extended RAM area size cannot be modified.

(4) Modifying user BIOS area size
CHGRAMD can also modify the user BIOS area size. See Section 4.6 for details.

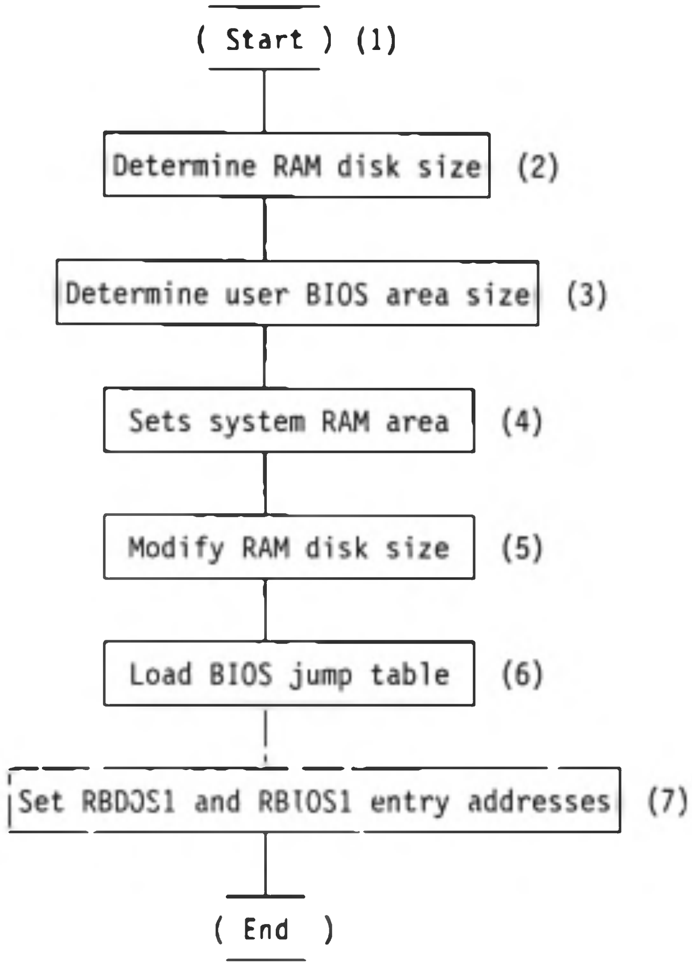


Fig 10.18 Modifying RAM disk size

Step : Explanation

(1) Determining RAM disk size
The user can know the current RAM disk size by referencing the following area contents.

Address : Variable name (Number of bytes)

F008H : SIZRAM (1)
RAM disk standard RAM area size (in units of Kbytes)

F060H : QT_RAM_IN (1)
Total RAM disk size (in units of Kbytes)

F068H : RAMD_SIZE (1)
Size of the extended RAM area used as the RAM disk area (in units of 32 Kbytes)

F069H : RAM_SET (1)
Total extended RAM size (in units of 32 Kbytes)

Determine new SIZRAM, QT_RAM_IN, and RAMD_SIZE values from the above values.

(2) Determining user BIOS area size
The user can know the current user BIOS area size from the contents of system area USERBIOS (F00CH). The unit is 256 bytes (one page).

(3) Setting system RAM area
Set the SIZRAM and USERBIOS values determined in Steps 1 and 2 and call SETRAMAD (0016H).
In this case, the following condition must be satisfied:

(SIZRAM) - (USERBIOS) $\leq$ 39.5 Kbytes (EHT-10) or 40.0 Kbytes (EHT-10/2)

(4) Modifying RAM disk size
Modify the RAM disk size according to the parameter values determined in Steps (1) and (2) .

C: New QT_RAM_IN value
B: New RAMD_SIZE value
A: 00H, 01H, or 02H

Call CHGRAMD by using these parameters.

(5) Loading BIOS jump table
Call BIOSJTLD (0013H).

(6) Set RBDOS1 and RBIOS1 entry addresses.
- BDSLAD (F005H) - 6, - 7 ---> (0006H, 0007H)
- BIILAD (F007H) - 3, - 4 ---> (0001H, 0002H)

[Function]
FMTRAMD formats the RAM disk.

[Entry parameters]
None

[Return parameters]
None

[Explanation]

If this function is executed, the entire RAM disk is formatted, erasing all files contained.

EPSPSND 001FH

[Function]

EPSPSND sends EPSP data from SIO.

[Entry parameters]

HL = EPSP send packet top address
A = Send mode
LSB = 0: Performs only send operation.
= 1: Performs both send and receive operations.

[Return parameters]

CY = return information
= 0: Processing completed
= 1: Processing suspended

A = return code
= 00H: Normal termination
= 61H: Device unconnected
= 62H: Communication error
= 63H: Time over
= 64H: Function key F6H

[Explanation]

(1) EPSPSND sends the packet data indicated by the HL parameter to a disk drive.

(2) If both send and receive operations are specified in the A register, the data received from the disk drive is stored in the area starting from the address indicated by the HL parameter.

(3) The packet data is transferred according to the EPSP (EPSON Serial Communication Protocol) protocol.

(4) The packet format is shown below. See Section 6.6.4 for details on the function.

FMT
DID
SID
FNC
SIZ
Text data

FMT: Head block format
00H: Indicates data sent from EHT-10/EHT-10/2
01H: Indicates data sent from the FDD

DID: ID of the destination device
SID: ID of the source device

Device IDs are as follows:

EHT-10/EHT-10/2: 23H
FDD (D: or E:): 31H

Therefore, the following is assumed:
When data is sent from EHT-10/EHT-10/2 to the FDD
DID=31H and SID=23H

When data is sent from the FDD to EHT-10/EHT-10/2
DID=23H and SID=31H

(5) Because the bank is not taken into account in send operation, the packet top address (HL register of the entry parameter) must be 8000H or later.

[Function]
EPSPRCV receives EPSP data from SIO.

[Entry parameters]
HL = EPSP receive packet top address

[Return parameters]
A = return code
= 00H: Normal termination
= 61H: Device unconnected
= 62H: Communication error
= 63H: Time over
= 64H: Function key F6H
B = received packet information (effective when A=00H)
= 00H: Data with a header was received.
= 01H: Data without a header was received.
(HL) = received data

[Explanation]

(1) EPSPRCV stores the data received from the disk drive in the packet indicated by the HL parameter.

(2) The receive packet format is the same as that of EPSPSND. See Section 6.6.4 for details.

(3) Because the bank is not taken into account in receive operation, the packet top address (HL register of the entry parameter) must be 8000H or later.

[Function]
MELODY sounds the specified melody.

[Entry parameters]
HL = Melody table top address
C = repeat count

[Return parameters]
None

[Explanation]

(1) MELODY sounds the specified melody the specified number of times.

(2) The melody table configuration is as follows.

Offset	Contents
(HL) ->	Tone length
+1	Tone
+2	Tone length
+3	Tone
⋮	⋮
+2n	00H

The tone length must be specified in units of 100 milliseconds.
The tone must be specified in the same way as for BIOS BEEP.

Set 00H as an end mark at the end of the table.

(3) The melody table must be allocated in RAM address 8000H or later.

(4) This routine runs using the BIOS BEEP function.

[Function]
WRT7508 sends a command to the slave 7508 CPU.

[Entry parameters]
C = command data to be sent to the slave 7508 CPU

[Return parameters]
None

[Explanation]

(1) The contents of all registers other than A are saved.

(2) WRT7508 sends a command to the slave 7508 CPU.

(3) See Section 7.6.3 for details on the commands for 7508 CPU.

(4) Before sending a command to the 7508 CPU, the interrupt by the 7508 CPU must be disabled.
BIOS MASKI is used to control interruption

[Function]
CMD7508 sends a command to the slave 7508 CPU and receives its response data.

[Entry parameters]
C = command data to be sent to the slave 7508 CPU

[Return parameters]
A = response data sent from the slave 7508 CPU

[Explanation]

(1) The contents of all registers other than A are saved.

(2) CMD7508 sends the specified command to the slave 7508 CPU, receives one-byte response data, and sets the response data in the A register.

(3) See Section 7.6.3 for details on the commands for 7508 CPU.

(4) Before using CMD7508, the interrupt by the 7508 CPU must be disabled. BIOS MASKI is used to control interruption.

(5) If multiple data items are returned from the 7508 CPU, the remaining data items must be read directly from the port.

[Function]
ENINTVL enables 1 ms or 8 ms timer interrupt.

[Entry parameters]
B = Specifies 1 ms or 8 ms interrupt.
  = 00H: 8 ms interrupt
  = 01H: 1 ms interrupt
C = 04H: Indicates that the interrupt is to be used by the user.

[Return parameters]
None

[Explanation]

See Section 8.8.2 for details.

[Function]
DISINTVL disables 1 ms and 8 ms timer interrupts.

[Entry parameters]
C = 04H: Indicates that using timer interrupt by the user is to be terminated.

[Return parameters]
None

[Explanation]

See Section 8.8.2 for details.

10.3.5 OS ROM jump table (II)

OS ROM jump table (II) is allocated starting from address 6000H in bank 2#1. The user can use only the following two routines through this table.

(1) ICDKMNT (IC card mount processing)
(2) ICDFMT (IC card formatting)

This section explains the specifications of each above routine.

[Function]
ICDKMNT checks whether an IC card has been mounted.

[Entry parameters]
None

[Return parameters]
CY = return information
= 0: Normal termination
= 1: Abnormal termination
A = return code (effective when CY=1)
= 01H: No IC card has been mounted.
= 02H: An error was detected during communications with the IC card.

[Explanation]

(1) ICDKMNT checks whether an IC card has been mounted and, if an IC card has been mounted, checks its capacity and sets the information in DPB (Disk Parameter Block).

(2) If ICDKMNT is called while power is being supplied to the IC card (open state), it returns control without performing any operation. If ICDKMNT is called while power is not being supplied to the IC card (close state), it supplies the power and the clock signal to the IC card and releases the reset status.

[Function]
ICDFMT formats the IC card so that the IC card can be used as a CP/M disk.

[Entry parameters]
None

[Return parameters]
CY = return information
= 0: Normal termination
= 1: Abnormal termination
A = return code (effective when CY=1)
= 01H: No IC card has been mounted.
= 02H: IC card formatting error

[Explanation]

(1) ICDFMT formats the IC card by writing E5H in all data area of the IC card.

(2) ICDFMT is used by an application program for formatting an IC card. The operating system calls this routine through the IC card format processing displayed on the system menu screen.

CHAPTER 11 System Expansion

11.1 Overview

This chapter mainly describes I/O devices for system expansion.

The following I/O devices are covered in this chapter.

1. Serial communication protocol
2. IC card
3. Cartridges
4. Bar code

Refer to the following chapters while reading this chapter for I/O device control and system expansion.

1. Chapter 7 I/O Interface overview
2. Chapter 8 Interrupt processing
3. Chapter 10 System Hook and Jump Table

11.2.1 Overview

EHT-10/EHT-10/2 supports communication procedures with protocol, but protocol may differ according to the host computer type.
The OS is structured to use protocol for easy expansion. The communication procedure can be selected from the following three types.

(1) No sequence (Data is sent/received without specific protocol)

(2) Filink (Common communication procedure for EPSON computers)

(3) Expansion sequence (protocol for original application is added and data is sent/received)

These protocols are used in:

(1) System utility DLL, DL/UL

(2) TCAM of the BIOS

To expand the communication procedure, expansion must be performed at the two processing locations mentioned above.
To operate the expansion procedure, select one of the following.

(1) Modify the value of TCAMPRM (F1CDH) + 0 to 03H.

(2) Select "Extend" of the CONFIG DLL item.

Then,

(1) DLLHK1 call in DLL processing

(2) ULHK1 call in UL processing

(3) DLHK1 call in DL processing

(4) TEXHK1 call in BIOS TCAM processing

are executed in the system by parameter modification as mentioned above.

The bank can be specified via these hooks, the application expands the communication protocol by adding an expanded section utilizing the hook necessary for the expansion procedure. (See "APPENDIX 9 SAMPLE 32")

11.2.2 Expansion Procedure

(1) Loading of expansion program

The program for protocol expansion can be placed;
1 In user BIOS area.
2 On application ROM.
3 In RAM (TPA).

See "10.2 System Hook" for each procedure and notes.

(2) Rewrite of Hook

Rewrite the contents of the hook corresponding to the process to be expanded. The communication protocol hook is configured of 3 bytes as follows.

Bank information (1 byte) + address information (2 bytes).

See "10.2 System Hook" for further information.

(3) Communication protocol expansion specifications

The communication protocol hook does not jump to the expansion process by only a rewrite of the hook. To start the expansion process, protocol expansion must be specified.

To specify expansion, select either of the following.

1 Modify the value of TCAMPRM+0 to 03H.
2 Set the protocol of the CONFIG DLL item to "Extend".

(Example)

```
TCAMPRM   EQU  0F1CDH

          LD   A,03H    } expansion specification code
          LD   (TCAMPRM),A
```

11.2.3 Description of Hook

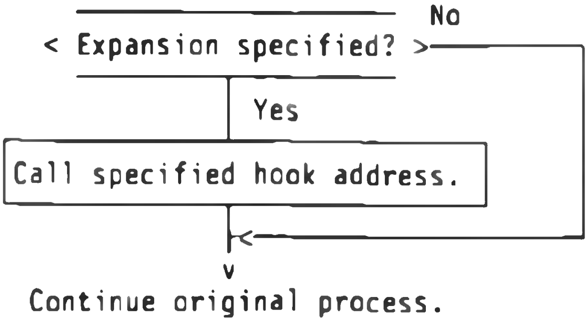
(1) Overview

The following 4 types of communication protocol hook are available.

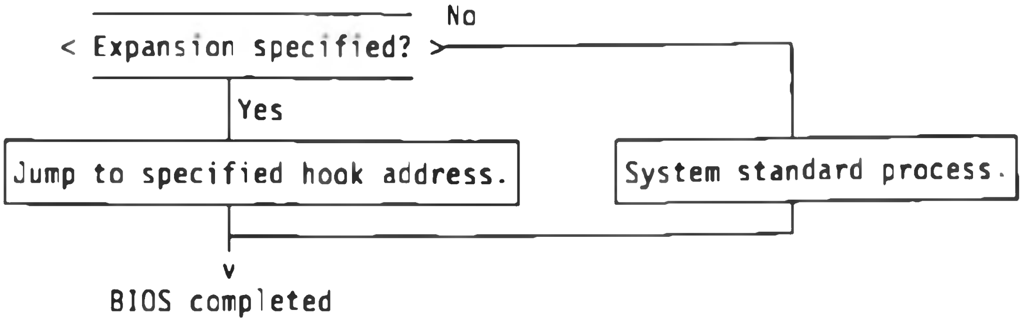
1 TEXHK1 (Expansion hook in BIOS TCAM)
2 DLLHK1 (Expansion hook in DLL process)
3 DLHK1 (Expansion hook in DL process)
4 ULHK1 (Expansion hook in UL process)

When using DLL or DL/UL process which is standard-support of the system, the hooks of (2) to (4) mentioned above must be used for expansion. When not used, expansion is not necessary, but correct operation of DLL and DL/UL is not guaranteed. (Interface may not match as BIOS TCAM is used in DLL and DL/UL.)

DLLHK1, DLHK1 and ULHK1 are types to call a specified hook address in the proper section of each process. The state of TEXHK1 stores the entry parameter of BIOS and uses it to jump to the specified hook address. (The processes after TEXHK1 are skipped.)



In case of DLLHK1, DLHK1, ULHK1.



In case of TEXHK1

(2) TEXHK1

1 Overview

TEXHK1 is used to expand communication protocol. This hook is included in the BIOS TCAM process and the following advantages are created by using this hook to expand the communication protocol.

* Communication with devices which have a different communication protocol is enabled.
* Protocol which expands the system utility (DLL,DL/UL) enables operation.
* Easy application development.

2 Hook location

Figure 11.1 shows the location of TEXHK1 during process.

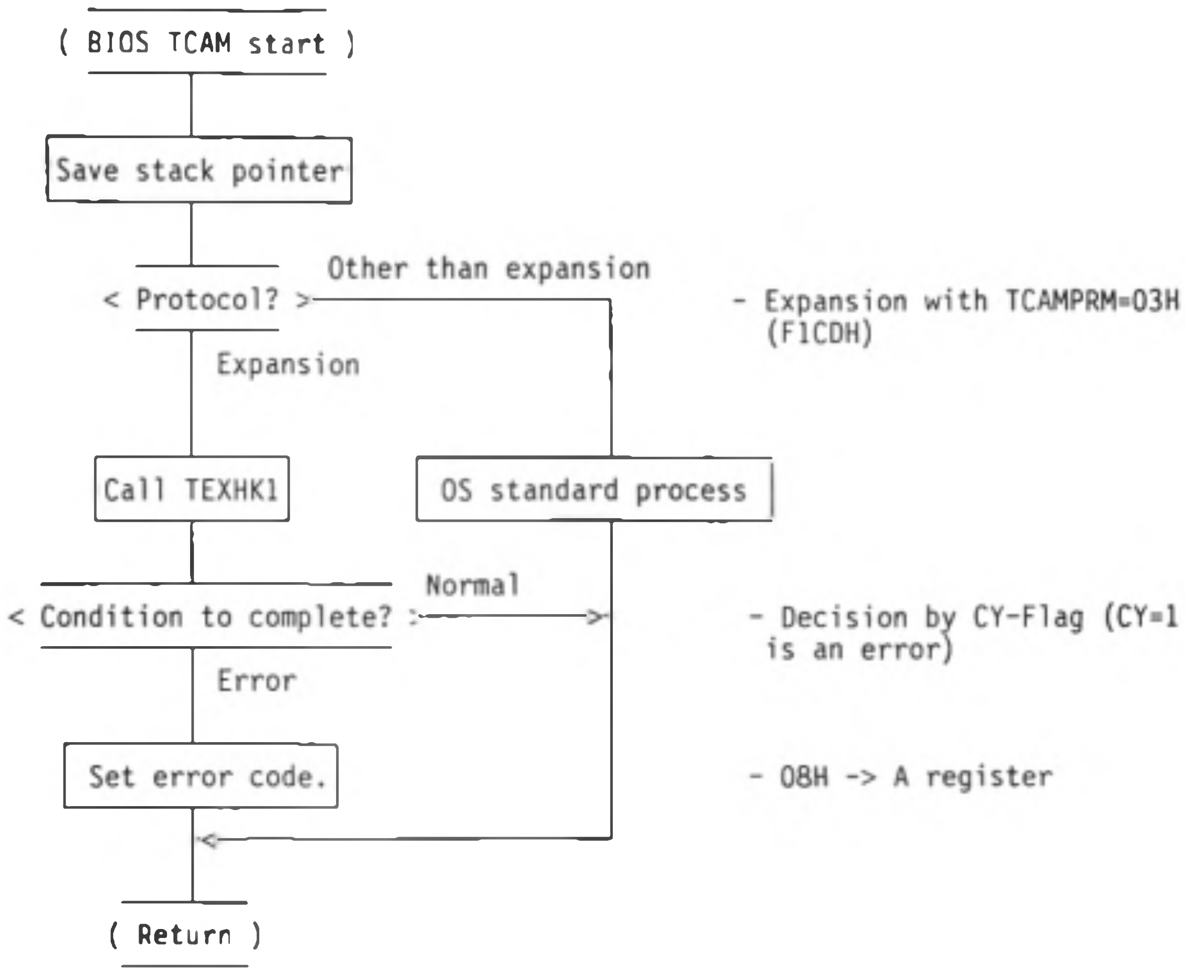


Fig. 11.1 Flow of BIOS TCAM

3 Description

(a) When TEXHK1 is called, TCAM entry parameter is stored excluding the flag register.

(b) When TEXHK1 is returned and the error process at OS is required to be invalid, process as follows.

* Request the location of the address returned from the hook by the value of the saved stack pointer.

* Rewrite the contents of the return address to an address with only a return instruction.

(Ex.)

```
TSPSAVE   EQU   0F6C3H
RETADR    EQU   0F000H
          LD    HL,(TSPSAVE)
          LD    DE,RETADR        - Address with RET instruction.
          DEC   HL
          LD    (HL),D           -- Rewrite the contents of the stack
          DEC   HL                  and omit OS error check process
          LD    (HL),E
```

(c) Input parameter at expansion process is as follows.

```
A register ... function specification
           A=01H : Connect
            =02H : Send
            =03H : Receive
            =04H : Disconnect
```

Other registers are different depending on the function. Further information is available in the item of each function.

(Note) When the addition of a function is necessary, add by the value of A register.

(d) Align return data of the expansion routine with the return data of the system. The system return data is as follows.

```
CY-Flag ... Indicates normal/abnormal end.
          CY=0 : Normal end
          CY=1 : Abnormal end
A register ... Indicates error data at abnormal end.
          A=01H : Parameter error
           =02H : Open
           =03H : Not open
           =04H : Forced end
           =05H : Received buffer overflow
           =06H : Timeout
           =07H : Protocol error
           =08H : Communication error
```

Other registers are different depending on the function, details are described later.

(Note) When the system returns by CY=1 at expansion, A register=08H is force set. Thus, when error data is required to remain in the application, the process of previous item (b) must be added.

(e) "Connect" function process

When entry becomes "A register=01H" for expansion routine, the loop is connected.

The expansion process uses the following data depending on requirements. (Details of each area is described in the work area item.)

```
Communication condition ... TCAMPRM(F1CDH - F1D3H)
Retry                   ... TDFLTCNT(F1D4H)
Timeout                 ... T1STTIME(F1D5H - F1D6H)
                            T2NDTIME(F1D7H - F1D8H)
```

The following items are to be processed when expansion is added.

```
<Parameter Check>
```

Send and receive process of Filink are different, the process is specified by the B register.

B register=00H : Send process
=01H : Receive process

When a distinction of send/receive data is required in the expansion process, refer to this data.

<Open Check>

To check whether it is open, if open, an error occurs.
See RSODEV to check whether it is open. If open, set value to RSODEV

RSODEV (F623H) =00H : RS-232C
=03H : Cartridge SIO
=FFH : Not open

<Communication Condition Set>

To set the bit rate, bit length, parity, stop bit and other status of the control line.
See "7.2 Serial Interface" for setting the communication condition.

(Note) Required processes for application is as follows.

1 Set value of 5 bytes of RS2CTLR1 (F19BH) to RS2SOUT (F19FH)
2. Set 00H to RSODEV (F623H)
3. Set 00H to SPMODE (F241H)
4. Call serial line switching process (SELSER) (entry parameter A=01H)

<Communication Interrupt Process>

The OS contains a standard receive interrupt process, but is used for BIOS RSIOX and cannot be used for application. Therefore, when receive interrupt for application is used, the interrupt vector must be rewritten or an interrupt expansion must be performed using the interrupt hook.
Interrupts used in applications must be enabled interrupts.
See "8.5 ART Interrupt" for interrupt process.

(Note) When a timer is necessary in communication process, 1 ms or 8 ms, 100 ms ,1 s are enabled (See "Chapter 8 Interrupt Process" for further information.)

<Connection With Host>

Loop connection with a host computer must be performed as a connect process.
Due to this, the expansion process does not only set serial communication, but also performs linkage with the host computer.
The utility (DLL,DL/UL) of the system mainly sends/receives data, a filename process is necessary for the first send/receive data. With the receive process, received filenames that follow must be set to TCAMAREA. With the send process, filename data is stored in TCAMAREA (F1CHDH), and data following is received. However, when operation of a utility is not necessary or filenames are not of concern, proper operation of data in TCAMAREA is only necessary.

(Note) When extension (9th to 11th byte of TCAMAREA) is "COM", DLL starts load to TPA, when "BAS", DLL starts as a BASIC program, other than these, DLL is stored in RAM disk by specified filename.
DL is stored in RAM disk by filename specified in TCAMAREA.
See MODEFG (F00AH) to distinguish whether DLL, DL/UL is being processed.

(f) "Send" function process

When entry is "A register=02H" in expansion routine, data send process is performed.

Input parameter at "Send":

BC register ... Number of bytes to be sent.
HL register ... Leading address of send data.

Expansion process sends data for the number of BC bytes from HL register.

<Open Check>

To check whether it has been opened, if not, an error (A=03H, CY=1) occurs.

<Serial Variance>

When IC cards are used at the same time, the serial line switches. Due to this, the serial line must be switched to the mode in present use, before "Send" process in expansion process. (See "7.2 Serial Interface") However, if other serials are not used between "Connect" and "Disconnect", this is not necessary.

<Send Data Process>

To send data for BC bytes from address indicated by HL to the host computer. At this time, data to be sent is assumed to be always in RAM.
The number of bytes to be sent by the system is always 128 bytes.

(Note) Since the location of send data may enter the end bank of the process routine, send data must be fetched with consideration for the bank.

<Send Data Completed>

When sending data completes normally, CY=0 is returned.

(g) "Receive" function process

When entry is "A register=03H" in expansion routine, data receive process is performed.

Input parameter at "Receive":
HL register .. Receive data stored at leading address. Expansion process stores the unaffected data received at addresses which follow the address indicated by HL, and return the number of bytes to BC.

<Open Check>

To check whether it has been opened, if not, an error (A=03H, CY=1) occurs.

<Serial Variance>

When IC cards are used at the same time, the serial line switches. Due to this, the serial line must be switched to the current mode before the "Receive" process in expansion process. However, if other serials are not used between "Connect" and "Disconnect", this is not necessary.

<Receive Data Process>

To store the data received at addresses which follow the address indicated by HL. Data storage is always performed in RAM; data is only unaffected data. The receive buffer of COMBUF (256 bytes) is used to receive control code.

(Note) Since storage of receive data is performed in RAM and may enter the last bank of the process routine, storage of receive data must be processed with consideration for the bank.

<Receive Data Completed>

When 1 block of receive data is completed, the number of receiving bytes is set to BC register and CY=0 is returned.

(Note 1) In the system (DLL,DL) process, the process is based on the following regulations.

* Completion of receive data is decided by BC=0. When receive data is completed, "Disconnect" function is called to complete the process.

* Receive process is performed in 1 record (128 byte) units. If it cannot be performed in 128 byte units, blocking process must be performed by the expansion process.

(Note 2) When expansion protocol operates DLL and DL, condition of (Note 1) mentioned above must be satisfied or the system process must be expanded using DLLHK1 or DLHK1.

(h) "Disconnect" function process

When entry becomes "A register=04H" in the expansion process routine, the loop is released.
The expansion process performs serial line close process after the process necessary for loop release from the host computer. The close processes to be performed at "Disconnect" in the expansion process are as follows.

<Close Process>

* Receive interrupt is disabled. When interrupts other than receive interrupt is used, it must be returned to the state prior to use.

* Set 00H to RSPSTS, RSINTST.

* Set FFH to SRMODE, RSODEV.

(Note) Close process must be always executed regardless of "Connect" function execution. ("Disconnect" is not error-returned.)

4 Remarks

(a) Development cartridge

The system forces communication through the development cartridge when the development cartridge is installed (in open state) at BIOS TCAM. Therefore, this process is not performed when TCAM expansion is processed, development with the development cartridge cannot be performed. However, as only DLL process out of TCAM relates to the development cartridge, the development cartridge operation is enabled by considering the system process to operate only at DLL process.

(b) Receive buffer

COMBUF (256 bytes) is used to receive protocol control code. This receive buffer is used to communicate at BIOS RSIOX and does not compete with other buffers during RSODEV check.

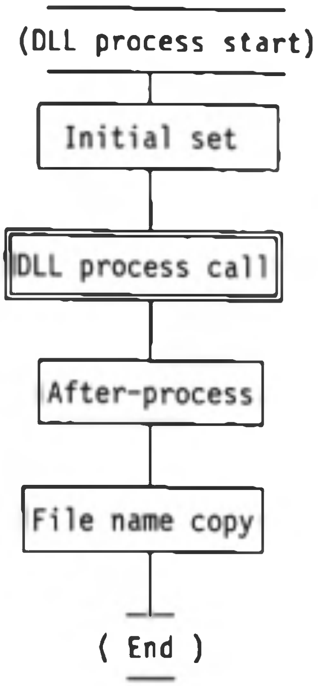
(3) DLLHK1

1 Overview

DLLHK1 is used to expand DLL (Down Line Loader) process. The DLL process loads the program from the host computer and starts it, when an application program does not reside in each drive of A:, B: or C:, it starts by turning on Restart. As DLL uses TCAM in BIOS, it is automatically expanded to TCAM expansion protocol along with TCAM expansion mentioned in the previous item. However, as DLL cannot operate well with TCAM expansion protocol, DLLHK1 is used for support.

2 Location of hook

The location of DLLHK1 during process is shown in Figure 11-4 to 11-6.



- Flag set during DLL process
- Cursor off

- Main process
- See Figure 11.3 for details

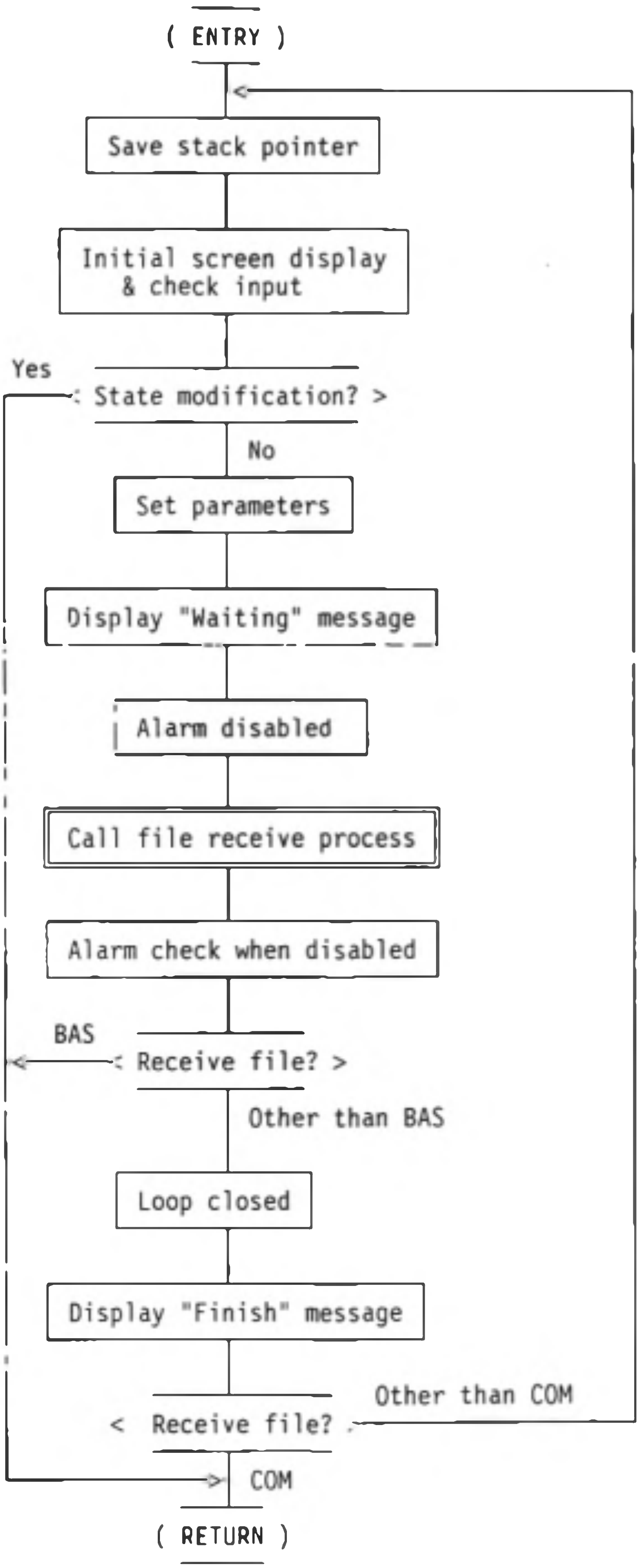
- Flag reset during DLL process
  (See note 1)

- Set file name for DLL to FILENAME using the value specified by DLLFILE
  (See note 2)
- ROM execution flag reset
  (Address with HL = FILENAME stored)

Fig 11.2 DLL Main Process

(Note 1) When the state of each type changes by system menu, it returns here.

(Note 2) Main process starts the received program, special consideration is not necessary in applications.



- Check whether state modification by system menu utility.

- Error process address setting when an error occurs.
- Set initial value to DLLFILE (DLLFILE = -1)

- See figure 11.4 for details.

- Alarm screen is displayed when alarm occurs during alarm disable.

- BAS file is received by DLLFILE = 1

- Use BIOS call TCAM disconnect

- COM file is received at DLLFILE = 0

Fig 11.3 DLL Process

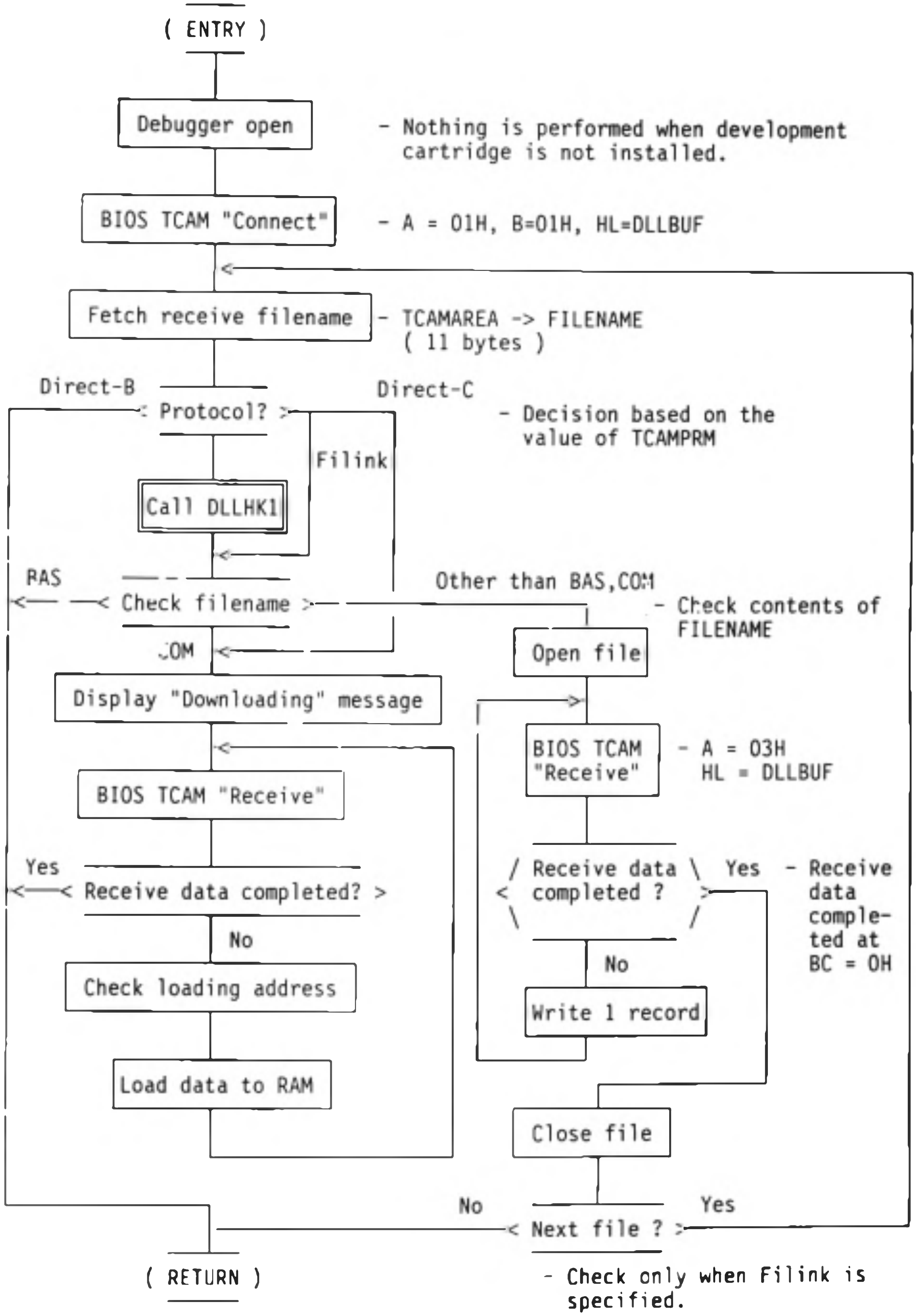


Fig 11.4 File Receive Process

3 Remarks

(a) DLL process uses BIOS TCAM to receive files. DLL process receives data in 1 record (128 byte) units. If the receive data process is not a 1 record unit, expansion using DLLHK1 is necessary.

(b) The conditions in which the main hook is not used at protocol expansion are as follows.

* File name must be stored in TCAMAREA at BIOS TCAM "Connect".

* Data must be received in 1 record units at BIOS TCAM "Receive".

* Receive data completion condition (BC=0) must be satisfied at BIOS TCAM "Receive".

* Loop cut process must be performed at BIOS TCAM "Disconnect".

* Normal/abnormal completion return parameter of BIOS TCAM must be satisfied.

(c) When an error occurs at the disk function of BIOS TCAM, BDOS, the process stops, loop cut and file close are processed.

(d) When BAS file is received, it goes through DLL process then gives control to BASIC. Basic receives data one record at a time storing them in RAM using BIOS TCAM "Receive".

(e) When a development cartridge is installed, receiving data is performed not from the RS-232C in the DLL process, but through the development cartridge. Thus, protocol expansion using a development cartridge with DLL process operation is impossible.

(4) DLHK1, ULHK1

1 Overview

DLHK1 and UKHK1 are used to expand DL (Down Loading) and UL (Up Loading) processes.
DL/UL process starts when the system menu DL/UL is specified. DL/UL process is to send/receive data to/from a file host computer for A drive (RAM disk).
As DL/UL process uses BIOS TCAM internally, it is expanded to TCAM expansion protocol automatically along with TCAM expansion mentioned in item (1).
However, TCAM expansion protocol may not operate DL/UL well, DLHK1/ULHK1 is used for support.

2 Location of hook

The location of DLHK1 and ULHK1 during processing is shown in Figure 11.5 to 11.7.

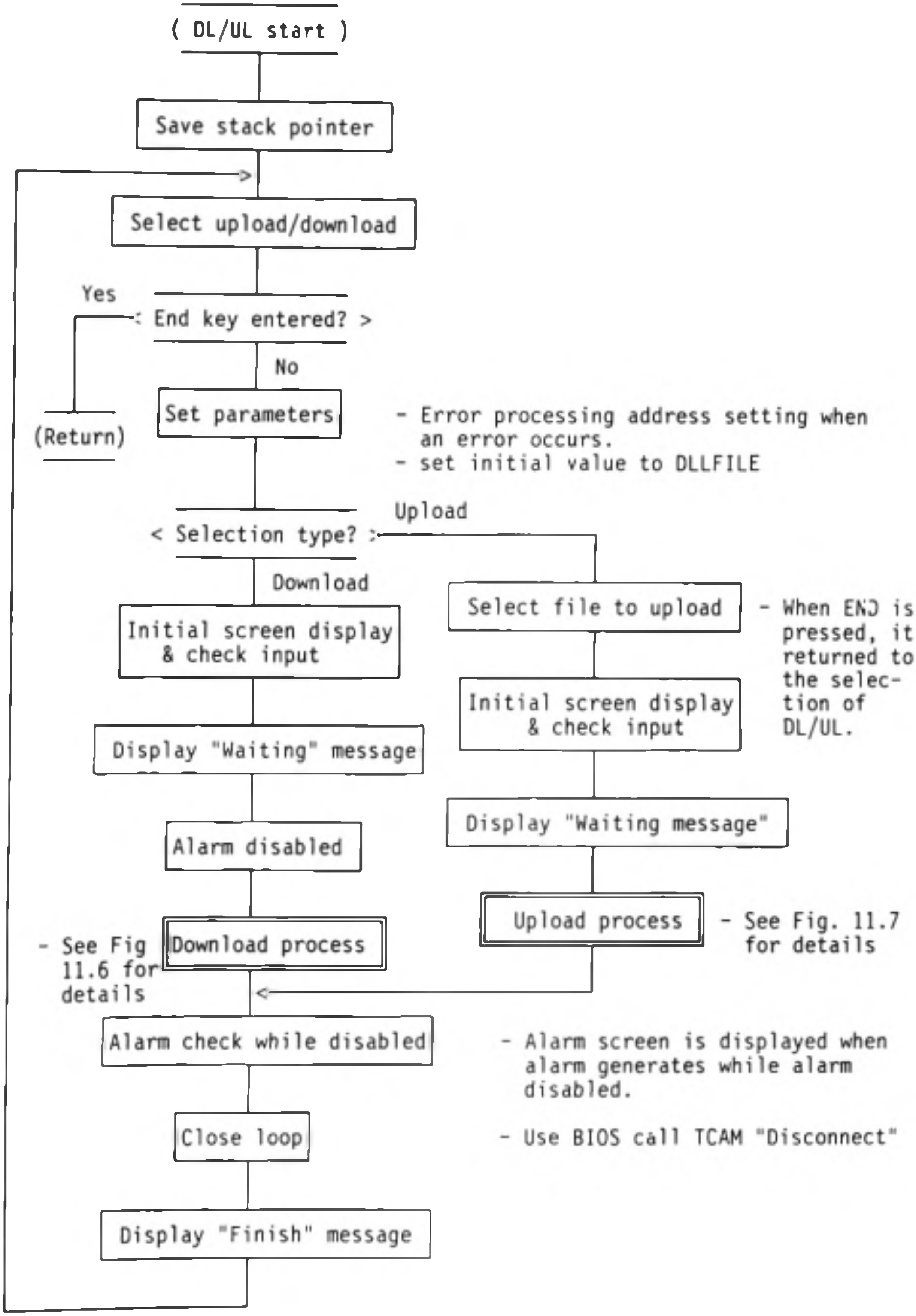


Fig 11.5 DL/UL Process

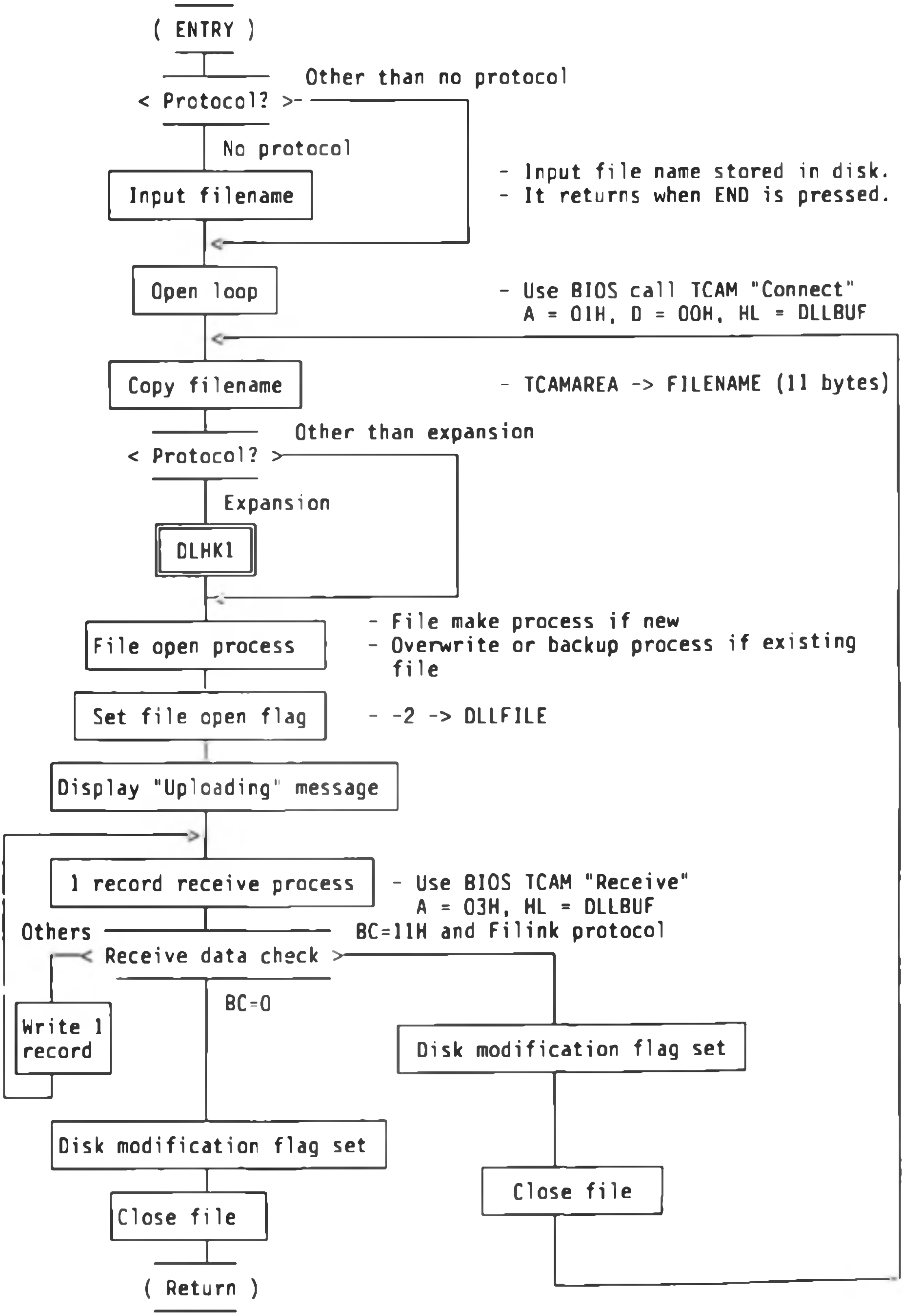


Fig 11.6 Download process

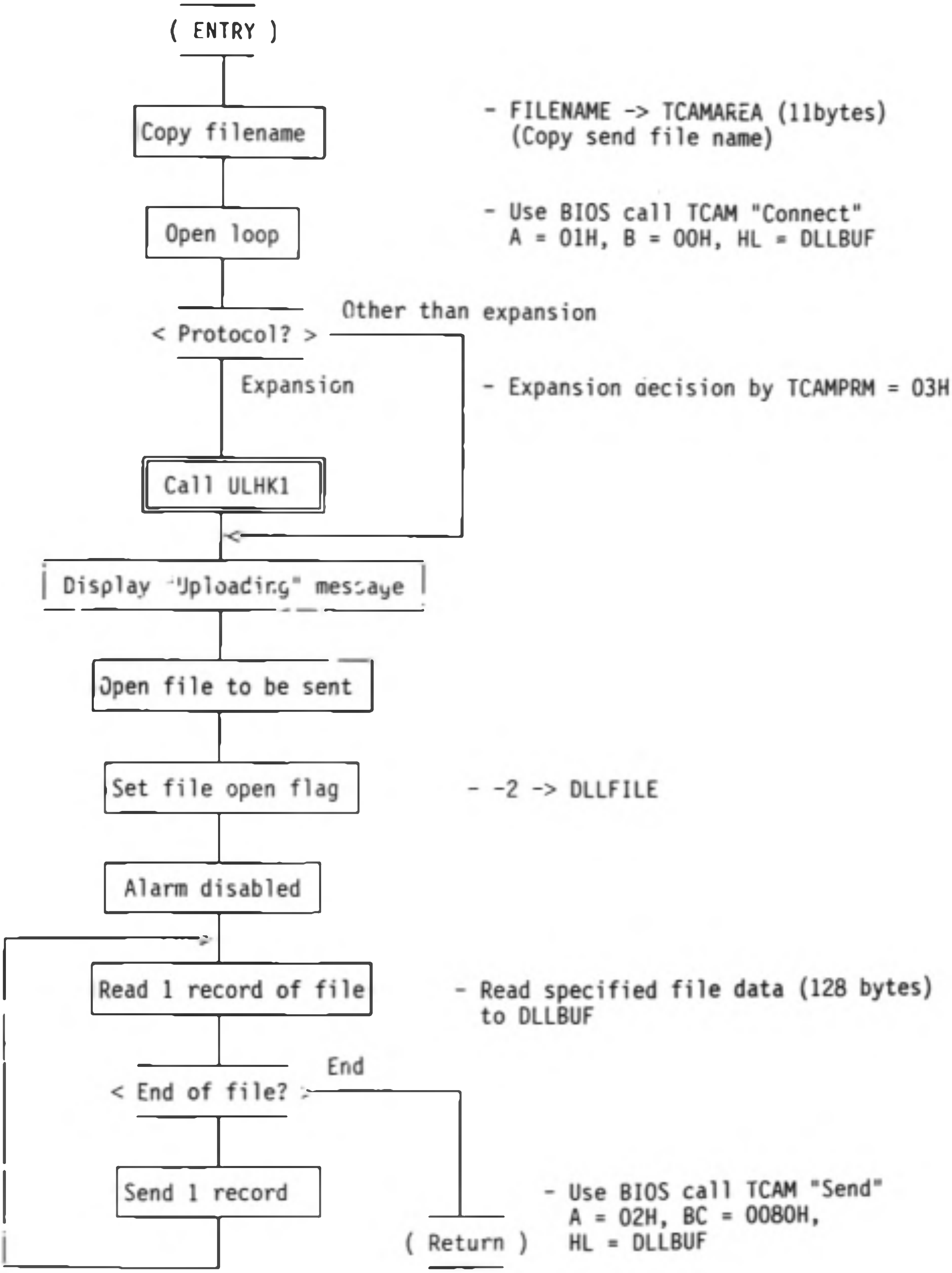


Fig 11.7 Upload Process

3 Remarks

(a) DL/UL process uses BIOS TCAM to send/receive data. DL/UL process sends/receives data in 1 record (128 byte) units. If send/receive process by expansion is not in 1 record units, expansion using DLHK1 or ULHK1 is necessary.

(b) The conditions in which the main hook is not used at protocol expansion are as follows.

* File name must be stored in TCAMAREA at BIOS TCAM "Connect" when data is received.
(Filename is stored in TCAMAREA at send data.)

* Data send in 1 record units at BIOS TCAM "Send" must be performed.

* Data receive in 1 record units at BIOS TCAM "Receive" must be performed.

* Receive data completion condition (BC=0) must be satisfied at BIOS TCAM "Receive".

* Loop cut process must be performed at BIOS TCAM "Disconnect".

* Normal/abnormal completion return parameter of BIOS TCAM must be satisfied.

(c) When an error occurs at the disk function of BIOS TCAM, BDOS, the process stops, loop cur and file close are processed.

(d) As the installed development cartridge does not effect DL/UL, special care is not necessary for applications.

11.2.4 Communication Relation Work Area Detail

Address : Variable name (Number of bytes)

F1CDH : TCAMPRM(7)

This area stores the initiation conditions used at TCAM open operation. The initiation conditions are set and modified by CONFIG but they can also be modified by an application program.

+0	Type
+1	Line
+2	Bit rate
+3	Character length
+4	Parity check
+5	Stop bit
+6	Special parameter

Type: Protocol type

=0: protocol direct-C

=1: protocol direct-B

=2: Filink (default)

=3: Extended protocol

Line: Serial line to be used for sending and receiving
=0: RS-232C (default)
=3: Cartridge I/F

The bit rate, character length, parity check, stop bit, and special parameter are same as that of BIOS RSIOX. The default values are 4800 Bps, 8 bits, Nonparity, and 2 stop bits.

(Note) When expansion protocol is specified by type without adding expansion protocol and this BIOS is executed, an error returns. See protocol expansion for details.

F1D4H : TDFLTCNT (1)

This area is used to specify the number of retry operations executed when the Filink protocol does not match with the host protocol. (The default is 3.)

F1D5H : T1STTIME (2)
F1D7H : T2NDTIME (2)

This area is used to specify the timer used for timeout during data receiving (unit: Seconds).
T1STTIME: Timer for the first data receive operation (30 seconds)
T2NDTIME: Timer for the second data receive operation (3 seconds)
In the first data receive operation, data up to file name is received when Filink protocol is used, or 1-byte data is received when protocol is not used (Direct-B or -C).

(Note) When expansion protocol is specified by type without adding expansion protocol and this BIOS is executed, an error returns. See protocol expansion for details.

F1D9H : T1STFLG (1)

Flag to identify "initial", "subsequent" when timeout time is determined.

=0 : "Initial"

=1 : "Subsequent"

= -1 : "For future receive"

F1D8H : DRCVCNT (2)

F1DDH : DRCVBUF (2)

Specifies the area for storing data at DLL, DL/UL.

DRCVCNT ... Size of data storage area. Default is 128 bytes.

DRCVBUF ... Leading address of data storage area. Default indicates DLLBUF.

F1DFH : DSKNUM (1)

Specifies the drive for file operation at DL/UL. Default is drive A. If the value of this area is "C" drive (=3) for example, drive for DL/UL becomes IC card and send/receive of IC card file is enabled.

F6C3H : TSPSAVE (2)

Area to save the stack pointer when BIOS TCAM starts.

F6C5H : TCNTDT (2)

Work area to check the number of times of retry when Filink protocol is specified.

F6C7H : TERRTIME (2)

Work area to check timeout time at BIOS TCAM.

F6C9H : TCAMAREA (12)

Work area to pass parameters of applications at BIOS TCAM. It has the following meaning at Filink protocol.

At file send ... Connect process is performed after filename (11 bytes) are stored in this area.

At file receive ... When connect process is performed, filename (11 bytes) received in this area is stored and F6C9H returns.

At no sequence:
F6C9H is not used at TCAM but is used as the area for filename memory in DL/UL or DLL process. (To equalize the interface with that at Filink.)

At expansion protocol:
It is desirable to use an identical interface as Filink.

F6D5H : TCAMIF (1)

Area to store whether connection is performed in send/receive mode when Filink protocol is specified at BIOS TCAM.

=0 : Send mode

=1 : Receive mode

F6D6H : DLLSVSP (2)

Work area to save the stack pointer when DLL process starts.

F6D8H : LOADAD (2)

Work area which stores the loading address of data which follows when loading a file to DLL process TPA area.

F6DAH : DLLRVS (1)

Flag to indicate reverse status of the message "Down Loading", "Up Loading" at DLL and DL/UL processes.

F6DBH : DLLFILE (1)

Work area to store file types received at DLL process.

=0 : COM file receive
=1 : BAS file receive
= -1 : For future receive
= -2 : File other than COM/BAS receive (While open)

F6DCH : ERRADR (2)

Indicates the execute address of the error process when an error occurs in the disk process during communication.

F6DEH : ULDLSVSP (2)

Work area to save the stack pointer when DL/UL process starts.

F6E0H : ULDLTYPE (1)

Flag to indicate which process is executing in DL/UL process.

=0 : Down Loading
= -1 : Up Loading

F6E1H : RENFLG (1)

Flag to indicate whether Rename process is executed at write process to disk in DL/UL process.

11.3 IC Card Protocol expansion

11.3.1 Overview

EHT-10/EHT-10/2 contains an IC card I/F and can read/write IC cards based on ISO standard. OS supports the following two types of usage as IC card use format.

(1) Use of IC card as disk
(2) Use of IC card as serial communication

(1) IC card as disk

IC card can be assumed to be a disk for R/W or R/O and can be used as a floppy disk or RAM disk. OS communicates with the IC card with Toppan printing protocol. (See Section 6.5) Therefore, "protocol expansion" must be performed for IC card with another protocol. Protocol expansion is performed using a hook arrangement in the OS process.

(2) IC card as serial communication

IC card is assumed to be media containing a CPU and only supports serial communication. OS supports this with BIOS ICCARD. Special care of protocol expansion is not necessary for applications, it can be assimilated in its process.

11.3.2 Support at OS

(1) Disk support

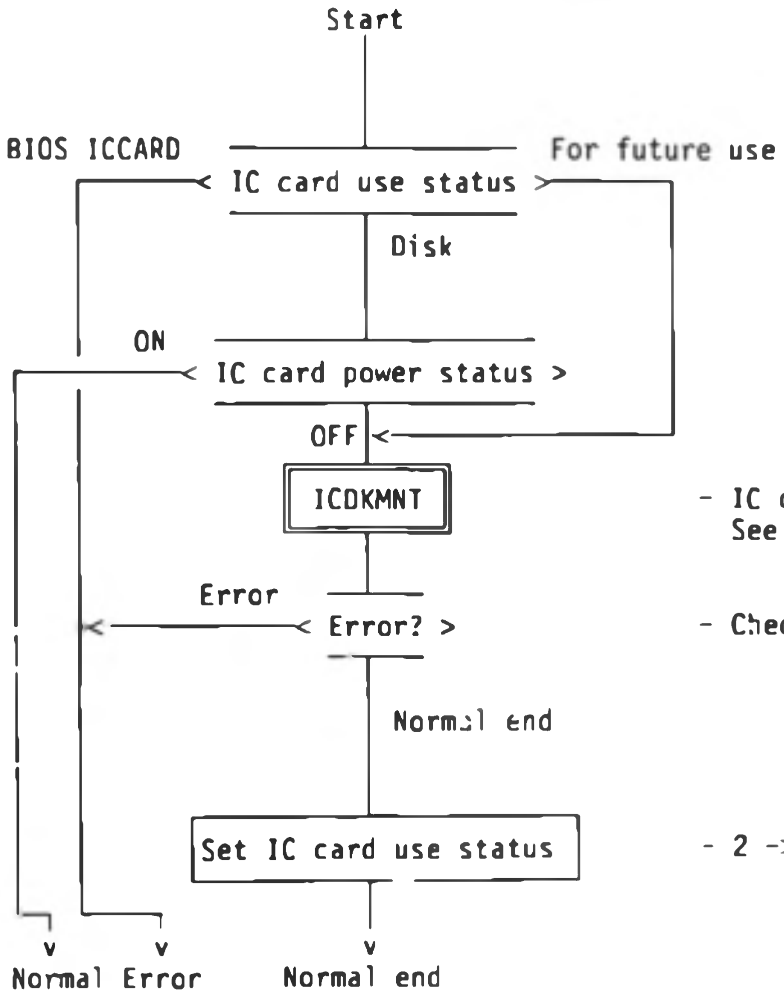
OS supports the use of the IC card as a disk. The IC card is assigned to drive C and is accessed using SELKSK, READ and WRITE of BIOS. (When BDOS is used, BDOS accesses using the BIOS of SELDSK, READ and WRITE in its process.)

```
SELDSK ... Drive selection (Fig. 11.8)
READ ..... Data read in 1 record units (128 bytes) ─┬──   Fig 11.9
WRITE .... Data write in 1 record units (128 bytes)─┘

ICDKMNT ... IC card installation check (See item 3)
ICRECRD ... 1 record read (See item 4)
ICRECWT ... 1 record write (See item 4)
```

The routines mentioned above are actually in charge of access to the IC card.

1 SELDSK (BIOS) at drive C specification



- ICUSEDV (F662H) = 0 : For future use
  = 1 : BIOS ICCARD
  = 2 : Use of disk
- ICONFLG = 1 : ON
  = -1 : ON -> OFF
  = 0 : OFF
- IC card installation check
  See item 3.
- Check CY flag
- 2 -> ICUSEDV (F662H)

Fig 11.8 SELDSK Process

2 READ / WRITE (BIOS) at drive C specification

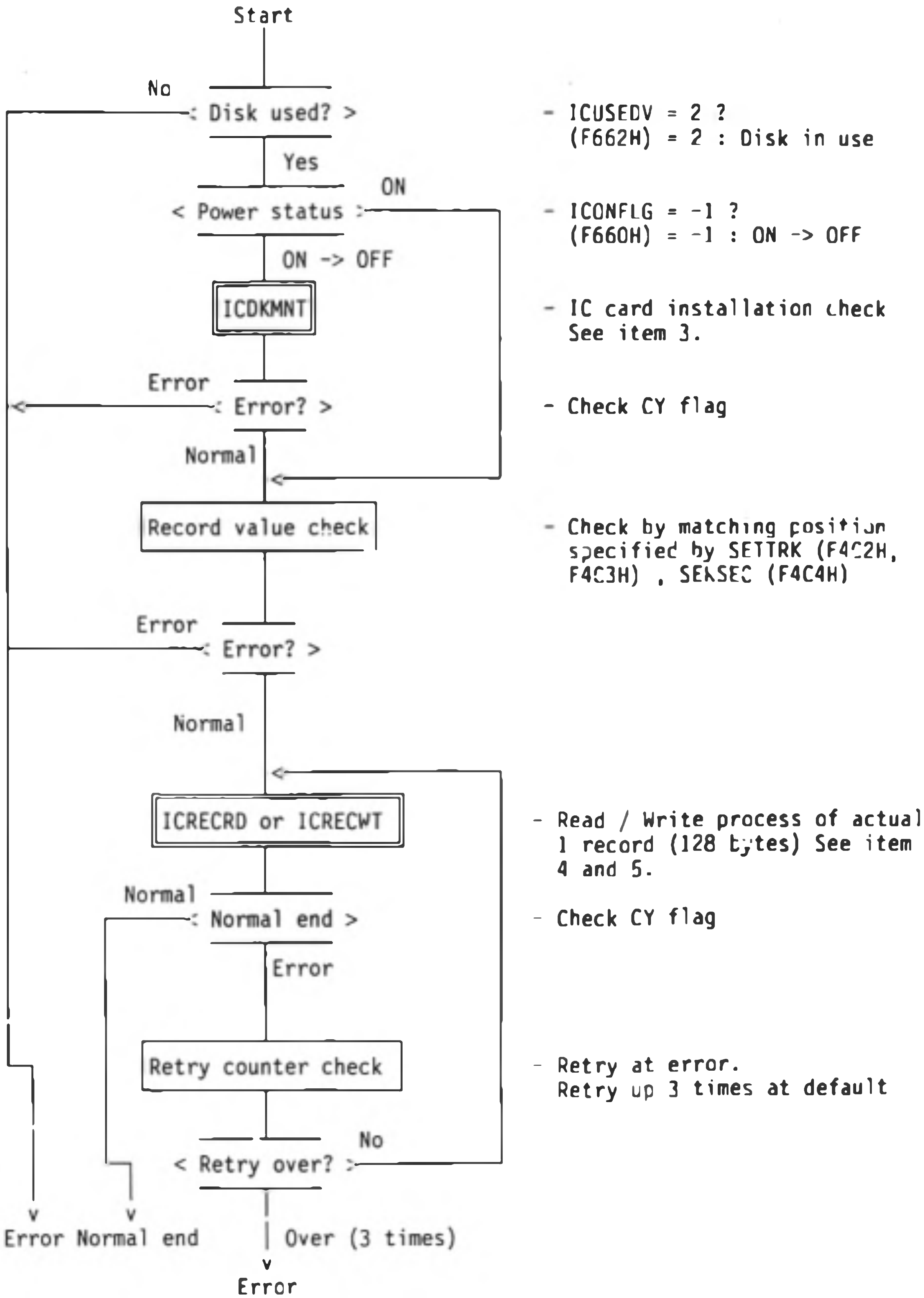


Fig 11.9 READ/WRITE process

3 ICDKMNT

(a) Overview

ICDKMNT starts at the first IC card access or at the IC card access after OFF operation is generated while IC card is in use. ICDKMNT checks IC card installation, IC card capacity and sets data to DPB (Disk Parameter Block). ICDKMNT supplies power and clock to the IC card, releases reset, accepts reset response and puts the IC card in the open state.

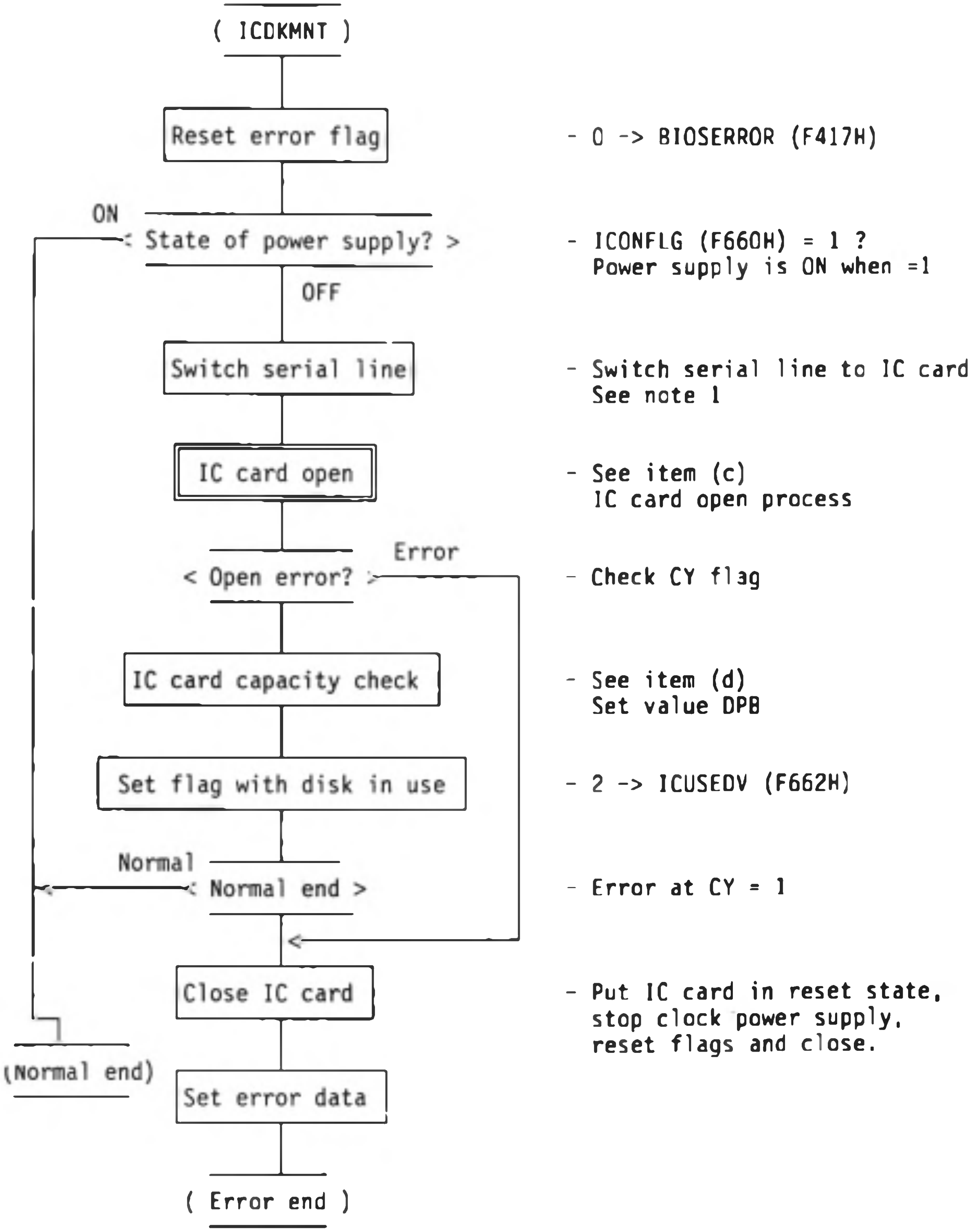
<Note> When OFF operation occurs while IC card is in use, one of the following conditions is satisfied.

* When the main power supply of the main unit is turned off (power switch, battery charge or auto-power is off.)

* When access to the IC card is not performed for a preset time (approx. 60 seconds as default).

* When the rear cover of the IC card opens.

(b) Main process



(Note 1) Serial line is performed at 9600 bps, 8 bit, even parity.
(Set values are changeable.)

Fig 11.10 IC Card Mount Process

(c) Open process

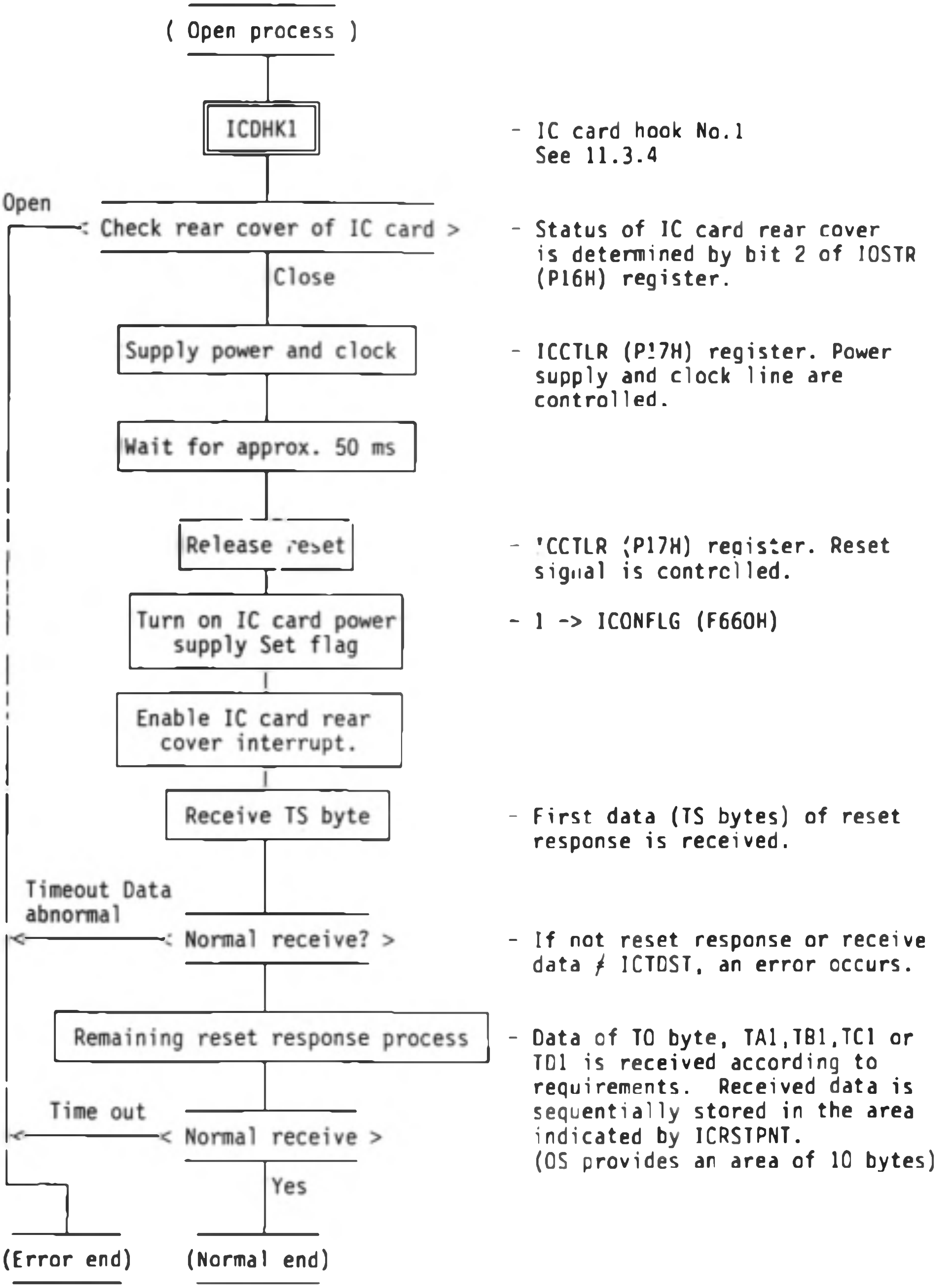
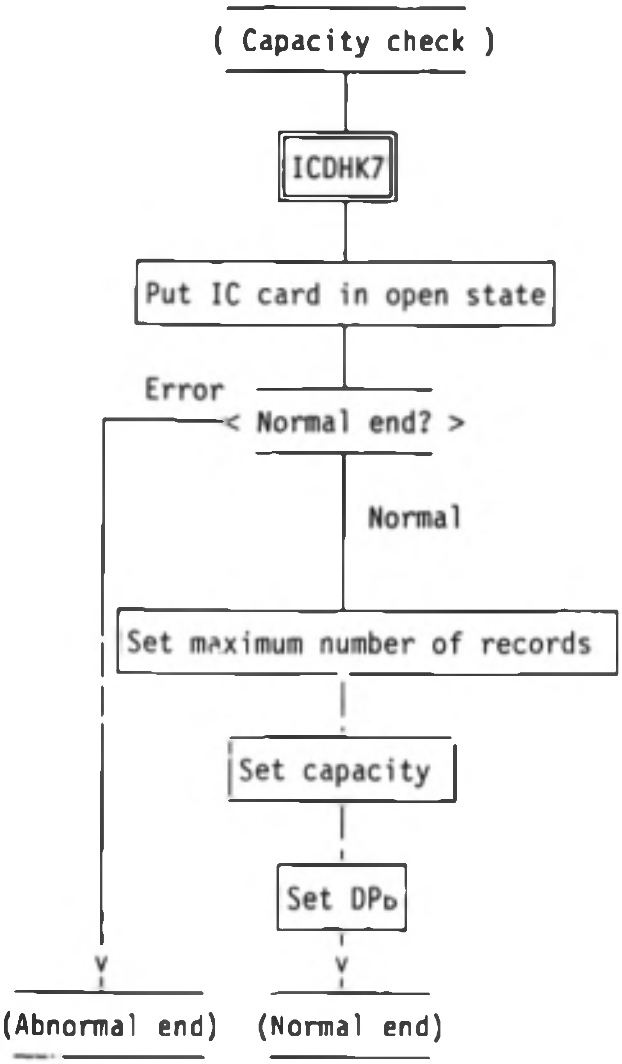


Fig. 11.11 IC Card Open Process

(d) Capacity check



- IC card hook No.7
  See section 11.3.4
- The code is sent/received to/from IC card in the open state. (See note 1)
- Number of 128 byte data records in queried by IC card data area size. This area is used at ICRECRD/ICRECWT.
  Maximum number of records -> ICMAXREC (F65CH)
- ICMAXREC /8 -> QTICCARD (F65EH)
  However, fraction is rounded.
- Set value of DPB based on the value of QTICCARD. (See note 2)

(Note 1) When IC card protocol is different, an error occurs in this process, thus, expansion using a hook is necessary. When this process is expanded, maximum number of records, capacity and DPB setting are necessary.

(Note 2) Set values corresponding to capacities are as follows.

QTICCARD - 1 -> DPB+5, 6
(Disk max. size)

QTICCARD	0 to 2 KB	2 to 8 KB	More than 8 KB
DPB2 + 7 (Directory size)	7	15	31
TPICCARD	2	4	8
DPB2 + 11, 12 (Check vector size	2	4	8

Fig 11.12 IC Card Capacity Check Process

4 ICRECRD, ICRECWT

(a) ICRECWD

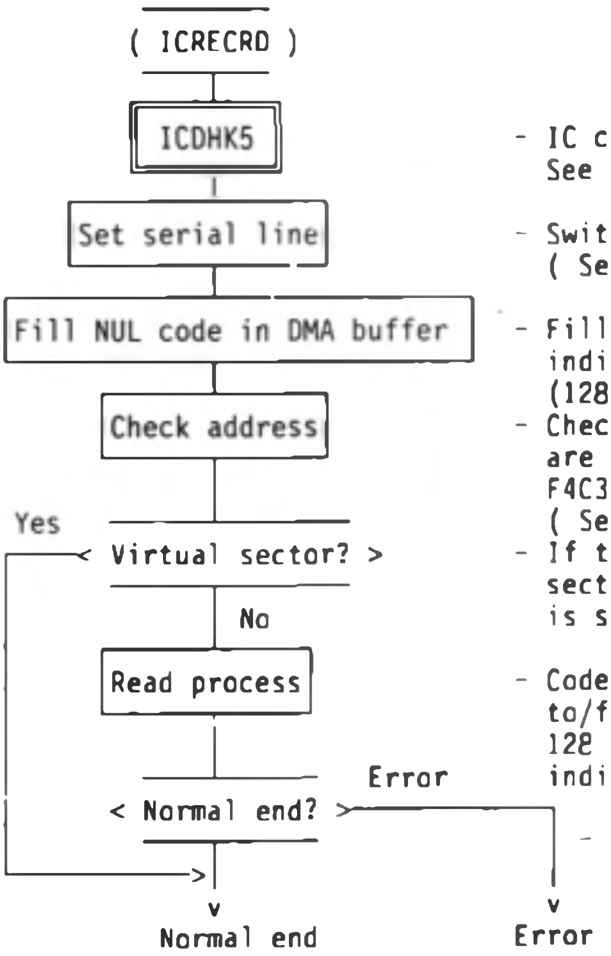
ICRECWD computes the physical address to access SEKTRK (track), SEKSEC (sector) IC card, reads data of 128 bytes from the computed address, stores in DMAADR (buffer) sequentially and returns. At his time, additional data of the header or checksum should not enter the 128 bytes and data stored in DMAADR is only unaffected data.

(b) ICRECWT

ICRECWT computes the physical address to access SEKTRK (track), SEKSEC (sector) IC card, then writes and returns data of 128 bytes stored in DMAADR (buffer), which follows the computed address. As data stored in DMAADR is unaffected data, header or checksum is added for processing, according to requirements.

<Note> When IC card is in close state and ICRECRD or ICRECWT is specified, ICDKMNT process is executed for open state before this process starts.

(c) ICRECRD Process



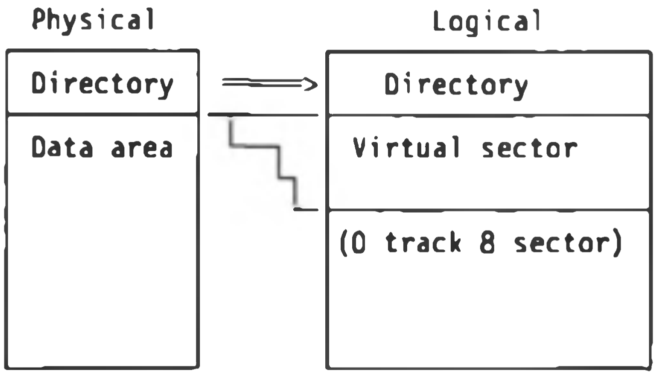
- IC card hook No. 5
  See section 11.3.5
- Switch serial line to IC card I/F.
  (See note 1)
- Fill NUL code (E5H) in DMA buffer indicated by DMAADR (F4D5H) (128 bytes)
- Check whether the records which are specified by SEKTRK (F4C2H, F4C3H), SEKSEC (F4C4H).
  (See note 2)
- If they do not reside, virtual sector is assumed and read process is skipped..
- Code is actually sent/received to/from IC card and data for 128 bytes which SEKTRK or SEKSEC indicates is read to DMAADR.
- Check CY flag.

Fig. 11.13 IC Card Read Process

(Note 1) See "7.2.4 Serial Communication by User" for serial line switching.

(Note 2) For example, directory is always assumed to have 8 sectors (1 Kbyte) in the system, the data section starts with 0 track 8 sectors. However, the directory area has 256 bytes or 512 bytes actually, if 0 track 2 sectors or 0 track 4 sectors are specified, it becomes virtual sector. (See below)

(Note 3) If protocol is different, expansion is performed using a hook.



(d) ICRECWT Process

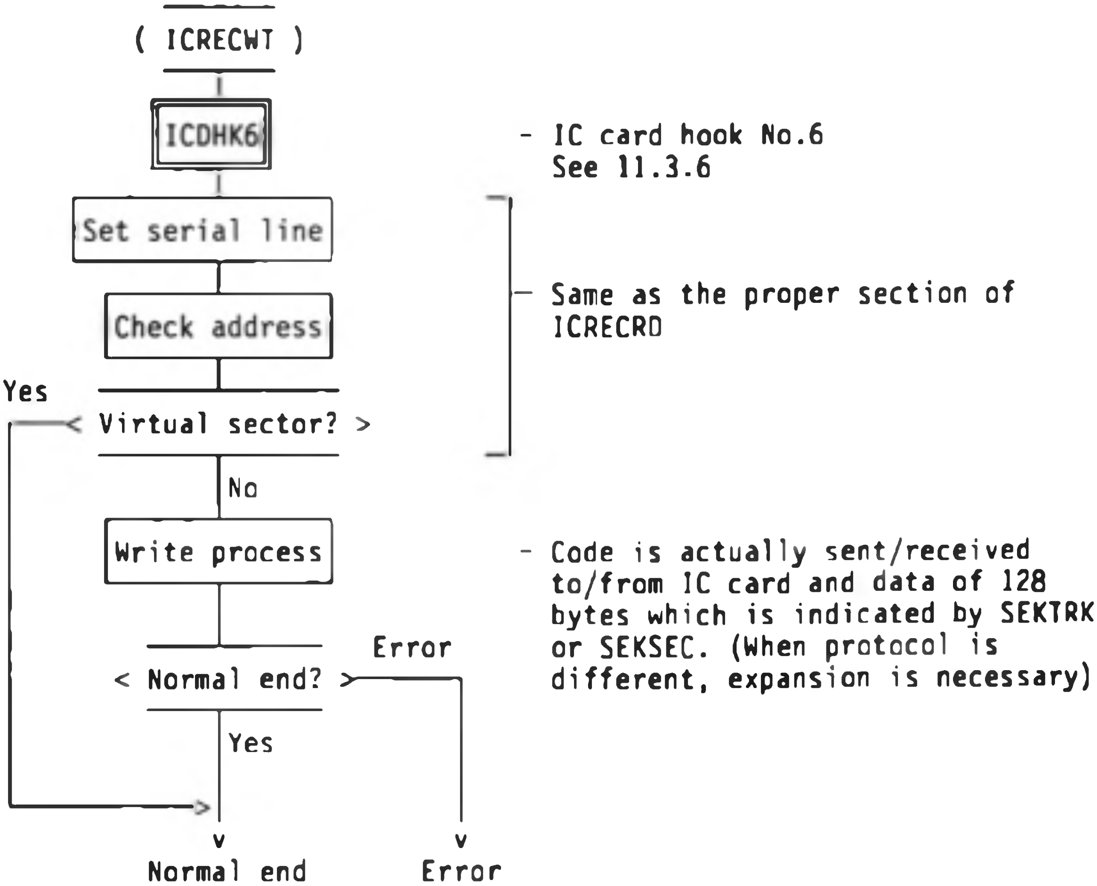
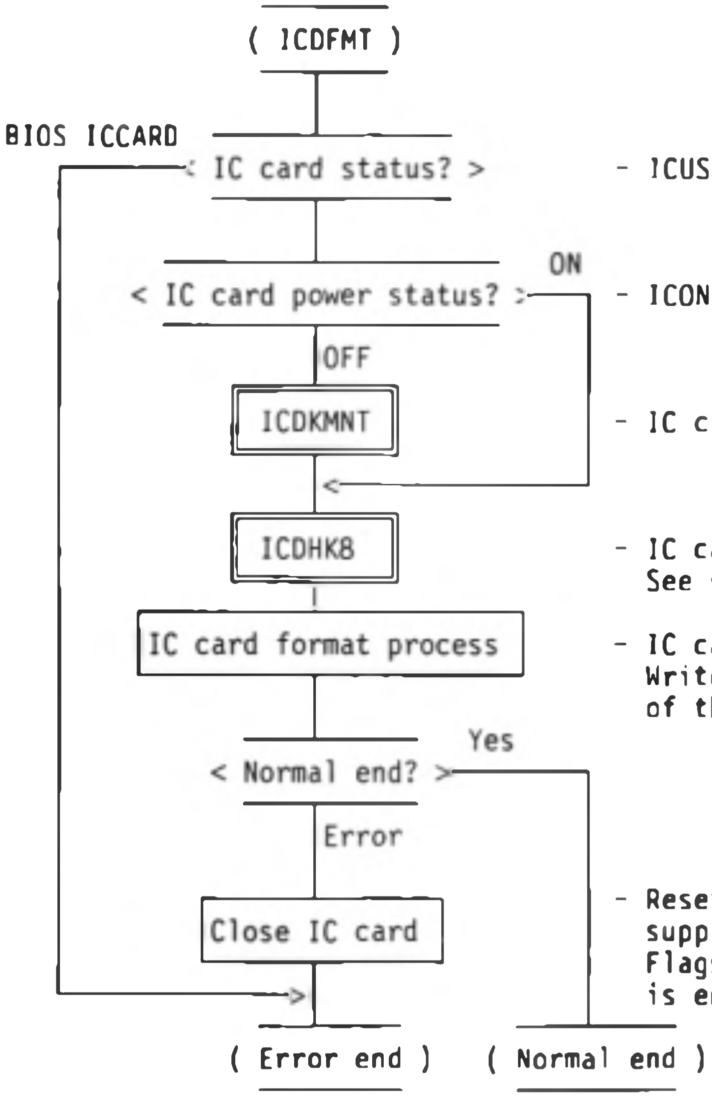


Fig 11.14 IC Card Write Process

5 ICDFMT

ICDFMT format-processes IC card to use IC card as CP/M disk. In format process, E5H is written to IC card data area using ICDKMNT, ICRECRD, ICRECWT as mentioned above. This process starts when the disk format of the system menu is specified.



- ICUSEDV(F662H) = 1 ?
  = 1 : BIOS ICCARD is in use

- ICONFLG(F660H) = 1 ?
  = 1 : ON

- IC card installation check.

- IC card hook NO. 8
  See section 11.3.7.

- IC card format process.
  Write E5H to the entire data area of the IC card.

- Reset signal and clock power supply are canceled to the IC card Flags are reset, and close state is entered.

(Note 1) When the protocol is different, expansion is performed using a hook.

Fig 11.15 IC Card Format Process

(2) Serial support

Due to IC card confidentiality of its original characteristics, OS supports BIOS ICCARD as standard, which supports only serial communication function.
The following 5 functions are provided for this BIOS.

Open (puts IC card in open state.)
Close (Puts IC card in close state.)
Read (1 byte data receive)
Write (1 byte data send)
Status (Status of receive buffer)

BIOS ICCARD is different from a disk and receives data using RXRDY interrupt. (Disk performs send/receive data in coded units, the software timer monitors timeout time.)
See item ICCARD in "4.2 BIOS Function Description" for further information.

11.3.3 Protocol Expansion

(1) Overview

This section describes expansion when using the IC card which utilizes a protocol other than the protocol OS supports as standard.
The following two types of usage of the IC card are as mentioned in the previous item.

1 Use of IC card as a disk
2 Use of IC card as serial communication

For the latter, special care of the protocol difference of the IC card is not necessary. (It can be assimilated in the application.)
However, with BIOS ICCARD open function, reset response check is executed after Reset is released. Thus, IC card of which process at reset is different must be expanded using ICDHK1 (open process hook).

For the former, expansion is performed depending on the difference of protocol from standard, but basically the following 4 types of hooks are used.

ICDHK7 ... Hook in ICDKMNT
ICDHK5 ... Hook in ICRECRD
ICDHK6 ... Hook in ICRECWT
ICDHK8 ... Hook in ICDFMT

(2) Expansion procedure

1 Expansion program loading

Program for protocol expansion can be placed in/on any of the following.

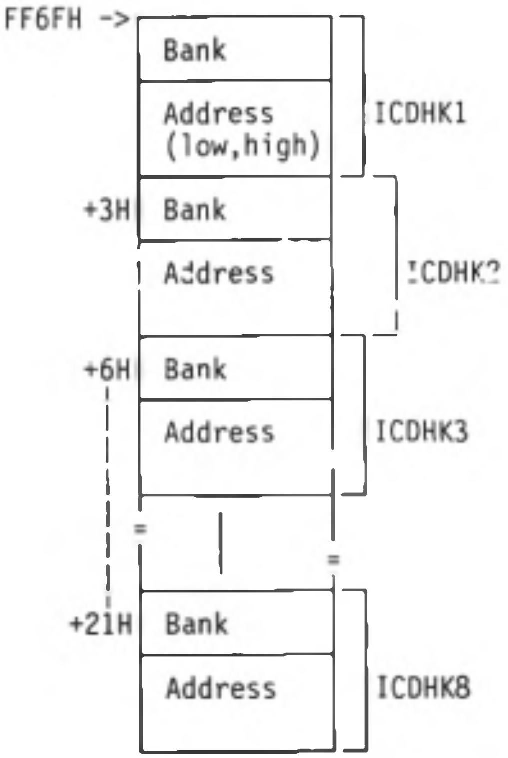
(a) In user BIOS area
(b) On application ROM
(c) In RAM (TPA)

With (a), see "4.6 User BIOS" for maintaining.
With (b), care of location (address) of expansion program in memory map is necessary. (When the hook is rewritten, it directly jumps to the specified ROM, relating to the location on ROM and memory map.)
With (c), no problems occur while the application is executing, but if another application starts, it may unconsciously speed-up . Expansion process must be closed in applications.

2 Rewrite of Hook

Rewrite the contents of the hook corresponding to the process to be expanded.

Address Bank



Sub-bank
Main bank
=0 : RAM
=1 : System ROM
=2 : Application ROM

(Ex. 1) When expansion process is at 6C80H or application ROM (27C256)
Bank = 20H
Address = 6080H

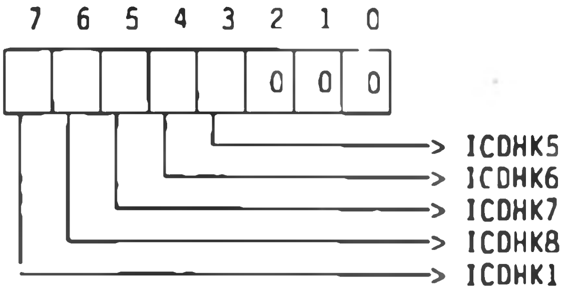
(Ex. 2) When expansion process is at D000H in user BIOS area in main RAM.
Bank = 00H
Address = D000H

Fig 11.16 Rewrite of Hook

3 IC card expansion specification

IC card hook does not jump to the expansion process only by rewrite of a hook. IC card expansion specification is also necessary.

ICHOKDT1 ... Specifies whether jump to IC card hook must be
(F1E6H) performed.



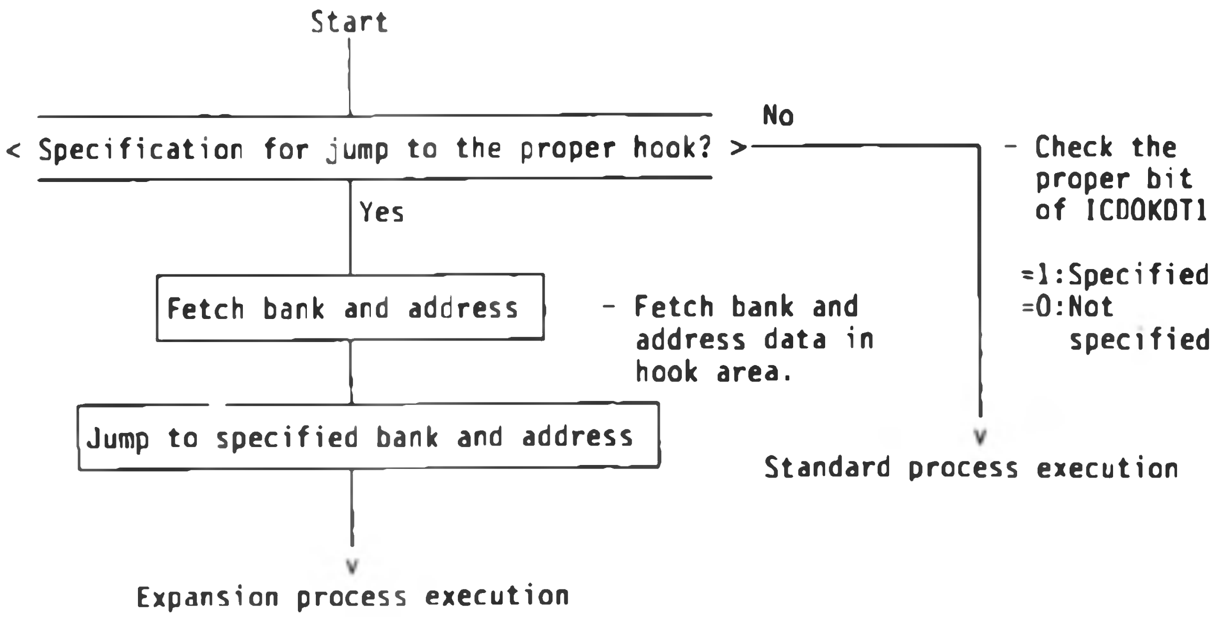
Bits correspond to hooks, specification is as shown below. Set bit corresponding to the proper process to 1 in order to jump from the next IC card access to the address specified by the bank which is set in item 2.

(Ex. 1) When specifying reset response process expansion.
ICHOKDT1=80H

(Ex. 2) When expanding IC card installation check, 1 record read/write and format processes.
ICHOKDT1=78H

(3) Process flow of each hook

Flow chart shows the positions of the hooks (ICKH1,ICDHK5 to 8). Each hook decides whether to jump to the specified address using the following process.



As control moves to the expansion process by jump instruction, if return is executed in the expansion process, original standard process is ignored and only the expansion process becomes optimizing.

11.3.4 Mount process Hook (ICDHK1, ICDHK7)

(1) Hook at reset response (ICDHK1)

1 Overview

This hook starts when access is going to execute to IC card in close state and puts IC card in open state. This hook opens IC card.

2 Items where expansion is required in this hook

(a) To check whether IC card rear cover is open.
When open, it must return with an error (CY=1, A=0AH)
(b) To enable IC card rear cover interrupt.
(c) To set IC card serial line to read mode.
(d) To supply power to IC card.
(e) To supply clock to IC card.
(f) To set IC card power ON flag (ICONFLG) after waiting for at least 50 msec and releasing the reset signal.
(g) To receive reset response from IC card. Return is performed by CY=1, A=06H at timeout error and by CY=0 at normal end.

3 Note

(a) Register does not need to be held.
(b) As communication conditions (transfer rate, parity ... etc.) have been already set (9600 bps, 8 bit, even parity), if modification is required, SYSCTLR1 TO SYSOUT must be modified in advance.

(2) Hook at IC card installation check (ICDHK7)

1 Overview

This hook starts when IC card is in the close state and access is executed to IC card. This hook computes IC card capacity and regulates DPB (Disk Parameter Block). (See ICDKMNT in the previous item.)

2 Items where expansion is required in this hook

(a) To put IC card in open state for software.
To perform specified collation and enable read/write execution of data that follows.
(b) To set IC card DPB (Disk Parameter Block) as drive "C" when IC card is in open state.

3 Note

(a) The following require area setting in this hook.

QTICCARD: IC card size (data area) Numeric value is specified in ICBLKSZ. When the value is 0, no IC card is assumed.
ICMAXREC: Number of access enable records in (F65CH) IC card data area, (the value that represents the net amount of data that can be held in 128 byte units).
TPICCARD: Number of records which is held as IC card directory area.

DPB3(F177H)... Disk parameter black (Drive C:)
DPB3 + 5, 6 (Disk size max)
DPB3 + 7 (Directory size)
DPB3 + 11, 12 (Check vector size)

(b) Return must be performed by CY=0 at normal end and by CY=1 at error end.

11.3.5 Hook at 1 Record Read (ICDHK5)

(1) Overview

This hook starts when IC card is specified by BIOS READ.

(2) Item where expansion is required in this hook

(a) To set serial line to SIO.
(b) To compute access address from SEKTRK, SEKSEC to IC card. To fill NUL code in DMA buffer for return with virtual record.
(c) To read data of 128 bytes from IC card using "read command" and store in DMA buffer.

(3) Note

(a) Each register of BE, DE, HL must be preserved.
(b) See "6.5 IC Card" for track/sector correspondence with IC card address.
(c) Return must be performed by CY=0 at normal end and by CY=1 at error end.

11.3.6 Hook at 1 Record Write (ICDHK6)

(1) Overview

This hook starts when IC card is specified by BIOS WRITE.

(2) Items where expansion is required in this hook.

(a) To set serial line to SIO (IC card).
(b) To compute access address from SEKTRK, SEKSEC to IC card. With virtual record, just return.
(c) To write DMA buffer data of 128 bytes to IC card using "write command".

(3) Note

(a) Each register of BC, DE, HL must be stored.
(b) See "6.5 IC Card" for track/sector correspondence with IC card address.
(c) Return must be performed by CY=0 at normal end and by CY=1 at error end.

11.3.7 Hook at Format (ICDHK8)

(1) Overview

IC card formatting is processed after installation check in a close state. This hook is in real formatting start point.

(2) Items where expansion is required in this hook.

(a) To write E5H to the entire area using "write command" to IC card.
(b) Return must be performed by CY=0 at normal end and by CY=1, A register=2 after IC card close process at error end.

(3) Note

IC card close process is executed as follows.

(a) Reset RXENB bit.
(b) Reset IC card.
(c) Disable IC card rear interrupt.
(d) Stop clock supply to IC card.
(e) Stop power supply to IC card.
(f) Clear each flag. (write 0) (ICCPWFLG, ICUSEDV, ICONFLG, ICCVFLG)
(g) Set serial line to RS 232C.
(h) Return RXENB bit to original.

11.3.8 IC Card Protocol Expansion Example

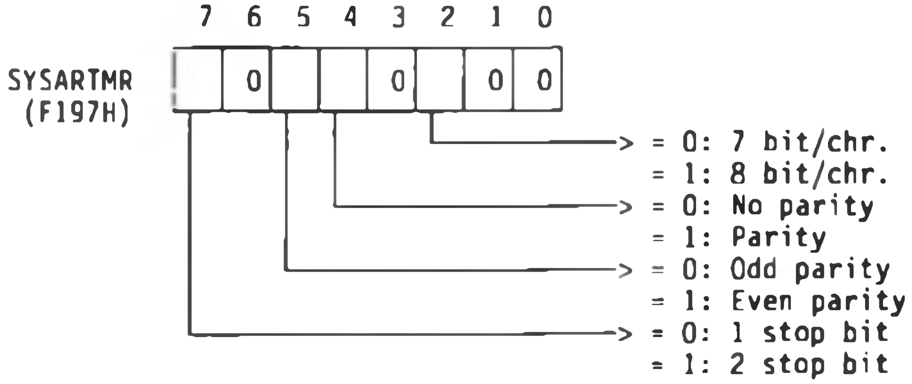
"Reference 9. Sample 33" describes expansion example using IC card.

11.3.9 Remarks

(1) Communication parameter modification

EHT-10/EHT-10/2 sends/receives data at 9600 bps, 8 bit, even parity as a communication parameter. To modify this communication parameter, follow the procedure as described below.

1 Set SYSCTLR1 and SYSARTMR to required communication parameter.



SYSCTLR1 specifies transfer rate, 9600 bps is constant for IC card.

2 When parity is modified, data for TS byte collation at reset response of ICTSDT(F1E9H) must be modified.

(Ex.) No parity ICTSDT = FBH
      Even parity ICTSDT = 3BH

(2) Interrupt

EI state is always held at jump to hooks, if another interrupt is issued in each process, malfunction may occur. Thus, IER (Interrupt Enable Register) operates and all interrupts are disabled in order to jump to a hook. Therefore, the following process must be inserted at return from each hook.

```
ZIER EQU  04H
     LD   A,(RZIER) ─┬─  Return interrupt to the state prior
     OUT  (ZIER),A  ─┘   to the hook jump.
```

(3) Remarks for using IC card

1. Error check at command send and response receive.
As IC card operates in DI state normally at command send and response receive, error check is performed as follows.

(a) Check open/close state of rear cover at command send and response receive, if open, an error occurs.

```
LD   A,(ICCVFLG)  ─┬─  Check rear cover interrupt of
OR   A             │   IC card
SCF                │
RET  NZ           ─┘
LD   A,(ICONFLG)  ─┬─  Check power off
INC  A             │
RET  Z            ─┘
OR   A
RET
```

(b) Check timeout in the process of which response receive loops.

(4) IC card power-on sequence

With EHT-10/EHT-10/2 OS, power-on sequence to IC card supplies power and clock, releases reset and resets response process by release of reset.
Reset response process is as follows.

1 Receive TS byte.

2 Check whether received TS byte data is equal to the value of ICTSDT(F1E9H). If different, an error occurs so do not execute the following processes.

3 Receive TO byte

4 Determine whether TA1 to TD1 that follow are received, following the data of received TO byte.

5 If there is data of TA1 to TD1, receive it.

6 Open process completes with end of data receive of all reset response.

TS byte, TO byte and data from TA1 are stored in the area that ICRSTPNT(F1AEH) points.
In expansion process, when either of the following conditions occurs, reset response process must be expanded using ICDHK1.

1 When timing of power supply, clock or reset is modified.

11.3.10 Work Area Detail Related to IC Card

This section describes the work area related IC card. "*" means that this variable is different from OS Version 1.0. For the difference of system work area between Version 1.0 and 2.0, see Appendix 10.

(Example)
Address(*): Name of Variable (Byte Qty.)

EAC0H*: ICBASIF (29)
Area for accessing the IC card from BASIC.

EADDH*: BDOSIF (33)
Area for accessing the IC card from BDOS.

EFC0H*: ICFILNAM (42)
Area for the file names of the IC card.

EFEAH*: ICDIDDAT (22)
Area for ID data of the IC card.

F196H: SYSCTLR1 (1)
F197H: SYSARTMR (1)
F198H: SYSSWR (1)
F199H: SYSARTCR (1)
Communication parameter with IC card when using IC card as a disk.
Default is 9600 bps, 8 bits, EVEN parity.
Bit configuration is the same as that of I/O port data.

```
SYSCTLR1 ----> CTLR1 (P01H)
SYSARTMR ----> ARTMR (P15H)
SYSSWR ------> SWR (P18H)
SYSARTCR ----> ARTCR (P16H)
```

F1ADH*: ICRSTCNT (1)
Specifies the number of receive bytes when the answer to reset.

= 00H : Ignore the answer to reset
≠ 00H : number of receive bytes. When answer to reset, receives this number of bytes, and stores the data to the area which is pointed by ICRSTPNT. If the time out error is detected during the receiving, system stores the data till the error is detected. default value is 09H.

F1AEH: ICRSTPNT (2)

Represents the storage area of the Reset response upon cancellation of the IC Card's Reset status.

```
+----------+       +------+
| ICRSTPNT |-----> |  n   | ....Size of area (in bytes)
+----------+       +------+     Default value = 10 bytes.
                   |  TS  |
                   +------+
                   |  T0  |
                   +------+
                   |      |
```

The procedure for storage to this area is specified by the value of ICRSTCNT.

F1B0H: IDFILCNT (1)
F1B1H: ICCDTIME (2)
F1B3H: ICCDPRM (5)

Area to regulate each access condition of BIOS ICCARD (independent from disk access)

IDFLTCNT: Specifies the number of times of retry when IC card error occurs. Default is 3 times.

ICCDTIME: Regulates timeout time when data is received from IC card. Default is approx. 3 seconds.

ICCDPRM : Has loop open parameter when ICCARD OPEN is specified by IC card.

(Note)
In BIOS ICCARD OPEN process, this section of communication parameter is copied after SPBADR+4 and BIOS RSIOX open process is executed using COMBUF area as receive buffer. Therefore, when other modules have already used RSIOX, open error occurs. It must be used after RSIOX close or ICCARD close process is executed.

ICCDPRM

+0	Bit rate
+1	Bit length
+2	Parity
+3	Stop bit
+4	Special parameter

Default value is 9600bps 8bit, Even parity, 1 stop bit.

F1E2H*: ICSRACD   (1)

SRA code (the 1st byte during command transmission)
Default value = 3AH

F1E3H*: ICFLCLS (1)

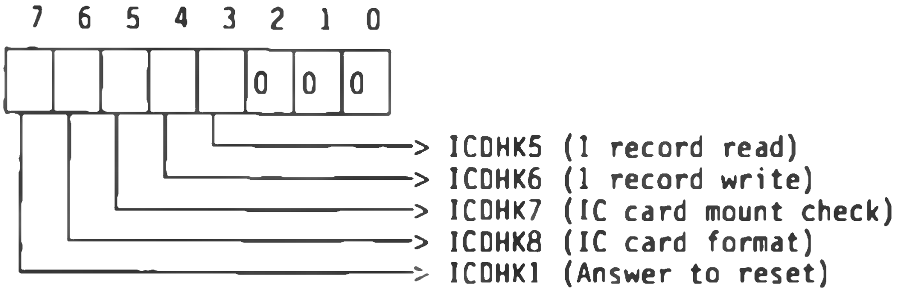
File class code. This is the class code when using file-related commands. Default value = 00H.
The File No. of the ISET function is entered here.

F1E4H*: ICCLASS (1)

System class code. Default value = 20H

F1E5H*: ICRECSZ (1)
Specifies the number of bytes of 1 record. Default value = 40H

F1E6H*: ICHOKDT1 (1)
Specifies whether execute the hook processing or not.
Bit = 1 : Jump to the hook.
= 0 : Do not jump (Default)



F1E7H*: ICWAIT1 (1)
Specifies the interval (in ms units) for awaiting the start of Reset response reception after the Reset status is canceled. Default value = 50 ms(32H)

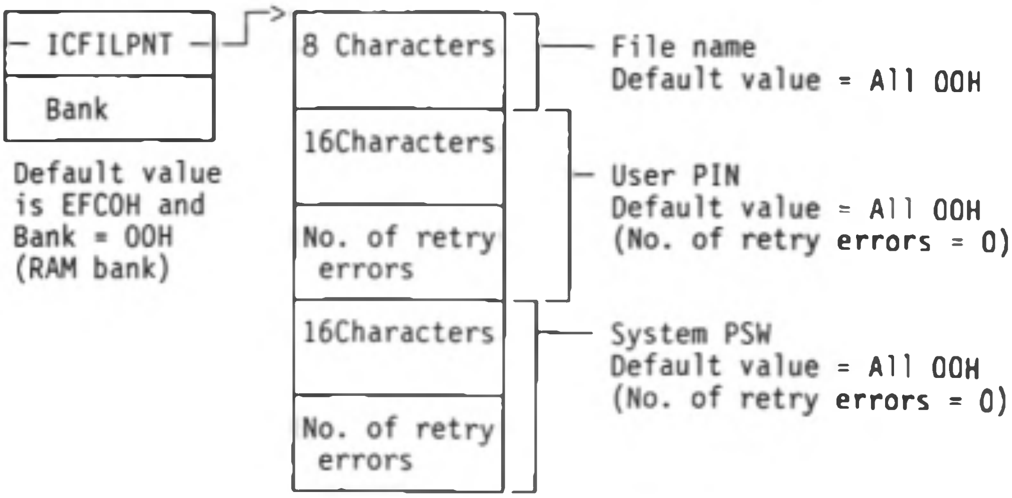
F1E8H*: ICWAIT2 (1)
Specifies the interval (in ms units) for awaiting the start of command transmission after the Reset response is received. Default value = 100 ms(64H)

F1E9H: ICTSDT (1)
Data used for TS byte collation during the reception processing of the Reset response.
During Reset response reception processing, only the TS byte is collated and the other data is merely received.
Default value = 3BH

F643H*: ICDIDPNT (4)
Represents the storage area for ID data during registration of the card ID. (Valid only in DISK Mode)
For the detailed information, see Section 6.5

F647H*: ICFILPNT (3)
Represents the storage area of the File Name data during file creation.

All of the above data are valid only in DISK Mode and represent the address of the data set by the ISET function.



F64AH*: ICFILSZ (1)
Specifies data area size of the IC card as a disk. Unit is 128bytes and default value is 38H (7 kbytes).
This variable is set by IDSK function.

F64BH*: ICDIRNO (1)
Specifies the number of directories of the IC card as a disk. Default value is 60H. This variable is set by IDSK function.

F64CH*: ICOPNFLG (1)
Flag for distinguish whether the class code of the command data is File class code or System class code.

F64DH*: ICDIRBOT (2)
Points the bottom address of the directory area of the IC card as a disk. Default value is 100H. This variable is set by IDSK function.

F64FH*: ICDATTOP (2)
Points the logical top address of the data area of the IC card as a disk. Default value is 400H.

F651H*: ICBLKSZ (2)
Specifies the block size of the IC card. Default value is 100H (256bytes)

F653H*: ICWAIT4 (1)
Specifies the waiting time between to supply the power to the IC card and to supply the clock to the IC card. Unit is mSec and default value is 01H(1mSec).

F654H*: ICSTATUS (3)
F657H*: ICOFFMD (1)
F658H*: RZICCTL1 (1)
F659H*: RZICCTL2 (1)

— Reserved.

F65CH: ICMAXREC (2)
F65DH*: QTICCARD (1)
F65FH: TPICCARD (1)

Indicates data for the IC card capacity.

ICMAXREC: Indicates maximum number of physical records of IC card. Default is 40H (64records)

QTICCARD: Indicates physical capacity of IC card. Unit is specified by ICBLKSZ. Default is 20H (32 x 256bytes).

TPICCARD: Indicates the number of records used for directory. Default is 08H (8 records).

F660H: ICONFLG (1)
F661H: ICCVFLG (1)
F662H*: ICUSEDV (1)

Flag to indicate each state while IC card is in use.

ICONFLG --- Indicates IC card power supply state.
- =0 : No power supply (close state)
- =1 : Power is supplied (open state)
- =-1: Power supply is temporarily canceled as there is no access for a present time.

ICCVFLG ----- Indicates state of the excess current while IC card is in use.
- =0 : Normal current
- =1 : Excess current goes through the IC card

ICUSEDV ----- Indicates type of module which is using the IC card.
- =-1: Command through mode uses IC card.
- =0 : Not used.
- =1 : BIOS ICCARD uses IC card.
- =2 : Disk mode uses IC card.

(Note) Continue to supply power to the IC card while IC card is used in BIOS ICCARD.

F663H: ICRSTDT (10)

Area to store data at reset response (Answer to reset)
This area is pointed by ICRSTPNT.

F66DH*: ICRETRY (1)

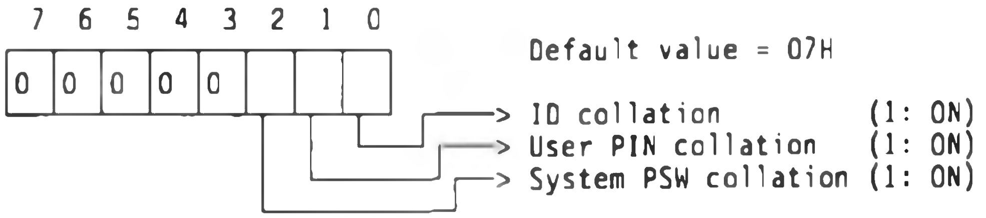
Reserved.

F66EH*: ICWAIT3 (2)

Specifies the interval (in 10-ms units) for a time out error for the response from the IC Card after command transmission. Default value = 00FFH (approx. 2.5 Sec)

F670H*: ICOPNSTS (1)

Specifies the processing during File OPEN status (after the IC Card power is ON). (Valid only in DISK Mode)



F671H*: ICFMTSTS (1)

Specifies the processing during formatting. (Valid only in DISK Mode)

7	6	5	4	3	2	1	0
0	0	0			0	0	0

Default value = 10H

Bits 4–3:

= 00 : Do nothing.
= 01 : Delete the specified file only.
= 1X : Create a file and format after deleted the specified file.

F8D2H*: ICERRSTS (3)

Returns the error information generated during access processing of the IC Card.

+0	Command code
+1	Status (1)
+2	Status (2)

For details on the data, see the following.

F8D5H*: ICDWORK (69)

Area for sending/receiving data to the IC card.

Command codes of ICERRSTS (F8D2H)

Command Code	Command Name(Remarks)
00H	Error unrelated to command processing
01H	Open Card ID
02H	Open System Password
03H	Open User Password
04H	Create File
05H	Directory Read
06H	Erase File
07H	Read
08H	Write
09H	Set Address Pointer

Status codes of ICERRSTS (EF8D2H)

Status 1		Status 2	
00H	OPEN error (EHT-10 side)	01H	Excess current (During Open)
		02H	Reserved.
		03H	TS byte not received (IC Card not installed)
		04H	TS bytes do not match
		05H	Reserved.
		06H	Reserved.
		07H	Reserved.
		08H	Reserved.
		09H	Power off.
		0AH	Excess current (During Write)
01H	Abnormal reception (EHT-10 side)	01H	Parity error (see NOTE below)
		02H	Framing error (see NOTE below)
		03H	Over-run error (see NOTE below)
		04H	BCC error
		05H	Reserved.
		06H	Time-out error
		07H	Format error(SRA code not match)

Status 1		Status 2	
E1H	CLS is abnormal	00H	
E2H	COD is abnormal	00H	
EFH	Others (IC Card side)	13H	Command mode error (for details see the IC card status)
		14H	Length error
		15H	Syntax error (parameter error)
		16H	Duplicate file names
		17H	Specified file does not exist
		19H	Formatting impossible (file already exists)
		20H	Format error
		21H	Sum error
		22H	Memory defect
		23H	Chain error
		24H	Memory full
		25H	Address overflow
		30H	Memory defect during formatting
		32H	Password collation error
		33H	Quantity of valid passwords exceeded
		34H	Card ID collation error
		35H	Comparison error
		41H	File has been forced OPEN (Chain error included)
		42H	File has been forced OPEN (Sum error included)
F0H	Abnormal reception (IC Card side)	01H	Parity error
		02H	Framing error
		03H	BCC error
		04H	Length error
		05H	Time-out error

NOTE: When the above three errors occur simultaneously, they are checked in the sequence of: Over-run error -> Framing error -> Parity error.

11.4.1 Overview

This chapter describes:

(1) Mount process expansion
(2) Interrupt process expansion

as expansion method of cartridge device control.

Be sure to read this chapter when you create a new cartridge device, control it or expand the standard device (printer unit) control.

See "7.3 Cartridge Interface" and "4.1 cartridge Interface" in Hardware Version for further understanding.

(1) Mount process

Mount process checks the cartridge device installation and sets the mode.
Mount process can add the expansion process of the cartridge device initialization using a system hook (CRGHOOK).

(2) Interrupt process

When creating a cartridge device for interrupt process, user's original interrupt process can be performed using a system hook (EXTHOOK).

11.4.2 Cartridge Mount Process

(1) Overview

EHT-10/EHT-10/2 performs cartridge device installation check (mount process) at power-on and reset of main unit and system initialization.
OS reads cartridge device code from the cartridge device, stores in memory and determines the cartridge mode using a device code at the same time in order to set the mode.
See "7.3 Cartridge Interface" for relation of device code and cartridge mode.
Mount process routine jumps to system hook (CRGHOOK) after all processes complete so that the cartridge device initialization is enabled at installation check by adding a hook process.

(2) Contents of mount process and location of hook.

Contents of mount process is shown in Figure 11.17.

1 Device code (CRGDEV)

Device code is configured of the cartridge device number and cartridge mode, and is stored in CRGDEV (F42FH) in memory.

```
           7   6   5   4   3   2   1   0
         +---+---+---+---+---+---+---+---+
CRGDEV   |   |   | - | - |   |   |   |   |
(F42FH)  +---+---+---+---+---+---+---+---+
         |_______|       |_______________|
             |                   |
             |                   +------->Device number
             |                            4 bits of high-order data when P13H is
             |                            read in DB mode.
             |
             +--> Cartridge mode
                    =00 : HS mode
                    =01 : IO mode
                    =10 : DB mode
                    =11 : OT mode
```

2 Decision of mode

Cartridge mode is determined by device number and CSEL (this P16H bit 6 data is set to DB mode and is read).
See "7.3 Cartridge Interface" for details.

3 Mode setting

Cartridge mode is set according to the device code. Mode in the development cartridge is modified at development cartridge connection.

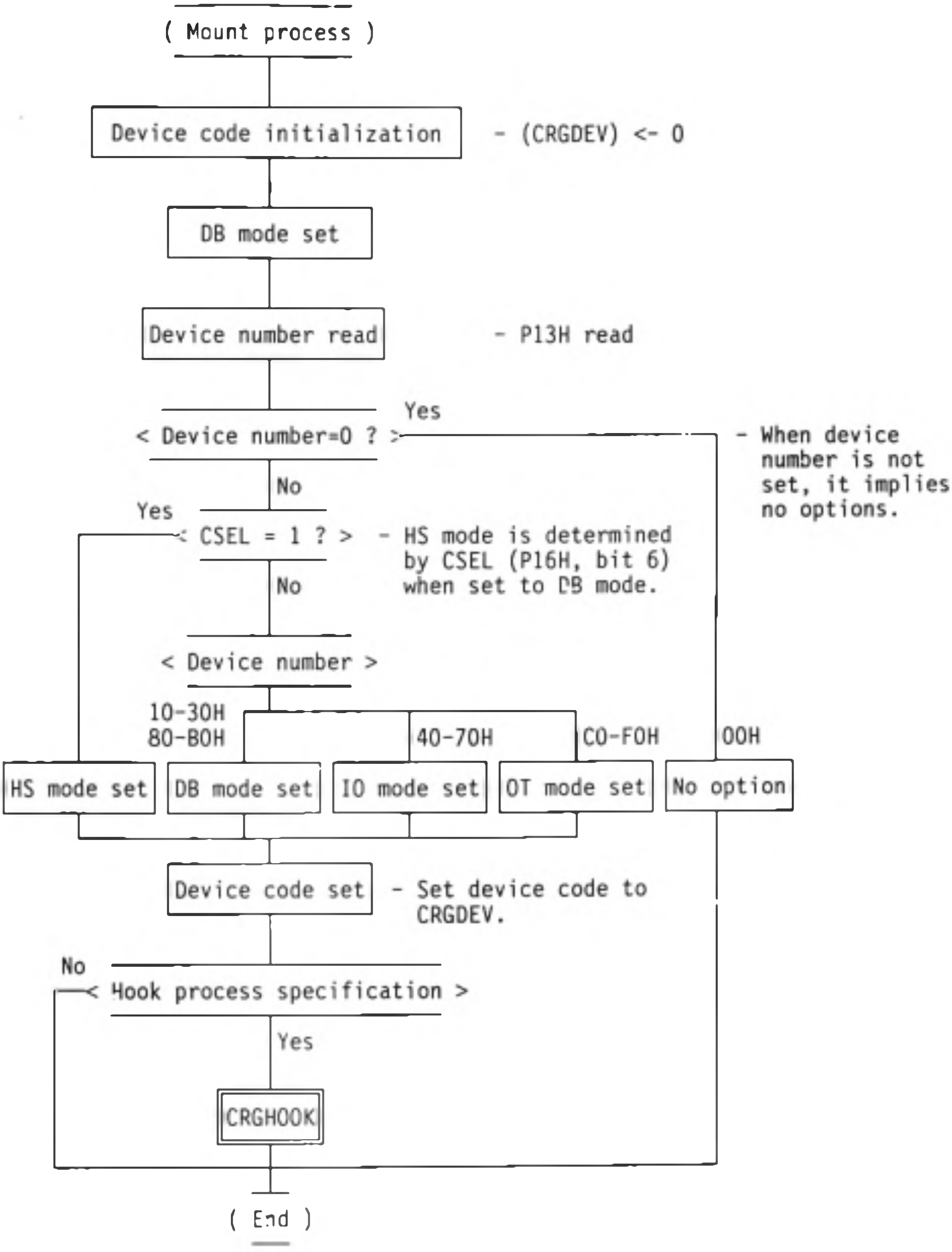


Fig 11 17 Cartridge Mount Process

(3) CRGHOOK expansion

CRGHOOK is used to expand the cartridge device mount process and initializes the cartridge device at power on.

1 Expansion program loading

Hook process program can be placed in the user BIOS area, on application ROM, or in RAM (TPA).
See "10.2 System Hook" for procedure and remarks.

2 Rewrite of hook

Rewrite the contents of a hook corresponding to the process to be expanded. CRGHOOK configuration is as follows.

CRGHOOK

FF6CH	+0	Bank information
	+1	Address information
	+2	

3 Hook process execution specification

To execute hook process, set hook process execution specification flag (USERCRG:F0DFH) in addition to items 1 and 2. Set value other than 0 to USERCRG to execute hook process.

11.4.3 Interrupt Process from Cartridge Device

(1) Overview

$\overline{\text{CINT}}$ terminal of cartridge I/F is connected to external interrupt signal line of EHT-10/EHT-10/2 main unit and can accept interrupt from

cartridge devices using $\overline{\text{CINT}}$. (See Hardware Version or Chapter 7 I/O Description, Chapter 8 Interrupt Process) OS uses Ready interrupt process of the printer unit for interrupt from a cartridge.

To create a new cartridge device for an interrupt process, expand the external interrupt hook (EXTHOOK).

This section describes the interrupt process from a cartridge device using EXTHOOK.

(2) I/O port related to interrupt

I/O port related to interrupt from the cartridge I/F is as described below.

Port Address (R/W) : Port name (RAM data address)

P16H (R) : IOSTR
    bit 0 : (PBUSY) Interrupt status
        0: Interrupt requested
        1: No interrupt requested

P17H (W) : ICCTLR (F0DBH)
    bit 6 : (PRIE) Interrupt mask
        0: Interrupt allowed
        1: Interrupt suppressed

P23H (R) : ITSR
    bit 1 : (IPBUSY) Status of interrupt from IC card or cartridge
        0: Interrupt requested
        1: No interrupt requested

P23H (W) : CTLR3 (F0DEH)
    bit 3 : (PINTDIS) Masking an interrupt sent from IC card or cartridge
        0: Interrupts allowed
        1: Interrupts suppressed
        (0 is generally set.)

(Note) PINTDIS of CTLR3 (P23H) must be always set to 0
    (enable) normally.

(3) EXTHOOK expansion

1 Expansion program loading

EXTHOOK automatically switches to the entire RAM (bank 0 # 0) to call the expansion process, thus, the expansion process routine must be loaded in RAM. When the expansion process is used in multiple application programs, use the user BIOS area. See "4.6 User BIOS" for further information for how to use the user BIOS area.
(See APPENDIX9 SAMPLE 30)

2 Rewrite of hook

EXTHOOK is a 3 byte configuration as shown below.
The address section is rewritten.

	EXTHOOK	
FFDBH +0	C3H	- JP instruction
+1 +2	Address information	- Expansion process address

3 Remarks in expansion process

(a) The expansion process is called in "DI" (Disable Interrupt). "EI" (Enable Interrupt) must not be called in the expansion process.

(b) When the stack is often used, set a new stack. In this case, the stack must be returned after the expansion process.

(c) EXTHOOK is called by the cartridge I/F interrupt, 1 msec / 8 msec interval interrupt and IC card interrupt. Therefore, the contents of the process must be classified by the interrupt cause in the expansion process. PBUSY (P16H bit 0) can acknowledge the cartridge I/F interrupt.

(d) Expansion process completes with the following format. System standard process continues after the end. (See Note 1.)

<Entry Parameter>

DE register : Interrupt state after the end of the interrupt process.

```
                DE=0DEBH ... Enable interrupt
                  =0DE1H ... Disable interrupt

    RETAD      EQU  EF4FH       Post-process address of EXT interrupt.

               LD   HL,RETAD    Exchange of post-process address with
                                return address.
               EX   (SP),HL
               PUSH DE          Save specification as enable/disable
                                interrupt.
               JP   (HL)        Jump to return address.
```

When loading the expansion process, modify a section of the system interrupt process as shown below.

Modification address	EFB9H
Before modification	18H (JR instruction)
After modification	C9H (RET instruction)

Following this, jump to enable/disable interrupt process which is set at the end of the expansion process. Return to the post-process routine following RET instruction of the jump target.

(Note 1) Return to the original state following the process shown above to obligingly disable the interrupt from the cartridge I/F in the system standard process.

11.5 Addition of Barcode Decoder

11.5.1 Overview

ICF interrupt is issued when the input polarity from barcode changes (0"*1 or 1"*0). ICF interrupt is issued when barcode is attached to white paper, the LED continuously lights then polarity changes from white to black. If a barcode is now scanned to read black bars, ICF interrupt is issued.
ICF interrupt routine ICFINT is as shown below.

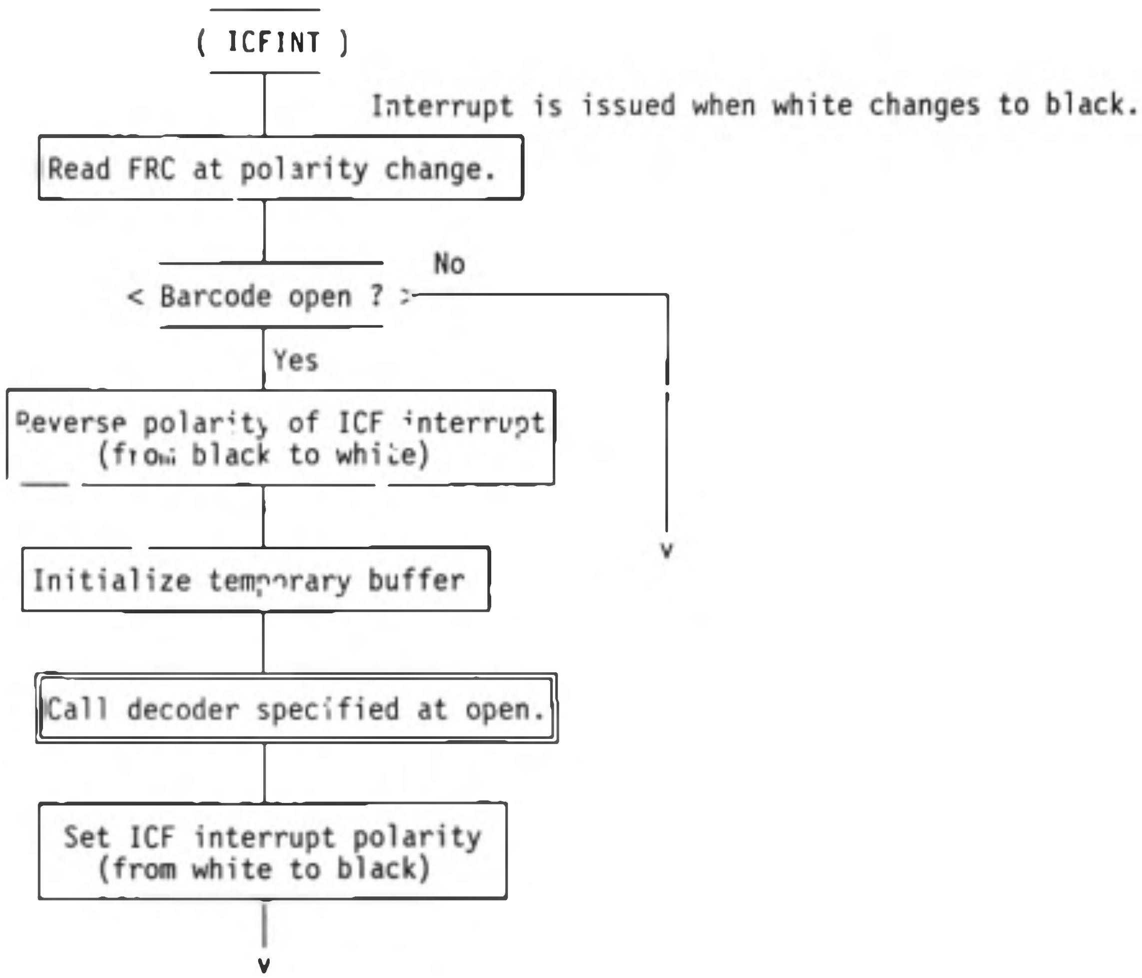


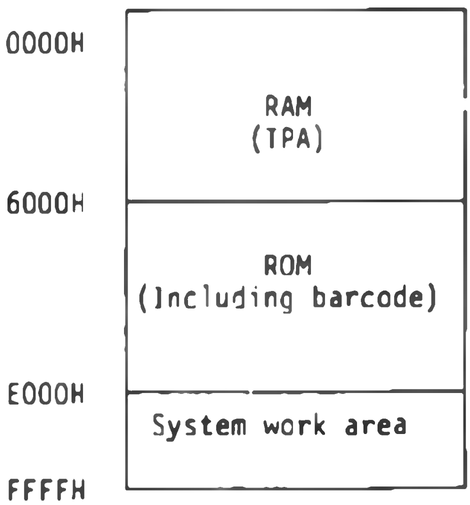
Fig. 11.18 ICF interrupt Process

Decoder for specified barcode is called in ICFINT routine. Called decoder correspond to types of barcode specified by BARCODE open function.
ICFDSP is the table which registers the vector (entry address) of the decoder corresponding to the types of barcode. ICFINT calls the decoder referring to the types of barcode and ICFDSP at open. ICFDSP of the vector (entry address) to 6 decoders is stored in a 12 byte table up to F6ECH-F6F7H. IF the vector is set to 0, the decoder is not called.

The vector to the user decoder of ICFDSP must be rewritten to add the barcode decoder. 6B vector is stored in ICFDSP as shown below.

Name	Type of barcode	Address	No. of bytes	Contents
ICFDSP	0	F6ECH	2	Reserved for system expansion
	1	F6EEH	2	Vector to decoder for JAN/EAN/UPC-A/UPC-E
	2	F6F0H	2	Vector to decoder for 3-of-9
	3	F6F2H	2	Vector to decoder of codabar
	4	F6F4H	2	Vector to decoder for interleaved 2-of-5
	5	F6F6H	2	Vector to decoder for user

ICFDSP is referred in INTICF routine and vectors are called. The memory map is as shown below.



Decoders for the user are created as described below.

(1) Time of white or black is queried using the value of FRC(Free Running Counter). This can be queried by calling RDTIMR routine. ICFINT routine is set so that the time of black can be read when RDTIMR is first called. The time of white can be read when RDTIMR is called for a second time. After this, every time RDTIMR is called, the time of black and white can be read alternately.

```
RDTIMR
Address                      : AF00H
Input condition              : HL = overflow time
Output condition             : HL = FRC value (when CY-Flag=0)
                               CY = 1 ... timeout
Description:  Overflow time is the maximum wait time till
              polarity changes and is given by the value of FRC.
              If polarity does not change after wait overflow
              time, the scan by the barcode  reader is assumed to
              have completed or canceled.
```

(2) Decode the bar code to character by the time ratio of black and white. Store the decoded character in the temporary buffer. This can be performed by calling TPUTQ routine. The temporary buffer is initialized by ICFINT routine.

```
TPUTQ
Address             : AD57H
Input condition     : A = character
Output condition    : CY = 0 ... Normal end
                           1 ... Buffer overflow
Description:  The temporary buffer stores the decoded character
              temporarily.  The temporary buffer shares the area
              with the character buffer.  BIOS BARCODE read
              function takes one character out of the character
              buffer.  The temporary buffer pointer must be
              reassigned to the character buffer using SETQ
              routine.
```

(3) When decode completes without an error, reassign the temporary buffer pointer to the character buffer to store it as data. This is performed in SETQ routine. SETQ appends CR to the end of data after referring to the specification whether CR must be appended at BARCODE open.

```
SETQ
Address             : ADA3H
Input condition     : None
Output condition    : CY = 0 ... Normal end
                           1 ... Buffer overflow
```

11.5.2 Hand scanner

ICF interrupt routine does not support a hand scanner, driver for a hand scanner must be created. The hand scanner serial-transfers in ASCII code after it reads a barcode and self-decodes. The parameter at serial transfer is as shown below.

```
Communication rate ... 1200 bps
Start bit ............ 1 bit
Data length .......... 8 bit/char
Parity bit ........... No parity
Stop bit ............. 1 bit
```

The program which converts to characters using the barcode routine can be created by input of serial-transferred data to the bar code terminal. Start bit 1, data length 8, stop bit 1, so 10 bits per character is transferred. As 1200 bps, pulse width per bit is;

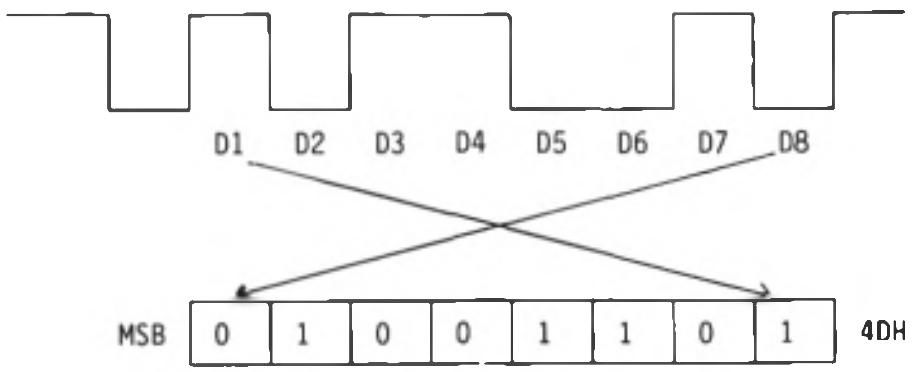
1/1200 = 833.3 micro Sec.

change this to the number of counts of the free running counter.

833.3 / 1.6276 = 512 counts

If "M" is transferred, the bit pattern is as shown below.

0 and 1 are reversed, so correct and convert to a character.



11.5.3 Work Area Details for Bar Code Decoder

Address : Variable name (Number of bytes)

F6ECH : ICFDSP (12)
- F6ECH : Reserved for system expansion (2)
- F6EEH : Vector to decoder for JAN/EAN/UPC-A/UPC-E (2)
- F6F0H : Vector to decoder for 3-of-9 (2)
- F6F2H : Vector to decoder for codabar (2)
- F6F4H : Vector to decoder for interleaved 2-of-5 (2)
- F6F6H : Vector to decoder for user (2)

F6F8H : BARFLG (1)
- Barcode open flag
  - 0 = Not open
  - 1 = Open

F6F9H : BCTYPE (1)
- Type of bar code
  - 0 = For system
  - 1 = JAN/EAN/UPC-A/UPC-E
  - 2 = 3-of-9
  - 3 = Codabar
  - 4 = Interleaved 2-of-5
  - 5 = For user

F6FAH : BCOPTN (1)

Option parameter at open

- bit 7 = Buzzer disable at normal read
  - =0 : Buzz
  - =1 : No buzz
- bit 6 = Delimiter disable
  - =0 : Delimiter enable
  - =1 : Delimiter disable
    (Use delimiter of TRMCHR)
- bit 5 = LED pulse control disable
  - =0 : Pulse control enable
  - =1 : Pulse control disable
- bit 4 = For future use
- bit 3 = For future use
- bit 2 = Zero addition specification
  (Valid only with UPC-E)
  - =0 : Zero not added
  - =1 : Zero added
- bit 1 = Full ASCII conversion specification
  (Valid only with 3-of-9)
  - =0 : Full ASCII conversion
  - =1 : No full ASCII conversion
- bit 0 = Check digit specification
  (Valid only with 3-of-9)
  - =0 : Last data is not assumed as check digit
  - =1 : Last data is assumed as check digit.

F6FBH : TRMCHR (1)

Used when delimiter, BCOPTN bit 6 = 0

F6FCH : SCNERR (1)

Error code detected at end

- bit 7 = Buffer overflow
- bit 6 = For future use
- bit 5 = For future use
- bit 4 = For future use
- bit 3 = For future use
- bit 2 = For future use
- bit 1 = For future use
- bit 0 = Scan error

F6FDH : LEDTIM (1)

Time for LED continuous lighting

- Default is 80H

F6FEH : LEDCNT (1)

Counter for LED continuous lighting

- 0 = Pulse lighting
- Other then 0 = continuous lighting

F6FFH : PLSFRQ (1)

Period of LED pulse flash

- Default is 8

F700H : PLSCNT (1)

Counter for LED pulse control

- 0 = LED on
- Other than 0 = LED off

F701H : PWRCNT (1)
Counter till logic power off
Logic power goes down 1 to 2 seconds after close When closed, this becomes 2

F702H : FRCBUF (2)
Address (FAAAH) of FRC (Free Running Counter) buffer address TIMBUF enters.

F704H : FRCSIZ (2)
Number of counters which can enter FRCBUF, it must be the value of 2^n, default is 64.

F706H : OVTIME (2)
Timeout counter value
Default is 65536/8

F708H : TIMCNT (2)
The last read FRC value

F70AH : TIMLOC (2)
Number of counters stored in FRC buffer

F70CH : BCGETP (2)
Get-pointer of character

F70EH : BCPUTP (2)
Put-pointer of character

F710H : BCLOC (1)
Number of characters in character buffer.

F711H : BCLOF (1)
Capacity empty in character buffer.

F712H : BCTPTP (2)
Pointer to store temporarily in character buffer.

F714H : BCTLOC (1)
Number of characters temporarily stored in character buffer.

F715H : BCTLOF (1)
Remaining capacity which can be temporarily stored in the character buffer.

F716H : SPCTHR (2)
Space threshold value (FRC value)

F718H : NRWSPC (2)
Narrow space value (FRC value)

F71AH : WIDSPC (2)
Wide space value (FRC value)

F71CH : BARTHR (2)
Bar threshold value (FRC value)

F71EH : NRWBAR (2)
Narrow space value (FRC value)

F720H : WIDBAR (2)
Wide space value (FRC value)

(F716H to F721H) : UPCBUF (12)
Work area for UPC-E zero addition

F722H : BITPIN (2)
Bit pattern of space and bar
SPCBIT (F722H) ... Space bit pattern
BARBIT (F723H) ... Bar bit pattern

F724H : DIRECF (2)
Direction flag or parity flag
LFLAG (F724H) ... Left parity for JAN
RFLAG (F725H) ... Right parity for JAN

F726H : MODULS (2)
Sum of module (JAN) or sum of check digit (3-of-9)

FA5AH : BCDBUF (80)
Character buffer Decoded bar code is stored in character (ASCII) units.

F822H : TIMBUF (128)
FRC (Free Running Counter) buffer. Value of FRC at change of black and white polarities is stored.

Part 2 HARDWARE

CHAPTER 1 Hardware Overview

1.1 Hardware Structure

1.1.1 Overview

EHT-10/EHT-10/2 uses 2 CPU types, C-MOS Z-80 compatible CPU (μPD70008) is used as the main CPU, and 4 bit C-MOS CPU7508 as a slave CPU. The slave CPU7508 controls key input, clock, power supply and performs serial communication with the main CPU. Main memory contains 256 KB (maximum) of RAM and 256 KB of ROM (128 KB for the system with a maximum of 128 KB for applications) for a total of 512 KB, switched by the bank register.

The main circuitry of the EHT-10/EHT-10/2 consists of 3 semi-custom LSIs (GAPNIO, GAPNIT, GAWIS), the I/O register in each custom LSI can be accessed by software using I/O instructions. Relation of the display and input of EHT-10 and EHT-10/2 is hardware dependent.

(1) EHT-10

Display : 154x84 dot LCD and LCD controller T6963.
2 KB is dedicated for VRAM.
Input : Touch panel with 14x5 input points

(2) EHT-10/2

Display : 120x32 dot LCD with the controller in GAWIS.
VRAM uses part of main RAM.
EHT-10/2 is available in a model (EHT-10/2B) with an EL back light.
Input : Keyboard with 34 keys and 3 LEDs

The EHT-10/EHT/10/2 includes RS-232C I/F, cartridge I/F, bar code I/F and IC card I/F for external interface connection. The cartridge I/F consists of 4 modes (HS mode, DB mode, IO mode, OT mode), and is structured for easy creation and control of each cartridge option.

The EHT-10 and EHT-10/2 power supply uses a nickel-cadmium battery pack as a main battery which is charged through an AC adapter . The nickel-cadmium sub-battery is built-in and performs RAM backup as a design safeguard, even when the main battery is discharged, RAM backup is supported.

Figure 1.1 shows EHT-10 and EHT-10/2 Hardware.

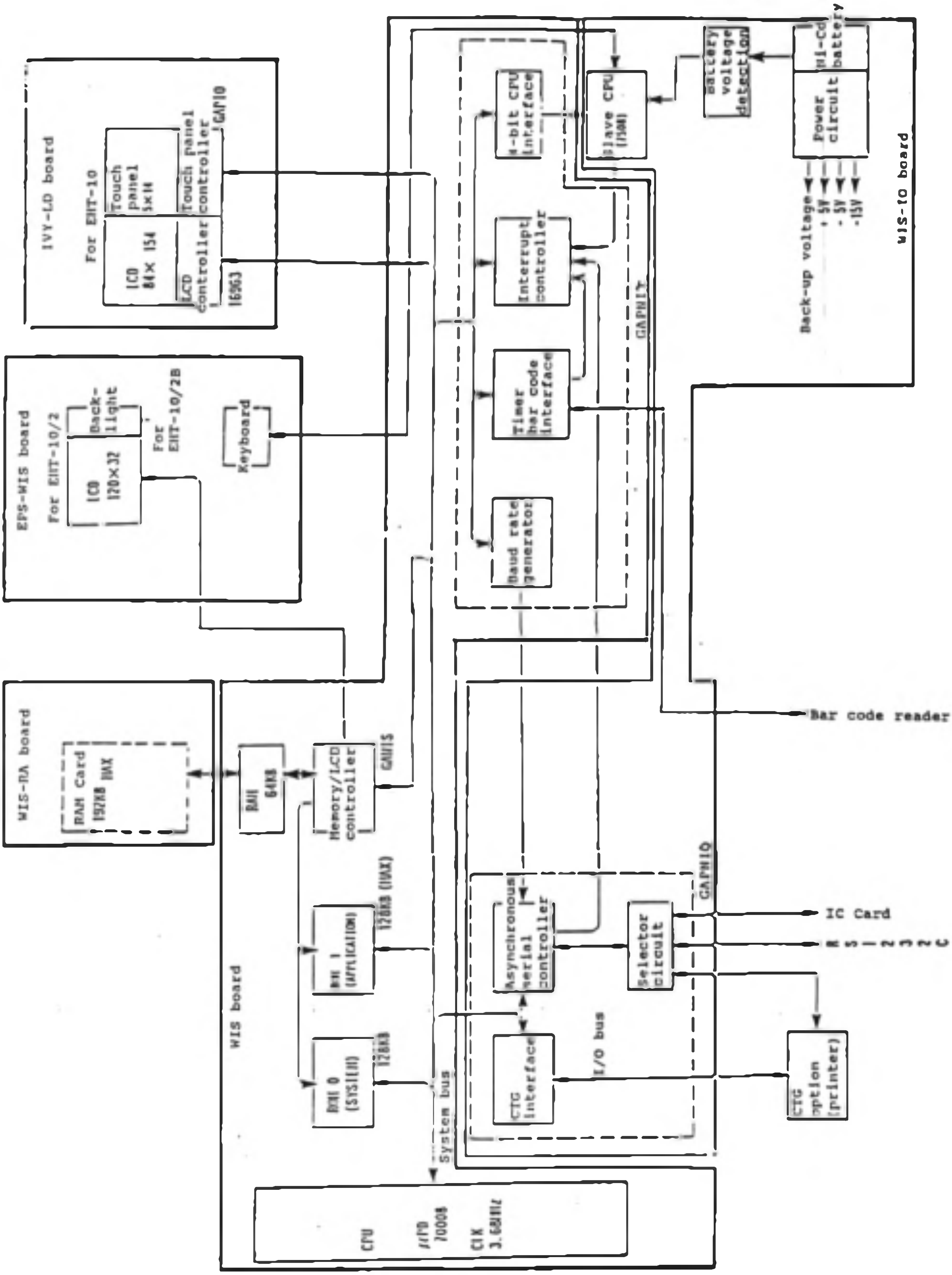


Fig 1.1 Hardware Structure

1.1.2 Hardware Structure

(1) Main CPU (µPD70008)

Z-80 compatible C-MOS CPU
CLK=3.6864 MHz

It sleeps via the HALT instruction optimizing power consumption. µPD70008 operates without WAIT, and neither DMA nor non-maskable interrupts are used.

(2) Memory

1. ROM

Mask ROM of 128 KB is available as the system ROM, mask ROM and EPROM of 128 KB, 64 KB and 32 KB are available as application ROM.

System ROM
µPD23C1000G ... CMOS 128 KB, mask ROM

Application ROM (use CMOS structure with access time faster than 200 ns).

128 KB EPROM : HN27C301G or ones based on this pin arrangement.
Mask ROM : HN62301P or ones based on this pin arrangement.

64 KB/32 KB EPROM : 27C256, 27C512
Mask ROM : Based on the EPROM pin arrangement mentioned above.

HN62301P HN27C301G

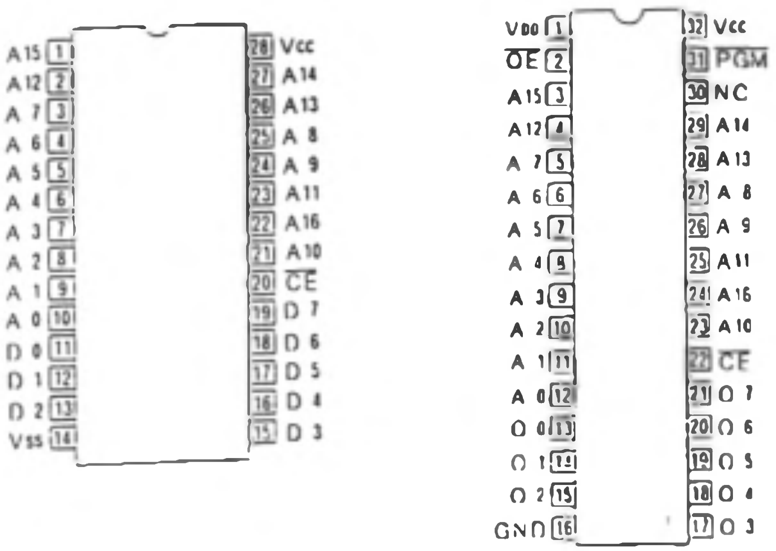


Fig 1.2 HN62301P/HN27C301G Pin arrangement (Top view)

2. RAM

CMOS 32 KB pseudo-static RAM is used. The standard arrangement of 64 KB with a maximum of 192 KB (every 64 KB additions are possible) using an additional RAM card brings the total of available memory to 256 KB.

Standard arrangement
CMOS 32 KB pseudo-static RAM μPD42832G x 2

RAM card
CMOS 32 KB pseudo-static RAM μPD42832G x 6

RAM controls read/write and refresh by G.A. (gate array) for memory control. Memory backup is performed for all RAM to avoid data loss at power-off.

(3) 7508 (slave CPU)

4 bit C-MOS CPU supports sleep timer functions.
CLOCK = approx. 270 kHz
Battery backup supports operation when the power switch is off.
Use: Clock function
Keyboard scan and control
Power on/off. reset signal control, battery voltage monitor
calendar, clock, alarm and timer function.

(4) GAWIS (memory control G.A.)

Control of pseudo-static RAM read/write and refresh. Read signal control of the system ROM and application ROM.
Timer interrupt
Control of LCD unit (EHT-10/2)

(5) GAPNIT (interrupt, timer control G.A.)

Interrupt controller
Timer & baud rate generator (with input capture function)
Interface with slave CPU (7508)
Barcode interface

(6) GAPNIO (I/O control G.A.)

ART (Asynchronous Receiver Transmitter)
Cartridge interface
RS-232C interface
LED interface
IC card interface

(7) GAPIO (touch panel control G.A. : Installed only in EHT-10

Gate array for touch panel interface.

(8) T6963 (LCD control installed only in EHT-10)

LCD controller

(9) RS-232C interface

Level : RS-232C level +-5 V
Bit transfer rate: 110, 150, 200, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 75 (bps)
Start bit : 1 bit
Stop bit : 1 or 2 bits
Parity : (Even, odd) parity/no parity
Error check : Parity error, framing error, overrun error
Communication Type : Full duplex

(10) Input

The touch panel for the EHT-10 and keyboard for the EHT-10/2 are available as input device. The major difference is as described below.

1 Touch panel (EHT-10)
Number of touch keys 5x14=70
Area of touch keys 10.5x9.35 mm

2 Keyboard (EHT-10/2)
Number of keys 34 keys
3 built-in LEDs indicate key mode display
7 character key buffer

Common functions of the EHT-10/EHT-10/2 touch panel and keyboard are as follows.
Auto repeat function
Repeat time reset function

(11) LCD display

LCD (liquid crystal) is used for the display of both the EHT-10 and EHT-10/2. The difference in LCDs for both devices is as described below.

1 EHT-10
1/48 duty liquid crystal
154x84 Dot matrix
Driver X:T7778x4
Y:T6961x1
Controller T6963
VRAM area 2 KB (independent of the memory in the main unit.)
Drive voltage Variable from 12 V to 18 V volume Controls view angle

2 EHT-10/2
1/32 duty liquid crystal
120x32 Dot matrix
Driver X:SED1180x2
Y:SED1190X1
Drive voltage Variable from 14 V to 19 V volume Controls view angle
VRAM area 512 bytes (of main memory)
Controller GAW1S (function of VRAM area redefinition and vertical dot scroll.)

(12) Buzzer

Piezo-electric buzzer is used with the selection of three frequencies; 512 Hz, 1024 Hz, 2048 Hz. In addition to this, various frequencies can be generated by writing "1" or "0". Buzzer drive voltage is +12 V.

(13) IC card interface

Read/write data is enabled by the IC card based on ISO DP7816. Write/read voltage is 5 V single.

(14) Interrupt

Mode 2 interrupt is used to support 5 types of interrupt in a prioritized order.

1.2 Address Map

The following 4 types of memory are placed in the memory address space of the EHT-10/EHT-10/2.

(1) RAM (64 KB) : Standard in the main unit
(2) System ROM : (128 KB)
(3) Application ROM (32/64/128 KB)
(4) Expansion RAM card (Maximum of 192 KB)

As the RAM capacity is 64 K + 192 K = 256 KB, the Z-80 with an address capacity of 64 KB cannot read/write the entire memory. The EHT-10/EHT-10/2 incorporates bank switching to alleviate this, supporting a large memory. The bank switching I/O port is shown in the table below above the actual description.

R/W	I/O Address	Register name	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Notes
W	P05H	BANKR	BANK 3	BANK 2	BANK 1	BANK 0	0	0	0	0	
W	P22H	SBKR	APBNK1	APBNK0	SYSBNK1	SYSBNK0	RAMBNK3	RAMBNK2	RAMBNK1	RAMBNK0	

Table 1.1 Bank Switch I/O Port

Care is required as there are 2 I/O ports for bank switching. Address map of EHT-10/EHT-10/2 changes according to the contents of the bank register.

1.2.1 Bank Switching

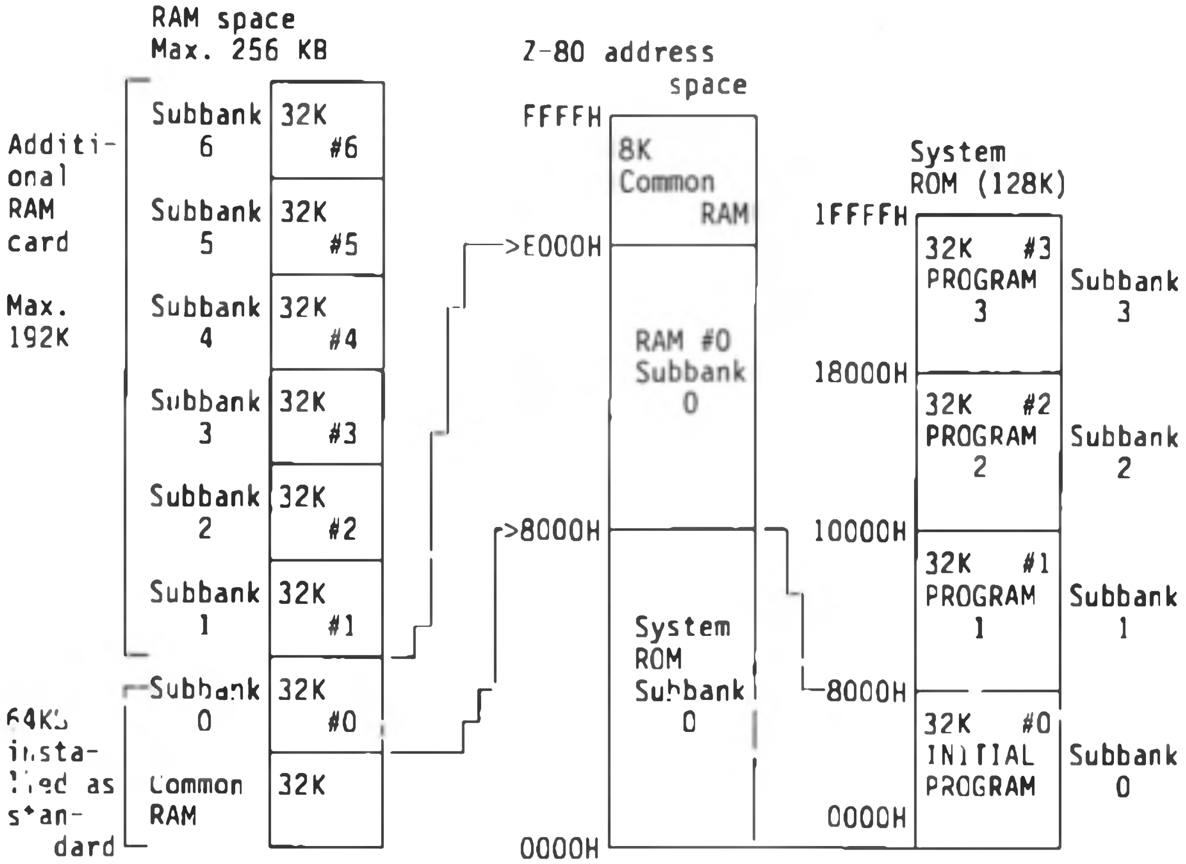
The memory concept divides the EHT-10/EHT-10/2 memory into 4, which gives an easy-to-operate environment to the OS and various application programs. These 4 environments are specified by I/O address P05H [bank register] and are named by each value as shown below.

BANK3	BANK2	BANK1	BANK0	Name
0	0	x	0	System bank
0	1	x	0	Bank 0
1	0	1	0	Bank 1
1	1	1	0	Bank 2

Either 1 or 0 can be entered in the boxes marked with an x. (Normally 0)

Table 1.2 Bank Name

(1) System Bank



(Notes)

1. RAM is divided into 8 areas every 32 KB and individually named.
2. Only 8 KB from 32 KB of common RAM can be used in the system.
3. System ROM of 128 KB is divided into 4 areas every 32 KB and named as sub-bank 0 to 3 as shown above.

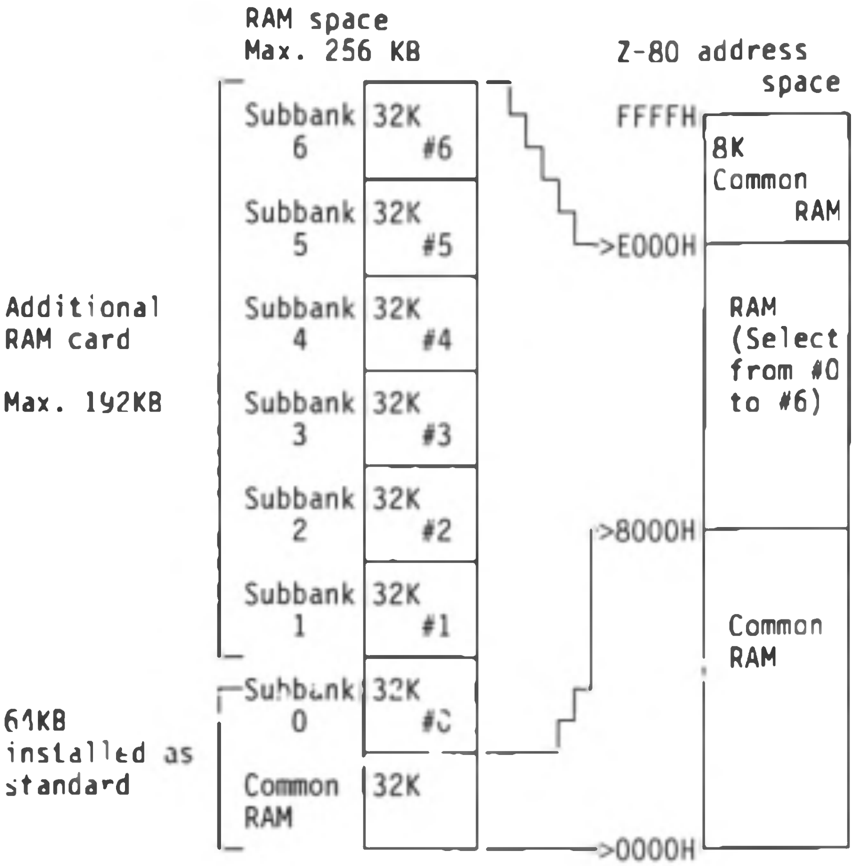
Fig 1.3 System Bank Memory Structure

The figure above is the memory structure of the system bank, showing 32 KB of system ROM from address 0000H to 7FFFH, and address 8000H to FFFFH is RAM area.

The system ROM is divided into 4 areas every 32 KB, when RAM is maximum (256 KB), it is divided into 8 areas every 32 KB.

The system ROM sub-bank 0, common RAM and RAM sub-bank 0 are selected for the system bank when EHT-10/EHT-10/2 is reset or the system bank is specified for the bank register (P05H).

(2) Bank 0



(Note)
One from RAM #0 to #6 is specified by sub-bank register [P22H]

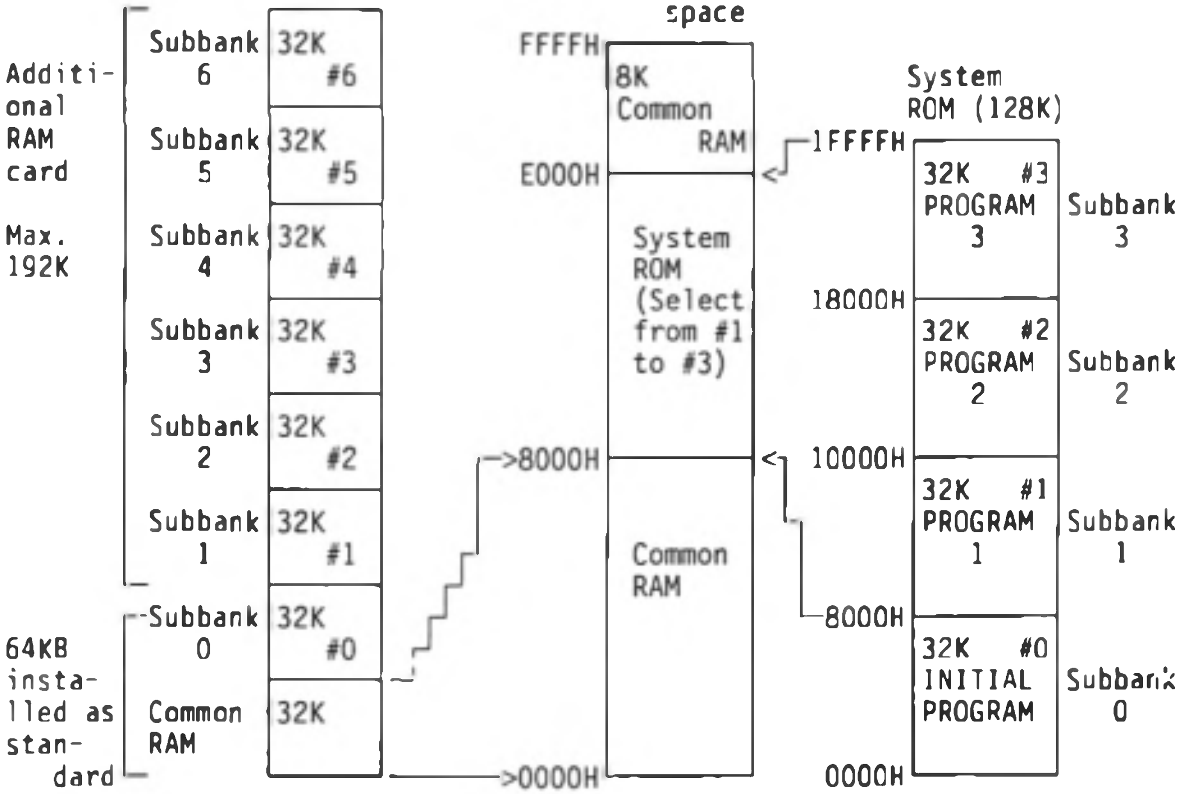
Fig 1.4 Bank 0 Memory Structure

The figure above shows the memory structure when bank 0. Bank 0 is sometimes called "all RAM mode" as the entire address space is RAM in the Z-80.
Address 0000H to 5FFFH and E000H to FFFFH is assigned to common RAM in this mode. One RAM (#0 to #6), divided into a maximum of 7 every 32 KB, can be assigned to address space of 6000H to DFFFH of the Z-80.
RAM is selected by the value of bit 0 to 3 (RAMBNK 0 to 3) of sub-bank register (P22H).
RAM (#0 to #6) selected by the value of RAMBNK 0 to 3 is shown in the next page.

RAMBNK3	RAMBNK2	RAMBNK1	RAMBNK0	Selected RAM
0	0	0	0	#0
0	0	0	1	#1
0	0	1	0	#2
0	0	1	1	#3
0	1	0	0	#4
0	1	0	1	#5
0	1	1	0	#6
0	1	1	1	This value should not be set.
1	x	x	x	

Table 1.3 sub-bank Selected by RAMBNK

(3) Bank 1



(Note)
1. RAM #0 to #6 are not used

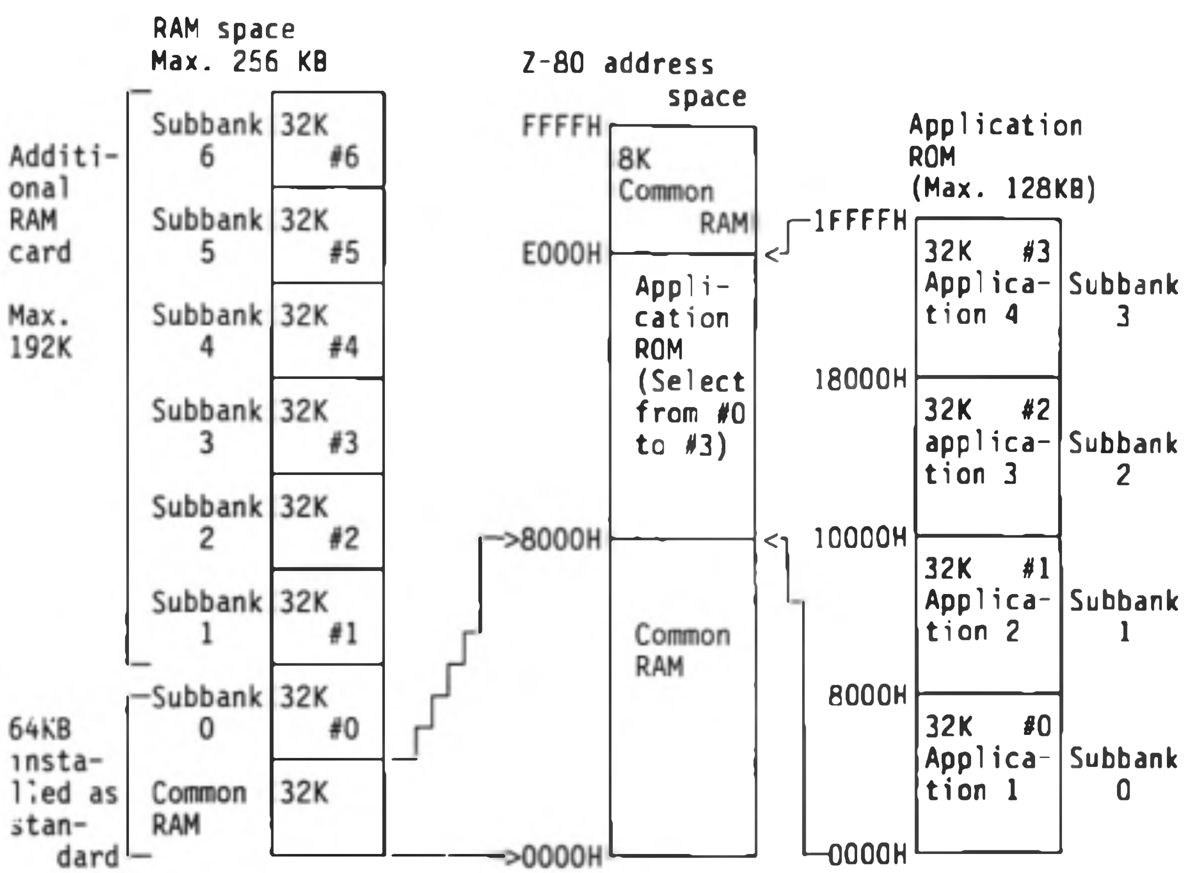
Fig 1.5 Bank 1 Memory Structure

The figure above shows the memory structure when bank 1. Address 0000H to 5FFFH and E000H to FFFFH is assigned to common RAM. One system ROM which has been divided into 3 every 32 KB can be assigned to the address space of 6000H to DFFFH. One sub-bank (1 to 3) is selected by the value of the sub-bank register (P22H) bit 4 or 5 (SYSBNK 0 or 1). The sub-bank selected by the value of SYSBNK 0 or 1 is shown in the table below. Both SYSBNK1 and SYSBNK0 cannot be 0 at the same time.

SYSBNK1	SYSBNK0	Selected sub-bank
0	0	This value should not be written
0	1	Sub-bank 1 (#1)
1	0	Sub-bank 2 (#2)
1	1	Sub-bank 3 (#3)

Table 1.4 Sub-bank Selected by SYSBNK

(4) Bank 2



(Note)
RAM #0 to #6 are not used

Fig 1.6 Bank 2 Memory Structure

The figure above shows the memory structure when bank 2. This structure is similar to bank 1 but the selected ROM is application ROM instead of system ROM. There are 3 application ROM types according to capacity; 32 KB, 64 KB and 128 KB. One sub-bank (0 to 3) is selected by the value of sub-bank register (P22H) bit 6 or 7 (APBNKO, 1). The sub-bank selected by the value of APBNKO or 1 is shown in the table below.

APBNK1	APBNKO	Selected sub-bank
0	0	Sub-bank 0 (#0)
0	1	Sub-bank 1 (#1)
1	0	Sub-bank 2 (#2)
1	1	Sub-bank 3 (#3)

Table 1.5 Sub-bank Selected by APBNK

1.2.2 Notes on Bank Switching

When a user's program is operating in RAM or ROM of EHT-10 and EHT-10/2 and the bank is switched, the bank register of the program must be in common RAM. However, common RAM is normally controlled by OS and a user's program cannot occupy common RAM. Therefore, when a user changes banks, OS BIOS call is used. See "Software 4.2 BIOS Details" for further information.

CHAPTER 2 I/O Register Description

2.1 I/O Address Table

A device placed in the I/O address space of the EHT-10/EHT-10/2 consists of GAPNIT, GAPNIO, GAPAWIS and GAPIO of 4 gate arrays, LCD controller T6963, and I/O device cartridge option.

Table 2.1 shows the contents of the I/O address space of EHT-10 and EHT-10/2. Access to addresses for future use from P00H to P26H in the table is disabled. Correct access of disabled addresses cannot be guaranteed. Port addresses and the corresponding device names are written in the table.

Table 2.1 EHT-10/EHT-10/2 I/O Address Table

I/O Address	Read 7	Read 6	Read 5	Read 4	Read 3	Read 2	Read 1	Read 0 (bit)	Write 7	Write 6	Write 5	Write 4	Write 3	Write 2	Write 1	Write 0 (bit)	Device
P00H	ICRL·C ICR·Low Command trigger								CTLR1 Control register								GAPNIT
	8-bit data								BRG3	BRG2	BRG1	BRG0	SWBCR	BCR1	BCR0	1	
P01H	ICRH·C ICR·HighCommand trigger								CMDR Command register								
	8-bit data													RES OVF	RES RDYSIO	SET RDYSIO	
P02H	ICRL·B ICR·Low Bar code trigger								CTLR2 Control register 2								
	8-bit data														0	ELON	
P03H	ICRH·B ICR·HighBar code trigger																
	8-bit data																
P04H	ISR Interrupt status register								IER Interrupt enable register								
				INT4	INT3	INT2	INT1	INT0				IER4	IER3	IER2	IER1	IER0	
P05H	STR Status register								BANKR Bank register								
	BANK3	BANK2	BANK1	BANK0	RDYSIO	RDY	BCRD		BANK3	BANK2	BANK1	BANK0	0	0	1	0	
P06H	SIOR Serial IO register								SIOR Serial IO register								
	8-bit data								8-bit data								
P07H																	NOTE: BANK3 and BANK2 are used by GAWIS.
P08H									VADR VRAMStart address register								GAWIS
										A14	A13	A12	A11	A10	A9		
P09H									YOFF Y offset register								
									DSP			Y4	Y3	Y2	Y1	Y0	
P0AH									FR Frame register								
													FR3	FR2	FR1	FR0	
P10H	CTG IF (CTG interface) address area								CTG IF (CTG interface) address area								GAPNIO
P11H																	
P12H																	
P13H																	

I/O Address	Read 7	6	5	4	3	2	1	0 (bit)	Write 7	6	5	4	3	2	1	0 (bit)	Device
P14H	ARTDIR ART data input register								ARTDOR ART data input register								GAPNIO
	7-,8-bit data								7-,8-bit data								
P15H	ARTSR ART status register								ARTMR ART mode register								
	RDSR		FE	OE	PE	Tx Emp	Rx RDY	Tx RDY	STOP		EVEN	PEN		DATA Length			
P16H	IOSTR IO status register								ARTCR ART command register								
		CSEL	RCTS	RCD	RXD	ICCLS		PBUSY			RRTS	ER	SBRK	RXE	RDTR	TXE	
P17H									ICCTLR IC Card control register								
									PW LCD	PR IE	IC ITE	OCTG	SC RP	ICDIR	ICC	ICCSC	
P18H									SWR Switch register								
												0	SSW1	SSW0	CSW1	CSW0	
P19H									IOCTLR IOControl register								
									SP	LED2	LED1	LED0	$\overline{PINI}$	ICR	$\overline{SLIN}$	PF	
P20H	LCDCDIR LCDC data input register								LCDCDOR LCDC data output register								LCDC
	8-bit data								8-bit data								
P21H	LCDCSTR LCDC status register								LCDCCR LCDC command register								
	8-bit data								8-bit data								
P22H									SBKR Sub Bank register								GAWIS
									APBN K1	APBN K0	SYS BNK1	SYS BNK0	RAM BNK3	RAM BNK2	RAM BNK1	RAM BNK0	
P23H	ITSR Interrupt status register								CTLR3 Control register								
							IPBU SY	OVF INT	SP1	SP0	OVFEN	OVF ITV	PINT DIS	W/P	OVF RESET		
P24H									TPSC Touch panel scan register								GAPIO
									DISC	0		TR5	TR4	TR3	TR2	TR1	
P25H	TPRT1 Touch panel return register																
	8-bit data																
P26H	TPRT2 Touch panel return register																
			6-bit data														

2.2 I/O Register Details

(1) P00H (Read)

ICRL.C (Input Capture Register Low Command Trigger)

bit	Name	Function
7	ICR7	Low order 8 bits of Input Capture Register
6	ICR6	
5	ICR5	
4	ICR4	
3	ICR3	
2	ICR2	
1	ICR1	
0	ICR0	

<Description>

1. Low order 8 bits of Input Capture Register. The moment this register is read, the contents of FRC (Free Running Counter) is latched (both high and low orders) to ICR (Input Capture Register). The value of high order is gained by reading address P01H (ICRH.C).

2. This register is used when reading the current FRC.

3. FRC and ICR are described in "3.5 OVF, ICF Interrupt".

(2) POOH (Write)

CRLR1 (Control Register 1)

bit	Name	Function
7	BRG3	Baud rate generator select
6	BRG2	
5	BRG1	
4	BRG0	
3	SWBCR	Power switch of +5V for barcode =1 : Power ON =0 : Power OFF
2	BCR1 (UP)	Barcode mode select
1	BCR0 (DOWN)	
0	-----------	Always set to 1

<Description>

1. BRG3 to 0 specifies serial transfer baud rate, BCR1 to 0 specifies ICR trigger polarity.

2. Baud Rate Generator Select

RBG	Send		Receive	
3 2 1 0	TXC	Bit rate	RXC	Bit rate
0 0 0 0	1.74545 K	110	1.74545 K	110
0 0 0 1	2.4	150	2.4	150
0 0 1 0	4.8	300	4.8	300
0 0 1 1	9.6	600	9.6	600
0 1 0 0	19.2	1200	19.2	1200
0 1 0 1	38.4	2400	38.4	2400
0 1 1 0	76.8	4800	76.8	4800
0 1 1 1	153.6	9600	153.6	9600
1 0 0 0	19.2	1200	1.2	75
1 0 0 1	1.2	75	19.2	1200
1 0 1 0	307.2	19200	307.2	19200
1 0 1 1	614.4	38400	614.4	38400
1 1 0 0	3.2	200	3.2	200

3. BarCode Mode Select

BCR1	BCR0	Trigger polarity
0	0	Trigger disable
0	1	Fall trigger
1	0	Rise trigger
1	1	Rise and fall trigger

<Notes>

In EHT-10 and EHT-10/2 OS, this register stores write data to CTLR1 in the system area RZCTLR1 (F0D6H) in RAM to be used at renew. Thus, when writing to CTLR, the value of RZCTLR1 is rewritten. Bit configuration of RZCTLR1 is the same as CTLR1.

(Ex.) This register turns on bar code.

```
LD          A,(RZCTLR1)
OR          08H
LD          (RZCTLR1),A
OUT         (CTLR1),A
```

In OS, this register modifies the value of the baud rate (BRG3 to 0) by BIOS RSIOX, floppy disk or IC card access.

(3) P01H (READ)

ICRH.C (Input Capture Register High Command Trigger)

bit	Name	Function
7	ICR15	High order 8 bits of Input Capture Register
6	ICR14	
5	ICR13	
4	ICR12	
3	ICR11	
2	ICR10	
1	ICR9	
0	ICR8	

<Description>

This register is used when reading the current FRC.
However, the values of ICR15 to ICR8 are latched at the moment ICRL.C (P00H) is read. Thus, ICRL.C must be the first one to read.

(4) P01H (Write)

CMDR (Command Register

bit	Name	Function
7	-----------	Don't care.
6	-----------	
5	-----------	
4	-----------	
3	-----------	
2	RESOVF	=1 : Resets OVF interrupt generated by FR overflow. =0 : Nothing is performed.
1	RESRDYSIO	=1 : Resets RDYSIO signal [P05H bit 3] used for communication with 7508. =0 : Nothing is performed.
0	SETRDYSIO	=1 : Sets RDYSIO signal used for communication with 7508. =0 : Nothing is performed. Normally this bit is not used.

<Notes>

In EHT-10 and EHT-10/2 OS, this register uses RES OVF in OVF interrupt process and uses RES RDYSIO in the communication process with 7580.

(5) P02H (Read)

ICRL.B (Input Capture Register Low Bar Code Trigger)

bit	Name	Function
7	ICR7	Low order 8 bits of Input Capture Register at signal change.
6	ICR6	
5	ICR5	
4	ICR4	
3	ICR3	
2	ICR2	
1	ICR1	
0	ICR0	

<Description>

1. The value of the low order 8 bits of ICR which is latched from FRC to ICR by a signal cnange (rise and fall trigger) from the bar code reader.

2. INT2 signal (ICF) becomes active when the signal changes from the bar code reader.

(6) P02H (Write)

CTLR2 (Control Register 2)

bit	Name	Function
7	-----------	Don't care.
6	-----------	
5	-----------	
4	-----------	
3	-----------	
2	-----------	
1	-----------	Always set to 0
0	ELON	EL panel ON/OFF =0 : OFF =1 : ON

<Description>

This register controls on/off of the back light (EL panel) for the EHT-10/2 in software. However, the power switch must be at B.L position to operate.

(7) P03H (Read)

ICRH.B (Input Capture Register High Bar Code Trigger)

bit	Name	Function
7	ICR15	High order 8 bits of Input Capture Register at signal change
6	ICR14	
5	ICR13	
4	ICR12	
3	ICR11	
2	ICR10	
1	ICR9	
0	ICR8	

<Description>

1. The value of the high order 8 bits of ICR which is latched from FRC to ICR by a signal change (rise and fall trigger) from the bar code reader.

2. Reading this register resets INT2 signal (ICF) generated by change of signal from the barcode reader.

(8) P04H (Read)

ISR (Interrupt Status Register)

bit	Name	Function
7	-----------	Always set to 0
6	-----------	
5	-----------	
4	INT4(EXT)	Interrupt signal from the cartridge, IC card, or timer of 1 msec or 8 msec. =1 : Interrupt request =0 : No interrupt request
3	INT3(OVF)	Interrupt signal generated by FRC overflow. Write 1 to RESOVF [P01H bit 2] to reset. =1 : Interrupt request =0 : No interrupt request
2	INT2(ICF)	This interrupt signal is generated at the same time when the value of FRC is latched to ICR by a change (rise and fall) from the bar code signal. However, this interrupt is not generated when both BCR1 and BCR0 [P00H] are 0. Read ICRH.B [P03H] to reset. =1 : Interrupt request =0 : No interrupt request
1	INT1(ART)	Interrupt signal generated when ART RXRDY=1 (receive data enters). Read receive data from ART to reset. =1 : Interrupt request =0 : No interrupt request
0	INT0(7508)	Interrupt signal generated from 7508. Respond to 7508 to reset. =1 : Interrupt request =0 : No interrupt request

<Description>

INT4 to INT0 can be read when an interrupt is masked. INT0 has the highest priority and INT4 the lowest priority of interrupt priority order.

See "Chapter 3 Interrupt Description" for details.

(9) P04H (Write)

IER (interrupt Enable Register)

bit	Name	Function
7	-----------	Don't care.
6	-----------	
5	-----------	
4	IER4(EXT)	INT4 (EXT) interrupt =1 : Enable =0 : Disable
3	IER3(OVF)	INT3(OVF) interrupt =1 : Enable =0 : Disable
2	IER2(ICF)	INT2 (ICF) interrupt =1 : Enable =0 : Disable
1	IER1(ART)	INT1 (ART) interrupt =1 : Enable =0 : Disable
0	IER0(7508)	INT0 (7508) interrupt =1 : Enable =0 : Disable

<Description>

Each bit corresponds to an interrupt and sets enable/disable.

<Notes>

In EHT-10 and EHT-10/2 OS, this register saves write data to IER in the system area RZIER (F42EH) in RAM to be used at renew. Thus, when writing to IER, the value of RZIER is rewritten. Bit configuration of RZIER is the same as IER.

(Ex.) This register enables EXT interrupt.

```
DI
LD      A, (RZIER)      RZIER:(F42EH)
OR      10H
LD      (RZIER), A
OUT     (IER), A
EI
```

In EHT-10 and EHT-10/2 OS, this register supports masked interrupt control using BIOS MASKI.
At normal state, 7508 and OVF interrupts are enabled, ART interrupt is enabled when OPEN is executed at BIOS RSIOX, it is disabled when CLOSE is executed.
ICF interrupt is disabled.
EXT can be masked individually by an interrupt factor.

Interrupt from cartridge -> ICCTLR [P17H bit 6]
Interrupt from IC card -> ICCTLR [P17H bit 5]
1 msec/8 msec timer interrupt -> [P23H bit 5]

(10) P05H (Read)

STR (Status Register)

bit	Name	Function
7	BANK3	Main memory bank register
6	BANK2	
5	BANK1	
4	BANK0	
3	RDYSIO	This serial bus control signal is an I/F with the 7508. =1 : Access enable to 7508 =0 : Access disable to 7508
2	RDY	RDY signal from 7508. Not used normally.
1	BCRD	Data input signal of bar code reader.
0	----------	Don't care.

<Description>

See "1.2 Address Map" for the value of BANK 3 to 0.

<Notes>

1. Value of RDYSIO is 1 directly after power on. RDYSIO signal must be reset using RESRDYSIO [P01H bit 1] when data or command is set using SIOR [P06H] or after data is received.

2. Signal from the bar code reader can be read through BCRD directly.

(11) P05H (Write)

BANKR (Bank Register)

bit	Name	Function
7	BANK3	Main memory bank register
6	BANK2	
5	BANK1	
4	BANK0	
3	-----------	Always set to 0
2	-----------	
1	-----------	Always set to 1
0	-----------	Always set to 0

<Description>

See "1.2 Address Map" for values of BANK 3 to 0.

<Notes>

1. In EHT-10 and EHT-10/2 OS, this register saves write data to BANKR in system area RZBANKR (F42CH) in RAM to be used at renew. Thus, when writing to BANKR, the value of RZBANKR is rewritten. Bit configuration of RZBANKR is the same as BANKR.

2. LOADX, STORX, LDIRX, JUMPX and CALLX are provided in EHT-10 and EHT-10/2 OS as BIOS for bank control.

(12) P06H (Read/Write)

SIOR (Serial I/O Register)

bit	Name	Function
7	SI07	Register for 7508 data send/receive.
6	SI06	
5	SI05	
4	SI04	
3	SI03	
2	SI02	
1	SI01	
0	SI00	

<Description>

At read : 8 bit data which is received after parallel conversion of serial data transferred from 7508.

At write : 8 bit data serial-transferred to 7508.

(13) P08H (Write)

VADR (VRAM Start Address Register (EHT-10/2 only)

bit	Name	Function
7	-----------	Don't care.
6	A14	VRAM start address
5	A13	
4	A12	
3	A11	
2	A10	
1	A9	
0	-----------	Don't care.

<Description>

VADR is the register to specify the VRAM area and specifies the high order 6 bits of the address.
Due to this, 512 bytes from (A14, A13, A12, A11, A10, A9 0 0000 0000) to (A14, A13, A12, A11, A10, A9 1 1111 1111) are assigned as the VRAM area.
VRAM address can be set in 512 byte units.

<Notes>

VRAM address must be fetched within 0000H to 5FFFH and E000H to FFFFH.

(14) P09H (Write)

YOFF (Y Offset Register) (EHT-10/2 only)

bit	Name	Function
7	DSP	This register controls the LCD panel display. =1 : Screen display =0 : No screen display
6	-----------	Don't care.
5	-----------	
4	Y4	Offset value in Y direction
3	Y3	
2	Y2	
1	Y1	
0	Y0	

<Description>

This register specifies the corresponding relation of VRAM and LCD in the vertical direction. Display starts at the point moved by YOFF dot of VRAM. When the display reaches the bottom of VRAM, it returns to the top of VRAM and displays YOFF=1 dot line to end a screen.

<Notes>

EHT-10/2 OS uses Y Offset Resister scrolling LCD screen vertically. The value of the current YOFF is saved in LVRAMYOFF (F266H).

(15) POAH (Write)

FR (Frame Register) (EHT-10/2 only)

bit	Name	Function
7	-----------	Don't care.
6	-----------	
5	-----------	
4	-----------	
3	FR3	Frame Register
2	FR2	
1	FR1	
0	FR0	

<Cescription>

This register regulates the LCD frame frequency. The relation of Frame Register and frame frequency is as listed below.

FR				LCD frame frequency (Hz)
FR3	FR2	FR1	FR0	3.68 M Hz
0	1	0	0	450
0	1	0	1	360
0	1	1	0	300
0	1	1	1	257
1	0	0	0	225
1	0	0	1	200
1	0	1	0	180
1	0	1	1	164
1	1	0	0	150
1	1	0	1	138
1	1	1	0	128

(0000) to (0011) and (1111) are disabled.

<Notes>

Fixed value OEH is set to FR in EHT-10 and EHT-10/2 OS at power on, reset or system initialization.

(16) P10H to P13H

<Description>

P10H to P13H is address space for the cartridge interface, the configuration differs depending on cartridge mode (HS, DB, IO and OT mode).

See "4.1 Cartridge Interface" for further information about P10H to P13H.

(17) P14H (Read)

ARTDIR (ART Data Input Register)

bit	Name	Function
7	RD7	Receive data
6	RD6	
5	RD5	
4	RD4	
3	RD3	
2	RD2	
1	RD1	
0	RD0	

<Description>

Serial data input from RXD is parallel-converted to be fetched.
When the data length is 7 bits, 0 is input to bit 7 (RD7).

<Notes>

1. In EHT-10/EHT-10/2 OS, Interrupt routine processes data receive and BIOS RSIOX interfaces with application of receive data.

2. EHT-10 and EHT-10/2 has 3 serial interfaces; RS-232C, IC card reader and cartridge interface. However, the register for send/receive data has only one send/receive data register. Thus, you must select which interface to use this register for before register access. SWR (Switch Register P18H) can be used for this occasion.

(18) P14H (Write)

ARTDOR (ART Data Output Register)

bit	Name	Function
7	TD7	Send data
6	TD6	
5	TD5	
4	TD4	
3	TD3	
2	TD2	
1	TD1	
0	TD0	

<Description>

This data is converted to serial data and is output from TXD. When the data length is 7 bits, bit 7 (TD7) is "Don't care".

<Notes>

1. BIOS RSIOX interfaces with the application for serial data send process in EHT-10 and EHT-10/2 OS.

2. See (17) <Notes>.

(19) P15H (Read)

ARTSR (ART Status Register)

bit	Name	Function
7	RDSR	DSR (Data Set Ready) signal. When RS-232C DSR terminal is active, RDSR=1.
6	-----------	Always 0
5	FE	Framing error (occurs when FE=1).
4	OE	Overrun error (occurs when OE=1)
3	PE	Parity error (occurs when PE=1).
2	TXEMP	This register indicates no data in the send section. It is set when send data buffer ARTDOR and the parallel/serial transducer are both empty.
1	RXRDY	When RXRDY=1, INT1 (ART) interrupt is requested to CPU (Z-80) and it indicates receive data is accepted. RXRDY is reset when the receive buffer ARTDIR (P14H) is read, it is also reset by reset input or error reset command.
0	TXRDY	This register is set when the output buffer ARTDOR is empty. It is reset when data is written to the output buffer ARTDOR.

<Description>

This register is equal to Status Register of 8251 (USART).

FE (bit 5) Framing error does not effect data receive operation, if the next data is fetched sequentially, framing error is checked for new data. FE is reset if the stop bit is correct.

OE (bit 4) Overrun error does not effect data receive operation, OE is not reset even if next data is fetched correctly. Error reset command (ER=1) or reset input is necessary for OE reset.

PE (bit 3) PE reset condition is the same as FE. Parity is checked only when PEN=1. PE is 0 when PEN=0.

(20) P15H (Write)

ARTMR (ART Mode Register)

bit	Name	Function
7	STOP	Specifies the number of stop bits. =1 : 2 bits =0 : 1 bit
6	-----------	Don't care.
5	EVEN	Specifies parity Even/Odd. (Valid only when PEN=1.) =1 : Even parity =0 : Odd parity
4	PEN	Specifies Parity/No parity. =1 : Parity =0 : No parity
3	-----------	Don't care.
2	DATA LENGTH	Specifies data length. =1 : 8 bit =0 : 7 bit
1	-----------	Don't care.
0	-----------	

<Description>

This register is equal to Mode Instruction Register of 8251 (USART).

<Notes>

In EHT-10 and EHT-10/2 OS, this register saves write data to ARTMR in the system area RZARTMR (F0D9H) in RAM to be used at renew. Thus, when writing to ARTMR, the value of RZARTMR is rewritten. Bit configuration of RZARTMR is the same as ARTMR.
Output to ARTMR is supported by BIOS RSIOX. Set TXE[P16H bit 0 W] and RXE[P16H bit 2 W] to 0 in advance to renew this register.

(21) P16H (Read)

IOSTR (IO Status Register)

bit	Name	Function
7	-----------	Don't care.
6	CSEL	Signal to identify the cartridge option type. =1 : HS (Hand Shake) mode =0 : Other mode
5	RCTS	RS-232C CTS signal This signal becomes 1 when RS-232C CTS signal is active.
4	RCD	RS-232C CD signal This signal becomes 1 when RS-232C CD signal is active.
3	RXD	Serial data input signal.
2	ICCLS	This signal becomes 0 when excess current goes through the IC card or IC card reader cover is open.
1	-----------	Don't care.
0	PBUSY	Interrupt request from the cartridge option can be checked using the cartridge interface CINT signal. =0 : Interrupt request from cartridge option =1 : No interrupt request from cartridge option

<Notes>

1. See "4.3.7 Notes" for RCTS, RCD and RXD signals. Serial data from RS-232C, IC card and cartridge interface can be directly read using RXD.

2. See "3.6 EXT Interrupt" for ICCLS and PBUSY.

(22) P16H (Write)

ARTCR (ART Command Register)

bit	Name	Function
7	-----------	Don't care.
6	-----------	
5	RRTS	RS-232C RTS signal RS-232C RTS signal becomes active when RRTS=1.
4	ER	OE, FE and PE are reset. ER is set to 1 when RXE=1. ER is pulse output, pulse generates only while writing with ER=1. Therefore, ER=0 does not need to be returned after ER=1.
3	SBRK	Break output TXD is forced set to 0 when SBRK=1. (Valid when TXE=1)
2	RXE	This enables data receive. =1 : Enable =0 : Disable
1	RDTR	RS-232C DTR signal RS-232C DTR signal becomes active when RDTR=1.
0	TXE	This enables data send. =1 : Enable =0 : Disable TXD=1 (mark) while TXE=0

<Description>

Bit 0 to 5 of this register is equal to Command Instruction Register of 8251 (USART).

<Notes>

In EHT-10 and EHT-10/2 OS, this register saves write data to ARTCR in the system area RZARTCR (F0DAH) in RAM to be used at renew. Thus, when writing to ARTCR, the value of RZARTCR is rewritten. Bit configuration of RZARTCR is the same as ARTCR.
Data output to ARTCR is supported at BIOS RSIOX.

(23) P17H (Write)

ICCTLR (IC Card Control Register)

bit	Name	Function
7	PLCD	LCD power supply =0 : Supplied =1 : Not supplied
6	PRIE	Signal from cartridge I/F CINT =0 : Enable =1 : Disable
5	ICITE	IC card interrupt enable =0 : Disable =1 : Enable
4	DCTG	Development cartridge switch =0 : Normal operation =1 : Development cartridge access enable
3	BCRP	Par code reader LED control =0 : OFF =1 : ON
2	ICDIR	IC card Read/Write switch =0 : Write =1 : Read
1	ICC	IC card power control =0 : OFF =1 : ON
0	ICOSC	IC card clock control =0 : Oscillation stops =1 : Oscillation

<Notes>

1. As this register is not effected by the reset signal, 54H must be set in the initialize routine.

2. Hold 15 msec from the IC card clock oscillation start till it becomes stable.

(24) P18H (Write)

SWR (Switch Register)

bit	Name	Function
7	-----------	Don't care.
6	-----------	
5	-----------	
4	-----------	Always set to 0
3	SSW1	Serial mode
2	SSW0	
1	CSW1	Cartridge interface mode
0	CSW0	

<Description>

This register sets the serial interface mode and cartridge interface mode.

Serial mode

SSW1	SSW0	RXD	TXD
0	0	Cartridge	Cartridge
0	1	IC card	IC card
1	0	RS-232C	RS-232C
1	1	This value should not be set.	

Cartridge interface mode

CSW1	CSW0	Mode
0	0	HS (Hand Shake) mode
0	1	IO (Input Output) mode
1	0	DB (Data Bus) mode
1	1	OT (Output port) mode

<Notes>

In EHT-10 and EHT-10/2 OS, this register saves write data to SWR in the system area RZSWR (F0DCH) in RAM to be used at renew. Thus, when writing to SWR, the value of RZSWR is rewritten. Bit configuration of RZSWR is the same as SWR.

(Ex.)

This register switches to IO mode.

```
  LD      A, (RZSWR)      RZSWR : (F0DCH)
  AND     0FCH            SWR   : (18H)
  OR      01H
  LD      (RZSWR), A
  OUT     (SWR), A
```

Serial mode switching is supported at BIOS RSIOX.

(25) P19H (Write)

IOCTLR (IO Control Register)

bit	Name	Function
7	SP	Output signal to speaker. Set SP=0 when not used.
6	LED2	LED output port
5	LED1	
4	LED0	
3	$\overline{\mathrm{PINI}}$	Cartridge set signal This signal is used to reset the cartridge for software. Cartridge is reset when $\overline{\mathrm{PINT}}$=0.
2	ICR	IC card reset signal =1 : Reset ON =0 : Reset OFF
1	$\overline{\mathrm{SLIN}}$	This signal controls cartridge interface CCTL1. Printer select-in output. Select-in when $\overline{\mathrm{SLIN}}$=0.
0	PF	This signal controls cartridge interface CCTL0. This form-feeds the printer.

<Notes>

1. In EHT-10 and EHT-10/2 OS, this register saves write data to IOCTLR in the system area RZIOCTLR (F0DDH) in RAM to be used at renew. Thus, when writing to IOCTLR, the value of RZIOCTLR is rewritten. Bit configuration of RZIOCTLR is the same as IOCTLR.
In OS, SP is used in BIOS BEEP process, LED 2 to 0 are used in key input process. Control LED 2 to 0 is supported by BIOS CONOUT (ESC+A0H to A5H)

2. $\overline{\mathrm{PINT}}$, $\overline{\mathrm{SLIN}}$ and PF are general purpose outputs and can be used when a user created cartridge is used. The functions mentioned above are enabled only when a dedicated printer for EHT-10 and EHT-10/2 is used.

3. When form-feed is performed in the printer, set $\overline{\mathrm{SLIN}}$ to disable, PF to enable; $\overline{\mathrm{SLIN}}$=1, PF=1. See "4.1 Cartridge Interface" if you would like to use $\overline{\mathrm{PINT}}$, SLIN or PF for general purpose output.

(26) P20H (Read) (EHT-10 only)

LCDCDIR (LODC Data Input Register)

bit	Name	Function
7	-----------	Data read register from LCD controller T6963
6	-----------	
5	-----------	
4	-----------	
3	-----------	
2	-----------	
1	-----------	
0	-----------	

<Description>

This register is to read data from LCDC controller T6963.

1. This register is valid for EHT-10 only.

2. Check LCDCSTR[P21H] status bit whether it can be read before data read from this register.

(27) P20H (Write) (EHT-10 only)

LCDCDOR (LCDC Data Output Register)

bit	Name	Function
7	-----------	Data write register from LCD controller T6963
6	-----------	
5	-----------	
4	-----------	
3	-----------	
2	-----------	
1	-----------	
0	-----------	

<Description>

This register is to write data to the LCD controller T6963.

<Notes>

1. This register is valid for EHT-10 only.

2. Check LCDCSTR[P21H] status bit whether it can be written before data write to this register.

(28) P21H (Read) (EHT-10 only)

LCDCSTR (LCD Status Register)

bit	Name	Function
7	STA7	To check the blink state (Approximate cycle 1 second duty 50%) =1 : Normal display =0 : OFF
6	STA6	If a point other than on the real screen is set while copy, the flag is set. If the instruction is executed, the flag is reset.
5	STA5	To check whether controller is operatable. =1 : Operatable =0 : Not operatable
4	STA4	Don't care.
3	STA3	To check whether data can be written (Valid only when AUTO). =1 : Write enable =0 : Write disable
2	STA2	To check whether data can be read (Valid only when AUTO). =1 : Read enable =0 : Read disable
1	STA1	To check whether data can be written or read. =1 : Data enable =0 : Data disable (during internal process)
0	STA0	To check whether the instruction is executable. =1 : Executable =0 : Not executable (during instruction execution).

<Description>

1. LCDC status can be read using this register. Check for STA0=STA1=1 when writing a command to LCDC [P21H W] for timing. Check for STA0=STA1=1 also when writing data to LCDC.

2. When T6963 (LCDC) is in AUTO mode, it can be performed only with identification of STA2 or STA3.

(29) P21H (Write) (EHT-10 only)

LCDCCR (LCDC Command Register)

bit	Name	Function
7	LCDCMD7	Command Register to LCD controller T6963.
6	LCDCMD6	
5	LCDCMD5	
4	LCDCMD4	
3	LCDCMD3	
2	LCDCMD2	
1	LCDCMD1	
0	LCDCMD0	

<Description>

This register gives command to the LCDC controller T6963.

<Notes>

1. This register is valid for EHT-10 only.

2. Check whether command write is enabled with LCDCSTR[P21H] status bit before command is given to this register.

See "4.6.4 T6963 Command" for T6963 command table.

(30) P22H (Write)

SBKR (SUBBANK Register)

bit	Name	Function
7	APBNK1	Application ROM sub-bank = 00 : sub-bank 0 = 01 : sub-bank 1 = 10 : sub-bank 2 = 11 : sub-bank 3
6	APBNK0	
5	SYSBNK1	System ROM sub-bank = 00 : sub-bank 0 = 01 : sub-bank 1 = 10 : sub-bank 2 = 11 : sub-bank 3
4	SYSBNK0	
3	RAMBNK3	RAM sub-bank = 0000 : sub-bank 0 = 0001 : sub-bank 1 = 0010 : sub-bank 2 = 0011 : sub-bank 3 = 0100 : sub-bank 4 = 0101 : sub-bank 5 = 0111 : sub-bank 6 = 0111 : sub-bank 7
2	RAMBNK2	
1	RAMBNK1	
0	RAMBNK0	

<Description>

This register switches memory space for the CPU (Z-80). See "1.2 Address Map" for further information.

<Notes>

1. This register is not affected by reset.

2. In EHT-10 and EHT-10/2 OS, this register saves write data to SBKR in the system area RZSBBNKR (F42DH) in RAM to be used at renew. Thus, when writing to SBKR, the value of RZSBBNKR is rewritten. Bit configuration of RZSBBNKR is the same as SBKR.

(31) P23H (Read)

ITSR (Interrupt Status Register)

bit	Name	Function
7	-----------	Don't care.
6	-----------	
5	-----------	
4	-----------	
3	-----------	
2	-----------	
1	IPBUSY	Interrupt request signal from IC card and cartridge option. =1 : No interrupt request =0 : Interrupt request
0	OVFINT	Interrupt request signal from timer of 1 msec or 8 msec. =1 : Overflow (interrupt request) =0 : No overflow (no interrupt request)

<Description>

1. See "Chapter 3 Interrupt Description" for OVFINT signal.

2. Interrupt from IC card and cartridge option, and timer interrupt of 1 msec and 8 msec cause INT4(EXT) interrupt. Even when INT4 (EXT) interrupt mask [P04H bit 4] is performed, interrupt can be checked using this status bit IPBUSY or OVFINT .

(32) P23H (Write)

CTLR3 Control Register 3

bit	Name	Function
7	SP1	Speaker ON/OFF and frequency switching. =00 : OFF =10 : 1024 Hz =01 : 512 Hz =11 : 2048 Hz
6	SP0	
5	OVFEN	1 msec, 8 msec interrupt enable =0 : Disable =1 : Enable
4	OVFITV	1 msec, 8 msec interrupt frequency select =0 : 8 msec =1 : 1 msec
3	PINTDIS	External interrupt enable =0 : Enable =1 : Disable
2	W/P	Always set to 1
1	OVFRESET	1 msec, 8 msec timer interrupt reset =1 : Reset
0	-----------	Don't care.

<Description>

1. SP0, SP1 Any of the 3 types of frequencies can be selected by changing this value. In this case, set SP[P19H bit 7] to 0.
When the signal is output to the speaker using SP, set SP0 and SP1 to 0.

2. OVFRESET Set this bit to 1 to reset timer interrupt (when OVFEN=1). When OVFRESET=1, write pulse generates, OVFRESET=0 does not need to be returned after OVFRESET=1.

<Notes>

These registers become all 0 after reset.
In EHT-10 and EHT-10/2 OS, this register saves write data to CTLR3 in the system area RZCTLR3 (F0DEH) in RAM to be used at renew. Thus, when writing to CTLR3, the value of RZCTLR3 is rewritten. Bit configuration of RZCTLR3 is the same as CTLR3.

(33) P24H (Write)

TPSC (Touch Panel Scan Register)

bit	Name	Function
7	DISC	Discharge of the touch panel charge
6	-----------	Don't care.
5	-----------	Always set to 0
4	TR5	Touch panel scan output data
3	TR4	
2	TR3	
1	TR2	
0	TR1	

<Description>

1. When the touch panel scan output data is entered TR1 to TR5, the bit is set to P25H or P26H corresponding to the pressed key. This shows which touch panel has been pressed.

Data to set

(a) When interrupt wait
Set TR1 to TR5 to 1.
DATA 10011111

(b) When scan
Set DISC to 1 and SET one bit of TR1 to TR5 to 1.
DATA 100 ....

(c) When discharge
Set all to 0
DATA 00000000

When scan, set one bit of TR1 to TR5 to 1 and set the remaining bits to 0. However, when writing 0 to this register, the corresponding bit does not become 0 but becomes high impedance, thus, the charge remains in polarity.
To discharge this, the operation in item (c) mentioned above is necessary.

See "4.5 Touch Panel I/F" for touch panel scan flow.

(34) P25H (Read)

TPRT1 (Touch Panel Return Register 1)

bit	Name	Function
7	TC8	Touch panel return register 1
6	TC7	
5	TC6	
4	TC5	
3	TC4	
2	TC3	
1	TC2	
0	TC1	

<Description>

See "4.5 Touch Panel I/F" for TC1 to TC8.

(35) P26H (Read)

TPRT2 (Touch Panel Return Register 2)

bit	Name	Function
7	-----------	Don't care.
6	-----------	
5	TC14	Touch panel return register 2.
4	TC13	
3	TC12	
2	TC11	
1	TC10	
0	TC9	

<Description>

See "4.5 Touch Panel I/F" for TC9 to TC14.

2.3 Notes

(1) I/O register initial reset

Some of the control registers included in EHT-10 and EHT-10/2 gate array set output directly after reset to 0 using a method to reduce the number of gates.
As shown in figure 2.1, mask register output with flip-flop to set output to 0 even if the contents of the register is unstable, in order to control the register output directly after the reset.
Register output mask release after the CPU starts and the contents of the register initialized by a program is described (See Figure 2.1). By doing this, register output operates similar to register reset. Hereafter this method is called pseudo-reset.
The EHT-10 and EHT-10/2 register has 3 types of state including this pseudo-reset at reset.

1. Pseudo-reset state
2. Reset state
3. State which is not reset and unstable

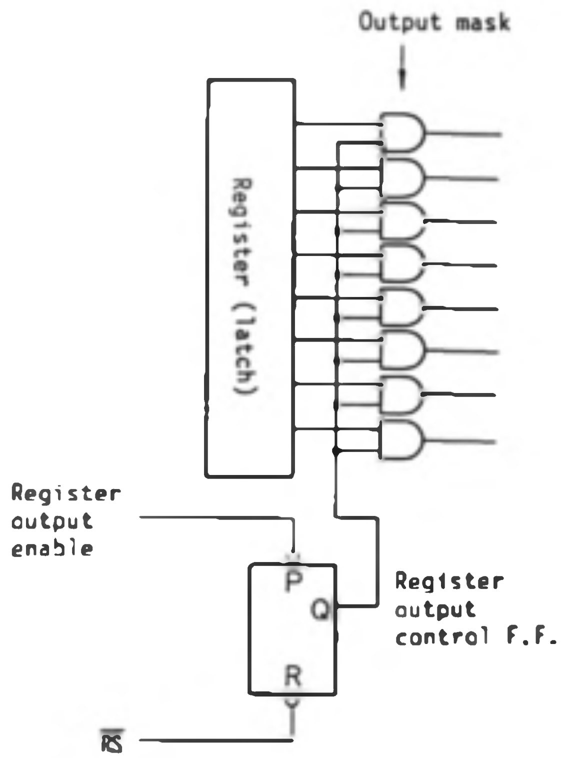


Fig 2.1 Pseudo-Reset

1. Pseudo-reset register bits

Register	7	6	5	4	3	2	1	0
CTLR1 [P00H]					SWBCR	BCR1	BCR0	
IER [P04H]				EXT	OVF	ICF	RXRDY	7508
BANKR [P05H]			BANK1	BANK0				

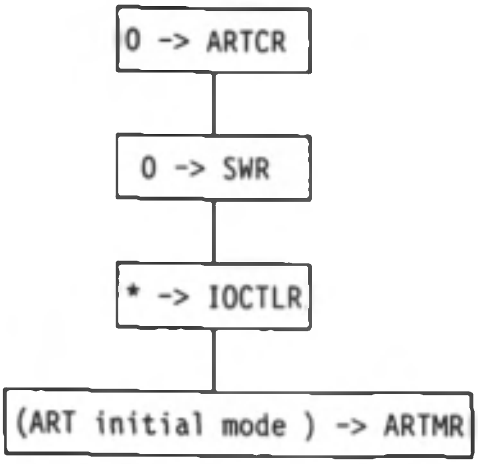
The 3 registers mentioned above have output masked with the same output control flip-flop directly after reset and the mask is released by writing to CTLR1 [P00H].

0 -> IER	IER Initial set
07H -> BANKR	BANKR Initial set
XXXX000X -> CTLR1	IER, BANKR, CTLR1 mask release

Mask release procedure

Register	7	6	5	4	3	2	1	0
ARTMR [P15H]	STOP		EVEN	PEN		DATA Leng.		
ARTCR [P16H]			RRTS	ER	SBKR	RXE	RDTR	TXE
SWR [P18H]				0	SSW1	SSW0	CSW1	CSW0
IOCTLR [P19H]	SP	LED2	LED1	LED0	$\overline{\text{PINT}}$	ICR	$\overline{\text{SLIN}}$	PF

The 4 registers mentioned above have output masked with the same output control flip-flop directly after reset and the mask is released by writing to ARTMR [P15H].



ARTMR, ARTCR, SWR, IOCTLR mask release.

Mask release procedure

*: Value which makes the control signal DISABLE.

2. Reset register bits

	7	6	5	4	3	2	1	0
CTLR2 [P02H]							0	ELON
BANKR [P05H]	BANK3	BANK2						
YOFF [P09H]	DSP							

3. Unstable register bits that are not reset.

	7	6	5	4	3	2	1	0
CTLR1 [P00H]	BRG3	BRG2	BRG1	BRG0				1
BANKR [P05H]							1	0
SIOR [P06H]	Transmit/Receive DATA to slave CPU							
VADR [P08H]		A14	A13	A12	A11	A10	A9	
YOFF [P09H]				Y4	Y3	Y2	Y1	Y0

FR [P0AH]					F3	F2	F1	F0

ARTDR [P14H]	7/8 Transmit / Receive DATA

ICCTLR [P17H]	PWLCD	PRIE	ICITE	DCTG	BCRP	ICDIR	IOC	ICOSC

(2) Output to I/O port

When EHT-10 and EHT-10/2 outputs data to the I/O port, it saves the contents of the current output I/O register in RAM so that a part of the I/O register can be modified and the I/O port state can be recovered at power on. Thus, when data is output to the I/O port, the contents of RAM must be modified at the same time.
Relation of I/O register and area in RAM is as listed below.

I/O address	Register name	RAM address	Variable	Notes
P00H	CTLR1	F0D6H	RZCTLR1	
P04H	IER	F42EH	RZIER	DI state at modification
P05H	BANKR	F42CH	RZBANKR	Further processing for actual bank switching.
P15H	ARTMR	F0D9H	RZARTMR	
P16H	ARTCR	F0DAH	RZARTCR	
P18H	SWR	F0DCH	RZSWR	
P19H	IOCTLR	F0DDH	RZIOCTLR	
P22H	SBKR	F42DH	RZSBBNKR	
P23H	CTLR3	F0DEH	RZCTLR3	

CHAPTER 3 Interrupt Description

3.1 Overview

5 types of interrupt are supported in the EHT-10/EHT-10/2. When an interrupt is issued, the vector is set corresponding to the interrupt vector and control moves to the interrupt routine.
CPU mode 2 is used for the interrupt, EHT-10/EHT-10/2 OS sets the interrupt vector at system initialization, reset, and power- on.
Interrupts the EHT-10/EHT-10/2 supports are as listed below.

(1) 7508 (4 bit CPU)
(2) ART (RXRDY)
(3) ICF (Input Capture)
(4) OVF (FRC overflow)
(5) EXT (External)

Factors of 7508 (4 bit CPU) interrupt are as listed below.

(1) Keyboard (EHT-10/2)
(2) Touch panel (EHT-10/2)
(3) Alarm time
(4) Power failure
(5) Power switch ON/OFF
(6) One second interrupt

Factors of EXT interrupt are as listed below.

(1) Cartridge option
(2) IC card reader
(3) 1 msec, 8 msec timer

3.2 Interrupt Vector

The EHT-10/EHT-10/2 has 5 types of interrupt, the interrupt vector Table is located at address FFF0H to FFFFH. The interrupts are prioritized, requested interrupts are accepted following the priority order. I/O address P04H [IER : Interrupt Enable Register] can disable/enable interrupts, when an interrupt is disabled, the interrupt request status can be checked by reading I/O address P04H [ISR: Interrupt Status Register].

Priority order	Interrupt cause	Vector	Address in RAM	IER	ISR	NAME
1 (Highest priority)	7508 (4bit CPU)	F0H	FFF0H,FFF1H	IER0	ISR0	INT0
2	ART (RXRDY)	F2H	FFF2H,FFF3H	IER1	ISR1	INT1
3	ICF (Input Capture)	F4H	FFF4H,FFF5H	IER2	ISR2	INT2
4	OVF (Overflow of FRC)	F6H	FFF6H,FFF7H	IER3	ISR3	INT3
5	EXT (timer, cartridge, and IC card)	F8H	FFF8H,FFF9H	IER4	ISR4	INT4

Table 3.1 Interrupt Priority Order

Write 0 to the corresponding IER for interrupt disable and write 1 for interrupt enable. "1" is set to each ISR at interrupt.

3.3 7508 Interrupt

Factors of 7508 (4 bit CPU) interrupts are as listed below.

(1) At key input (EHT-10/2)
(2) When the touch panel is pressed (EHT-10)
(3) One second interval interrupt
(4) At power failure (when main battery is discharged)
(5) Specified alarm time
(6) At power switch ON/OFF

Read status (command code 2) is issued to 7508 to check the factor of interrupt, the interrupt factor is analyzed and interrupt process is performed in the interrupt routine.
See "Software Version 7.6 7508 Command" for interface and command of 7508 and CPU.

3.4 ART Interrupt

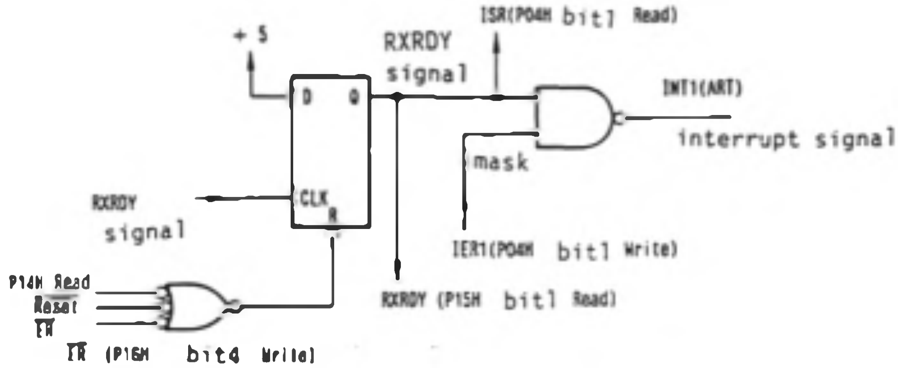


Fig. 3.1 ART Interrupt Generation Block

RS-232C interface is built-in the EHT-10/EHT-10/2, it issues ART (RXRDY) interrupt when receive data enters. When the interrupt is not used, write "0" to IER1 so that the interrupt can be masked and the status can be read by ISR and RXRDY. Interrupt signal reset is performed when I/O port P14H is read (receive data is read), 1 is written to error reset command ER or reset signal is input.

3.5 OVF Interrupt, ICF Interrupt

ICF is set when the input signal changes from the barcode reader and is reset when input capture register [ICRH.B: P03H] is read.

OVF is set when FRC (Free Running Counter) overflows and is reset by RESOVF command [CMDR: Write 1 to P01H bit 2]. As FRC is 16 bits and the input clock is 614.4 kHz, frequency is approx. 106.67 msec.

The interrupts are enabled when IER [P04H] corresponding bit is set to 1 and are disabled when it is set to 0. Interrupt request can be checked by reading ISR [P04H] regardless of the value of IER. While the value of IER is rewritten, the CPU must be in the interrupt disable state.

IER is initial-reset. Neither ICF nor IVF is initial-reset. Input capture register ICR contains 16 bits and FRC of 16 bits is fetched. There are 2 types of timing to fetch. One is according to a read instruction, when ICRL.C [P00H] read instruction is issued, FRC of 16 bits at that time is latched. This operation is used when the value of FRC is read. Read ICRH.C [P01H] to read the remaining high order 8 bits. The other is according to the barcode. When the bar code signal changes, ICR latch pulse generates and FRC 16 bits are fetched to ICRL.B [P02H], ICRH.B [P03H]. ICF is set at this time. ICF is reset at ICRH.B read. The bar code signal is directly read by reading STR [P05H] bit 1.

Figure 3.2 shows OVF, ICF interrupt generation block.

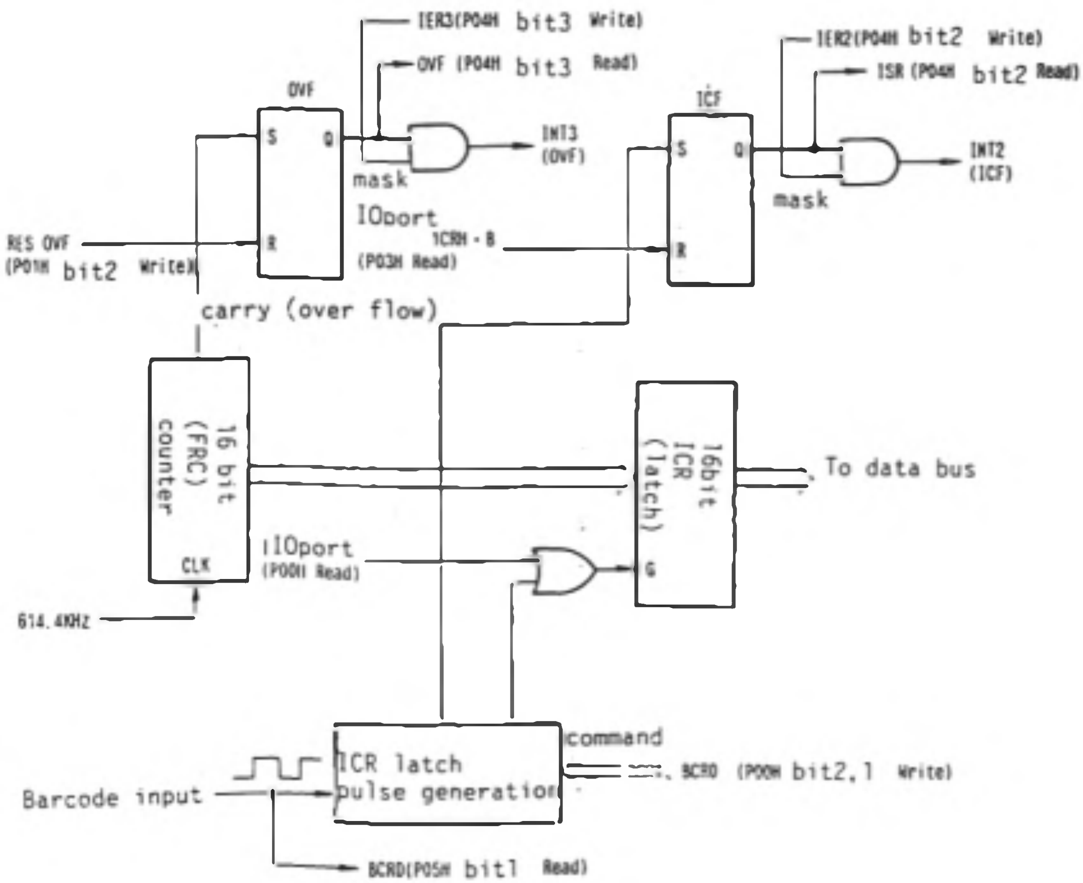


Fig 3.2 OVF, ICF Interrupt Generation Block

3 factors of EXT interrupt are as listed below.

(1) Interrupt from cartridge option ($\overline{\text{CINT}}$ signal)
(2) Interrupt from IC card reader
(3) 1 msec, 8 msec timer interrupt

These interrupts are logical-added (ORed) and become one EXT interrupt. They can be individually enabled or disabled.
Even when an interrupt is disabled, reading the I/O port status can check the interrupt request. Figure 3.3 shows EXT interrupt block and Table 3.2 describes the terminals.

(1) IC card reader

When excess current goes through the IC card or IC card cover is open, the IC card interrupt is issued. This state reflects ICCLS [P16H bit 2]. When this interrupt is not required, write "0" to ICITE [P17H bit 5]. Turn off IC card power supply or close the IC card cover to release the interrupt signal.

(2) Cartridge option

The EHT-10/EHT-10/2 accepts interrupts from the cartridge option. Set PRIE [P17H bit 6] to "0" so that this interrupt is accepted. This interrupt signal reflects PBUSY [P16H bit 0]. Ways of releasing the interrupt signals are different according to the cartridge option.

(3) Timer

When 1 msec or 8 msec is counted by the timer, the timer interrupt is issued.
Interrupt of 1 msec or 8 msec is selected by OVFITV [P23H bit 4], writing "1" for 1 msec, "0" for 8 msec. Write "0" to OVFEN [P23H bit 5] when this interrupt is not required.
Write "1" to OVFRESET [P23H bit 1] to release the interrupt.

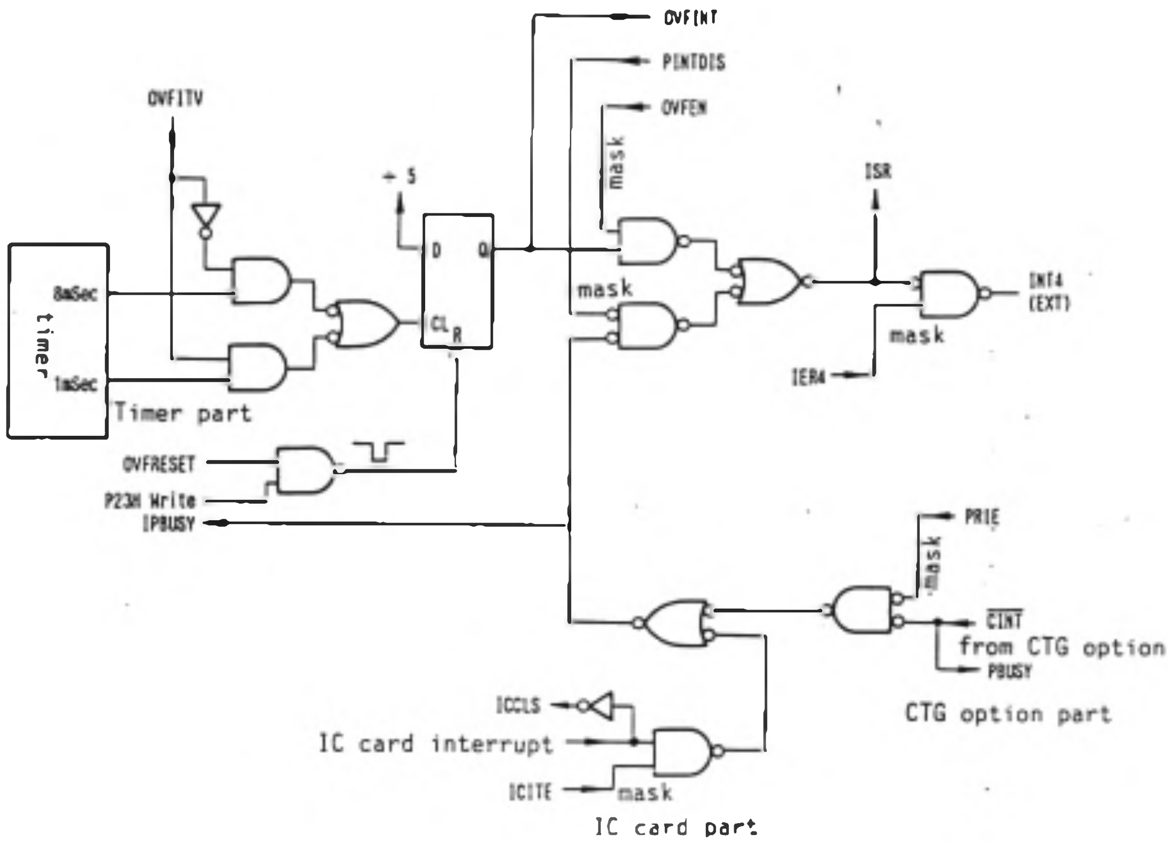


Fig 3.3 EXT (External) Interrupt Block

Terminal	Description	I/O port	R/W
IC card interrupt	This becomes 1 when IC card cover is open or excess current goes through the IC card.	---------	-
ICCLS	IC card interrupt signal ICCLS=$\overline{\text{IC card interrupt}}$	P16H bit2	R
ICITE	IC card interrupt mask.	P17H bit5	W
$\overline{\text{CINT}}$	Interrupt signal from cartridge option.	---------	-
PBUSY	$\overline{\text{CINT}}$ signal can be read.	P16H bit0	R
PRIE	$\overline{\text{CINT}}$ interrupt signal mask.	P17H bit6	W
OVFITV	Frequencies of interrupt 1 msec and 8 msec are set.	P23H bit4	W
OVFRESET	Interrupts of 1 msec and 8 msec are reset.	P23H bit1	W
IPBUSY	IC card interrupt and $\overline{\text{CINT}}$ interrupt status can be read.	P23H bit1	R
OVFINT	Interrupt status of 1 msec and 8 msec can be read.	P23H bit0	R
PINTDIS	Mask of IC card interrupt and $\overline{\text{CINT}}$ interrupt.	P23H bit3	W
OVFEN	Mask of interrupt 1 msec and 8 msec.	P23H bit5	W
ISR	Signals of IC card interrupt, $\overline{\text{CINT}}$ interrupt, 1 msec and 8 msec interrupt.	P04H bit4	R
IER4	EXT interrupt signal mask.	P04H bit4	W
INT4(EXT)	CPU mode 2 Interrupt signal.	---------	-

Table 3.2 EXT Interrupt terminal

CHAPTER 4 Interface Description

4.1 Cartridge Interface

4.1.1 Overview

EHT-10/EHT-10/2 contains a cartridge interface so that various types of option cartridges can be connected. The following 4 modes are available for cartridge interface and can correspond to various options.

(1) Hand Shake mode (HS mode)

This CPU to CPU mode interfaces with devices which use the CPU as an option, similar to 8255 hand shake mode. HS mode handles data through the input and output buffer. Data handling is controlled by flag.

(2) Input Output mode (IO mode)

This mode uses an interface format of 4 bit input port and 4 bit output port.

(3) Data bus mode (DB mode)

The option looks like a normal IO device in this mode viewing from the main unit. The cartridge I/F only connects the main unit data bus directly to the cartridge data bus.

(4) Output Port mode (OT mode)

This mode interfaces at the 8 bit output port.

Functions with (1) to (4) mentioned above are performed using gate array GAPNIO.

The cartridge I/F consists of a value signal line for modes mentioned above, serial communication signal line and battery outputs of -5 V and +5 V.

4.1.2 Cartridge I/F circuitry

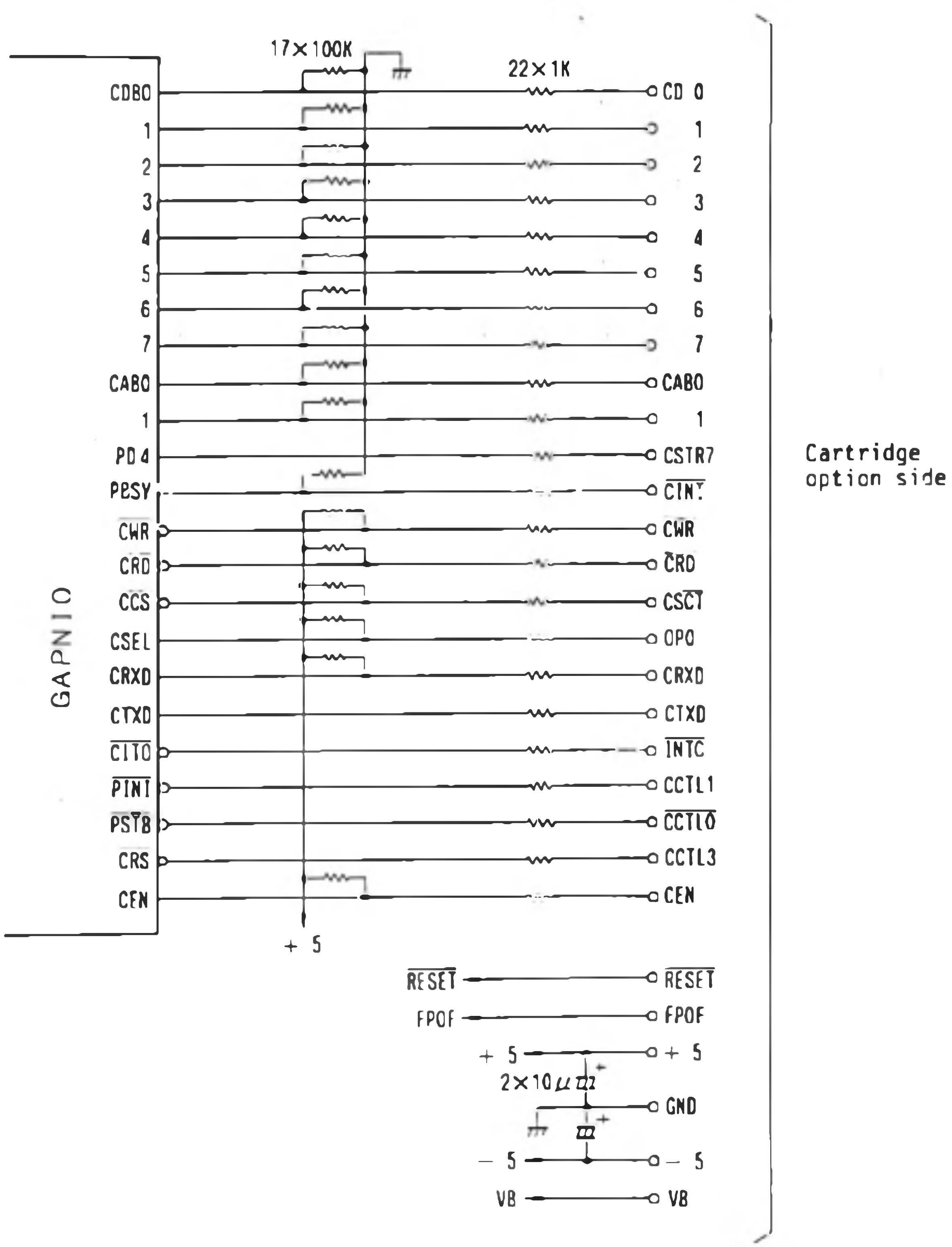


Fig 4.1 Cartridge I/F Circuitry

4.1.3 Connector in use

The following connector is used for the cartridge I/F.

Connector	PICL-30P-LT
Ground metal fitting	Cartridge ground metal fitting A (Seiko Epson)

Use the following connector for the other side

Connector	PICL-30S-LT
Ground metal fitting	Micro cassette ground metal fitting B (Seiko Epson)

4.1.4 Cartridge Connector Terminal Name and Function

Pin arrangement (front of main unit connector)

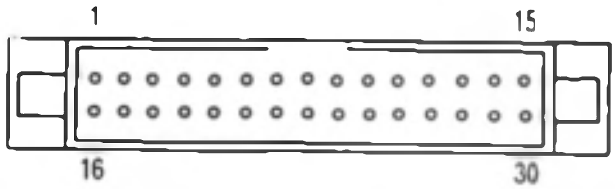


Fig 4.2 Cartridge Connector Pin Arrangement

Table 4.1 Cartridge I/F Pin Arrangement

Signal	Pin No.	I/O	Description	At main unit reset
FPOP	1	O	When main battery voltage is less than 3.6 V, more then 3 V of H level is output. This signal is used for cut-off when the printer is in low voltage.	Not effected
+5	2	-	+5 V is output at power on.	-----
$\overline{\text{CINT}}$	3	I	General use input terminal with interrupt function. When "L" level is input, interrupt generates. Status read is enabled by the I/O port. This signal is used for ready detection of the printer, care is required for other uses.	-----

Signal	Pin No.	I/O	Description	At main unit reset
CCTL1	4	O	General use output.	"L" at reset.
CCTL3	5	O	General use output.	"L" at reset.
$\overline{CWR}$	6	I/O	Write pulse input from the cartridge option in HS mode. Write pulse output to the cartridge option in DB mode.	Input state
CEN	7	I	When CPU 6301 is used in the cartridge, 6301E is connected. Other than that "H" level is fixed.	-----
CD0	8	I/O	The data bus I/O with cartridge option in HS mode and DB mode. They are port outputs in OT mode, CD7 to CD4 are port outputs in IO mode and CD3 to CD0 are port inputs.	They are set to HS mode and are in the input state.
CD1	26			
CD2	25			
CD3	11			
CD4	10			
CD5	28			
CD6	27			
CD7	13			
$\overline{CSCT}$	9	I/O	Chip select input from the cartridge option in HS mode. Chip select output to the cartridge option in DB mode. It is for future use in OT or IO mode.	Input state
CRXD	12	I	This Serial receive line is connected to ART section through the serial switch and can be used for general input when set to SSW1=0 and SSW0=0.	-----
CTXD	14	O	This serial send/receive line is connected to ART section through the serial switch. (See 4.3.1)	"H" level
OP0	18	I	Cartridge option type identification signal. OP0=1 in HS mode, OP0=0 in other modes.	-----

Signal	Pin No.	I/O	Description	At main unit reset
CSTR7	19	O	Development cartridge switch signal. Normal operation at "0", development cartridge access enable at "1".	Inconsistent output at power on Reset does not effect this signal
$\overline{RESET}$	20	O	Reset signal of the main unit system	-----
$\overline{INTC}$	21	O	Interrupt signal used only in HS mode. When the main unit outputs data or command to the output buffer for HS mode, $\overline{INTC}$=0 and it requests interrupt to the cartridge option. When the cartridge option reads data the request is canceled and $\overline{INTC}$=1.	"H" at reset.
$\overline{CCTLO}$	22	O	General use output	"H" at reset.
CAB0	23	I/O	Address input from the cartridge option in HS mode. Address output to the cartridge option in DB mode. It is for future use in OT or IO mode.	Input state.
CAB1	29	I/O	General use input line from cartridge option in HS mode. Address output to cartridge option in DB mode. It is for future use in OT or IO mode.	Input state.
$\overline{CRD}$	24	I/O	Read pulse input from the cartridge option in HS mode. Read pulse output to the cartridge option in DB mode.	Input state.
GND	16	-	Signal ground	-----
- 5	17	-	This signal outputs -6 $\pm$1 V at power on.	-----
VB	15	-	This signal is connected to (+) of the main battery.	-----
VG	30	-	This signal is connected to (-) of the main battery.	-----

4.1.5 Cartridge I/F I/O Address Map (Table 4.2)

I/O address	Read								Write							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
P10H	CTG I/F Address space								CTG I/F Address space							
P11H																
P12H																
P13H																
P16H		CSEI (OPO)						BUSY (CIN1)								
P17H										PRIE		DCIG (CSIR?)				
P18H													SSW1	SSW0	CSW1	CSW0
P19H													PINT (CCIL3)		SLIN (CCIL1)	PI (CCIL0)
P1AH							PBUSY						PINT DIS			

(Note)Terminal names are written in ( ) under bit names for I/O address bit names which are different from I/F terminal names.

4.1.6 Mode Setting

The cartridge I/F mode is set by cartridge switch CSW1, CSW0 (bit 1, 0) in the switch register (SWR:P18H).
CSW1 and CSW0 are set to HS mode (CSW1, CSW0=0) at reset.

CSW1	CSW0	Mode
0	0	HS mode
0	1	IO mode
1	0	DB mode
1	1	OT mode

Set to HS mode for mode setting after initialization (in HS mode) and for mode switching at cartridge option exchange.

4.1.7 Mode Description

(1) Symbols with meaning that are changeable according to modes. Table 4.3 shows signals with meaning that are changeable according too modes (HS, DB, IO, OT).

Signal / Mode	HS	DB	IO	OT
$\overline{CSCT}$	(Input)Chip select input	(Output)Chip select output	unused	unused
CAB1	(Input) General use input line	(Output) Address output	unused	unused
CAB0	(Input) Address input	(Output) Address output	unused	unused
$\overline{CRD}$ $\overline{CWR}$	(Input) Read/write pulse	(Input) Read/Write pulse	unused	unused
CB7 - 0	(I/O) Data bus I/O	(I/O) Data bus I/O	(Output) Port output (CDB7 to 4) (Input) Port input (CDB3 to 0)	(Output) 8 bit port output
$\overline{INTC}$	(Output) Interrupt signal	unused	unused	unused

Table 4.3 Mode Signal Name

(Input) cartridge --> EHT-10 and EHT-10/2
(Output) EHT-10 and EHT-10/2 --> cartridge

(2) Signals enabled for general use I/O

Table 4.4 shows the signal names enabled as general use input or output.

Signal	I/O	Port and bit position	Note	At reset
$\overline{\mathrm{CCTL0}}$	O	P19H bit0 (PF)	$\overline{\mathrm{CCTL0}}$ terminal becomes 0 when 1 is written.	1
CCTL1	O	P19H bit1 ($\overline{\mathrm{SLIN}}$)		0
CCTL3	O	P19H bit3 ($\overline{\mathrm{PINI}}$)		0
CRXD	I	P16H bit3 (RXD)	Select SSW0=SSW1=0 at switch register P18H.	
CAB1	I	P11H bit2 (CAB1)	Only HS mode is valid.	
$\overline{\mathrm{CINT}}$	i	P16H bit0 (PBUSY)	With interrupt function When $\overline{\mathrm{CINT}}$=0, PBUSY=0	

(Input) cartridge --> EHT-10/EHT-10/2
(Output) EHT-10/EHT-10/2 --> cartridge

Table 4.4 General Use I/O Signal

(Note 1) See "3.6 EXT Interrupt" for $\overline{\mathrm{CINT}}$.

(Note 2) $\overline{\mathrm{CCTL0}}$, CCTL1, CCTL3 and $\overline{\mathrm{CINT}}$ are used for the printer in OS, when used for general use I/O, care is suggested. See "Software Version 8.8 EXT Interrupt" for further information.

(3) HS mode

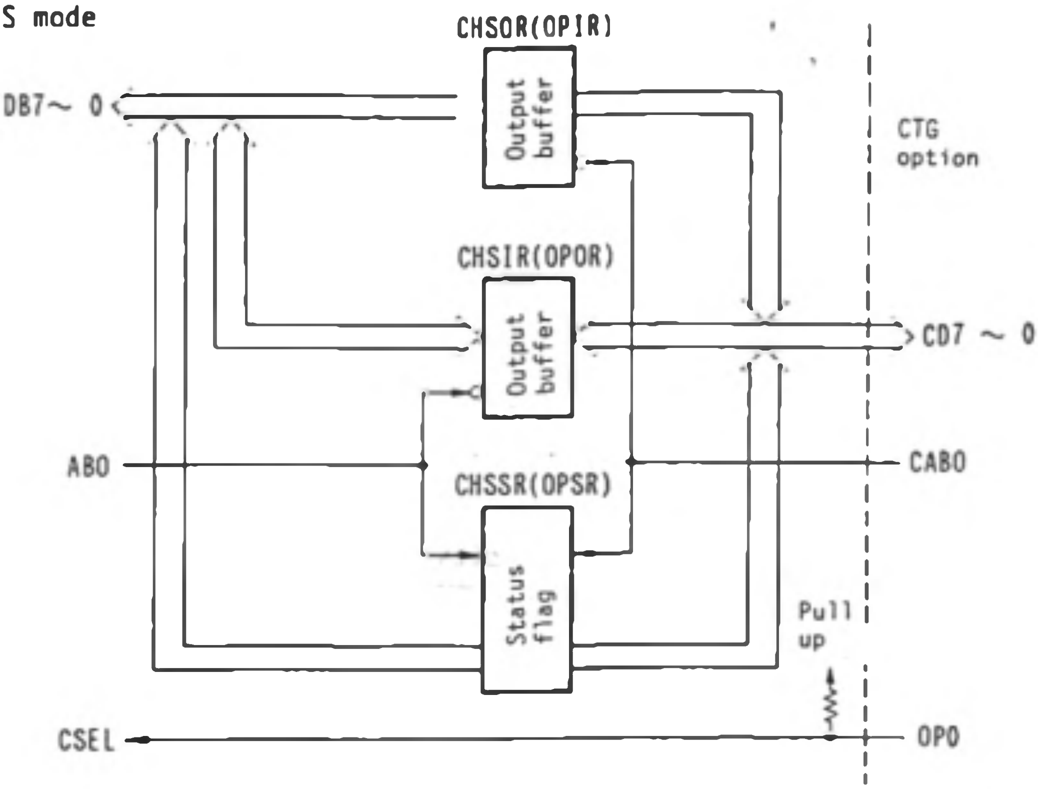


Fig 4.3 HS Mode I/F Diagram

1. Description

The main unit interfaces with the cartridge option through the output or input buffer in HS mode. When data is written to the output buffer by the main unit, it is read by the option. When data is written to the input buffer it is read by the main unit. Hand shake is performed by OBF (Output Buffer Full) and IBF (Input Buffer Full).

Option CD 7 to 0, CAB0 $\overline{CRD}$, $\overline{CWR}$, $\overline{CCS}$ are output from the option.
When the main unit writes to the output buffer, AB0=1 sets the command and AB0=0 sets the data. The value of AB0 is a status flag and is fetched by F0, this is shown from the option side by status read. A write to the output buffer regardless of the command data sets OBF=1

and $\overline{INTC}$=0, and an interrupt to the option is requested.
The value of OBF can be shown from either the main unit or the option by status read. (However, OBF and IBF are reversed between the main unit and the option, OBF viewed from the main unit is the option for IBF.)
The option acknowledges, write is performed from the main unit to the buffer by interrupt or status read, and the data is read with command/data flag F0. When the option reads the output buffer, it returns to OBF=0, INTC=1. The option reads the value of the output buffer as CAB0=0 and reads the status as CAB0=1.
When the option writes to the input buffer (command/data are not distinguished when written from the option, thus, the value of CAB0 is originally "Don't care" but must be used as CAB0=0), IBF=1 and the main unit checks this value using the status read and reads the input buffer.
IBF=0 is returned by input buffer read. Data (input buffer) is read as AB0=0 from the main unit and the status is read as AB0=1.

CAB1 is not an address in HS mode but a general use input from the option. The value of this CAB1 can be read from the main unit as a part of the status.

2. I/O address space

R/W	IO address	Register	7	6	5	4	3	2	1	0	Flag
R	P10H	CHSIR	8 bits data								IBF reset
	P11H	CHSSR						CAB0	IBF	OBF	
	P12H P13H	For future use (access disable).									
W	P10H	CHSOR	8 bits data								OBF set, FO=0
	P11H	CHSOR	8 bits command								OBF set, FO=1
	P12H P13H	For future use (access disable)									

Note When OBF=1, $\overline{INTC}$=0.

View from the cartridge option

R/W	CAB0	Register	7	6	5	4	3	2	1	0	Flag
R	0	OPIR	8 bits data								OPIBF reset
	1	OPSR					FO		OPI FB	OPO BF	
W	0	OPOR	8 bits data								OPOBF reset
	1		For future use (access disable)								

OPIR=CHSOR, OPOR=CHSIR, OPIBF=OBF, OPOBF=IBF. Data when FO is 0, Command when FO is 1.

3. AC characteristic (Main unit protection point of 1 K ohm resistance)

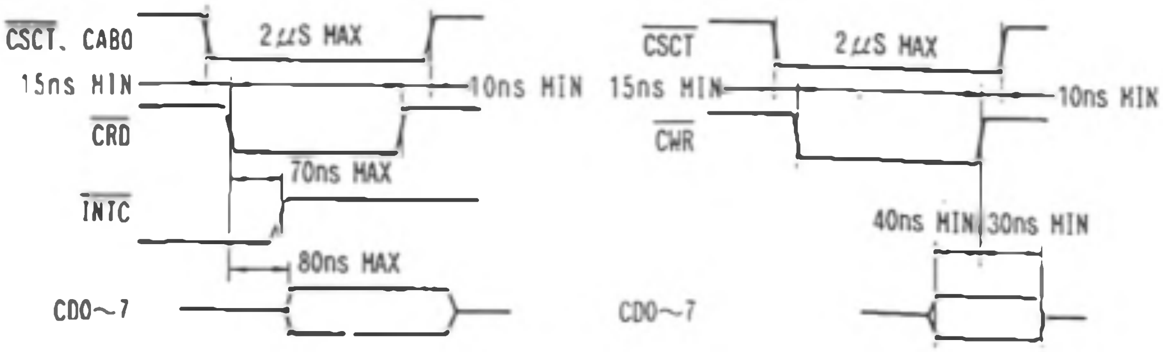


Fig 4.4 HS Mode AC Characteristic

(4) IO mode

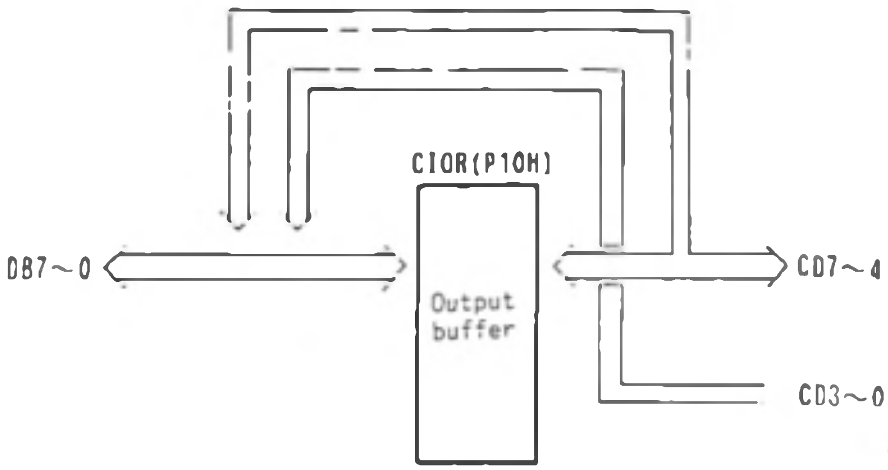


Fig 4.5 IO Mode I/F Diagram

1. Description

The IO mode consists of a 4 bit output port (CD7 to CD4) and input port (DC3 to DCO). The output side consists of an 8 bit latch (CIOR), the value of terminal CD3 to CDO is input without the latch at input side. I/O address is P10H. The contents of the 8 bit data bus is written to CIOR at output to P10H. CIOR high order 4 bits are output directly to CD7 to CD4, low order 4 bits are not connected. After P10H is read, the values of CD terminal 3 to 0 are input directly to the data bus low order 4 bits of the main unit, the values of the output buffer high order 4 bits are input to the high order 4 bits.

$\overline{CSCT}$, CAB1, CABO, $\overline{CWR}$ and $\overline{CRD}$ are high impedance in IO mode, $\overline{CSCT}$, $\overline{CWR}$

and $\overline{CRD}$ are pulled up, CAB1 and CABO are pulled down. $\overline{INTC}$ output becomes 1.

2. I/O address space

R/W	IO address	Register	7	6	5	4	3	2	1	0
R	P10H	CIOR	(Contents of output port) CB 3 to 0							
	P11H	For future use (access disable)								
	P12H									
	P13H									
W	P10H	CIOR	4 bits data				Don't care.			
	P11H	For future use (access disable)								
	P12H									
	P13H									

Table 4.5 IO Mode I/O Address Space

3. AC characteristic (Main unit protection pcint of 1 K chm resistance)

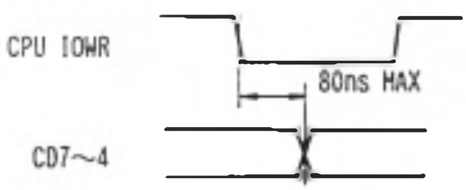


Fig 4.6 IO Mode AC Characteristic

4. IO Mode I/F Circuitry

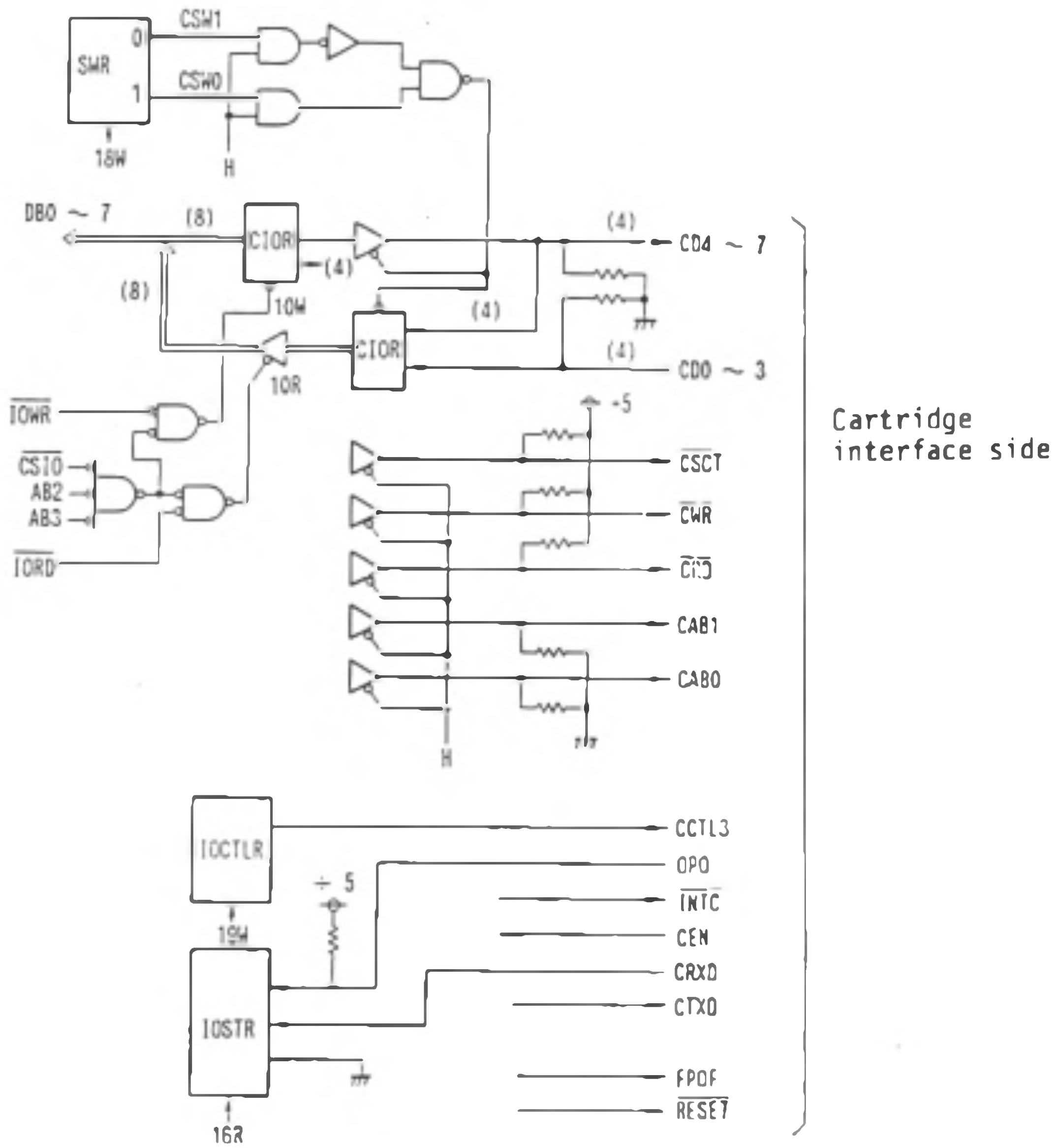


Fig 4.7 I/O Mode Interface

(5) DB mode

DB 7 - 0 <==========================> CD 7 - 0

Fig 4.8 DB Mode I/F Diagram

1. Description

The DB mode uses the cartridge option as normal I/O device, the option has the appearance from the main unit, of an I/O device with 4 address spaces. CD7 to CD0 is directly connected to the system data bus, $\overline{CSCT}$, $\overline{CRD}$, $\overline{CWR}$, CAB1, CAB0 are all supplied from the main unit, control lines are all output. $\overline{CSCT}$ is 0 for address 10H to 13H, CAB1 and CAB0 are address low order 2 bits. $\overline{CRD}$ and $\overline{CWR}$ are I/O read/write pulse from the main unit. $\overline{INTC}$ output is 1 in DB mode.

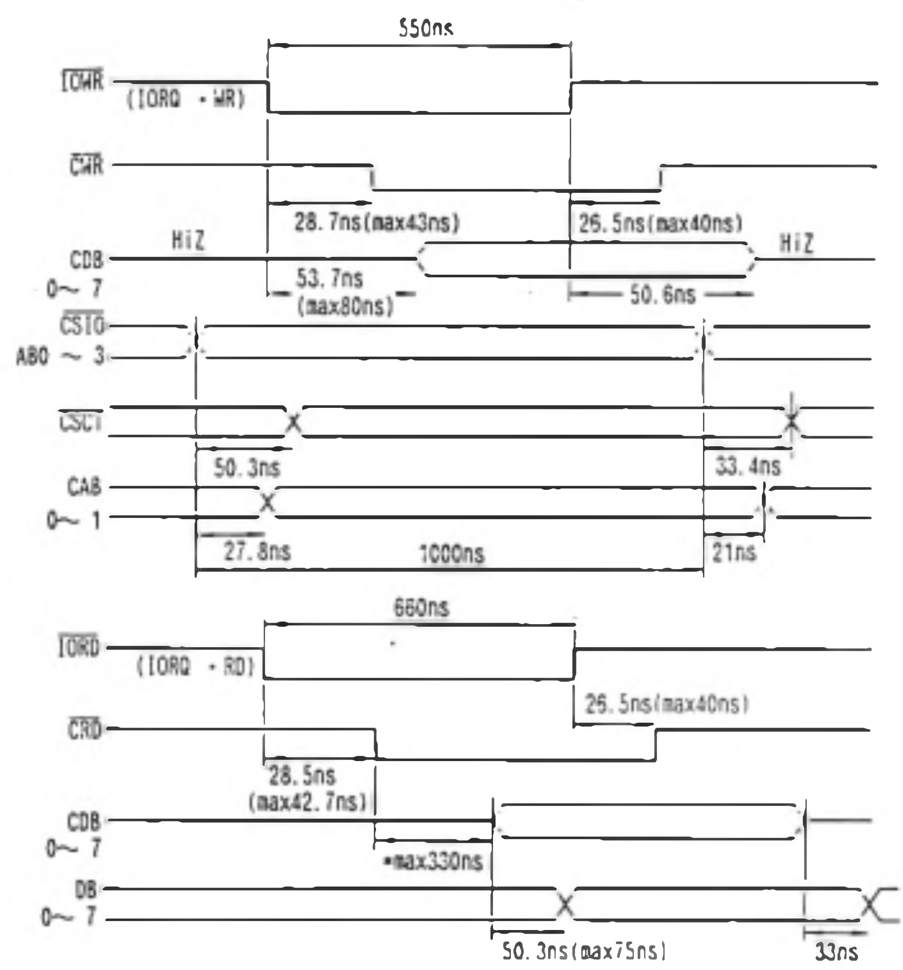
2. I/O address space

R/W	IO address	Register	7	6	5	4	3	2	1	0
R	P10H		Defined at cartridge option side							
	P11H									
	P12H									
	P13H	Disabled								
W	P10H		Defined at cartridge option side							
	P11H									
	P12H									
	P13H									

Table 4.7 DB Mode I/O Address Space

<Note> I/O address P13H READ is used by OS to get the device address, thus, circuitry must be structured as the cartridge option does not operate by P13H read.

3. AC characteristic (Main unit protection point of 1 K ohm resistance)



* -> Value determined by option

Fig 4.9 DB Mode AC Characteristic

4. DB Mode I/F Circuitry

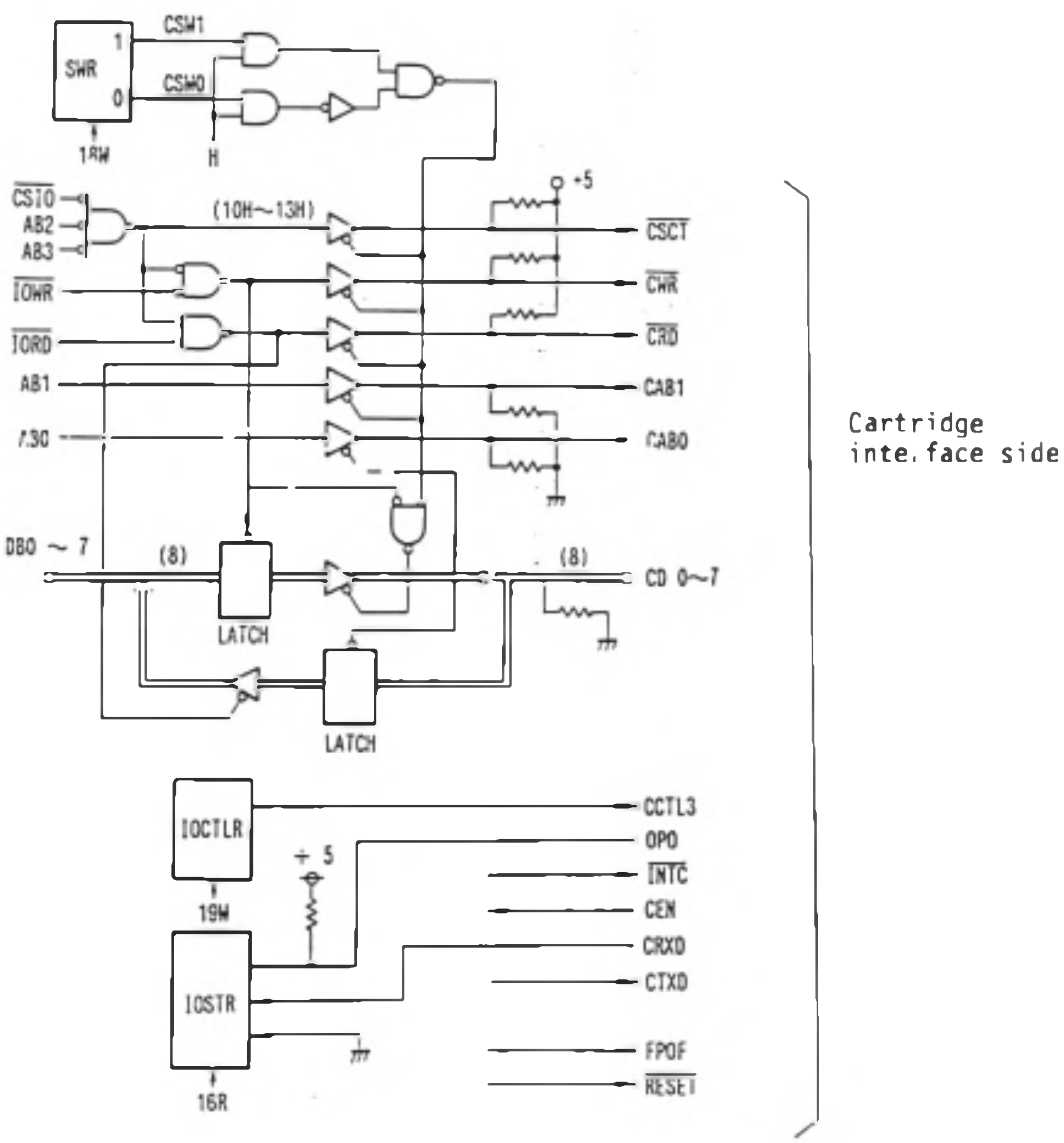


Fig 4.10 DB Mode Interface

(6) OT mode

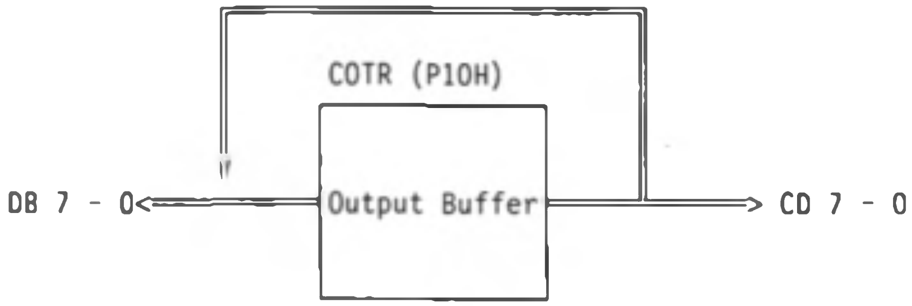


Fig 4.11 OT Mode/IF Diagram

1. Description

OT mode consists of an 8 bit output port (latch). The output buffer (COTR) address is P10H, Contents of the data bus DB7 to DB0 are fetched to COTR at output to P00H and are output to CD7 to CD0 at the same time. Contents of COTR is read by reading P10H.

$\overline{CSCT}$, CAB1, CAB0, $\overline{CWR}$ and $\overline{CRD}$ are high impedance, $\overline{CSCT}$, $\overline{CWR}$ and $\overline{CRD}$ are pulled up, CAB1 and CAB0 are pulled down. $\overline{INTC}$ output is 1.

2. I/O address space

R/W	IO address	Register	7	6	5	4	3	2	1	0
R	P10H	COTR	(Contents of output port)							
	P11H	For future use (access disable)								
	P12H									
	P13H									
W	P10H	COTR	8 bits data							
	P11H	For future use (access disable)								
	P12H									
	P13H									

Table 4.8 OT Mode I/O Address Space

<Note> Register CHSOR, CHIOR and COTR are physically the same.

3. AC characteristic (Main unit protection point of 1 K ohm resistance)

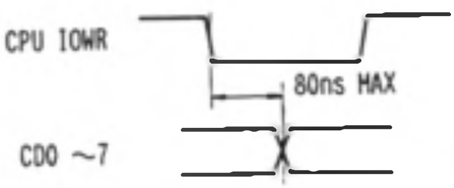


Fig 4.12 OT Mode AC Characteristic

4. OT mode I/F circuitry

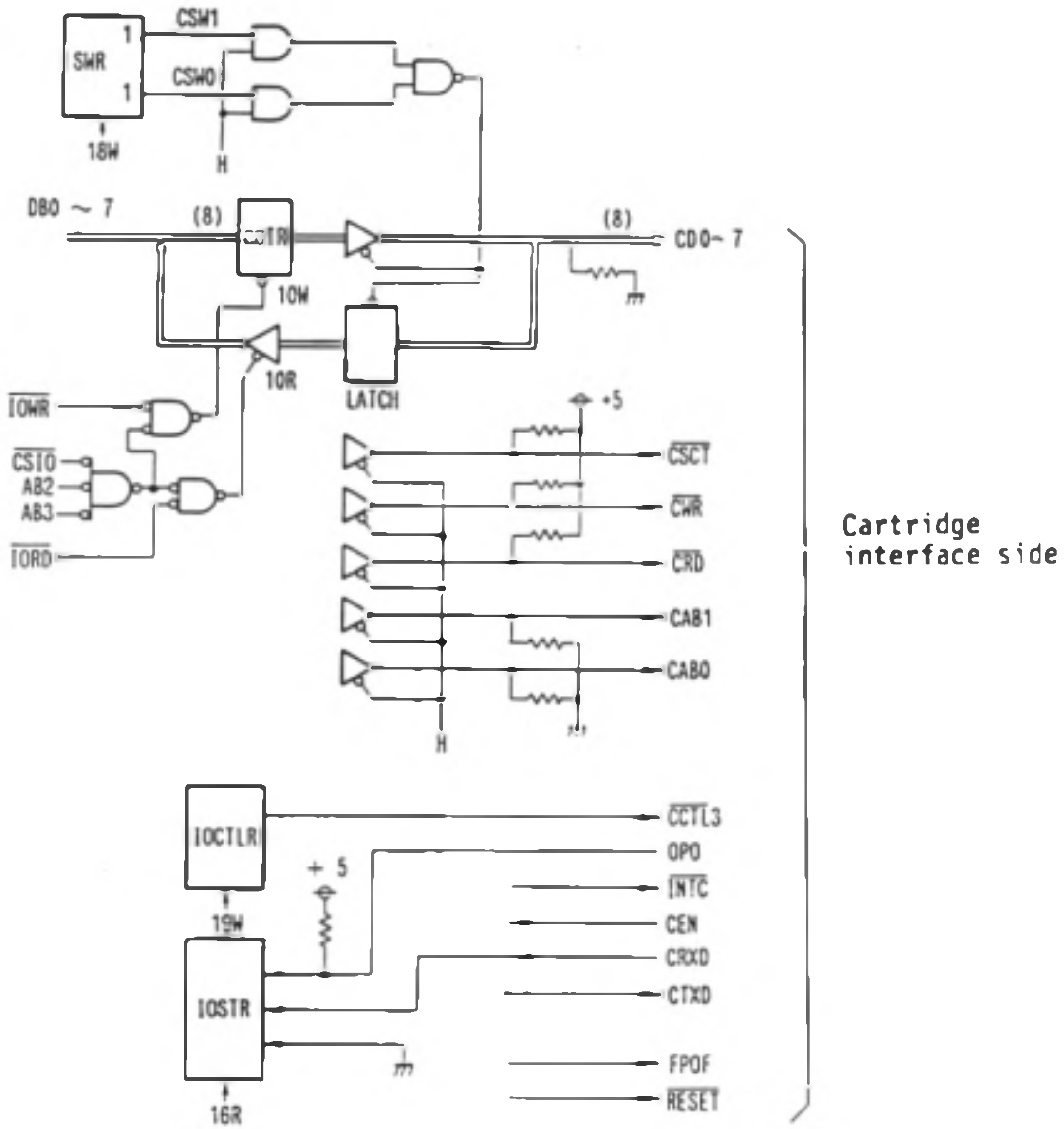


Fig 4.13 OT Mode Interface

4.1.8 Device Address

Cartridge optional devices are assigned specific addresses for cartridge type recognition when connected. The OS identifies the type by putting this device address main unit in DB mode and reading I/O port P13H.
Some of the device addresses have already be defined as in Table 4.4 , most are available for future expansion. A user should utilize an available address if a device address is required. In this case, OS only sets the cartridge I/F mode to switch register (SWRP18H) following modes in the table, it does not access the cartridge, but must access the cartridge in the user's application program. Timing is used by OS to identify the cartridge option device address and set to switch registers after power switch on , reset and at system initialization.

(1) How to set device address

When a user sets the device address, pull up the data buss CD4 to CD7 at cartridge option by 10 Kohm resistance. Fix terminal OPO (CSEL) to high or low. CD0 to CD7 have been already pulled down by 100 k ohm resistance by the main unit, busses that are not pulled up are assumed as "L".

(2) Device address set example

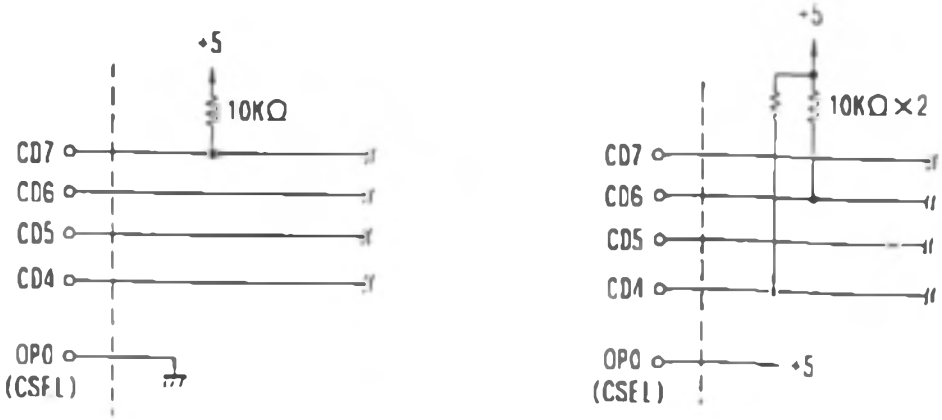
Ex. 1 : Figure 4.14 (a) shows when cartridge ID=8 in DB mode.

Ex. 2 : Figure 4.14 (b) shows when cartridge ID=5 in HS mode.

<table>
<tr><th>Device number</th><th colspan="2">CSEL-i</th><th colspan="2">CSEL=0</th></tr>
<tr><td>0H</td><td colspan="4">No option</td></tr>
<tr><td>1H
2H
3H</td><td>Printer unit</td><td>HS mode</td><td></td><td>For DB mode extension</td></tr>
<tr><td>4H
5H
6H
7H</td><td>(M160 Printer)</td><td rowspan="3">For HS mode extension</td><td></td><td>IO mode</td></tr>
<tr><td>8H
9H
AH
BH</td><td></td><td></td><td>DB mode</td></tr>
<tr><td>CH
DH
EH
FH</td><td></td><td></td><td>OT mode</td></tr>
</table>

Table 4.9 Device Address Table

(Note) As cartridge ID0, 3 and 7 are defined in EHT-10/EHT-10/2, they should not be used.



(a) When ID=8 in DB mode (b) When ID=7 in HS mode

Fig 4.14 How to Set Device Address

4.1.9 Cartridge I/F DC Characteristic

(1) Signal line other than $\overline{\text{RESET}}$ FPOF

Item		Symbol	Condition	Specification			Unit
				Min	Typ	Max	
Output voltage (See Note 1)	H level	VOH	IOH=-0.4mA	3.7		5.25	V
	L level	VOL	IOL=2mA	0		0.4	V
Input voltage (See Note 1)	H level	VIH		2.6			V
	L level	VIL				0.7	V
Input leakage current (See Note 2)			VI=0 to 5.25	-10		10	μA

(Note 1) Value at main unit protection point of 1 K ohm resistance.

(Note 2) Value at IC9 input terminal (including I/O common terminal at input time) excluding resistance leakage of pull-up and pull-down.

(2) $\overline{\text{RESET}}$

Item		Symbol	Condition (See note)	Specification			Unit
				Min	Typ	Max	
Output voltage	H level	VOH	IOH=-0.35mA	4.2			V
	L level	VOL	IOL=2.7mA			0.4	V

(Note) Power consumption at main unit is excluded.

(3) FPOF

Item		Symbol	Condition (See note)	Specification			Unit
				Min	Typ	Max	
Output voltage (See Note 1)	H level	VOH	IOH=-5mA		3.4		V
	L level	VOL	IOL=5mA		0		V

(Note) Power ON/OFF depends on battery voltage.

(4) +5 V

Voltage range	Output current (See Note 1)
5 V ± 10%	200 mA MAX

(Note 1) main unit power consumption is included.

<Reference> Main unit power consumption at normal operation.:
Approx. 70 mA
At IC card in use (64 KB).: Approx. 170 mA

(Note 2) Power consumption at power ON/OFF switching (during reset) must be as small as possible. V BK (backup voltage) may drop at switching, contents of RAM cannot be guaranteed.

(Note 3) Rush current at power on must be as small as possible.

(4) -5 V

Voltage range	Output current (See Note)
-6 V ± 1 V	10 mA MAX

(Note) Power consumption at main unit is included.

<Reference> Power consumption at RS-232C in use approx. 6 mA.

(5) VB

Voltage range (See Note)
4.6 to 6.0 V

(Note) It may be less than 4.6 V at printer operation.

4.1.10 Application Circuit Introduction

This chapter briefly introduces application circuits in IO mode, DB mode and OT mode from the cartridge interface mode. Please check operation first to use these circuits.
See the comment on each application circuit as reference. Read thoroughly "Chapter 6 Notes on Circuit Design".

(1) IO mode application circuit

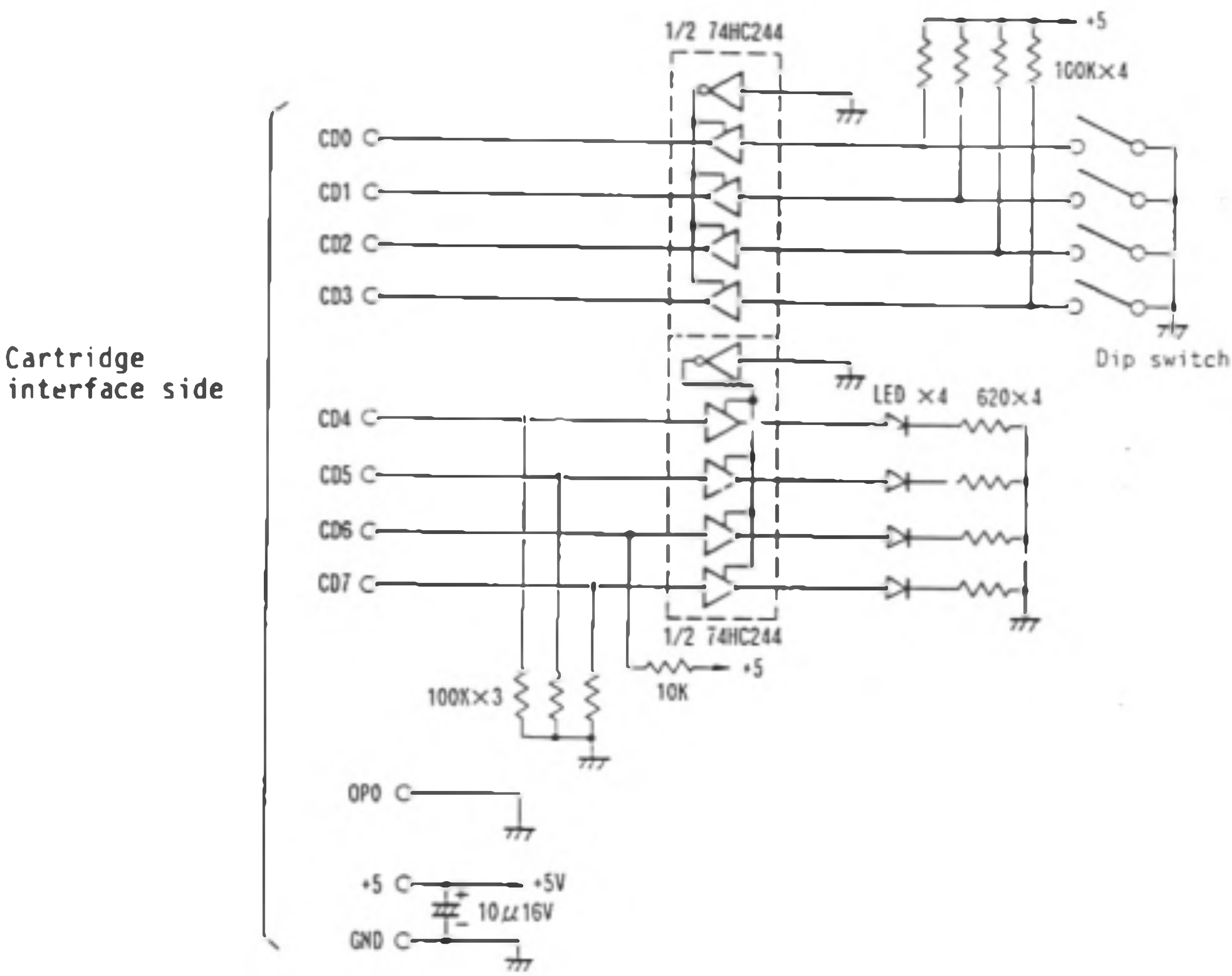


Fig 4.15 IO Mode Application Circuit

Simple circuit using the buffer, LED and dip switch is introduced as an IO mode application. CD6 is pulled up to +5 V by resistance of 10 K ohm and OP0="L" to obtain the device address 40H in this example. When this circuit is connected to the cartridge I/F of EHT-10/EHT-10/2 and the main unit power is turned on, OS automatically sets IO mode to switch register [P18H] so that the application program can read the contents of the dip switch and turn on/off the LED by P1CH read/write. CD4 to CD7 pull-down resistance of 100 k ohm is for input protection (74HC244 in this case). Resistance value of 100 K ohm to 1 M ohm is suggested.

(2) DB mode application circuit

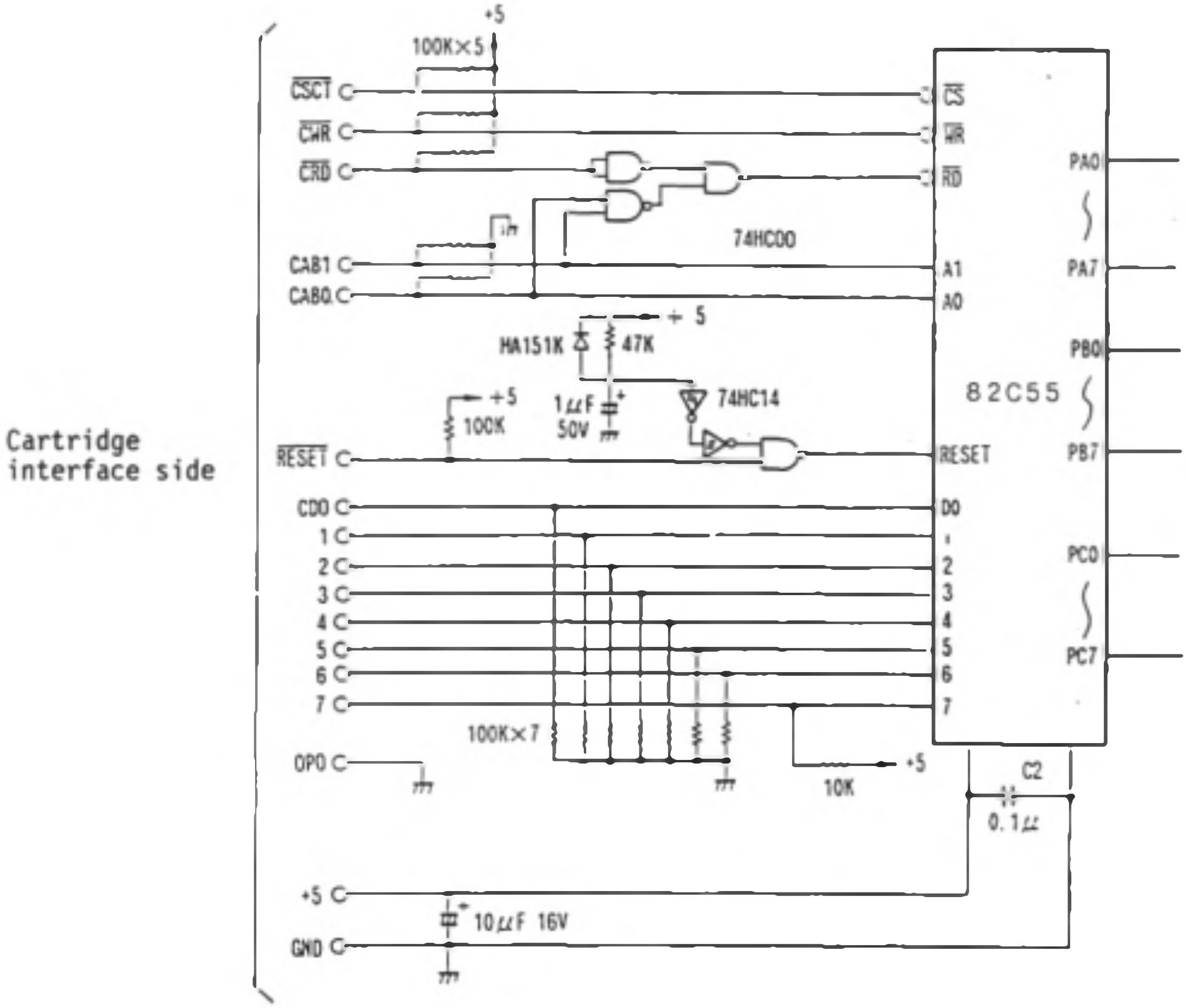


Fig 4.16 DB Mode Application Circuit

We will connect 82C55 to DB mode interface as an application example. EHT-10/EHT-10/2 OS reads the contents of I/O address P13H and checks the device address. This operation is an 82C55 access disable condition, CRD disable gate is necessary to avoid malfunction. When the device address is necessary in the application program, pull-up CD4 to CD7 to +5 V using resistance (about 10 K ohm). Device address is

set to 80H (Cartridge ID=8, CSEL=0) $\overline{\text{CSCT}}$, $\overline{\text{CWR}}$ and $\overline{\text{CRD}}$ are output when I/O address P10H to P13H is specified.

(3) OT mode application circuit

Cartridge interface side

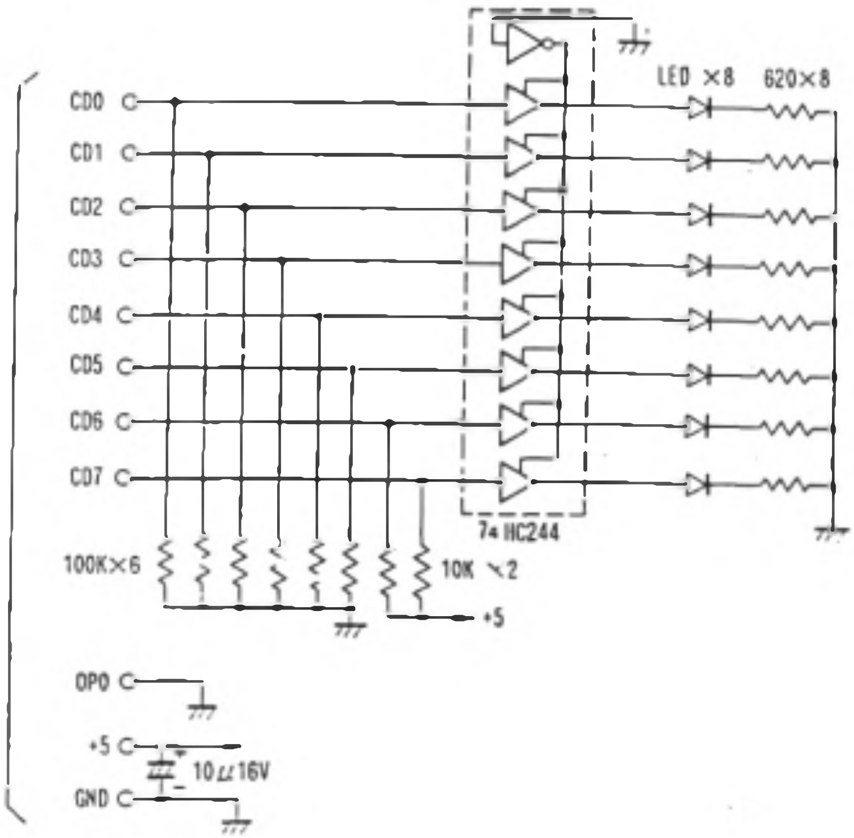


Fig 4.17 OT Mode Application Circuit

A simple circuit using buffer and LED is introduced as the OT mode application example. CD7 and CD6 are pulled up to +5 V by resistance of 10 K ohm and OP0="L" to obtain the device address C0H in this example. Writing P10H can turn on/off LED individually.

(Notes)

(1) When EHT-10/EHT-10/2 turns off the power supply in the continue mode, the reset signal is input to 82C55, thus, the application program cannot operate correctly.
See "Software Version 11.4 Cartridge Device Expansion" to operate the application program in continue mode.

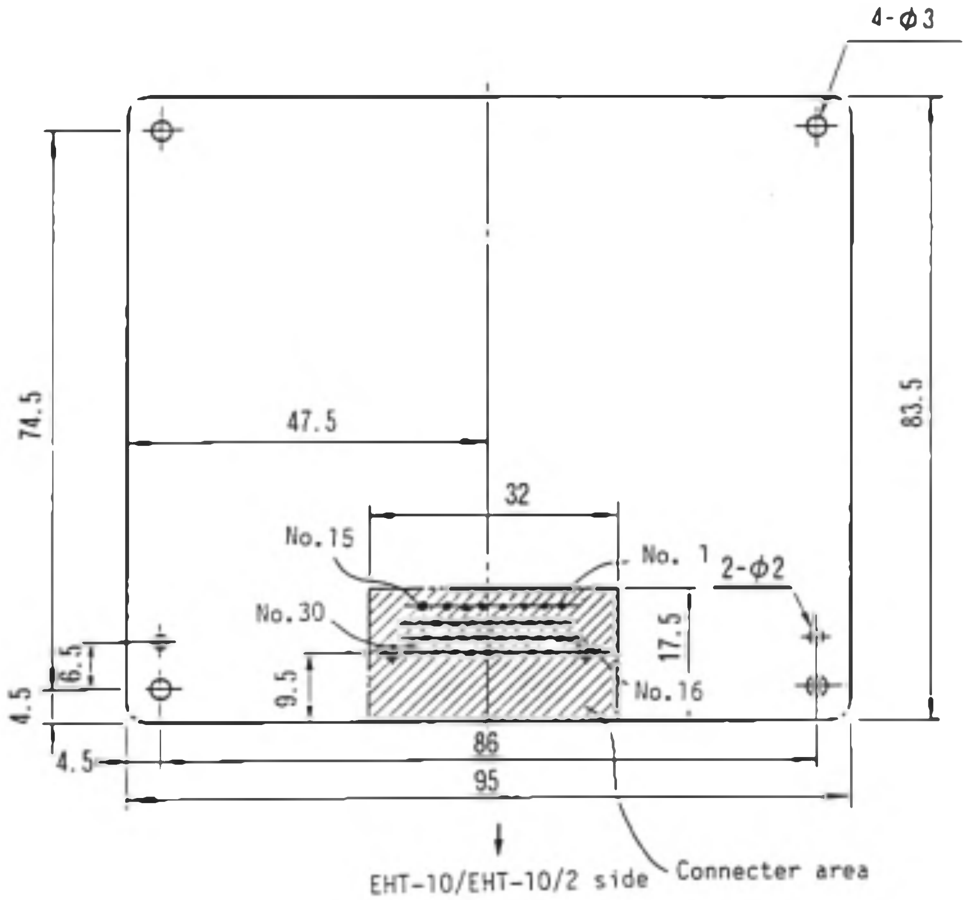
(2) Use CMOS-IC as the circuit to connect to the cartridge option as much as possible in order to conserve power consumption.

4.1.11 Universal Unit Circuit Board Size

To design various option unit using cartridge IO, the universal unit is prepared.

The circuit board size for the universal unit is described below.

Circuit board	1.6 mm glass
Pitch	1/10 inch (2.54 mm)



Unit mm

Fig 4.18 Universal Unit Circuit Board Size

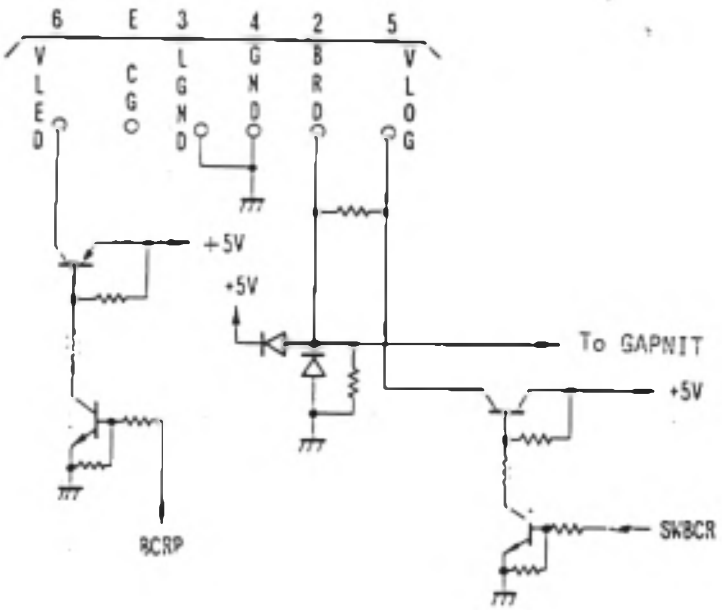


Fig 4.19 BarCode Reader Interface Circuit

Barcode can be decoded using the timer/counter in gate array GAPNIT. In addition to barcode data input, the light emitting diode power supply and logic power supply are barcode reader interfaces which can be controlled individually. NP-410 type barcode reader is used for EHT-10/EHT-10/2.

4.2.1 Connector in Use

The following barcode reader I/F connector is used.

TCS7560-01-101

4.2.2 Terminal and Function

Terminal	Pin number	I/O	Contents
BRD	2	I	Barcode reader read signal "L" when white
LGND	3	-	Logic ground
GND	4	-	LED ground
VLOG	5	-	Barcode reader logic section power supply.
VLED	6	-	Barcode reader LED power supply.
CG	E	-	Barcode reader case ground

Table 4.10 Barcode reader I/F Terminal Function

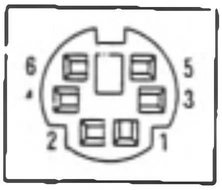


Fig 4.20 Bar Code Reader I/F Pin Arrangement

4.2.3 I/O Register

(1) Barcode Reader LED Power Control

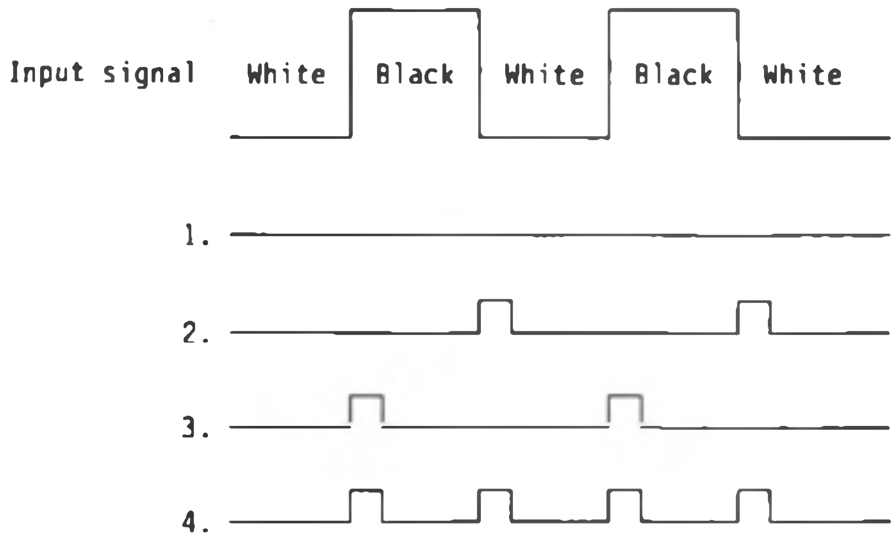
Write "1" to ICCTLR [P17H] bit 3 in order to supply power to the barcode reader LED. As this register is not effected by the reset signal, care is required.

(2) Barcode Reader Logic Power Supply Control

Write "1" to CTLR1 [P00H] bit 3 in order to supply logic power to the barcode reader. This register becomes all "0" by reset.

(3) Input signal BRD

The signal input by the barcode reader is fetched at gate array GAPNIT. This signal line is directly read by reading the I/O register STR [P05H] bit 1.
As Input Capture Register (ICR) latches the value of Free Running Counter (FRC) every time the signal changes, the time difference between the latch pulse and the next latch pulse can be measured. The latch pulse generation can be acknowledged by Input Capture Flag ICF [P04H bit 2], this ICF is reset by reading ICRH.B [P03H].
Latch pulse generation method can be selected by setting control register CTLR1 [P00H] as follows.



	CTLR1 (POOH)	BCR1 (bit2)	BCR0 (bit1)	
1.		0	0	No trigger
2.		0	1	Fall trigger
3.		1	0	Rise trigger
4.		1	1	Fall/rise trigger

Fig 4.21 latch Pulse Generation Method

4.2.4 Power Characteristic

Power characteristic of the barcode reader interface section is as listed below.

LED power voltage : 5V ± 10%
Service current to LED power supply : 50 mA MAX (See note)
Logic power voltage : 5V ± 10%
Service current to logic power supply.: 50 mA MAX (See note)

(Note) This value is determined by the transistor characteristic for control, not provided power capacity for barcode. When this I/F is used with another load (IC card for example), it is limited by the suppliable capacity at the power supply.

4.3 RS-232C Interface

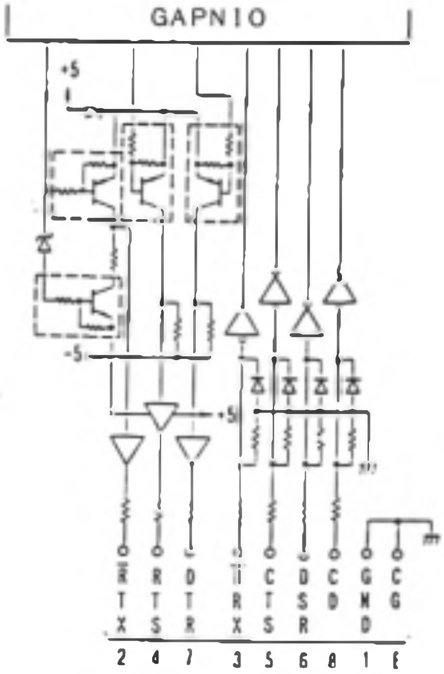


Fig. 4.22 RS-232C I/F Circuit

EHT-10/EHT-10/2 serial interface has built-in ART (Aperiodic Receiver/transmitter) which is the same level as 8251. This ART consists of the functions from 8251 functions, which are only necessary for EHT-10/EHT-10/2. The send data section is independent from the receive data section and clocks sent from the baud rate generator are individually input.
Send/receive data sections have a double buffer structure.
If buffer control function is used, even after host requests to stop send data, EHT-10/EHT-10/2 may output data of 2 to 3 bytes before send data stops. Receive buffer holds 256 bytes in main memory in addition to this.
ART outputs interrupt signal RXRDY to the main CPU after receiving data.

4.3.1 Serial Mode Switching

Gate array GAPNIO has 3 systems of serial interface; RS-232C, IC card, cartridge SIO (Serial IO). Use the Switch Register SWR [P18H] to select.
Selected serial interface is connected to ART (same level as 8251) in the gate array. This is called serial mode switching. Serial mode switching by switch register SWR is as listed below. Pay attention to the change in the contents of RCTS [P16H bit 5].

Switch Register SWR Write

Serial mode	SSW3(Bit 3)	SSW2(Bit2)	Contents I/O	Port P16H Value of RCTS
0	0	0	Cartridge SIO	1
1	0	1	IC card	
2	1	0	RS-232C	Reversed RS-232C CTS terminal

Serial mode switching block is as shown below.

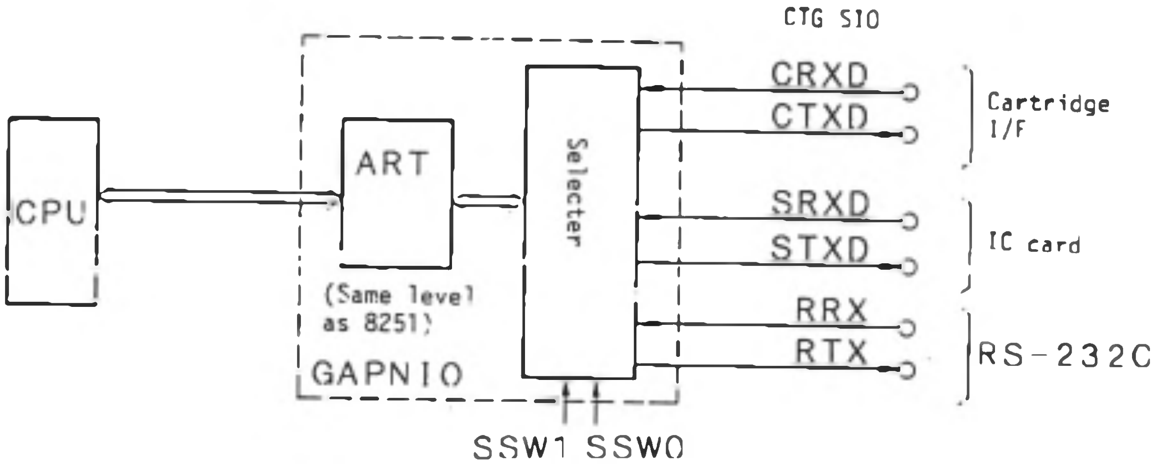


Fig 4.23 Serial Mode Switching Block

4.3.2 Connector in Use

The following connector is used for RS-232C I/F.

TCS7580-01-101

4.3.3 Terminal and Function

Table 4.11 Terminal functions of RS-232C interface

Terminal	Pin number	I/O	Contents
GND	1	-	Ground
RTX	2	O	Serial data output
RRX	3	I	Serial data input
RTS	4	O	Request to send
CTS	5	I	Clear to send
DSR	6	I	Data set ready
DTR	7	O	Data transmit ready
CD	8	I	Carrier detect
CG	E	-	Case ground

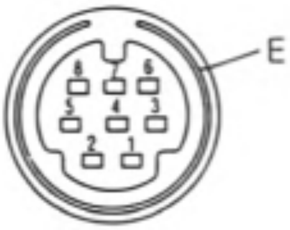


Fig 4.24 RS-232C I/F Pin Arrangement

4.3.4 Function Overview

Table 4.12 Specification of RS-232C Interface

Signal level	RS-232C level (±5 V)
Bit transfer rate	110,150,200,300,600,1200,2400,4800,9600,19200 38400,75 (bps)
Start bit	1 bit
Stop bit	1 or 2 bit
Parity	Even or odd parity/no parity
Error check	Parity error, framing error, overrun error
Communication format	Full duplex
Receive level	Within ±15 V

Set Switch Register SWR [P18H] SSW1 and SSW0 to SSW1=1 and SSW0=0 respectively to use the RS-232C interface.

* Send level of -5 V fluctuates slightly depending on the other load condition (-6 $\pm$1 V).

4.3.5 Difference From 8251 (USART)

The major differences of the 8251 (USART) from the EHT-10/EHT-10/2 RS-232C interface are as listed below.

Contents	8251	EHT-10/EHT-10/2 RS-232C Interface
Communication format	2 types: Synchronous and asynchronous	Only asynchronous
Stop bit	3 types: 1, 1 1/2, and 2bits	2 types: 1 and 2 bits
Data length	4 types: 5, 6, 7, 8 bits	2 types: 7 and 8 bits
Clock rate	3 types of clock: 1, 16, and 64 times	Fixed to clock of 16 times
CTS terminal	Sent when CTS=0. It is hardware controlled.	Send data can be controlled by reading the CTS signal through the I/O port. It is not hardware controlled.
Enter mode (EH)	Enabled	Disabled as only asynchronous is supported.
Internal reset(IR)	Enabled	Disabled (See Note 1)
Break Detect (BD)	Enabled	Disabled (See Note 2)

Table 4.13 Difference From 8251 (USART)

(Note 1) IR is used in 8251 for return from command instruction to mode instruction. AS EHT-10/EHT-10/2 RS-232C I/F has a different address for Command Register [P16H] and Mode Register [P15H], Internal Reset is not necessary.
Set TXE=0, RXE=0 before rewrite of the mode register.

(Note 2) 8251 status register bit 6 is BD (Break Detect) but EHT-10/EHT-10/2 RS-232C I/F does not support this function by the hardware.

4.3.6 Error Check During Receive Data

The EHT-10/EHT-10/2 RS-232C interface can acknowledge through the I/O port that an error has occurred while receiving data the same as 8251. FE (Framing Error) corresponds to I/O port P15H bit 5, OE (Overrun Error) corresponds to bit 4 and PE (Parity Error) corresponds to bit 3. They operate as follows.

FE (Framing Error) ... Even when a framing error occurs, data receive operation is not effected and continues. When successive data is fetched, the newly received data is checked for framing error and FE is reset if the stop bit is correct. Reset is also performed by reset input or error reset command (ER=1).

OE (Overrun Error) ... Even when overrun error occurs, the data receive operation is not effected and continues. OE is not reset even if successive data is correctly fetched. To reset OE, error reset command (ER=1) or reset input is necessary.

PE (Parity Error) ... PE reset condition is the same as FE. Parity is checked only when PEN=1. PE is 0 when PEN=0.

4.4 IC Card Interface

The EHT-10/EHT-10/2 can read/write data from/to the IC card based on the ISO DP7816. However, to write, only the IC card can be used as program voltage is +5 V constant.
Close the rear cover of the main unit when using IC card. The IC card connector is stored in the IC card, power is supplied and data is sent/received by connecting to the IC card connector, thus, avoiding an increase of resistance by touch. Set switch register serial mode to 1 (See 4.3.1) when using the IC card.

4.4.1 Connector in Use

The following IC card I/F connector is used.

ICC-8P

4.4.2 Terminal and Function

Table 4.14 Pin contact function of IC card I/F

Terminal	Pin number	I/O	Description
RST	2	O	Reset signal to IC card which resets the CPU in the IC card.
VCC	1	-	IC card CPU power supply
GND	5	-	Ground
VPP	6	-	IC card EPROM read/write power supply
I/O	7	I/O	Serial data I/O line which handles data with the IC card.
CLK	3	O	Clock supplied in the IC card 4.9152 MHz

(Note) Neither pin 4 nor pin 8 is connected.

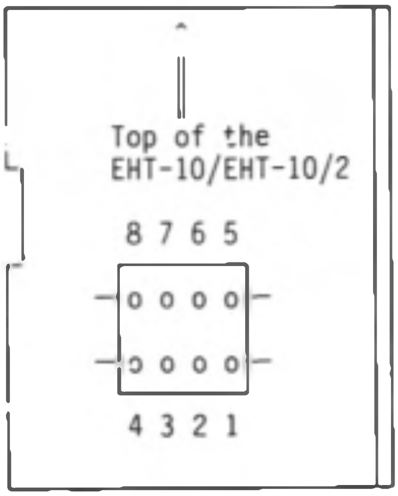


Fig 4.25 IC Card Pin Arrangement

4.4.3 IC Card Interface Block

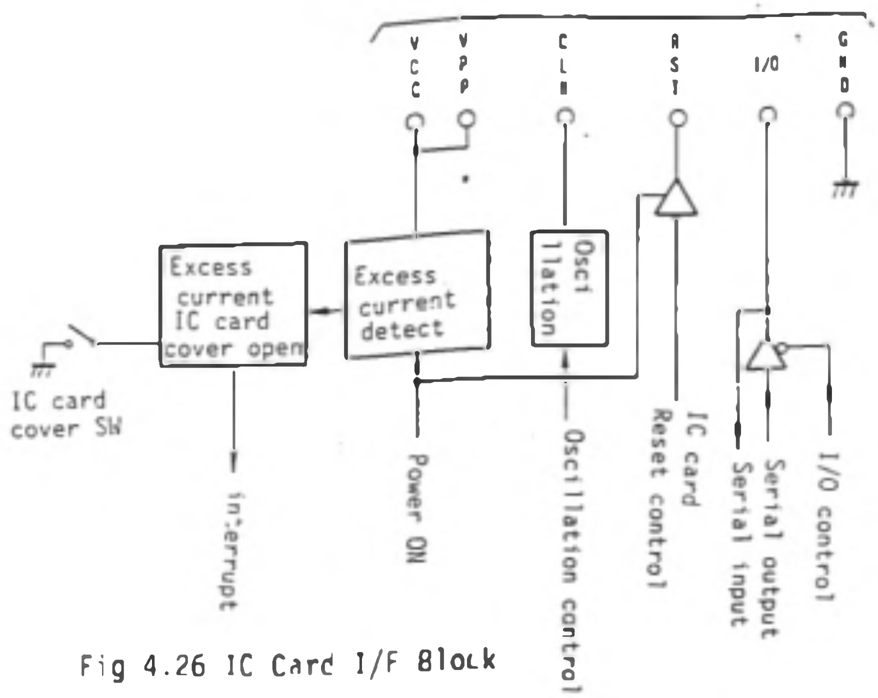


Fig 4.26 IC Card I/F Block

(Note)

Interrupt : Two factors of interrupt from the IC card to the EHT-10/EHT-10/2 are as follows.

(1) EXT (external) interrupt
(2) ART interrupt

(1) EXT interrupt issues when excess current goes through the IC card and IC card cover is open.

(2) ART interrupt issues when data is received from IC card. These interrupts can be read as status through the I/O port, but when interrupt enable, it can be processed in the interrupt program. See "Chapter 3 Interrupt Description" for further information.

4.4.4 Power Characteristics

Power characteristics of the IC card interface section is as listed below.

CPU drive voltage (Vcc)	5 V ± 5 %
IC memory, read/write voltage (Vcc)	5 V ± 5 %
Clock frequency (CLK)	4.9152 MHz
Data transfer rate	9600 bps
Supply current	140 mA MAX (See note)
Excess current detect current	200 mA

(Note) This is a suppliable current value when only the IC card is in use, when this I/F is used with another load (bar code reader for example), it is limited by suppliable current at the power supply, so care is required.

Characteristics other than mentioned above are based on ISODP7816.

4.5 Touch Panel Interface (EHT-10)

EHT-10 touch panel interface is configured as shown below.

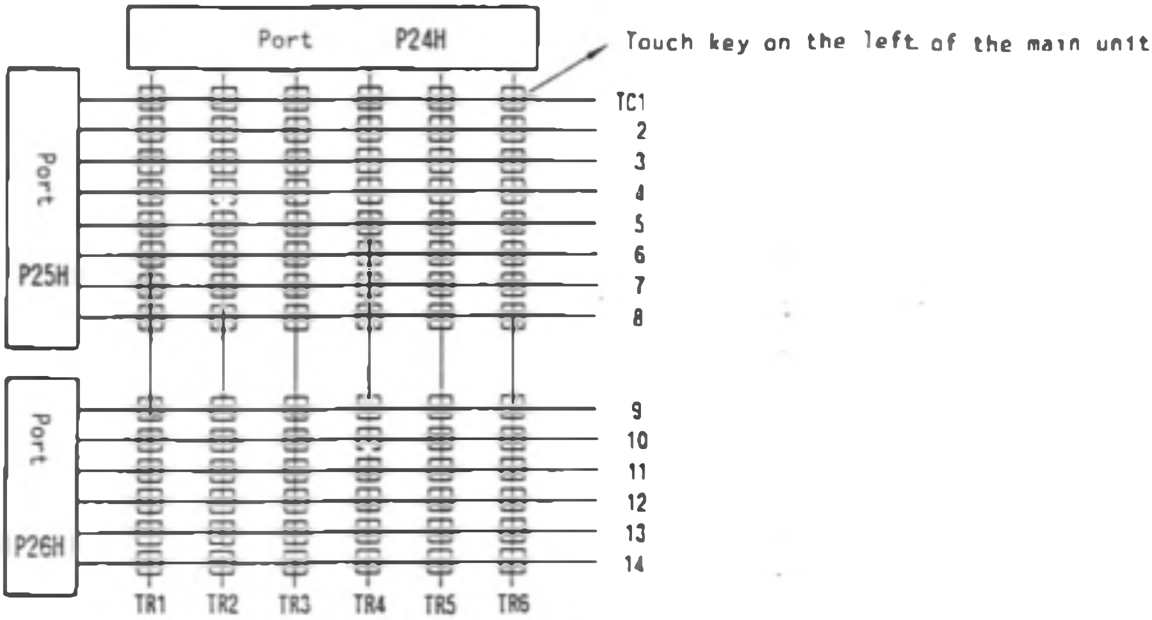


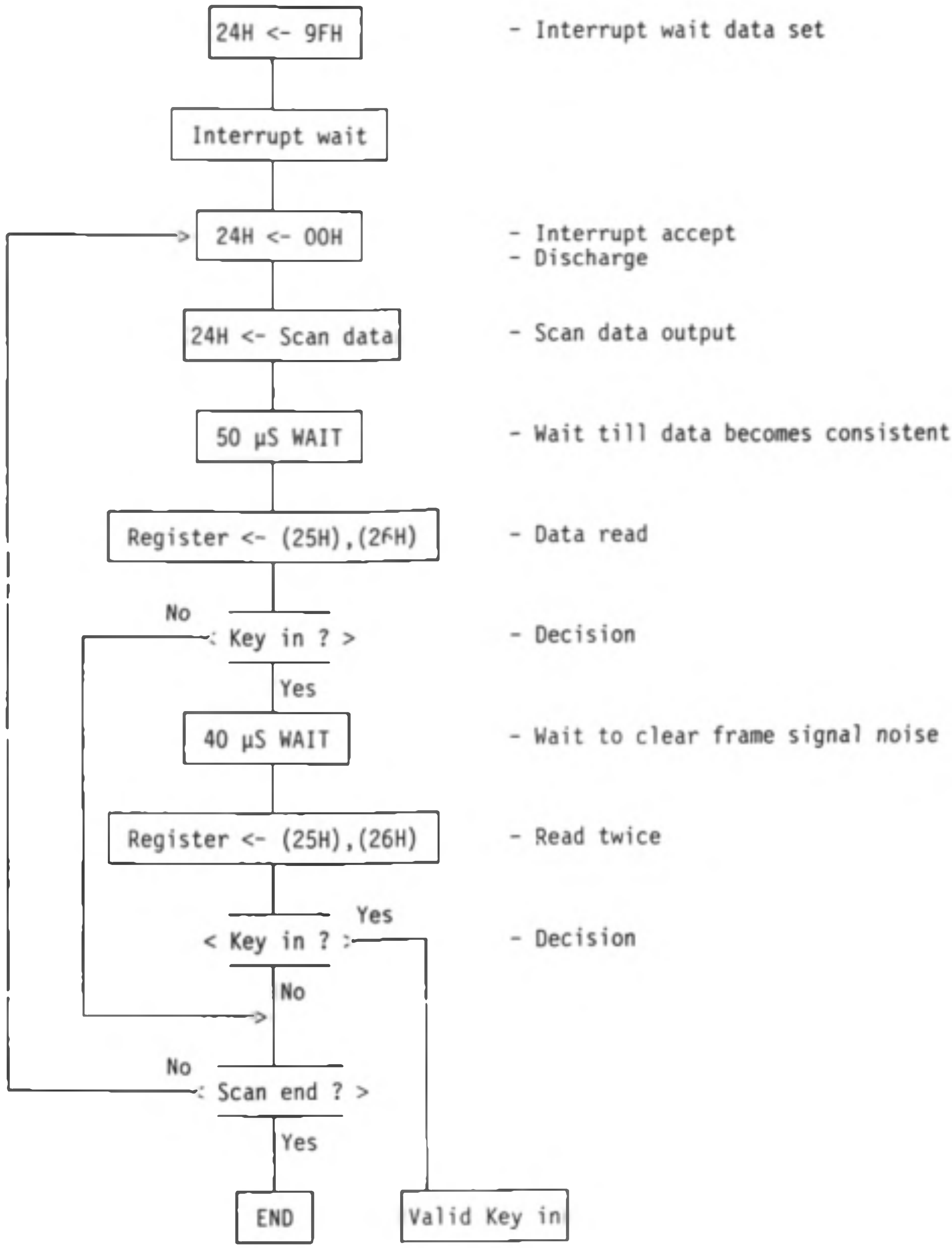
Fig 4.27 Touch Panel I/F Block

1 is written to all bits of port P24H. When any switch of the touch panel is pressed, it transmits to 7508 after the keyboard interface scan line is fetched and ANDed, key code "80H" is assumed to be pressed.

Key interrupt routine clears port P24H, raises 1 every 1 bit, reads touch key data of port P25H and P26H, scans thoroughly and detects the pressed key. After scan, write "1" to all bits of P24H again.

(Note) 7508 key scan starts after 16 msec after key interrupt issues, if it overlaps the touch panel scan, the touch panel switch may be assumed to separate.

Touch panel input method



Reading the key twice avoids the LCD frame signal from entering the touch panel as noise. This noise width is about 20 micro-seconds, therefore, read 40 micro-seconds of gap twice. As the interval of noise is about 35 ms, the number of times to read twice per scan is once at most.

4.6 LCD Control (EHT – 10)

EHT-10 LCD panel is a full graphic panel of 84 dots wide by
154 dots long which uses T6963 as an LCD controller.
Display duty is 1/48.

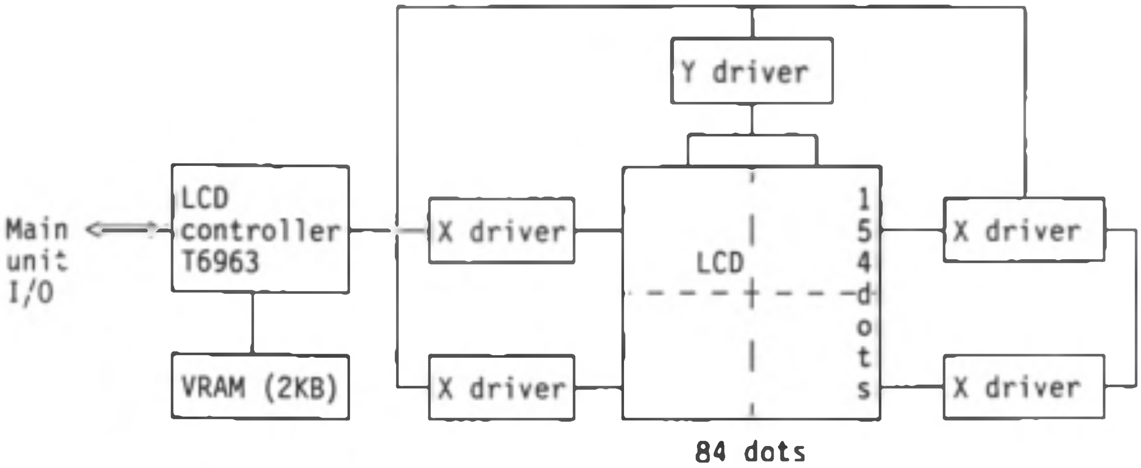


Fig 4.28 LCD Panel Block

For controller convenience, the 84 dots are configured as 48 dots x 2, a section of 6 dots x 2 is outside the display area which is assigned to VRAM (write 0 to reduce power consumption).
The display is entirely performed as graphics. T6963 has a built-in character generator, but VRAM address is different from the actual display position as shown in Table 4.15, character output using the character generator is not enabled.

4.6.1 Hardware configuration

EHT-10 LCD hardware configuration is as shown below.

LCD controller ..... T6963 x 1
driver .... T6961 x 1
.... T7778 x 4
VRAM .... CMOS 16 K bit SRAM

Relation of the position between VRAM address and display is listed below (address is HEX).

	1	2	…………	41	42	43	…………	84	
MBS	0	28	…………	668	14	3C	…………	67C	1
LSB	1	29		669	15	3D		67D	9
	2	2A		66A	16	3E		67E	17
	3	2B		66B	17	3F		67F	25
	4	2C		66C	18	40		680	33
	5	2D		66D	19	41		681	41
	6	2E		66E	1A	42		682	49
	7	2F		66F	1B	43		683	57
	8	30		670	1C	44		684	65
	9	31		671	1D	45		685	73
	A	32		672	1E	46		686	81
	B	33		673	1F	47		687	89
	C	34		674	20	48		688	97
	D	35		675	21	49		689	105
	E	36		676	22	4A		68A	113
	F	37		677	23	4B		68B	121
	10	38		678	24	4C		68C	129
	11	39		679	25	4D		68D	137
	12	3A		67A	26	4E		68E	145
	13	38	…………	67B	27	4F	…………	68F	153
									154
									160

Table 4.15 Relation of Position between VRAM Address and Display

VRAM valid addresses are B6 and B7 for data in the address (13H, 3BH ...) which is the same level as the bottom line in 0FH to 68FH.

4.6.2 VRAM Address Map

Address	Area
7FFH	
	128 bytes (1)
780H	
	240 bytes (2)
690H	
	1680 bytes (3)
000H	

(1) Any data can be written.
(2) Data is sent, but not displayed.
Write 0 to reduce power consumption.
(3) LCD display valid area.

With T6963, this VRAM is virtual memory area and the area actually displayed on screen is the real area (1680 bytes fixed). The real area can be variable according to command.
The address map on the left shows the set values of EHT-10 and EHT-10/2 OS.

4.6.3 I/O Port

I/O port	READ	WRITE
P20H	LCDC data	LCDC data
P21H	LCDC status	LCDC command

LCDC T6963 register is assigned to I/O port P20H and P21H. When these ports are accessed in an application program, LCDC status must be read and timing of read/write is necessary (however, LCDC status read or timing is not necessary with a relatively slow language such as BASIC).
Table 4.16 shows T6963 status and Figure 4.29 shows the communication flow with CPU.

Status	Contents
STA0 ($\overline{\text{BUSY1}}$)	To check whether instruction is executable. =1 : Executable =0 : Non-executable (during instruction execution)
STA1 ($\overline{\text{BUSY2}}$)	To check whether data can be written/read. =1 : Data enable =0 : Data disable (during internal process)
STA2 (DAV)	To check whether data can be read (valid only when AUTO). =1 : Read enable =0 : Read disable
STA3 (DAV)	To check whether data can be written (valid only when AUTO). =1 : Write enable =0 : Write disable
STA4	Don't care.
STA5 ($\overline{\text{CLR}}$)	To check whether the controller is operatable. =1 : operatable =0 : Not operatable
STA6 (ERROR)	If a point other than on the real screen is set while copy, the flag is set. Then if the instruction is executed, the flag is reset.
STA7 (BLINK)	To check blink state (approximate cycle 1 second duty 50%) =1 : Normal display =0 : OFF

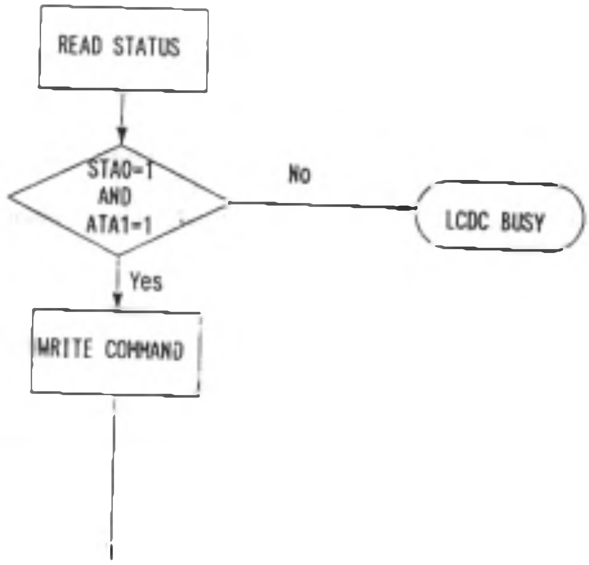
Table 4.16 T6963 Status Table

MSB							LSB
STA7	STA6	STA5	STA4	STA3	STA2	STA1	STA0

(Description)

STA5 : The clock is unstable for 1 to 2 ms after HALT release. T6963 does not operate during this time.

(a) Command write



(b) Data write

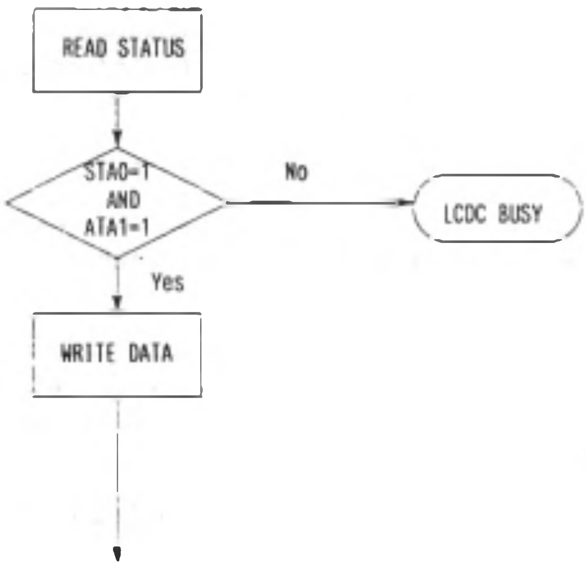


Fig 4.29 Communication Flow of T6963 with CPU

4.6.4 T6963 Command

The T6963 contains a great number of commands, but the commands related to text display cannot be used due to the hardware of the EHT-10/EHT-10/2. Only the commands used in the EHT-10/EHT-10/2 are described.
Initial data must be written before writing commands (WTRG, WTRM, DR/W), necessary for the T6963.

(1) Internal Register Write (WTRG)

	7	6	5	4	3	2	1	0
Command	0	0	1	0	0	1	0	0
Data D1	Address (low)							
D2	Address (high)							

A user uses this to specify a RAM point when writing data to RAM or reading data from RAM.
(OH < address < 68FH)

(2) Internal RAM Write (WTRM)

7	6	5	4	3	2	1	0	
0	1	0	0	0	0			D1,D2
						N		

N	Meaning	D1	D2
10	Display graphic home address	Address(low)	Address(high)
11	Number of graphic columns	Column	0

Display graphic home address: Specifies the point in the virtual memory graphic area to place the home position on the real screen. EHT-10 and EHT-10/2 initial value is D1=0, D2=0.

Number of graphic columns : Specifies the number of columns per line to use the real graphic area in the virtual memory graphic area. The EHT-10/EHT-10/2 initial value is D1=(160 dots x 2)/8=40. A user should not set values without proper preparation for the number of graphic columns and graphic home address.

(3) Display Mode Set (DSPM)

7	6	5	4	3	2	1	0
1	0	0	1	N	0	0	0

N	Meaning
1	Displays graphics
0	Disables graphic display

No data. This command enables LCD screen on/off.

(4) Data Read/Write (DR/W)

7	6	5	4	3	2	1	0
1	1	0	0	0			

D1 (at write)

N

N	Meaning
000	Data write (Address pointer up after execution)
001	Data read (Address pointer up after execution)
010	Data write (Address pointer down after execution)
011	Data read (Address pointer down after execution)
1*0	Data write (Address pointer change after execution)
1*1	Data read (Address pointer change after execution)

Data write executes instructions after data set.

(5) Auto Mode (AS/R)

7	6	5	4	3	2	1	0
1	0	1	1	0	0		

N

N	Meaning
00	Data auto write set
01	Data auto read set
1*	Auto reset

Data auto write sends data after executing this instruction.
Address pointer goes up every time data is written (read) till auto reset is accepted.

(6) Screen Peek (PEEK)

7	6	5	4	3	2	1	0
1	1	1	0	0	0	0	0

If the point specified by the address pointer equals the point of the graphic pointer on the real screen, 1 byte of displayed data on the aligned point is transferred to the stack so it can be read by a user. If the pointer specified by the address pointer is not in the graphic pointer area on the real screen, this instruction is ignored and the status flag is set.

(7) Screen Copy (COPY)

7	6	5	4	3	2	1	0
1	1	1	0	1	0	0	0

If the point specified by the address pointer equals the point of the graphic pointer on the real screen, 1 byte of data is written at a time sequentially to the aligned graphic pointer RAM area, this data is displayed for one line of real screen that follows the aligned point. If the point specified by the address point is not in the graphic pointer area, this instruction is ignored and the status flag is set.

(8) Bit Set/Reset (BS/R)

7	6	5	4	3	2	1	0
1	1	1	1				

N (bits 2–0)
1: Set (bit 3)
0: Reset

N	Contents
000	Bit 0 (LSB)
001	1
010	2
011	3
100	4
101	5
110	6
111	7 (MSB)

This command sets (1) and resets(0) bit specified by memory N which is specified by the address pointer.

(C) Notes

When auto commands of data auto write set and data auto read set are executed, all commands that follow are ignored. Therefore, if a new command is issued, auto reset must be executed.

4.6.5 Sample Program

Three programs which write "1" (black) over the entire LCD screen, put it in the auto mode, and write "A" in the upper left corner using the write command are shown below.

Sample program 1 Screen filled with 1. (When Data Write command is used.)

```
100  CLS                   'Auto reset
110  OUT &H21,&H2B
120  OUT &H20,0
130  OUT &H20,0
140  OUT &H21,&H24         'Set address pointer
150  FOR I=1 TO 20*84
160       OUT &H20,&HFF    'Data written to screen
170       OUT &H21,&HC0    'Data Write command
180  NEXT
190  END
```

Sample program 2 Screen filled with 1. (When Auto Mode command is used.)

```
100  CLS                    'Auto reset
110  OUT &H21,&H28
120  OUT &H20,0
130  OUT &H20,0
140  OUT &H21,&H24          'Set address pointer
150  OUT &H21,&HB0          'Data auto write set
160  FOR I=1 TO 20*84
170       OUT &H20,&HFF     'Data written to screen
180  NEXT
190  END
```

Sample program 3 Display "A" in the upper left corner of the screen

```
100  CLS
110  OUT &H21,&HB2          'Auto reset
120  ADRS=0                 'Address pointer initial setting
130  GOSUB 260
140  FOR I=0 TO 260
150       READ A$
160       D=VAL("&H" + A$)
170       OUT &H20,D
180       OUT &H21,&HC4     'Data Write command
190       ADRS=ADRS+40      'Pointer is moved to the next line
200       GOSUB 260
210  NEXT
220  END
230  '
240  DATA 3E,48,88,48,3E
250  '
260  HI=INT(ADRS/256)       'Address of pointer write
270  LO=ADRS-HI*256
280  OUT &H20,LO
290  OUT &H20,HI
300  OUT &H21,&H24
310  RETURN
```

T6963 status is not checked when data and command are written to T6963 in the sample program 1 to 3 as it is not necessary due to the process rate of BASIC. If a user writes in assembler, the status must be checked before data and command read/write.

4.7 Buzzer

EHT-10 and EHT-10/2 used a piezo-electric buzzer. Various frequencies can be selected through software using BIOS BEEP, and various frequencies can be hardware generated by writing "1" or "0".

The circuit for buzzer is as shown below.

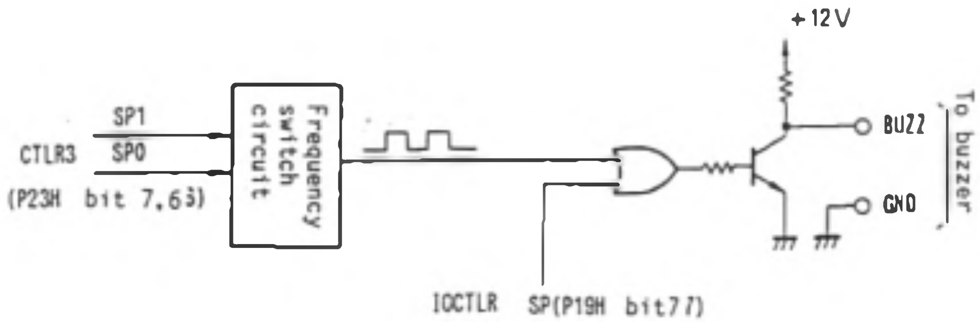


Fig. 4.30 Buzzer Block

Set everything to "0" (SP1=SP0=SP=0) when the buzzer does not sound, SP and CRLR3 should not mask each other.

4.8 Counter

4.8.1 1 Second Counter

This counter is a memory area which counts 1 second signals of the calendar clock. It counts through the OS and can be set to any as desired. Normally, these counters are used at auto power off.

```
F02EH,F02DH : Up count
F030H,F02FH : Down count

10    A=PEEK(&HF02F)
20    B=PEEK(&HF030)
30    PRINT B*256+A
40    GOTO 10
```

4.8.2 1/10 Second Counter

The 16 bit timer counter issues an interrupt when FFFFH -> 0000H. This interrupt is issued every 106.667 ms and is used as a count lock. The interrupt flag (F093H) monitors whether the interrupt is issued. This flag must be reset after reading.

```
F093H bit 7: 7508
          6: ART
          4: ICF
          3: OVF  Timer counter OVF
          2: EXT

10    A=PEEK(&HF093)
20    B=A AND &H08
30    IF B=0 GOTO 10
40    POKE &HF093,&H0C
50    C=C+1
60    PRINT C*(65.563/614.4)
70    GOTO 10
```

4.9 Dip Switch

The dip switch state is acknowledged by memory address F5F2H (value should not be written in this address).

The dip switch corresponds to F5F2H bits as shown below. However, DP1 is different from the other switches. If DP1=ON, bit 3=0, if DP1=OFF, bit 3=1. DP2 to DP4 are set to 1 at ON, they are set to 0 at OFF. When a user changes the dip switches, press the Reset button or turn the Power switch on again. If not, the contents of the dip switch is not reflected to F5F2H.

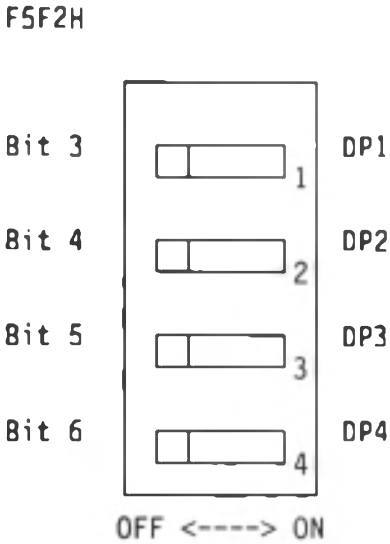


Fig. 4.31 Dip Switch

4.10 LED (EHT-10/2)

The EHT-10/2 has 3 LEDs on the keyboard. The LEDs can be controlled by IOCTLR [P19H], but may destroy other bits. Due to this, OS writes the same contents written to IOCTLR to memory address F0DDH. Therefore, when a user controls LED on/off, set the bit in the content of F0DDH, write to IOCTLR and return the value to F0DDH. When 1 is written to the corresponding bit, the LED turns on.

```
        MSB                             LSB
       +---+---+---+---+---+---+---+---+
F0DDH  | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
       +---+---+---+---+---+---+---+---+
             |   |   |
             |   |   +------> LED2
             |   +----------> LED1
             +--------------> LED0
```

When turning on LED 2.

```
10    IO=PEEK(&HF0DD)
20    LED=IO OR &H40
30    POKE &HF0DD,LED
40    OUT &H19,LED
50    END
```

When turning off LED 2.

```
10    IO=PEEK(&HF0DD)
20    LED=IO AND &HBF
30    POKE &HF0DD,LED
40    OUT &H19,LED
50    END
```

5.1 Overview

In the EHT-10/EHT-10/2 power supply section, the step-up circuit and series regulator generate backup power and +5 V. DC-DC converter generates +12 V for the sub-battery charge, -5 V for serial communication and -15 V for LCD display. (See Figure 5.1.)

At power on, step-up circuit operates and steps up battery voltage to approx. 5.3 V and controls this to 5 V using a series regulator. DC-DC converter operates and outputs the voltages.

At power off, DC-DC converter and step-up circuit stop, the battery voltage which by-passes the step-up circuit becomes backup power through the series regulator.

Battery voltage is always monitored, when the voltage goes less than approx. 4.8 V, power failure signal generates and supply from the sub-battery starts.

The main battery and sub-battery are recharged using an AC adapter. Values of charging current are shown in the table on the next page, the AC adapter charges only the battery and does not supply the main unit. Thus, if the main unit consumes more than charging current, the battery supplements with extra current.

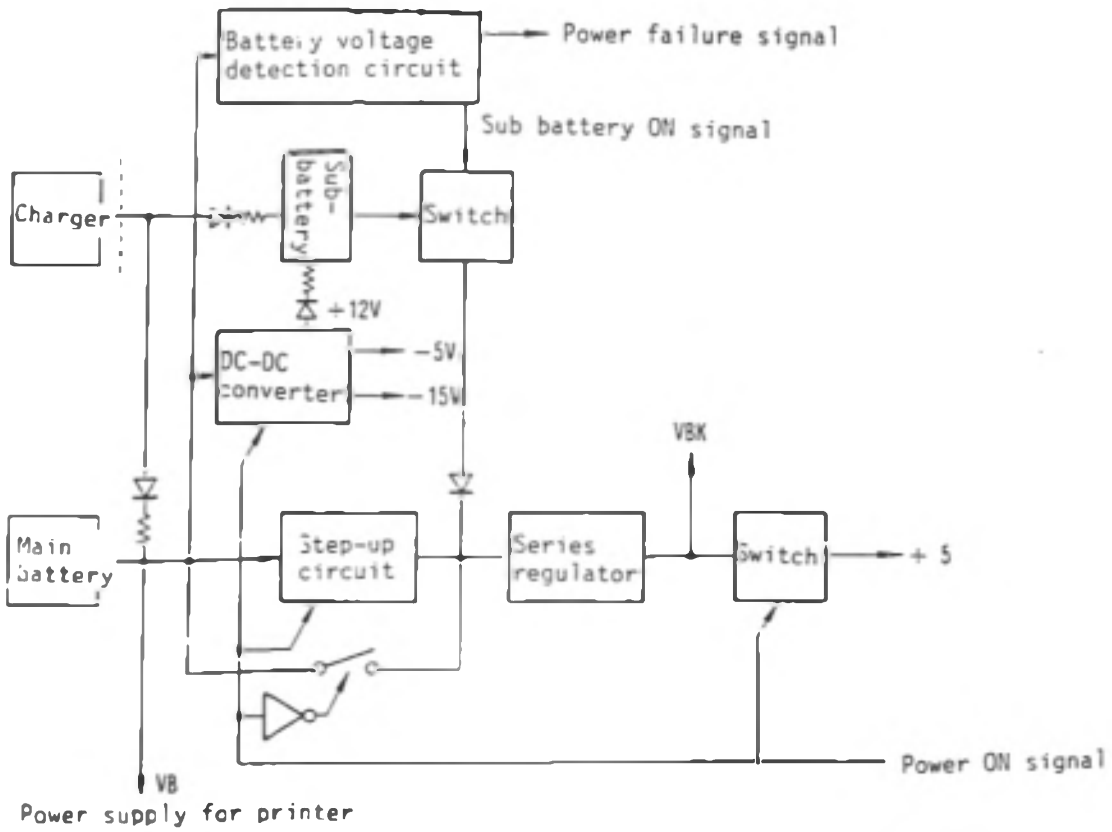


Fig. 5.1 EHT-10/EHT-10/2 Power Source Block

Table 5.1 shows power supply specifications.

Input voltage range DC 4.5 V to DC 6 V
Output voltage

Voltage		Standard	Current capacity
+5		+5±10%	Total of 200 mA MAX
VBK	Power ON	+5±5%	
	Power OFF	+5+5%-8%	5 mA MAX
-5		-6V±1V	10 mA MAX
-15		-15V±0.5V	5 mA MAX
+12		+12V±0.5V	5 mA MAX

Table 5.1 EHT-10/EHT-10/2 Power Source Section Specification

(Note) VBK, -15, +12 are not user available.

Power failure detection voltage: Approx. 4.7 V

Power failure release voltage after power failure occurs : Approx. 5.05 V

5.2 Charging Current to Battery by adaptor

Charging current by the AC adaptor (standard charger) is as listed below.

Main battery		Approx. 120 mA
Sub-battery	At power ON	Approx. 3 mA
	At power OFF	Approx. 1.5 mA

5.3 Battery Capacity

Main battery 700 mAH Nominal voltage 4.8 V

Sub-battery 45 mA Nominal voltage 4.8 V

Memory backup time (256 KB)

Approx. 700 hrs. with main and sub-batteries full.

Approx. 45 hrs. with sub-battery full.

5.4 Charger and Charging Time

EHT-10/EHT-10/2 uses H00CA* (AC adaptor/standard charger). Charging time is normally about 10 hours when using the AC adaptor, it takes 45 hours to charge the sub-battery for initial use or after not using for an extended period of time.

Power failure is detected when the battery voltage is less than approx. 4.7 V, CHARGE BATTERY is displayed. If not charged, memory contents are eventually destroyed. If the memory contents are destroyed, when the battery recovers, system initialization starts.

1. Cartridge interface signal guarantees operation for circuits which can be fixed to the main unit.

2. RS-232C interface is configured of CMOS circuitry, and does not use a line driver or receiver for the RS-232C. Therefore, a 10 to 20 m cable can be connected and operated, but capability cannot be guaranteed.

3. Power supply can be obtained from the main unit to an external circuit, but the supply capacity is limited, care is required.

4. Use CMOS or NMOS for external circuit ICs.

5. Cartridge interface output terminal CSTR7 is dedicated to the EHT-10/EHT-10/2 development cartridge and should not be used for other purposes by a user. This terminal must be left open (NC).

6.

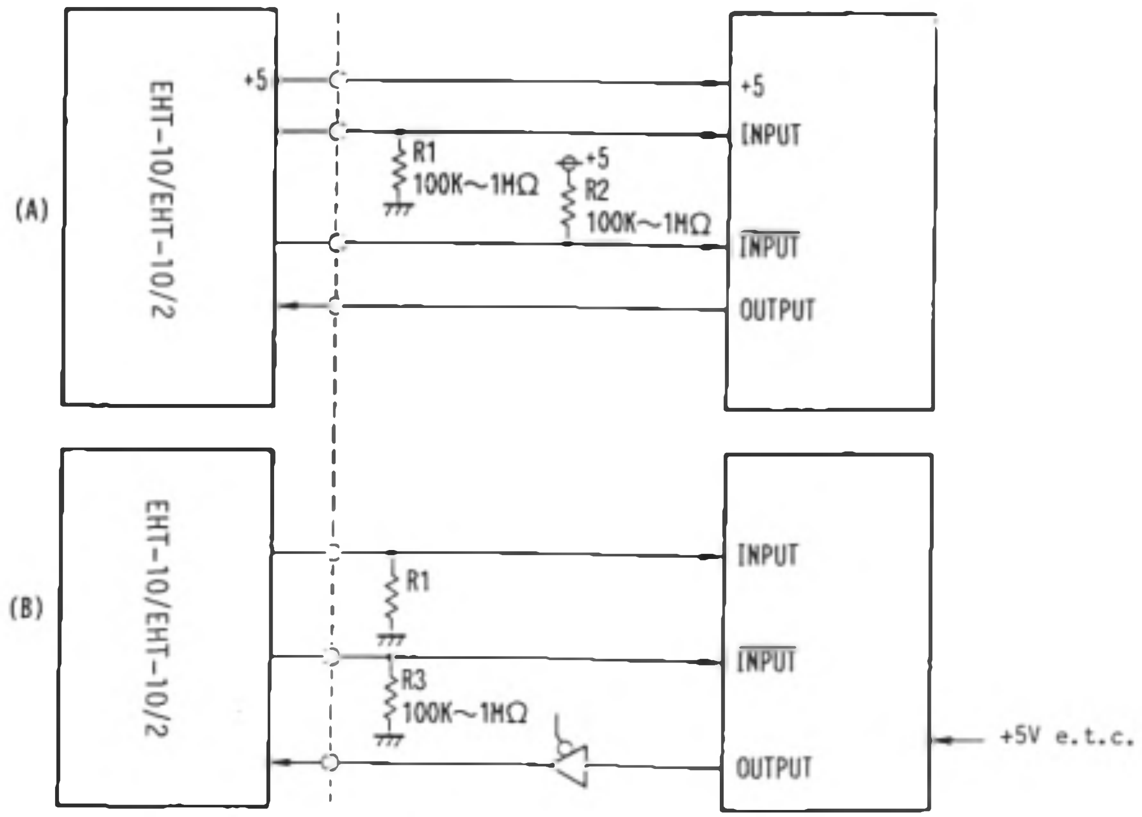


Fig 6.1 Cartridge Option Connection

When the power supply is supplied from the main unit to an external circuit (when A connection), an external circuit pull-up and pull-down must be performed in the same way as at the main unit side.

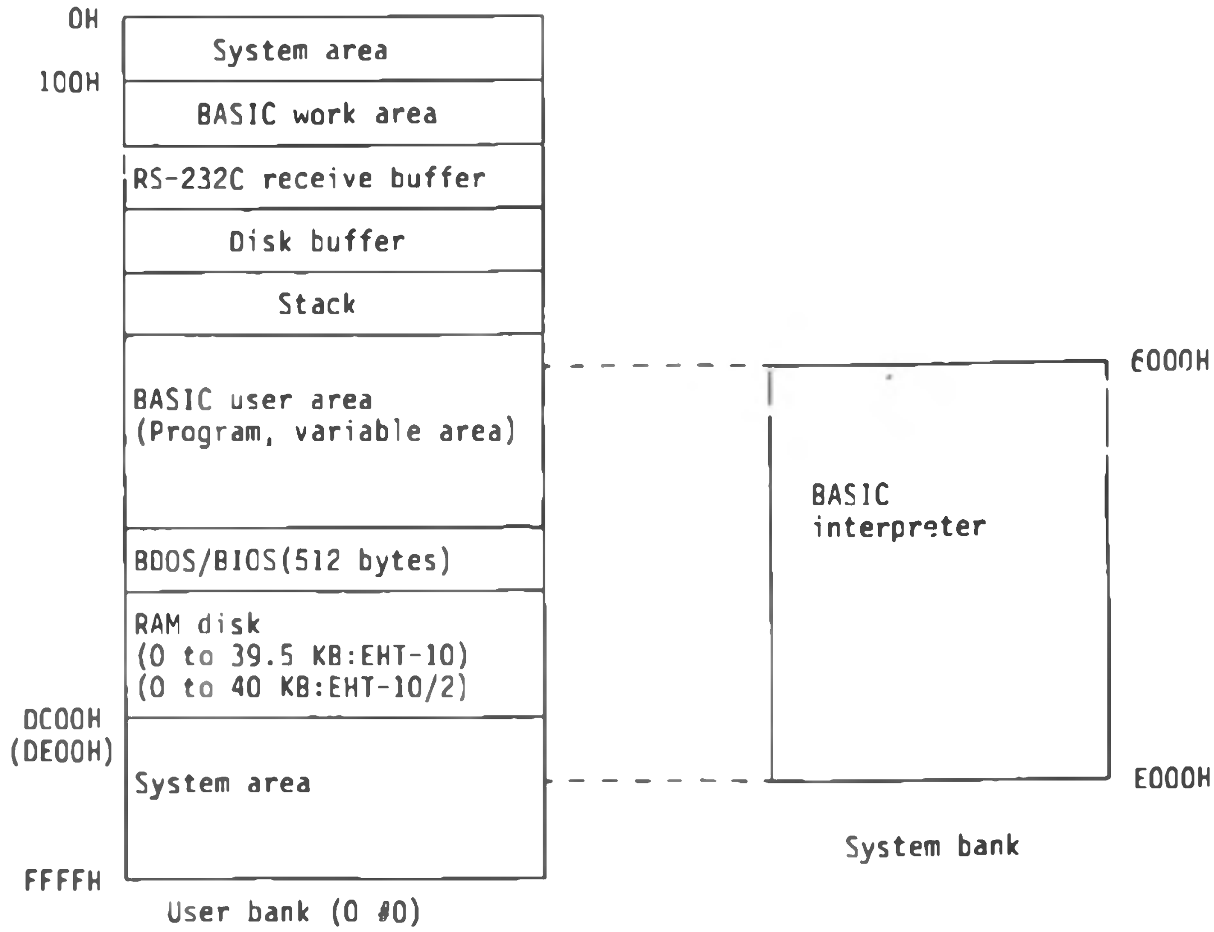
When an outside power supply is supplied to an external circuit (when B connection), the input signal must be pulled down. The external circuit output signal is tri-state output. This can avoid unnecessary current to flow from the external circuit when the main unit power is off. R1 to R3 are resistance for input terminal protection.

Part 3 BASIC

Chapter 1 BASIC Memory Management

1.1 Overview

The following memory map is used at the start of BASIC. The BASIC interpreter is in ROM (bank 1 #2), and BASIC program and variable area are in RAM.



(Note) Words and numbers in parentheses are for the EHT-10/2.

Fig 1.1 Memory Map at BASIC Start

1.2 Memory Map in RAM

The BASIC interpreter controls the area from address 100H in the user bank to directly before BDOS. However, as the high order address is variable depending on the CLEAR statement, it can be used as machine language area. When CLEAR, xxxx, yyyy is used, the user bank memory map is as shown below. With the CLEAR statement, xxxx indicates the high order address and yyyy indicates the stack size.

Address	Area	Pointers held in work area(Address)
0000H		
0100H	Work area	
		<- CCMBUF (099AH,099BH)
	RS-232C receive buffer	
		<- FILPTR (04A6H,04A7H)
	Disk buffer	
		<- LOWMEM (0962H,0963H)
	Stack (yyyy byte)	
		<- TXTTAB (03D9H,03DAH)
	Program area	
		<- VARTAB (07C6H,07C7H)
	Simple variable	
		<- ARYTAB (07C8H,07C9H)
	Array variable	
		<- STREND (07CAH,07CBH)
		<- FRETOP (079FH,07A0H)
	Character string	
xxxxH		<- HIGMEM (077AH,077BH)
	User area (machine language ... etc.)	
		<- MEMSIZ (0964H,0965H)
	BDOS/BIOS	

Fig 1.2 User Bank Memory Map

1.3 Options at BASIC Start

The following can be optionally set by CONFIG at the start of BASIC, this is equal to rewriting the following addresses. When BASIC starts, il references these areas, so if rewritten after the start of BASIC, it has no effect.

Option (Variable)	Address	NO. of bytes	Contents	Default
BASIC_F (BASICF)	F07CH	1	Number of files to open at the same time (Maximum number is 15.)	3
BASIC_S (BASICR)	F07DH	2	Maximum value of random file buffer at open in "R" mode	128
BASIC_C (BASICB)	F07FH	2	RS-232C receive buffer size	256

Table 1.1 Options at BASIC Start

As options BASIC_F, BASIC_S and BASIC_C effect the user area size (program, variable) according to the values set, care is required.

1.4 Disk Buffer

The disk buffer size is determined by option BASIC_F or BASIC_S and cannot be modified by a BASIC statement. The disk buffer is used as shown below. File 0 is used for commands such as LOAD, SAVE. Files starting from file 1 are used according to the file number specified by the OPEN statement.

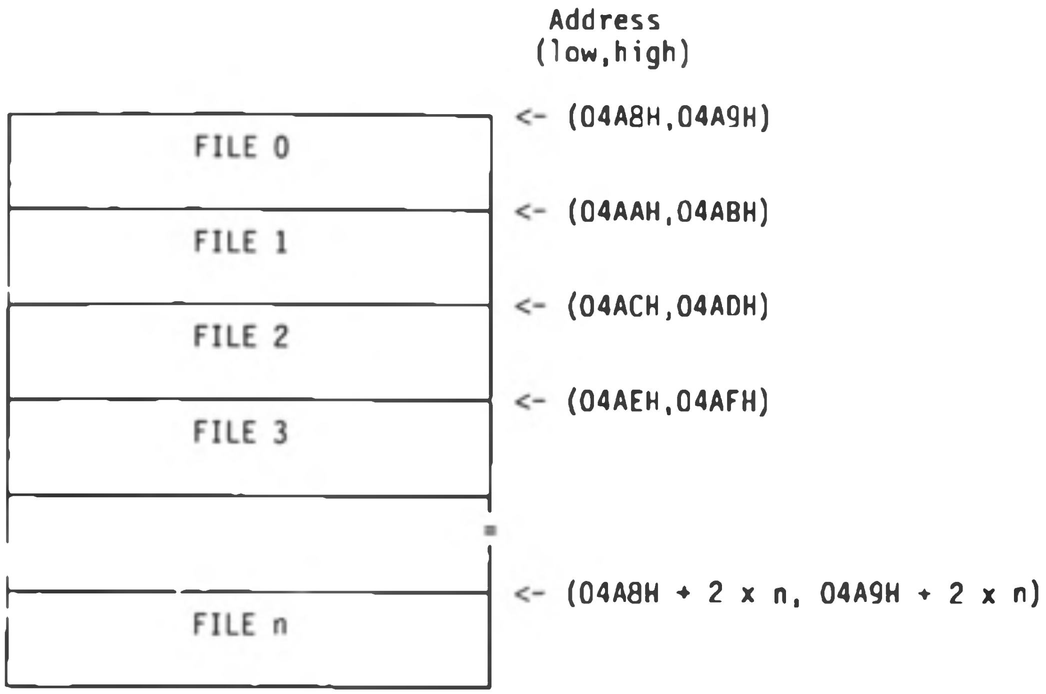


Fig 1.3 Disk Buffer Size

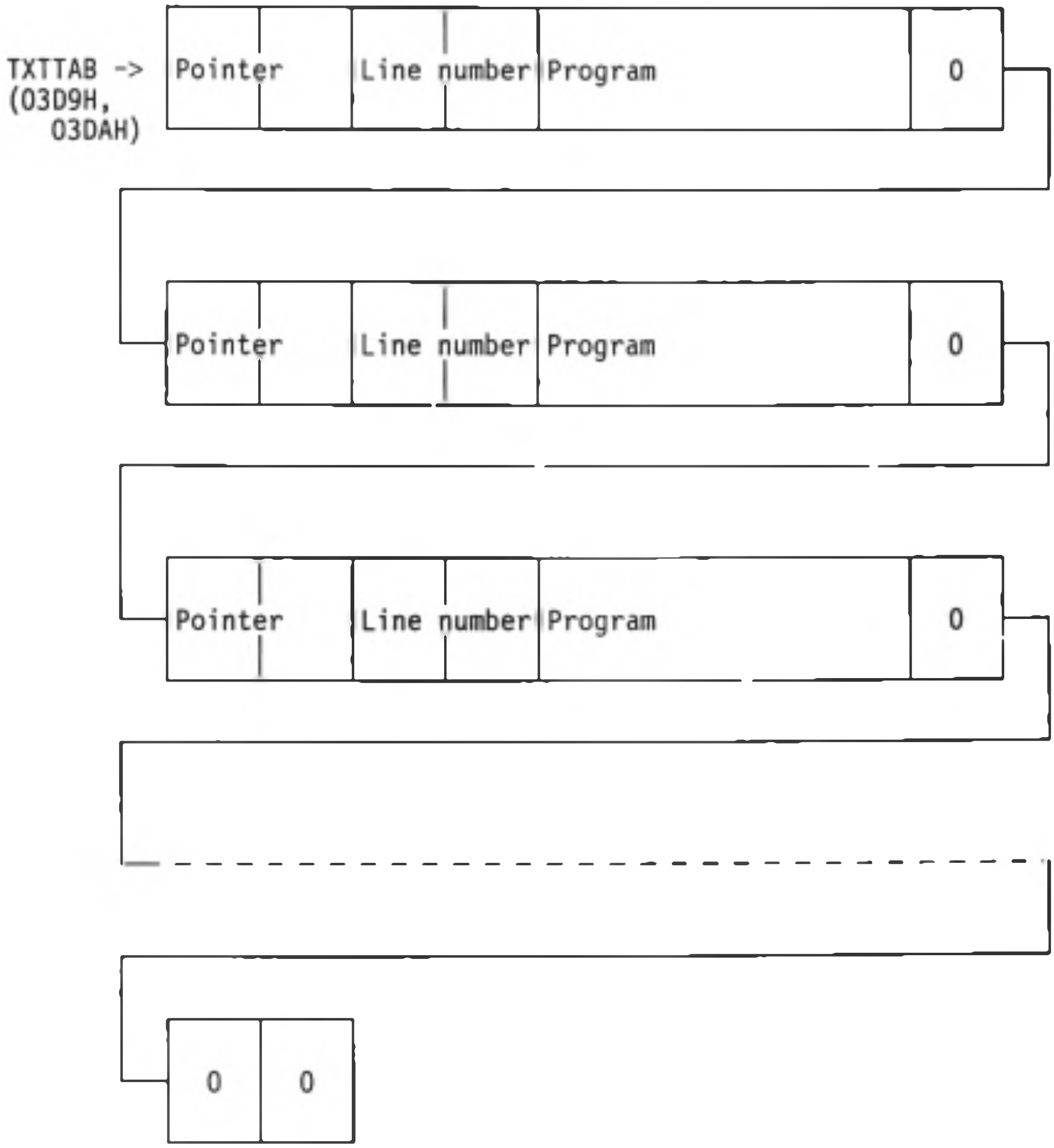
The file buffer for each file is used as shown below. A offset from the leading address of each file buffer is used. File 0 consists of 177 bytes from offset 0 to B0H. Other files consist of 186+n bytes. The VARPTR function points to the random buffer address at open in the "R" mode, otherwise at sequential buffer address.
The character variable specified in the Field statement points to an address in the random buffer.

Offset	No. of bytes	Contents
00H	01H	File mode = 0: Not open = 1: "I" mode = 2: "O" mode = 3: "R" mode = 4: "A" mode
01H	24H	FCB (CP/M format)
25H	02H	Offset to sequential buffer
27H	01H	Pointer for INPUT#
28H	01H	Number of characters or output digits remaining in the buffer
29H	03H	Unused
2CH	01H	Device number = FFH : KYBD: = FEH : SCRN: = FDH : LPT0: = FCH : COM0: = FBH : COM1: = FAH : COM2: = F9H : COM3: = F8H : BRCD: = F7H : DLL0:
2DH	01H	Maximum number of digits at output
2EH	01H	Unused
2FH	01H	Flag for INPUT#
30H	01H	Flag for PRINT#
31H	80H	Sequential file buffer
81H	02H	Record size
83H	02H	Current physical record number
85H	02H	Current logical record number
87H	01H	Flag
88H	02H	Number of output digits
8AH	n	Random file buffer n is the number of bytes specified by option BASIC_S.

Table 1.2 File Buffer

1.5 Text (Program)

The text (program) is chain structured as shown below. Pointers indicate the pointer address of the next line. Line numbers follow the pointers and text is entered after the line number. The end of the line is recognized by the pointer becoming 0,0 (2 bytes of 0).

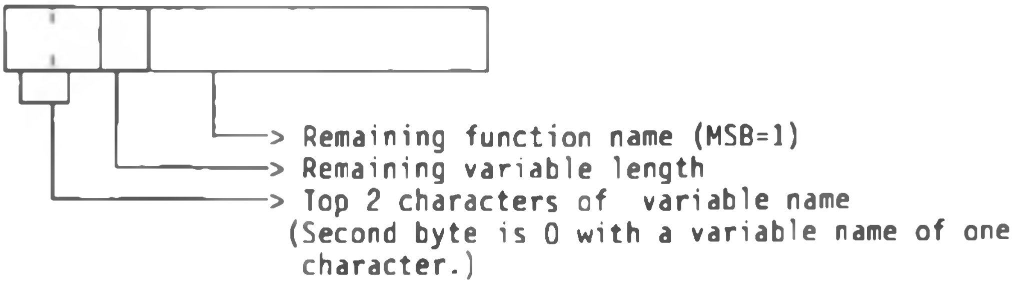


1.6 Simple variables

When simple variables in a program are used, they are indexed in the order of use and in the format corresponding to type. Variables are indexed in the following format. VARPTR function indicate the first byte of data.

Integer type	2	Variable name	Data 2 bytes
Single-precision type	4	Variable name	Data 4 bytes
Double-precision type	8	Variable name	Date 6 bytes
Character type	3	Variable name	String descriptor 3 bytes

* Variable name

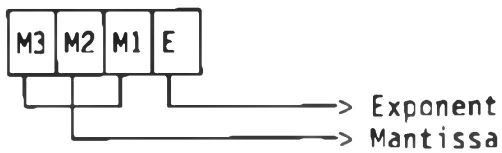


* String descriptor

* Data

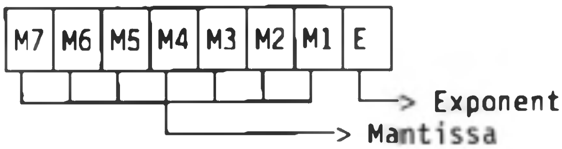
Integer type

Single-precision type



(M1 is high order, M3 is low order and M1 MSB is the sign.)

Double-precision type



(M1 is high order, M7 is low order and M1 MSB is the sign.)

1.7 Array Variables

When the DIM statement is executed in a program or the definition of an array is less than 10 elements, the array is indexed in the order of use. Array variables are indexed in the following format.

Type	Variable name	Size	Dimension

No. of arrays	----	No. of arrays	Element	----	Element

* Type ... Same as single variable (Value of 2, 3, 4 or 8)
* Variable name ... Same as single variable
* Size ... Memory capacity used after dimension (2 bytes)
* Dimension ... Array Number (1 byte)
* Number of arrays ... Single-dimension (dimension x 2 bytes)
* Element ... One element is 2, 3, 4 or 8 bytes according to variable type.

Address : Variable name (Byte length)

036FH : LPWAIT (1)
This variable specifies the wait time to which LPRINT and LLIST refer when printer is not ready. The default value is 30. ("DT error" will be displayed on the LCD after waiting 30 seconds.) Unit is second. But, system will wait eternally if you set this value to 0.

03F7H : RSWAIT (1)
This variable specifies the wait time when COM n: device has opened and control line check is ON. The default value is 30. ("DT error" will be displayed on the LCD 30 seconds after, if the control line of RS-232C have not changed to READY.) Unit is second. But, system will wait eternally if you set this value to 0.

03FAH : DEVNUM (2)
Pointer for extend sequential device. See Chapter 3.

044BH : DCBTAB (2)
Pointer for DCB of extend sequential device. See Chapter 3.

0995H : BLDADR (2)
Loading address for BLOAD.

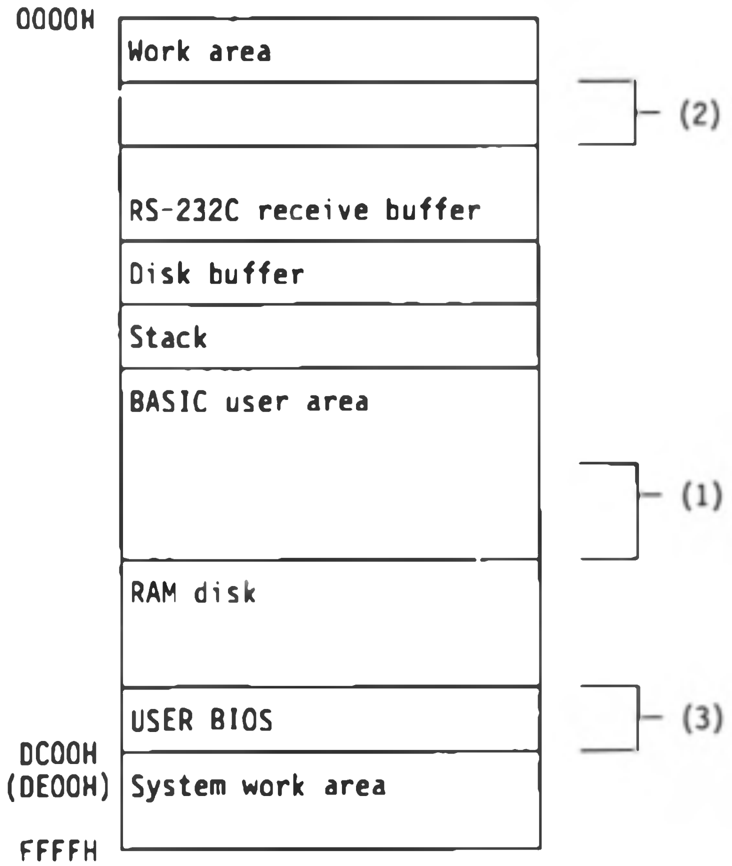
0997H : BLDLNG (2)
Loaded data length when BLOAD is executed.

CHAPTER 2 Command Expansion

2.1 Machine Language Hold

The following three ways to hold machine language are available.

(1) Reduce the BASIC user area by the CLEAR statement to hold an area which the BASIC interpreter does not access.
(2) Hold the area between the BASIC work area and RS-232C receive buffer, where the BASIC interpreter does not access, using a hook at the start of BASIC. (See 2.2 for further information.)
(3) Use the user BIOS area.



(Note) The word in parentheses is for EHT-10/2.

Fig 2.1 Machine Language Hold

2.2 Hook at BASIC Start

Call the hook after initialization of the work area at the start of BASIC. The default jump target of this hook is an address with the RET instruction. The user routine in the user BIOS area can be called by rewriting this to the address in the user BIOS area.

```
BASHOOK | C3H |     |     |----> Inside the User Bank (User BIOS Area)
(FFEAH)
```

(0000H to FFFFH) in the user bank can be called by BASHOOK, but only the user BIOS area can actually be used. Hooks controlled by BASIC can be rewritten in the routine called by BASHOOK. One hook, INIT, enables to move the location that follows the RS-232C receive buffer.

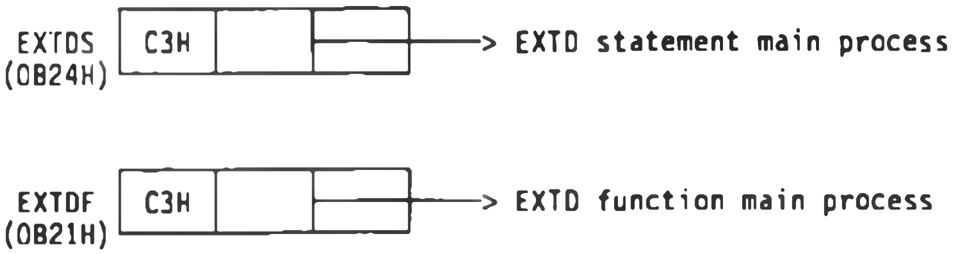
```
 INIT   | C3H |     |     |----> Inside the User Bank (00C05H - 0CCFH)
(0BE7H)
```

Input condition : ML - Leading address of RS-232C receive buffer (always 0D17H)
Output condition : HL - Same as above
(However, HL at output $\geq$ HL at input)

0000H to 5FFFH and E000H to FFFFH in the user bank can be called by INIT, (0C50H to 0CCFH) temporary area for hooks in BASIC work area is actually used. Therefore, load the necessary program into this area by BASHOOK. The area where the BASIC interpreter does not access can be held by changing the leading address of the HL register RS-232C receive buffer in the routine called by INIT.

2.3 Reserved Word EXTD

Reserved word EXTD is provided for command (statement, function) expansion. After EXTD is interpreted, the BASIC interpreter jumps to the EXTDS routine when used as a statement and jumps to the EXTDF routine when used as a function. Both EXTDS and EXTDF routines consist of 3 bytes and can write the JUMP instruction. Default is JMP FCERR (FC error).



EXTD syntax must be:
    EXTD p1, p2 ....
to use EXTD as a statement, parameters p1 and p2 ... are analyzed in the EXTD statement main unit.

EXTD syntax must be:
    V-EXTT (p1, p2 ....)
to use EXTD as a function, (p1, p2 ...) are analyzed in the EXTD function main unit.

Place the EXTD statement and function main unit between the BASIC WORK area and RS-232C receive buffer. Syntax analysis routine in the BASIC interpreter can be used while doing this. See 2.4 for Syntax Analysis Routine.

(1) SYNCHK
Address: 09ACH (or 000BH)
Input: HL = Text pointer
Output: Same as CHRGET
Description: This routine checks syntax.
SN error occurs if the character which the current text pointer indicates is different from the character to be checked. Let's assume the character to be checked is xx and call as follows.

```
     CALL  SYNCHK
     DB    xx
```

When the character the text pointer indicates is xx, advance the text pointer to the next character and return. As SYNCHK has entry at address 8, the following call is enabled.

```
     RST   08H
     DB    xx
```

SYNCHK is the same as the following routine.

```
     LD          A, (HL)
     EX          (SP),HL
     CP          (HL)
     JP          NZ,SNERR  ;give "SN Error"
     INC         HL
     EX          (SP),HL
     JP          CHRGET
```

(2) CHRGET
Address: 7007H (or 0010H)
Input: HL = Text pointer
Output: A = Character
HL = Text pointer

Z flag = 1 ... When the end of the statement is reached.

Description : This routine returns the character before the one indicated by the text pointer to the A register. Space is skipped. When the text pointer reaches the end of the statement, Z flag = 1 is returned.

(3) CHROUT
Address : BE78H (or 0018H)
Input : A = character
Output : No output
Description : This routine outputs one character to the device currently selected. Output target is checked as follows.

```
                  ≠ 0
< PTRFIL >--------------> File
    |
    | = 0
    |             ≠ 0
< PRTFLG >--------------> Printer
    |
    | = 0
    |
 Screen
```

* FCB leading address is stored in PTRFIL when executing the PRINT# statement. As PTRFIL≠0 in CHROUT, the character is output to the file. PRTFIL=0 at the end of the PRINT# statement.
* PRTFLG≠0 when executing LPRINT statement. As PTRFIL=0 and PRTFLG≠0 in CHROUT, the character is output to the printer. PRTFLG=0 at the end of the LPRINT statement.
* Both PTRFIL and PRTFLG are 0 when executing the PRINT statement. Thus, the character is output to the screen in CHROUT.

Address : Name (Number of bytes)

03D3H : PTRFIL (2)
FCB leading address

03CCH : PRTFLG (1)
Flag at printer output

(4) COMPAR
Address : C9A6H (or 0020H)
Input : HL, DE
Output : Z flag, CY flag

HL > DE --- Z=0,CY=0
HL = DE --- Z=1,CY=0
HL < DE --- Z=1,CY=1

Description : This routine compares the HL register with DE register. The A register is cleared.

(5) GETYPE
Address : 7A52H (or 0028H)
Input : No input
Output : Flag (Z flag, CY flag, P flag, S flag)

	Z	CY	P	S
Integer type	0	1	E	M
Character type	1	1	E	P
Single-precision type	0	1	O	P
Double-precision type	0	0	E	P

Description : This routine returns FAC type. Variable VALTYP has FAC type, the flag can acknowledge this.

VALTYP...2 = Integer type
(076BH) 3 = Character type
4 = Single-precision type
8 = Double-precision type

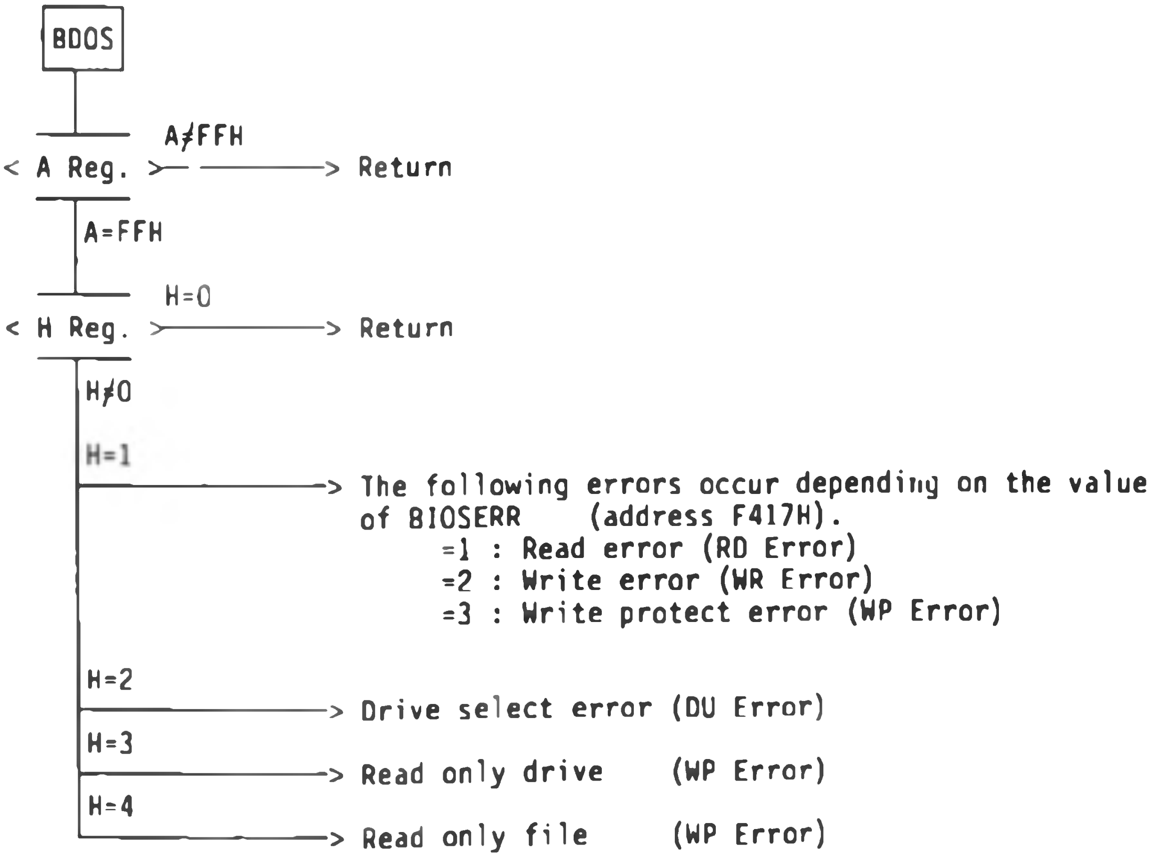
(6) GOBDOS

Address : A861H (or 0030H)
Input : Same as BDOS input parameter
Output : Same as BDOS output parameter

Description : This routine calls BDOS. Function number n BDOS is called as follows. The function number does not need to be set to the C register.

```
     CALL   GOBDOS
     DB     n
```

BASIC causes an error in the following sequence when BDOS reads an error.



(7) FRMEVL

Address : 76F1H
Input : HL = Text pointer
Output : HL = Text pointer
Description : This routine evaluates an expression, sets the value to FAC and sets the type to VALTYP.

(8) FRCINT
Address : 870AH
Input : No input
Output : HL = Integer value
Description : This routine converts FAC to an integer and returns it to the HL register.

(9) GETBYT
Address : 7D79H
Input : HL = Text pointer
Output : E = A = Value (0 to 255)
HL = Text pointer
Z flag = 1 When the end of the statement is reached.
Description : This routine evaluates an expression and returns the value to E(A) register if numeric. An error occurs if not numeric or greater than 256. GETBYT is the same as the following routine.

```
CALL    FRMEVL    ;evaluate formula
PUSH    HL        ;save text pointer
CALL    FRCINT    ;convert to integer
EX      DE,HL     ;integer to DE reg
POP     HL        ;restore text pointer
LD      A,D       ;get high order
OR      A,A       ;is it 0?
JP      NZ,FCERR  ;no, give "FC Error"
DEC     HL        ;back text pointer
CALL    CHRGET    ;set condition on terminator
LD      A,E       ;return the result in A and E
RET
```

(10) MAKINT
Address : B767H
Input : HL = Integer
Output : No output
Description : This routine sets an integer value to FAC.

(11) FRESTR
Address : CF1AH
Input : No input
Output : HL = String descriptor
Description : This routine puts the string descriptor to the HL register. "TM Error" occurs when not a character.

Ex.) This routine evaluates an expression, puts the character string length in A register and the character string address in DE register.

```
CALL    FRMEVL    ;Formula evaluate
PUSH    HL        ;Save text pointer
CALL    FRESTR    ;get string descriptor
LD      A,(HL)    ;A = string length
INC     HL
LD      E,(HL)
INC     HL
LD      D(HL)     ;DE = address
POP     HL        ;restore text pointer
RET
```

(12) ERROR
Address : 69E9H
Input : E = Error code
Description : Error handler for BASIC

A AUTO, AND, ABS, ATN, ASC, ATTR$, ALARM

B BEEP, BLOAD, BSAVE, BACKLIGHT , BRCD

C CLOSE, CONT, CLEAR, CINT, CSNG, CDBL, CVI, CVS, CVD, COS, CHR$, CALL, COMMON, CHAIN, CLS, COLOR, CIRCLE, COPY, CSRLIN, COM

D DELETE, DATA, DIM, DEFSTR, DEFINT, DEFSNG, DEFDBL, DEF, DAY , DATE, DSKF

E ELSE, END, ERASE, EDIT, ERROR , ERL, ERR, EXP, EOF, EQV, EXTD

F FOR, FIELD, FILES, FN, FRE, FIX, FONT

G GOTO, GOSUB, GET

H HEX$

I INPUT, IF, INSTR, INT, INP, IMP, INKEY$

J

K KILL, KEY, KANJI, KINPUT$

L LPRINT, LIST, LLIST, LPOS, LET, LINE, LOAD, LSET, LIST, LOCATE, LOG, LOC, LEN, LEFT$, LOF , LKANJI, LFONT, LOGIN

M MERGE, MOD, MKI$, MKS$, MKD$, MID$, MENU, MOUNT, MOTOR

N NEXT, NAME, NEW, NOT

O OPEN, OUT, ON, OR, OCT$, OPTION, OFF

P PRINT, PUT, POKE, POS, PEEK, PSET, PRESET, PAINT, POINT, POWER, PCOPY

Q

R RETURN, REND, RUN, RESTORE, REM, RESUME, RSET, RIGHT$, RND, RENUM, RESET, RANDOMIZE, REMOVE

S STOP, SWAP, SAVE, SPC(, STEP, SGN, SQR, SIN, STR$, STRING$, SPACE$, SYSTEM, SOUND, SCREEN , SET, STAT

T THEN, TRON, TROFF, TAB(, TO, TAN, TIME, TITLE, TAPCNT

U USING, USR

V VAL, VARPTR, VIEW

W WIDTH, WAIT, WHILE, WEND, WRITE, WINDOW, WIND

X XOR

Y

Z

2.5.1 Internal Code (Statement)

(HEX)	(HEX)	(HEX)	(HEX)
80 -	A0 -	C0 - FIELD	E0 - USR
81 - END	A1 - WIDTH	C1 - GET	E1 - FN
82 - FOR	A2 - ELSE	C2 - PUT	E2 - SPC(
83 - NEXT	A3 - TRON	C3 - CLOSE	E3 - NOT
84 - DATA	A4 - TROFF	C4 - LOAD	E4 - ERL
85 - INPUT	A5 - SWAP	C5 - MERGE	E5 - ERR
86 - DIM	A6 - ERASE	C6 - FILES	E6 - STRING$
87 - READ	A7 - EDIT	C7 - NAME	E7 - USING
88 - LET	A8 - ERROR	C8 - KILL	E8 - INSTR
89 - GOTO	A9 - RESUME	C9 - LSET	E9 - '
8A - RUN	AA - DELETE	CA - RSET	EA - VARPTR
8B - IF	AB - AUTO	CB - SAVE	EB - INKEY$
8C - RESTORE	AC - RENUM	CC - RESET	EC - OFF
8D - GOSUB	AD - DEFSTR	CD - CLS	ED -
8E - RETURN	AE - DEFINT	CE - LOCATE	EE -
8F - REM	AF - DEFSNG	CF - BEEP	EF - >
90 - STOP	B0 - DEFDBL	D0 - SOUND	F0 - =
91 - PRINT	B1 - LINE	D1 -	F1 - <
92 - CLEAR	B2 -	D2 - COLOR	F2 - +
93 - LIST	B3 -	D3 - PSET	F3 - -
94 - NEW	B4 - WHILE	D4 - PRESET	F4 - *
95 - ON	B5 - WEND	D5 - CIRCLE	F5 - /
96 -	B6 - CALL	D6 - PAINT	F6 - ^
97 - WAIT	B7 - WRITE	D7 -	F7 - AND
98 - DEF	B8 - COMMON	D8 -	F8 - OR
99 - POKE	B9 - CHAIN	D9 - COPY	F9 - XOR
9A - CONT	BA - OPTION	DA - KEY	FA - EQV
9B - BSAVE	BB - RANDOMIZE	DB - COM	FB - IMP
9C - BLOAD	BC -	DC - TO	FC - MOD
9D - OUT	BD - SYSTEM	DD - THEN	FD - ¥
9E - LPRINT	BE -	DE - TAB(	FE -
9F - LLIST	BF - OPEN	DF - STEP	FF - (function)

2.5.2 Internal Code (Function)

Function use a 2 byte code. The first byte is always FFH and the second byte and corresponding reserved words are shown below.

(HEX)	(HEX)	(HEX)	(HEX)
80 -	A0 -	C0 -	E0 - ALARM
81 - LEFT$	A1 -	C1 -	E1 - WIND
82 - RIGHT$	A2 -	C2 -	E2 - EXTD
83 - MID$	A3 -	C3 -	E3 - MOTOR
84 - SGN	A4 -	C4 -	E4 - KANJI
85 - INT	A5 -	C5 -	E5 - LKANJI
86 - ABS	A6 -	C6 -	E6 - FONT
87 - SQR	A7 -	C7 -	E7 - LFONT
88 - RND	A8 -	C8 -	E8 - VIEW
89 - SIN	A9 -	C9 -	E9 -
8A - LOG	AA -	CA -	EA - WINDOW
8B - EXP	AB - CVI	CB -	EB - SET
8C - COS	AC - CVS	CC -	EC - ATTR$
8D - TAN	AD - CVD	CD -	ED - KINPUT$
8E - ATN	AE -	CE -	EE - BACKLIGHT
8F - FRE	AF - EOF	CF -	FF - BRCD
90 - INP	B0 - LOC	D0 - CSRLIN	F0 -
91 - POS	B1 - LOF	D1 - POINT	F1 -
92 - LEN	B2 - MKI$	D2 - DAY	F2 -
93 - STR$	B3 - MKS$	D3 - DATE	F3 -
94 - VAL	B4 - MKD$	D4 - TIME	F4 -
95 - ASC	B5 -	D5 - SCREEN	F5 -
96 - CHR$	B6 -	D6 - DSKF	F6 -
97 - PEEK	B7 -	D7 - MENU	F7 -
98 - SPACE$	B8 -	D8 - LOGIN	F8 -
99 - OCT$	B9 -	D9 - TITLE	F9 -
9A - HEX$	BA -	DA - STAT	FA -
9B - LPOS	BB -	DB - PCOPY	FB -
9C - CINT	BC -	DC - MOUNT	FC -
9D - CSNG	BD -	DD - POWER	FD -
9E - CDBL	BE -	DE - REMOVE	FE -
9F - FIX	BF -	DF - TAPCNT	FF -

CHAPTER 3 Sequential Access Device Expansion

3.1 Overview

Data is input/output to a file based on the data block called DCB (Device Control Block) in devices which access sequentially. DCB is required in each device such as COM0. DCB must be set and the device name and number must be indexed at the same time to expand sequential access of the device. These can be performed by creating or adding a table. Required tables are as follows.

* Device table ... Device names and numbers are indexed.
* DCB table ...... DCB addresses for devices are stored in this table.
* DCB .... This table lists routine address which perform data I/O.

3.2 Device Table

This table indexes device names and numbers. The device name and number must be indexed to expand the device. BASIC supports 9 sequential devices and the device names and numbers are as shown below.

DEVNUM -> (03FAH)	09H				
DEVTAB ->	K	Y	B	D	FFH
	S	C	R	N	FEH
	L	P	T	0	FDH
	C	O	M	0	FCH
	C	O	M	1	FBH
	C	O	M	2	FAH
	C	O	M	3	F9H
	B	R	C	D	F8H
	D	L	L	0	F7H

Fig 3.1 Device Name and Device Number

DEVNuM indicates the number of devices indexed in the device table. The device table consists of 4 character device names and numbers par device, with a maximum of 16 devices to index.
Increase DEVNUM to expand the device and index the devices of device number F6H - in the device table.

3.3 DCB Table

This table stores the address of DCB. DCB address for each device must be stored to expand the device.

		Device No.
DCBTAB -> (044BH)	-> DCB address of KYBD	FFH
	-> DCB address of SCRN	FEH
	-> DCB address of LPTO	FDH
	-> DCB address of COMO	FCH
	-> DCB address of COM1	FBH
	-> DCB address of COM2	FAH
	-> DCB address of COM3	F9H
	-> DCB address of BRCD	F8H
	-> DCB address of DLLO	F7H

Fig 3.2 DCB Table

Indexing to the DCB table must be performed in the position corresponding to the device numbers indexed in the device table.

3.4 DCB (Device Control Block)

This table stores the entry address of routines such as each device open, close, data I/O. 10 entry addresses per device are required. DCB configuration and contents are as shown below.

Item	Offset	Size	Name	Contents
1	0,1	2	OPEN	Open
2	2,3	2	CLOSE	Close
3	4,5	2	OUTPUT	One character output
4	6,7	2	INPUT	One character input
5	8,9	2	LOC	For LOC function
6	10,11	2	LOF	For LOF function
7	12,13	2	EOF	For EOF function
8	14,15	2	PUT	Saves preread data
9	16,17	2	WIDTH	Maximum number of digits at output
10	18,19	2	RND	For GET# and PUT#

Table 3.1 DCB Contents

(1) OPEN

Function : Device open process

```
Input Condition : D = Device name
                  E = Open mode
                      (1="I", 2="O", 4="R", 8="A" mode)
                 HL = FCB leading address
               (SP) = File number
             (SP+2) = Text pointer
Output condition : HL = Text pointer
```

Procedure :

1. Check whether open mode is correct.
(If "O" mode is specified in the device only with input, an error occurs.)
If "R" mode is specified, the mode ("I" or "O" or both) which can open the device is assumed specified.

2. Actual open process must be performed for the device.

3. If open is performed without an error, PTRFIL and FCB must be initialized.

a. Set FCB leading address to PTRFIL (03D3H, 03D4H).
b. Initialize the area for FCB sequential device.

```
FCB + 0     Open mode
    + 28H   Initial value of digit position at output
    + 2DH   Maximum number of digits at output
(See WIDTH)
```

However, the digit position and maximum number of digits at output are meaningless in the "I" mode.

4. Remove the file number text pointer from the stack. Put the text pointer in HL.

* File name can be specified in the Open statement.

For example:

OPEN "I", #1, "DEVO:(ABC)"

is executed, character string of "(ABC)" enters

FILNAM+1 (11 bytes 0504H to 050EH)

and device number enters FILNAM (0503H).

(2) CLOSE

Function : Device close process
Input condition : (SP) = FCB leading address
(SP+2) = Text pointer
Output condition : HL = Text pointer

Procedure :

1. Perform close process for the device.

2. Remove the FCB address from the stack clearing the 49th byte to 0, from the address.

3. Set PTRFIL (address 03D3H, 03D4H) to 0.

4. Remove the text pointer from the stack, put it in HL and return.

(3) OUTPUT

Function : One character output
Input condition : (SP) = Output character
HL = FCB leading address
Output condition : None

Procedure :

1. Remove the character to be output from the stack and output to the device.

2. Remove AF, BC, DE and HL respectively from the stack and return.

(4) INPUT
Function : One character input
Input condition : HL = FCB leading address
Output condition : A = Input area
CY = 0 : When normally input.
1 : When there is no data for input (EOF) or forced to cancel input.

Procedure :

1. Check whether there is data saved by PUT before input from the device, if there is, return the value.

2. Input one character from the device.

(5) LOC
Function : LOC function
Input condition : None
Output condition : Set value to FAC.

(6) LOF
Function : LOF function
Input condition : None
Output condition : Set value to FAC.

(7) EOF
Function : EOF function
Input condition : None
Output condition : Set value to FAC.
0 = Not EOF
1 = EOF
MAKING routine is suggested for use to set the return parameter to FAC (Floating Accumulator) in LOC, LOF or EOF for convenience.

(8) PUT
Function : This DCB saves preread data.
Input condition : C = Data to be saved
Output condition : None

Description :
Data preread in the INPUT# statement is saved in PUT. 1 byte per device is required to save the data. This area holds 16 bytes (device number FFH to F0H respectively) from PUTBUF (0A9AH)
Access to PUTBUF is as follows.

* OPEN ..... Clears PUTBUF.
* PUT ...... Saves data to PUTBUF.
* INPUT .... Returns value of PUTBUF as data when PUTBUF is not 0.
0 must be set after PUTBUF read.

Offset	Address	Device number	Device name	Note
0	0A9AH	FFH	KYBD	
1	0A9BH	FEH	SCRN	Not used
2	0A9CH	FDH	LPT0	Not used
3	0A9DH	FCH	COM0	
4	0A9EH	FBH	COM1	
5	0A9FH	FAH	COM2	
⋮	⋮	⋮	⋮	

(Note) PUT is not required for a device with output only.

(9) WIDTH
Function : This DCB saves the maximum number of digits at output.
Input condition : C = Maximum number of digits
Output condition : None

Description :
This DCB is called at execution of WIDTH "device" statement. WIDTH saves the maximum number of digits (WIDTH "device", W of W) at output to enable use of this value in OPEN or OUTPUT. The user must hold the area to save the maximum number of digits. The maximum number of digits which is saved by WIDTH is accessed as follows.

* OPEN ..... Sets maximum number of digits to FCB.
* WIDTH .... Obtains maximum number of digits from WIDTH statement.
* OUTPUT ... Outputs CR and LF after the output of the number of characters in the statement with the maximum number of digits set in FCB. However, if the maximum number of digits is FFH, neither CR nor LF is output.

Default value must be set in advance as the maximum number of digits is used at OPEN.

(Note) WIDTH is not necessary for a device with input only.

(10) RND
Function : Block I/O
Input condition : (SP) = Text pointer
(SP+2) = 0 : GET#
Other than 0 : PUT#
HL = FCB leading address
Output condition : None

Description : The device which BASIC supports as standard does not perform block I/O (GET#, PUT#). (FC error occurs.) GET# and PUT# for sequential device are handled as follows.

GET# n,m...Input m characters in the file buffer of file number n.

PUT# n,m...Outputs m characters in the file buffer of file number n.

In GET# and PUT# statement, syntax analysis is only performed till file number (directly before ","), analysis must be performed in RND after ",".

PART 4 APPENDIX

1. CHARACTER CODE TABLE

(1) Character generator code table

Lower byte \ Upper byte	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	à	Æ	SP	0	@	P	`	p	▁	┴	SP	ー	タ	ミ	円	
1	°	Ø	!	1	A	Q	a	q	▂	┬	。	ア	チ	ム	年	
2	ç	Å	"	2	B	R	b	r	▃	┤	「	イ	ツ	メ	月	
3	§	æ	#	3	C	S	c	s	▄	├	」	ウ	テ	モ	日	
4	é	ø	$	4	D	T	d	t	▅	▔	、	エ	ト	ヤ		
5	ù	å	%	5	E	U	e	u	▆	─	・	オ	ナ	ユ		
6	è	É	&	6	F	V	f	v	▇	│	ヲ	カ	ニ	ヨ		
7	¯	â	'	7	G	W	g	w	█	▕	ァ	キ	ヌ	ラ		
8	Ā	¤	(	8	H	X	h	x	▏	┌	ィ	ク	ネ	リ		
9	Ö	ô	)	9	I	Y	i	y	▎	┐	ゥ	ケ	ノ	ル		
A	Ū	ı	*	:	J	Z	j	z	▍	└	ェ	コ	ハ	レ		
B	ā	Pt	+	;	K	[	k	{	▌	┘	ォ	サ	ヒ	ロ		
C	ō	¡	,	<	L	¥	l	¦	▋	╭	ャ	シ	フ	ワ		
D	ū	Ñ	-	=	M	]	m	}	▊	╮	ュ	ス	ヘ	ン		
E	ß	¿	.	>	N	^	n	~	▉	╰	ョ	セ	ホ	゛		
F	£	ñ	/	?	O	_	o	\	┼	╯	ッ	ソ	マ	゜		

The shapes of some character fonts displayed on the LCD screen are partially different from those printed by the printer. See the font table in (3) for details.

(2) Character Code Table

The basic character codes are as listed below. The character codes indicated by shaded portions depend on the countries as listed in the table on the next page.

Lower byte \ Upper byte	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0			SP	0		P		p	▁	┴						
1			!	1	A	Q	a	q	▂	┬						
2			"	2	B	R	b	r	▃	┤						
3				3	C	S	c	s	▄	├						
4				4	D	T	d	t	▅	▔						
5			%	5	E	U	e	u	▆	─						
6			&	6	F	V	f	v	▇	│						
7			'	7	G	W	g	w	█	▕						
8			(	8	H	X	h	x	▏	┌						
9			)	9	I	Y	i	y	▎	┐						
A			*	:	J	Z	j	z	▍	└						
B			+	;	K		k		▌	┘						
C			,	<	L		l		▋	╭						
D			-	=	M		m		▊	╮						
E			.	>	N		n		▉	╰						
F			/	?	O	_	o	SP	┼	╯						

	ASCII	France	Germany	England	Denmark	Sweden	Italy	Spain	Norway
23H	#	#	#	£	#	#	#	Pt	#
24H	$	$	$	$	$	¤	$	$	¤
40H	@	à	§	@	@	É	@	@	É
5BH	[	°	Ā	[	Æ	Ä	°	¡	Æ
5CH	\	Ç	Ō	\	Ø	Ō	\	N̄	Ø
5DH	]	§	Ū	]	Å	Å	é	¿	Å
5EH	^	^	^	^	^	Ū	^	^	Ū
60H	`	`	`	`	`	é	ù	`	é
7BH	{	é	ā	{	æ	ā	à	¨	æ
7CH	¦	ù	ō	¦	ø	ū	ò	n̄	ø
7DH	}	è	ū	}	å	å	è	}	å
7EH	~	¨	ß	~	~	ū	ì	~	ū

Notes:

- The printer unit (cartridge printer) for EHT-10/EHT-10/2 does not support the Norwegian font. If Norway is selected, ASCII is assumed.

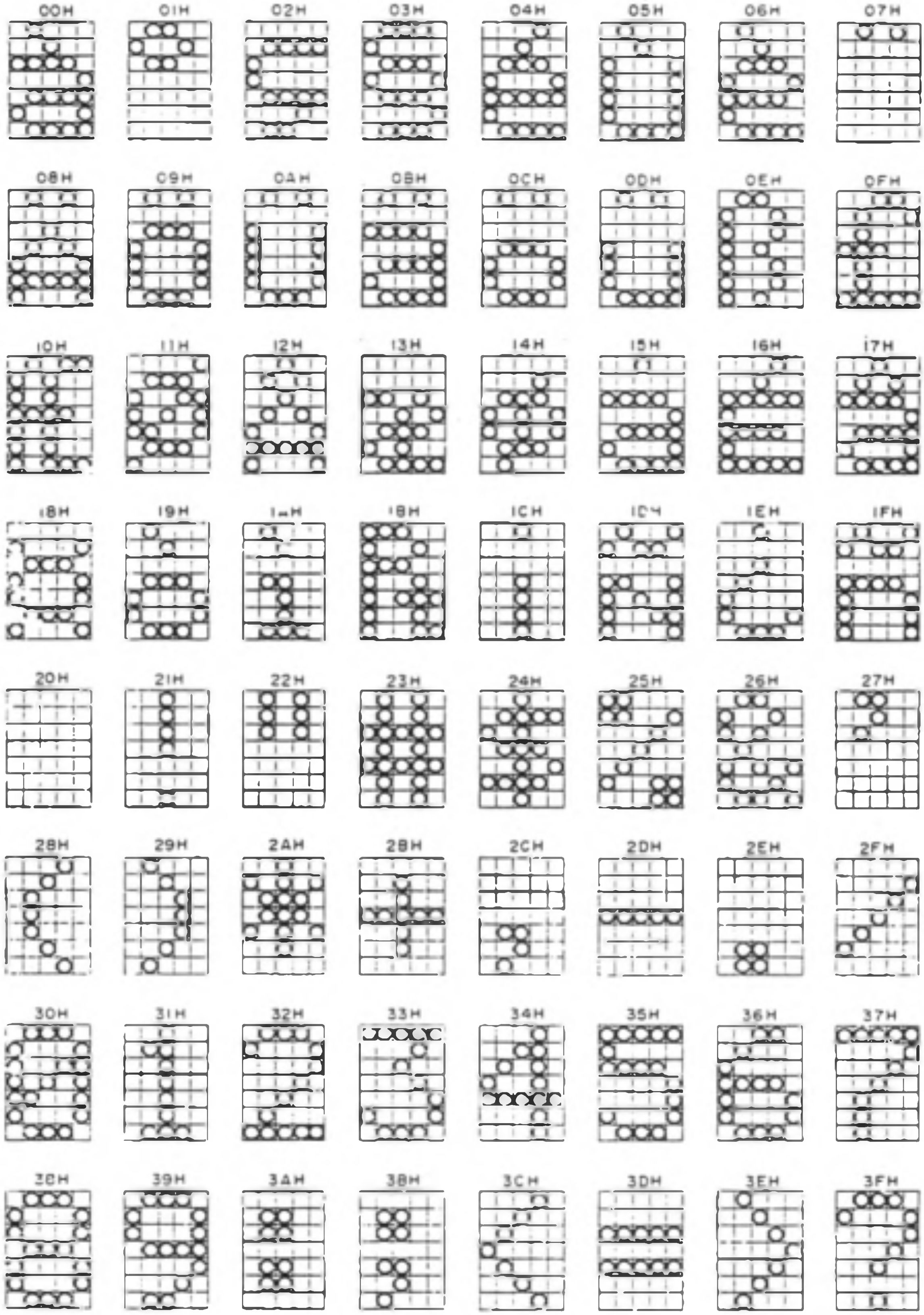
- The terminal printer connected to the system through an RS-232C interface can only normally print front characters of a specific country normally that it supports.

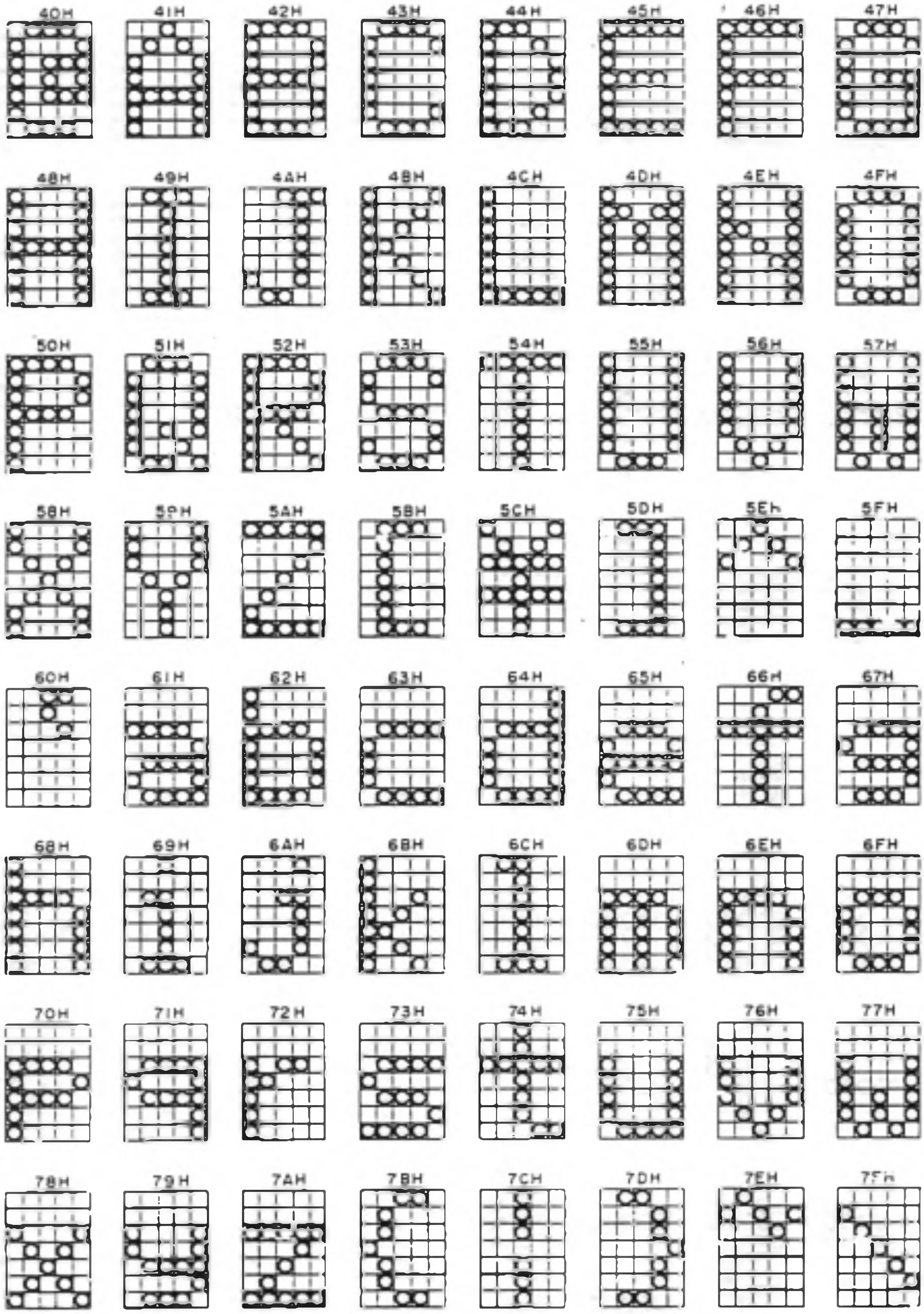
2. FONT TABLE

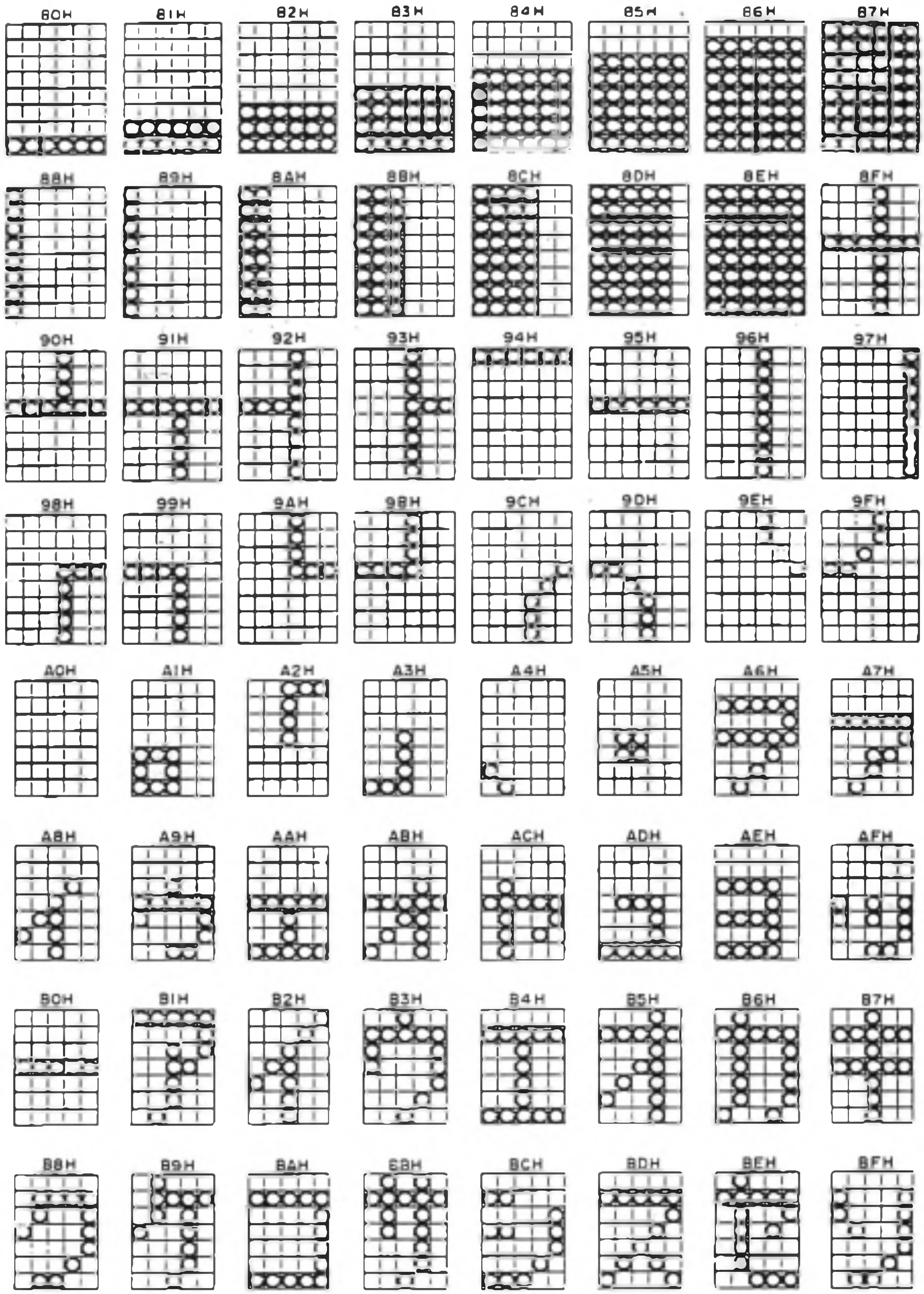
In EHT-10/EHT-10/2, the character and graphic patterns supported by the mainframe are the same as those supported by the printer as far as the CG codes are the same. However, the shapes of some character and graphic patterns displayed on the LCD screen are partially different from those printed by the printer. These character and graphic patterns are listed in the following table.

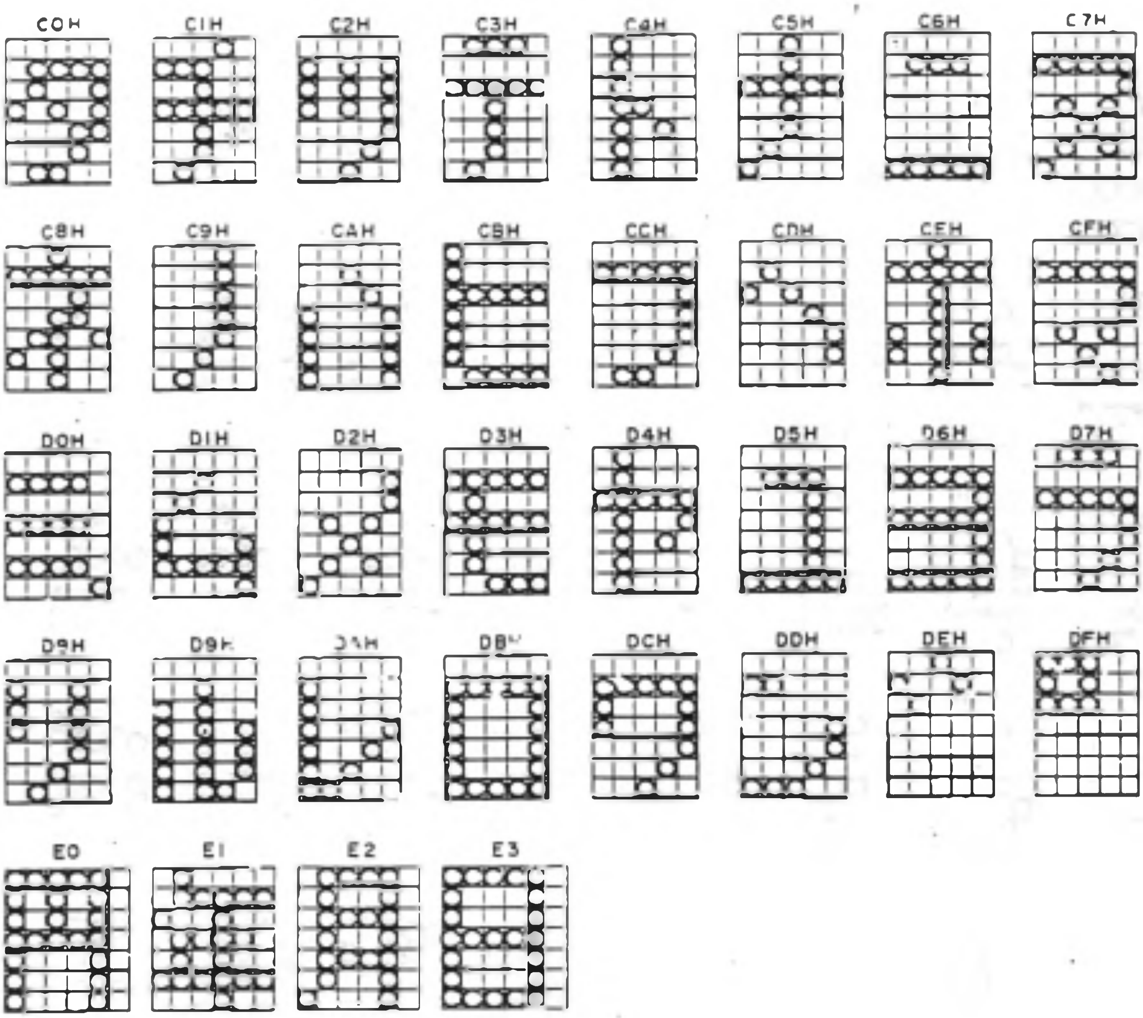
00H	a	12H	Å	68H	h	85H	█	8FH	┼	B2H	イ
02H	ç	15H	å	6DH	m	86H	█	91H	┬	C4H	ト
04H	e	1FH	ñ	6EH	n	87H	█	92H	┤	C8H	ネ
05H	ù	26H	&	73H	s	88H	│	93H	¦	CDH	ヘ
06H	è	2AH	*	75H	u	89H	│	96H	│	DAH	レ
08H	ä	37H	7	80H	▁	8AH	▌	97H	│	E0H	
0DH	û	40H	@	81H	▂	8BH	▌	98H	┌		
0EH	β	59H	¥	82H	▃	8CH	▌	99H	┐		
10H		61H	a	83H	▄	8DH	█	9CH	╭		
11H	Φ	65H	e	84H	▅	8EH	█	9DH	╮		

(1) EHT-10/EHT-10/2 font table



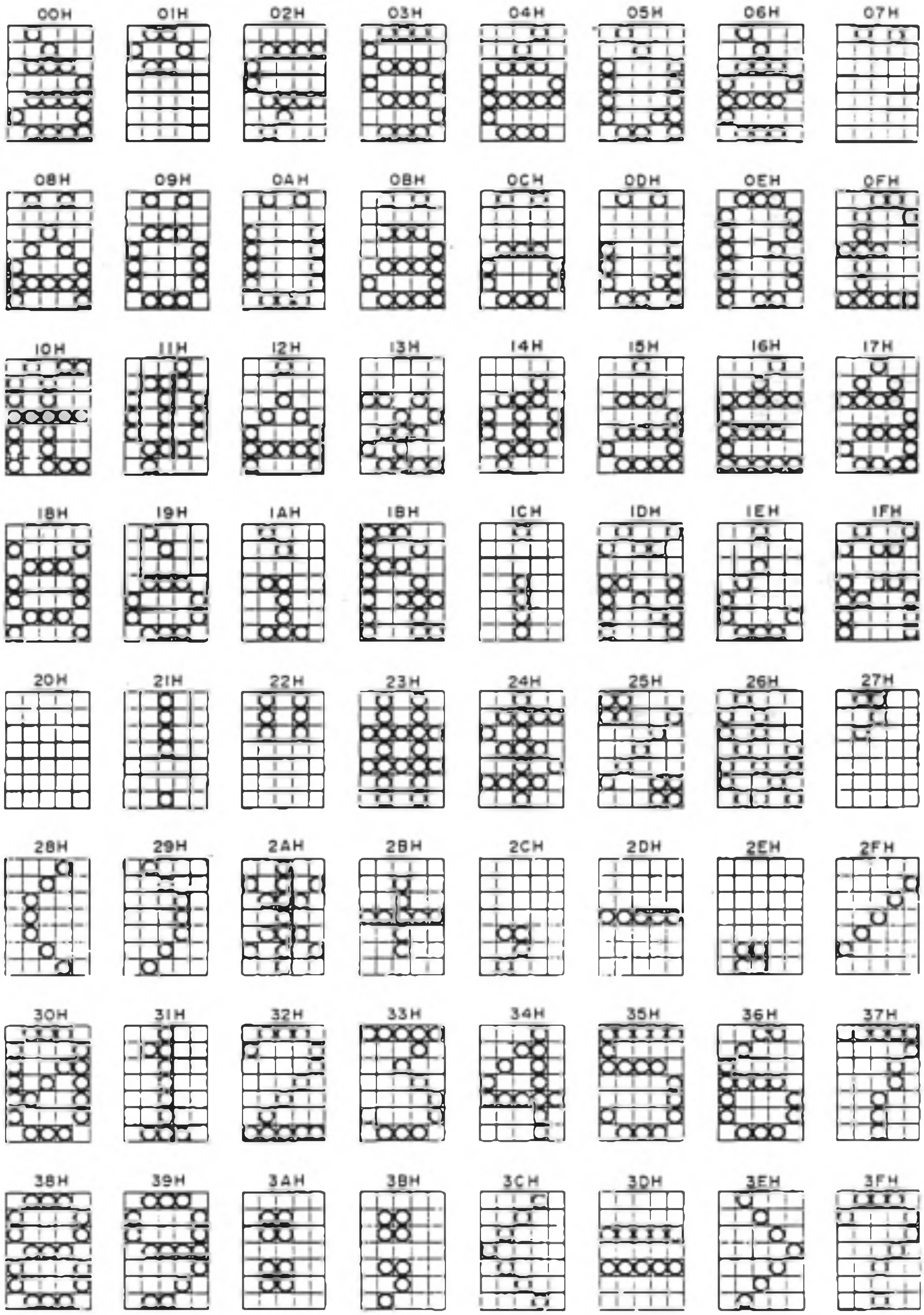

40H 41H 42H 43H 44H 45H 46H 47H
48H 49H 4AH 4BH 4CH 4DH 4EH 4FH
50H 51H 52H 53H 54H 55H 56H 57H
58H 59H 5AH 5BH 5CH 5DH 5EH 5FH
60H 61H 62H 63H 64H 65H 66H 67H
68H 69H 6AH 6BH 6CH 6DH 6EH 6FH
70H 71H 72H 73H 74H 75H 76H 77H
78H 79H 7AH 7BH 7CH 7DH 7EH 7FH

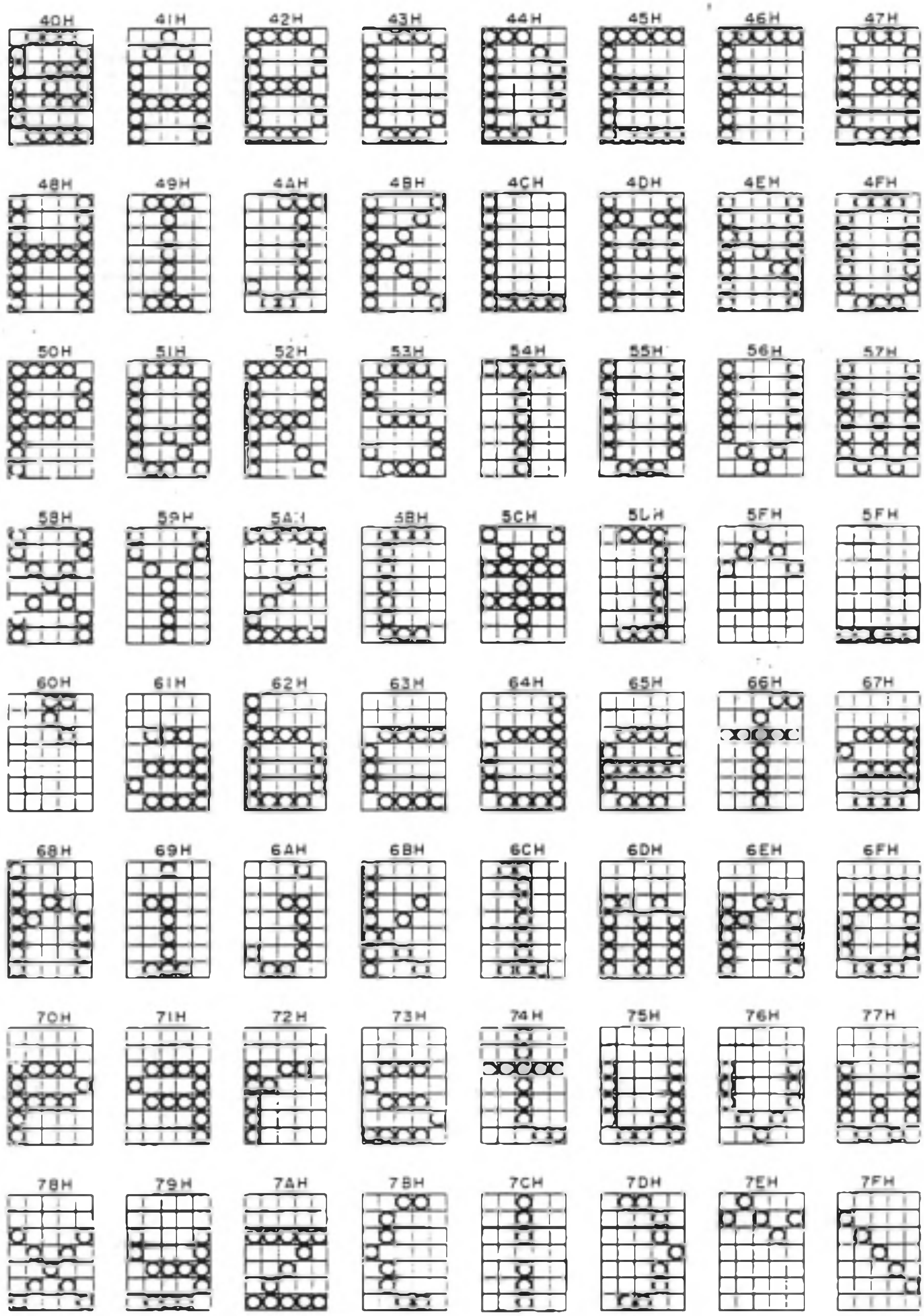

80H 81H 82H 83H 84H 85H 86H 87H
88H 89H 8AH 8BH 8CH 8DH 8EH 8FH
90H 91H 92H 93H 94H 95H 96H 97H
98H 99H 9AH 9BH 9CH 9DH 9EH 9FH
A0H A1H A2H A3H A4H A5H A6H A7H
A8H A9H AAH ABH ACH ADH AEH AFH
B0H B1H B2H B3H B4H B5H B6H B7H
B8H B9H BAH BBH BCH BDH BEH BFH

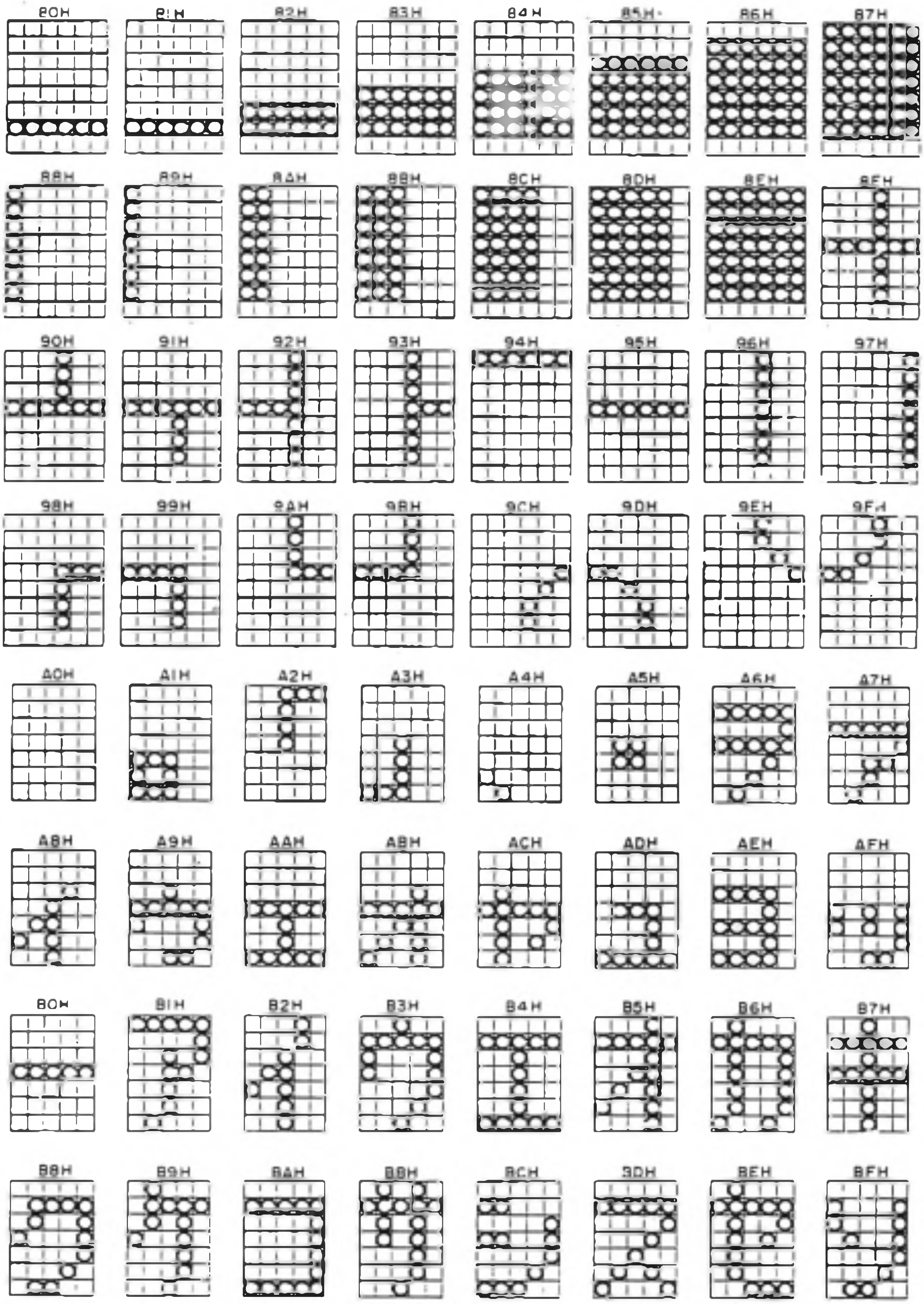

C0H C1H C2H C3H C4H C5H C6H C7H
C8H C9H CAH CBH CCH CDH CEH CFH
D0H D1H D2H D3H D4H D5H D6H D7H
D8H D9H DAH DBH DCH DDH DEH DFH
E0 E1 E2 E3

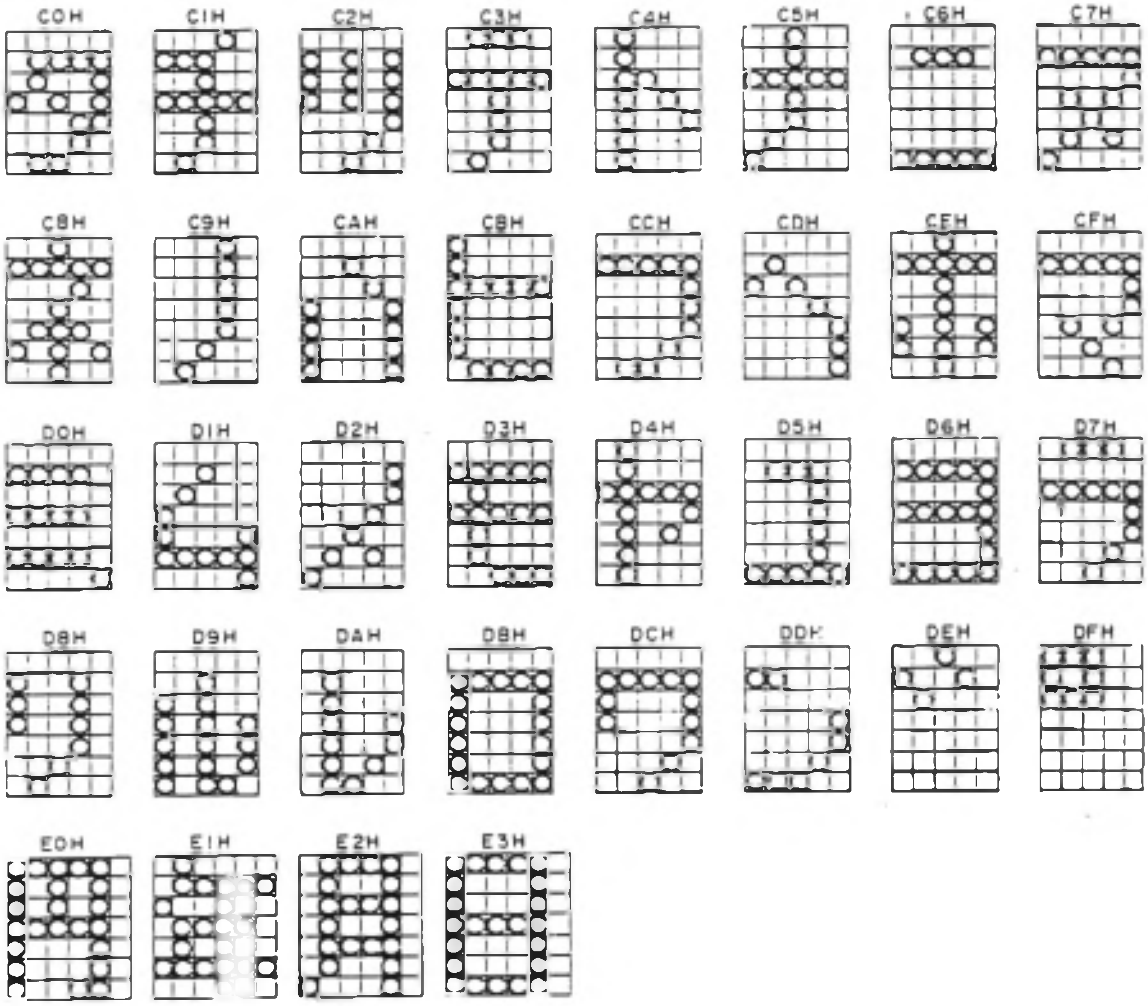
(2) Printer unit font table

Note: The patterns listed in this table correspond to the EHT-10/EHT-10/2 mainframe front pattern codes.




40H 41H 42H 43H 44H 45H 46H 47H
48H 49H 4AH 4BH 4CH 4DH 4EH 4FH
50H 51H 52H 53H 54H 55H 56H 57H
58H 59H 5AH 5BH 5CH 5DH 5EH 5FH
60H 61H 62H 63H 64H 65H 66H 67H
68H 69H 6AH 6BH 6CH 6DH 6EH 6FH
70H 71H 72H 73H 74H 75H 76H 77H
78H 79H 7AH 7BH 7CH 7DH 7EH 7FH


80H 81H 82H 83H 84H 85H 86H 87H
88H 89H 8AH 8BH 8CH 8DH 8EH 8FH
90H 91H 92H 93H 94H 95H 96H 97H
98H 99H 9AH 9BH 9CH 9DH 9EH 9FH
A0H A1H A2H A3H A4H A5H A6H A7H
A8H A9H AAH ABH ACH ADH AEH AFH
B0H B1H B2H B3H B4H B5H B6H B7H
B8H B9H BAH BBH BCH 3DH BEH BFH


C0H
C1H
C2H
C3H
C4H
C5H
C6H
C7H
C8H
C9H
CAH
CBH
CCH
CDH
CEH
CFH
D0H
D1H
D2H
D3H
D4H
D5H
D6H
D7H
D8H
D9H
DAH
DBH
DCH
DDH
DEH
DFH
E0H
E1H
E2H
E3H

3. *Version*

(1) CP/M Serial Number

The device and version number is located in the serial number (6 bytes) of BDOS CP/M.
The serial number is located in the leading 6 bytes of RBDOS1 and RBDOS2.
To refer to the serial number of RBDOS1, check the leading 6 bytes before the BDOS entry address located at address 0006H and 0007H. To refer to the serial number of RBDOS2, check the 6 byte area that follows address EA00H. The same value is stored at these two address areas.

Thc address which is pointed by 0006H and 0007H (RBDOS1) or EA00H (RBDOS2).

Offset	Value	
-->	DCH	Fixed data
+01H	16H	16H means CP/M Version 2.2
+02H	03H	02H means this machine is EHT-10/EHT-10/2
+03H	00H	
+04H	02H	
+05H	Version	---> Version=41H : Vers. 1.0 =42H : Vers. 2.0
	BDOS	

(2) ROMID

ROMID indicates the OS ROM state which is located in the 8 byte area that follows 7FF8H of OS ROM.
To check ROMID in an application, use BIOS LDIRX and read the data from OS ROM.
ROMID configuration is as shown below and is set by ASCII code.

7FF8H	'W'	-> Machine code ='WIS' : EHT-10/EHT-10/2
9H	'I'	
AH	'S'	
BH	'I'	-> Country code ='I' : International
CH	Prog. Vers.	-> All 3 bytes are ='1'(31H) : Vers. 1.0 ='2'(32H) : Vers. 2.0
DH	Data Vers.	
EH	Area Vers.	
FH	Reserved	

4. Table of System Work Area

<How to read the table>

Address : Indicates the address of each variable. Addresses are arranged from low to high order for easy location of the address contents when the address is known.

Variable name: Label name to refer to each variable in the system. When the variable is known and its address and contents is desired, refer to the list of labels at the end of this table. Only the first 8 letter are written for variables with more than 8 letters.

Number of bytes : Indicates variable size in bytes (decimal notation).

Type : Indicates kind of variables.

R/W : A user can rewrite, changing the state of the system as a result.

R/O : A user cannot rewrite, but reference enables to check the state of the system.

-: Used as temporary work of the system. A user should not rewrite.

Initial value : Indicates initial value of each variable. However, the system may rewrite some variables but they are meaningless. Variables without an initial value written are clarified in the description. (Some are not described.)

Reference : Indicates chapters in the Software Version which give further information of each variable. When parentheses are used, a description of the variables is not written but a related discussion is given by the indicated chapter.

(1) System Work Area I (RSYSAR1)
Variables in this area are initialized only when the system initialization process is executed.

Address : Variable (Number of bytes) : Initial value (Type) : Reference

F000H : RETADD (1) : C9H (R/O): 10.2 System Hook
RET instruction (C9H) is set and system hook initial value indicates this address.

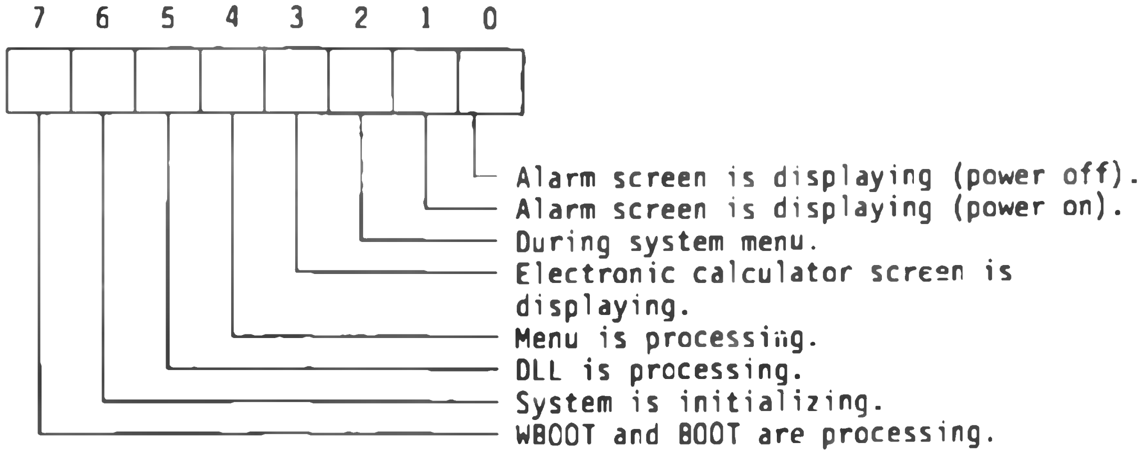
F001H - F004H : For future use.

F005H : BDSLAD (2) : 5E00H (EHT-10),6000H (EHT-10/2) (R/O) : (5.2) BDOS Process
Leading address of RBDOS1.
This address changes according to RAM disk and size of user BIOS area.

F007H : BI1LAD (2) : 5F00H (EHT-10),6100H (EHT-10/2) (R/O) : (4.1.2) BIOS Process
Leading address of RBIOS1.
This address changes according to RAM disk and size of user BIOS area.

F009H : For future use.

F00AH : MODEFG (1) : 00H (R/O) : (2.2) System State Transition
This flag indicates the system module currently executing in the system.



F00BH : SIZRAM (1) : 1FH (R/O) : (6.3) RAM Disk
Indicates size of standard RAM of the RAM disk, unit is 1 KB.

F00CH : USERBIOS (1) : 00H (R/O) : 4.6 User BIOS
Indicates size of user BIOS area, unit is 1 page (256 bytes).

F00DH : RXON (1) : 11H : 4.2 BIOS (RSIOX)
Code XON used in the XON/XOFF process when XON/XOFF is specified by BIOS RSIOX.

F00EH : RXOFF (1) : 11H (R/O) : 4.2 BIOS (RSIOX)
Code XOFF used in the XON/XOFF process when XON/XOFF is specified by BIOS RSIOX.

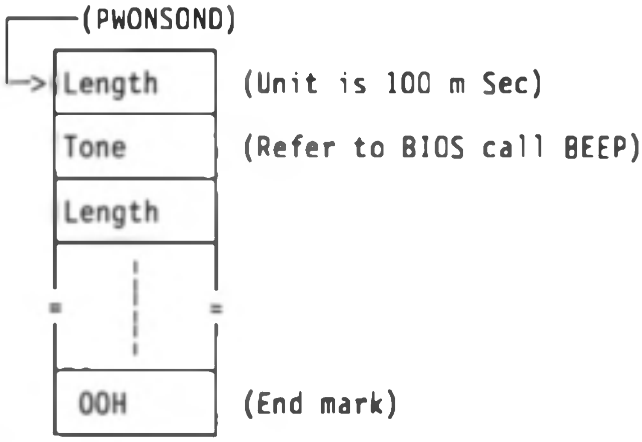
F00FH : For future use.

F010H : SRSADR (9) : See below (R/W) : 4.2 BIOS (PUNCH)
Open parameter used when RS-232C I/F is used with an I/O device such as PUN:, LST:, RDR:, CON:. The configuration is the same as BIOS RSIOX open parameter.

Address:	Contents	:Initial value	:Meaning
F010H	Receive buffer address	F95AH	
F012H	Receive buffer size	0100H	
F014H	Transmission rate	0DH	4800 BPS
F015H	Bit length	03H	8 BITS
F016H	Parity	00H	NON Parity
F017H	Stop bit length	03H	2 BITS
F018H	Special parameter	FFH	DTR/RTS Active

F019H : EXECTYPE (1) : 00H (R/W) :9.9.2 CONFIG Parameter Setting
Indicates starting method of an application program.
=00H : Automatic decision specification
=01H : Forced DLL specification
=02H : Forced execution specification

F01AH : PWONSOND (2) : 36B7H (R/W) : (10.3) System Jump Table
Indicates the data address of sound generation when the restart mode is on. Sound generation data must be after RAM address 8000 for a user to rewrite and the data configuration is the same as MELODY in the system jump table. The initial value indicates the data table of OS ROM, the contents are 01H, 37H, 00H.



Sound Generation Data Table

F01CH : CNTNSOND (2) : 36BAH (R/W) : (10.3) System Jump Table
Indicates the sound generation data address when the continue mode is on. Table configuration is the same as PWONSOND.
The initial value indicates the data table of OS ROM, the contents are 03H, 37H, 00H.

F01EH : ALRMSOND (2) : 36BDH (R/W) : (10.3) System Jump Table
Indicates the sound generation data address at alarm/wake. Table configuration is the same as PWONSOND. However, with alarm/wake, the sound generation data pattern repeats 3 times. The initial value indicates data table of OS ROM, the contents are 01H, 25H, 01H, 29H, 00H.

F020H : ATSHUTOFF (1) : 05H (R/W) : 2.3.5 Sleep Function and Auto Power Off Function.

Indicates the time of auto power off in minutes, auto power is not off when 00H. As this variable is not used to check the time of real auto power off, ATSOTIME (F021H, F022H) must be modified when the time is changed.

F021H : ATOSTIME (2) : 012CH (R/W) : 2.3.5 Sleep Function and Auto Power Off Function

Indicates the time of auto power off in seconds. The contents of ATSHUT OFF (F020H) must be multiplied by 60 to set.

F023H : ELOFTIME (2) : 00B4H (R/W) : 2.3.5 Sleep Function and Auto Power Off Function

Indicates the time of auto back light off in seconds. Auto back light does not turn off when 0000H.

F025H : PRNPWTN (1) : B4H (R/W) : 4.5.4 Printer Unit

Indicates the time of the power-down mode of the printer unit in seconds. When 00H is specified, power-down mode is disabled.

F026H : ICCPWTN (1) 3CH (R/W) : (11.3) IC Card Protocol Expansion

Indicates the time in seconds after IC card access until the current supplied to IC lead is cut. When 00H is set, current is supplied until close.

F027H : For future use.

F028H : ALRMTP (1) : 00H (R/W) : 4.2 BIOS (TIMDAT)

Indicates the status of alarm/wake.

=00H : Not set
=01H : Alarm set
=02H : Wake set

F029H : ALRMAD (2) : F32FH (R/W) : 4.2 BIOS (TIMDAT)

This pointer indicates the address of the alarm message or wake string. This must be set after RAM address 8000H for user modification.

F02BH : ALRMST (1) : 00H (R/O) : 4.2 BIOS (TIMDAT)

Indicates generation state of alarm/wake.

=00H : For future generation
=01H : Generated

This is set to 00H by alarm/wake of BIOS TIMDAT or read alarm/wake is set to 01H by alarm interrupt.

F02CH : ALRMFG (1) : 00H (-) : 4.2 BIOS (TIMDAT)

This flag checks whether the alarm interrupt is initial in the system. 7508 CPU returns the status of alarm generation at every second interrupt for 10 seconds after alarm interrupt is issued, the system ignores this by using this flag.

=00H : Alarm interrupt is not ignored.
=01H : Alarm interrupt is ignored.

F02DH : TIMERO (2) : 0000H (R/O) : 8.4 7508 Interrupt Process

This 16 bit up-counter is counted through a second interrupt and should not be rewritten by a user.

F02FH : TIMER1 (2) : 0000H (R/O) : 8.4 7508 Interrupt Process
This 16 bit down-counter is counted through a second interrupt and should not be rewritten by a user.

F031H : ISTS7508 (1) : 0AH (R/O) : 4.2 BIOS (MASKI)
This flag indicates the current state of interrupt disable/enable related to 7508.
See MASKI in "4.2 BIOS Function" for further information.

F032H : TIMER1M (2) : 0000H (R/O) : 8.7 EXT Interrupt Process
This 16 bit up-counter is counted by 1 msec/8 msec. The unit is always 1 msec, a user should not rewrite this counter.

F034H : BNKDTTBL (40) : See below (R/W) : (1.4) Memory Map
This table stores the bank data which is set to Bank Register (BANKR), Sub-bank Register (SUBBNKR). Refer to this table for bank switching in the system.

Address	Main bank data	Sub-bank data	Bank number
F034H,35H	02H	00H	System bank
36H,37H	02H	00H	System bank
38H,39H	02H	00H	System bank
3AH,3BH	02H	00H	System bank
3CH,3DH	42H	00H	Bank 0 # 0
3EH,3FH	42H	01H	Bank 0 # 1
40H,41H	42H	02H	Bank 0 # 2
42H,43H	42H	03H	Bank 0 # 3
44H,45H	A2H	00H	(Dummy)
46H,47H	A2H	10H	Bank 1 # 1
48H,49H	A2H	20H	Bank 1 # 2
4AH,4BH	A2H	30H	Bank 1 # 3
4CH,4DH	E2H	00H	Bank 2 # 0
4EH,4FH	E2H	40H	Bank 2 # 1
50H,51H	E2H	80H	Bank 2 # 2
52H,53H	E2H	C0H	Bank 2 # 3
54H,55H	42H	04H	Bank 0 # 4
56H,57H	42H	05H	Bank 0 # 5
58H,59H	42H	06H	Bank 0 # 6
5AH,5BH	42H	07H	(Dummy)

BNKDTBL Initial Value

F05CH : TOPRAM (2) : DC00H (EHT-10),DE00H (EHT-10/2) (R/O) :4.6 User BIOS
Indicates leading address of user BIOS area. Initial value is the value when user BIOS size is page 0.

F05EH : For future use.

F05FH : QT_ROM_CP1 (1) : 00H (R/O) : (6.4) ROM Socket
Indicates ROM capacity which is set to ROM socket (ROM disk) in 1KB units.

F060H : QT_RAM_IN (1) : 00H (R/O) : (4.6) User BIOS
Indicates RAM disk capacity in 1 KB units.

F061H : DR_ROM_CP1 (1) : 00H (R/O) : (6.4) ROM Socket
Indicates the number of ROM disk directories.

F062H : DR_RAM_IN (1) : 00H (R/O) : (6.3) RAM Disk
Indicates RAM disk directory area size in 128 B units.

F063H : AD_ROM_CP1 (2) : 0000H (R/O) : (6.4) ROM Socket
Indicates the leading address (header position) of ROM disk.

F065H : AD_RAM_IN (2) : 0000H (R/O) : (6.3) RAM Disk
Indicates the leading address of the RAM disk. When the expansion RAM card is used, this indicates the leading address of standard RAM. If the expansion RAM disk is not added, the address is 0000H.

F067H : CS_RAM_IN (1) : 00H (R/O) : (6.3) RAM Disk
Indicates RAM disk checksum area size in 128 B units.

F068H : RAMD_SIZE (1) : 00H (R/O) : (6.3) RAM Disk
Indicates the expansion RAM size in 32 KB units, which is used in the RAM disk.

F069H : RAM_SET (1) : 00H (R/O) : (1.4) Memory Map
Indicates the expansion RAM size in 32 KB units.

F06AH - F070H : This is used as a work disk for the read/write process.

F071H : QT_RAM_OLD (1) : 00H (R/O) : (4.6) User BIOS
Stores the previous size at RAM disk size modification, the unit is 1 KB.

F072H : AD_RAM_OLD (2) : 0000H (R/O) : (4.6) User BIOS
Indicates the leading address of the previous RAM disk at RAM disk size modification and user BIOS size modification.

F074H : DSPFLAG (1) : 00H (R/O) : 4.4.7 Work Area Detail Related to Display
Indicates the state of the LCD screen on/off.
=00H OFF
=80H ON

F075H : BLNKSTAT (1) : 00H (R/O) : 4.4.6 Cursor Display
This flag indicates whether blink is specified.
=00H Blink
=01H No blink

F076H : BLNKCNT (1) : 00H (-) : 4.4.6 Cursor Display
This work area is used in the system in order to measure the blink interval.

F077H : BLNKRVS (1) : 00H (R/O) : 4.4.6 Cursor Display
Indicates the current cursor display state
=00H : Displayed
=FFH : Not displayed

F078H : BLNKTIME (1) : 04H (R/W) : 4.4.6 Cursor Display
Specifies the cursor blink interval. The unit is 100 ms.

F079H : BTRYALM (1) : 00H (R/W) : (2.3.6) Power Failure
This flag indicates whether the alarm should be disabled while power is off after power failure.
=00H : Disabled
=01H : Enabled

F07AH : DSKTFLG (1) : 01H (R/W) : 9.2.2 CONFIG Parameter Setting
This flag indicates whether RAM disk checksum should be performed at power on.
=01H : Checksum
=00H : No checksum

F07BH : RAMTFLG (1) : 01H (R/W) : 9.2.2 ConFIG Parameter Setting
This flag indicates whether RAM checksum should be performed at power on.
=01H : Checksum
=00H : No checksum

F07CH : BASICF (1) : 03H (R/W) : 9.2.2 CONFIG Parameter Setting
Indicates the number of files which can be opened in BASIC at a time.

F07DH : BASICR (2) : 0080H (R/W) : 9.2.2 CONFIG Parameter Setting
Random record size of files handled in BASIC.

F07FH : BASICB (2) : 0100H (R/W) : 9.2.2 CONFIG Parameter Setting
Size of data send buffer handled in BASIC.

F081H : SYSINFLG (1) : FFH (R/W) : (2.3.1) System Initialize
This flag indicates system initialization has been processed.
=FFH : System initialization has been processed.
=00H : System initialization has not been processed.

F082H : SSYSMMOD (1) : 0FH (R/W) : (9.2.1) System Menu Control
This flag disables the system menu functions during the MENU and DLL processes and is copied to SYSMMOD (F6BBH) at WBOOT. See SYSMMOD for bit configuration.

F083H : USYSMMOD (1) : 01H (R/W) : 9.2.1 System Menu Control
This flag disables the system menu functions during application execution and is copied to SYSMMOD (F6BBH) at application start. See SYSMMOD for bit configuration.

F084H : SCONFMOD (2) : 07FFH (R/W) : 9.2.1 System Menu Control
This flag disables CONFIG functions during the Menu and DLL processes and is copied to CONFMOD (16BCH) at WBOOT. See CONFMOD for bit configuration.

F086H : UCONFMOD (2) : 07EFH (R/W) : 9.2.1 System Menu Control
This flag disables CONFIG functions during application execution and is copied to CONFMOD (F6BCH) at application start. See CONFMOD for bit configuration.

F088H - F08FH : For future use.

(2) System Work Area II (RSYSAR2)

Variables in this area are initialized when the system initialization or reset process is executed.

Address : Variable (Number of bytes) : Initial value (Type) : Reference

F090H : For future use.

F091H : INBIOS (1) : 00H (R/W) : (4.1.2) BIOS Process
This flag indicates whether the BIOS process is being executed.
=00H : BIOS not processing
=FFH : BIOS processing

F092H : INTLEVEL (1) : FFH (R/O) : 8.4 7508 Interrupt Process
This flag indicates 7508 interrupt level.
=FFH : Not 7508 interrupt processing
=00H : 7508 interrupt processing

F093H : INTTYPE (1) : 00H (R/W) : 8.9 Interrupt Process Expansion
This flag indicates interrupt generation state.

F094H : BUZ_FLG (1) : FFH (R/W) : 4.3 Key Input
This flag disables key input sound.
=00H : Disabled
=FFH : Enabled

F095H : KDFLTBZ (2) : 0064H (R/W) : 4.3 Key Input
Key input sound length (1 ms unit).

F097H : BUZZHZ (1) : 80H (R/W) : 4.3 Key Input
Key input sound frequency
=40H : 512 Hz
=80H : 1024 Hz
=C0H : 2048 Hz

F098H : INTVLFG (1) : 00H (R/W) : 8.8 EXT Interrupt Process
This flag indicates the module which uses the timer interrupt of 1 msec/8 msec.

F099H : INTS7508 (10) : See below (R/O) : (8) Interrupt Process
Interrupt stack area leading address table

Address	Variable	Value	Contents
FF99H,9AH	INTS7508	FE3FH	For 7508 interrupts
9BH,9CH	INTS8251	FDC7H	For ART interrupts
9DH,9EH	INTSICF	FD5FH	For ICF interrupts
9FH,A0H	INTSOVF	FD97H	For OVF interrupts
A1H,A2H	INTSEXT	FD2FH	For EXT interrupts

F0A3H : TBL7508 (16) : - (R/O) : (8.4) 7508 Interrupt Process
Interrupt type process distribution table at 7508 interrupt generation.

F0B3H : BTRYFG (1) : 00H (R/O) : (2.3.6) Power Failure
Power failure interrupt generation flag.
=00H : For future generation
=FFH : Generated

F0B4H : YPOFDS (1) : 00H (R/W) : 8.4 7508 Interrupt Process
This flag temporarily disables the power-off process. This flag is copied to YPOFST (F0B5H) at the power-off interrupt process.

F0B5H : YPOFST (1) 00H (R/W) : 8.4 7508 Interrupt Process
This flag indicates power-off interrupt generation state when the power-off process is disabled by YPOFDS. The value of YPOFDS (F0B4H) is copied at power-off interrupt with this flag.

F0B6H : YALMDS (1) : 00H (R/W) : 8.4 7508 Interrupt Process
This flag temporarily disables the alarm/wake process. The alarm interrupt process copies this flag to YALMST (F0B7H).

F0B7H : YALMST (1) : 00H (R/W) : 8.4 7508 Interrupt Process
This flag indicates the alarm interrupt generation state when the alarm/wake process is disabled by YALMDS (F0B6H). The alarm interrupt process copies the value of YALMDS of this flag.

F0B8H : SMENUFLG (1) : 00H (-) : (9.3) MENU Process Expansion
This flag indicates whether there is modification of the disk state or execution of the system menu functions during the MENU process.

F0B9H : MDRV (5) : See below (R/W) : 9.3 MENU Process Expansion
Specifies a maximum of 4 drives for the MENU process.
Initial values are as follows.

Address	Initial value	Meaning
F0B9H	01H	Drive A
BAH	02H	Drive B
BBH	03H	Drive C
BCH	FFH	End mark
BDH	FFH	End mark

F0BEH : MFTYP (10) : See below (R/W) : 9.3 MENU Process Expansion
Specifies a maximum of 3 file extension names for the MENU process.
Initial values are as follows.

Address	Initial value
F0BEH - C0H	"COM"
F0C1H - C3H	"BAS"
F0C4H - C6H	FFH,FFH,FFH
F0C7H	FFH

When not specified, end mark (FFH) is filled.

F0C8H - F0D5H : For future use.

F0D6H : RZCTLR1 (1) : 61H (R/W) : Hardware Part Chapter 2
Area to store the output state of control register 1 (CTLR1 :P00H), bit configuration is the same as CTLR1.

F0D7H : For future use.

F0D8H : RZCTLR2 (1) : 03H (R/W) : Hardware Part Chapter 2
Area to store the output state of control register 2 (CTLR2 :P02H), bit configuration is the same as CTLR2.

F0D9H : RZARTMR (1) : 84H (R/W) : Hardware Part Chapter 2
Area to store the output state of the ART mode register(ARTMR : P15H), bit configuration is the same as ARTMR.

F0DAH : RZARTCR (1) : 00H (R/W) : Hardware Part Chapter 2
Area to store the output state of ART command register (ARTCR: P16H), bit configuration is the same as ARTCR.

F0DBH : RZICCTLR (1) : C4H (R/W) : Hardware Part Chapter 2
Area to store the output state of IC card control register(ICCTLR : P17H), bit configuration is the same as ICCTLR.

F0DCH : RZSWR (1) : 08H (R/W) : Hardware Part Chapter 2
Area to store the output state of switch register (SWR :P18H), bit configuration is the same as SWR.

F0DDH : RZIOCTLR (1) : 06H (R/W) : Hardware Part Chapter 2
Area to store the output state of IO control register (IOCTLR : P19H), bit configuration is the same as IOCTLR.

F0DEH : RZCTLR3 (1) : 04H (R/W) : Hardware Part Chapter 2
Area to store the output state of control register 3 (CTLR3: P23H), bit configuration is the same as CTLR3.

F0DFH : USERCRG (1) : 00H (R/W) : 11.4 Cartridge Device Expansion
This flag indicates whether the mount check process hook(CRGHOOK) must be processed when a user expands the cartridge device.

=00H : No hook process
≠00H : Hook process

F0E0H : AHSTWRT (1) : 00H (R/O) : (6.6.4) Protocol
Indicates whether write pending data to FDD (drive D:, E:) resides.

=00H : No pending data
=31H : Pending data

This flag is set when write/read to/from FDD is performed.

F0E1H : ADSKOPN (1) : 00H (R/O) : (6.6.4) Protocol
This flag indicates the open state of FDD (drive D:, E:).

=00H : Open
=FFH : Not open

This flag is set when BIOS SELDSK is executed to FDD.

F0E2H : DSKDID (1) : 31H (R/O) : (6.6.4) Protocol
Indicates FDD device code in DID code of EPSP for FDD send/receive.

F0E3H : DSKSID (1) : 23H (R/O) : (6.6.4) Protocol
Indicates EHT-10/EHT-10/2 code in SID code of EPSP for FDD send/receive.

F0E4H : DSKTBL (5) : - (R/W) : 6.2 Logical Drive and Physical Drive
Logical drive and physical drive corresponding table.

F0E9H : DISKROV (2) : 0002H (R/W) : 6.2 Logical Drive and Physical Drive
Indicates disk read only status.

F0EBH : FTSTAB (10) : - (R/W) : (6) Disk System Description
Jump table for the initial select process of each drive at BIOS SELDSK.

F0F5H : READTAB (10) : - (R/W) : (6) Disk System Description
Jump table to the read process routine of each drive at BIOS READ.

F0FFH : WRTTAB (10) : - (R/W) : (6) Disk System Description
Jump table to the write process routine of each drive at BIOS WRITE.

F109H : DPBASE (80) : See below (R/O) : (6) Disk System Description
Disk parameter header corresponding to each drive has a 16 byte configuration and is sequentially set from drive A:.
Disk parameter header configuration is as shown below.

Relative address		
+0	Sector translate table address	Fix to 0000H to avoid skew
+2	Used as a scratch pad by BDOS	
+8	Directory buffer address	Fix to DIRBUF (F724H)
+10	Disk Parameter Block	
+12	Check Vector Address	
+14 +15	Allocation vector	

F159H : DPB0 (60) : - (R/O) : (6) Disk System Description
Disk parameter block corresponding to each drive is sequentially set from A: in a 15 byte configuration for each drive after DPB0.
However, drive D and E have the same configuration and use the same parameter block.

F195H : For future use.

F196H : RSTABL (15) : - (R/W) : 7.2 Serial Interface
This serial data send/receive parameter packet modifies the serial switch in the system.
Configuration is shown in 7.2.4.

F1A5H : EPTRCN (1) : 03H (R/W) : (6.6.4) Protocol
Specifies the number of retries at no response at EPSP data send/recei-ve.

F1A6H : EPTIMO (1) : 0AH (R/W) : (6.6.4) Protocol
Specifies the number of retries at time-over at 1 byte of EPSP data receive.

F1A7H : EPETMO (1) : 64H (R/W) : (6.6.4) Protocol
Specifies the number of retries at time-over at 1 block of EPSP data receive.

F1A8H : EPATMO (1) : 0AH (R/W) : (6.6.4) Protocol
Specifies the number of retries at time-over when ACK is received by EPSP protocol.

F1A9H : EPMODE (1) : 00H (R/W) : (6.6.4) Protocol
This flag specifies the mode at EPSP data send/receive.
=00H : Master mode (when header is sent)
≠00H : Slave mode (when header send is omitted)

F1AAH : For future use.

F1ABH : ET1BRTO (2) : 1C2FH (R/W) : (6.6.4) Protocol
Specifies the time until time-over at 1 byte of EPSP data receive.
Unit is approx. 13.86 micro-seconds, initial value is approx 100 msec and is time-over.

F1ADH : ICRSTCNT (1) : 09H (R/W) : 11.3 IC card Protocol Expansion
Specifies the number of receive bytes when the answer to reset.
= 00H : Ignore the answer to reset
≠ 00H : number of receive bytes.

F1AEH : ICRSTPNT (2) : F663H (R/W) : 11.3 IC card Protocol Expansion
This pointer indicates the data address for IC card reset response check.

F1B0H : IDFLTCNT (1) : 03H (R/W) : 11.3 IC card Protocol Expansion
Specifies the number of retries at IC card send/receive.

F1B1H : ICCDTIME (2) : 0003H (R/W) : 11.3 IC card Protocol Expansion
The timeout time when no response from the IC card (unit 1 sec).

F1B3H : ICCDPRM (5) : See below (R/W): 11.3 IC Load Protocol Expansion
Serial parameter packet for send/receive to/from IC card.
Initial value is 0EH,03H,03H,01H,FFH

F1B8H : RLCGENX (3) : See below (R/W) : 4.4.5 Character Generator
F1BBH : RLCGENN (3)
F1BEH : RLCGENG (3)
F1C1H : RLCGENK (3)
4 types of character generator table pointer data.
See 4.4.5 Character Generator for further information.

F1C4H : LTOUCHLD (3) : - (-) : (4.2) BIOS (TOUCH)
Indicates routine address for TOUCH redisplay (used only in the system).

Address	Meaning
F1C4H,C5H	Routine address of TOUCH redisplay.
C6H	For future use.

F1C7H - F1C8H : For future use

F1C9H : THSYSFLG (1) : 00H (R/W) : (4.2) BIOS (TOUCH)
This flag speeds up the BIOS TOUCH process.
=00H : Normal mode
=01H : Retrieves display data from the system bank.
(For use by a user, data must be retrieved after RAM address 8000H.)
=FFH : Omits key definition besides the process mentioned above.-(01H)

F1CAH : ALMTIME : 32H (R/W) : (2.4) Alarm/Wake
Specifies alarm screen display time (unit : second).

F1CBH : For future use

F1CCH : PRTINIT (1) : C0H (R/W) : (4.5.3) Destination Process
Specifies whether the initial printer process corresponding to each country is performed.
bit 7 RS-232C
bit 6 Printer unit
= 1: Initialization
= 0: No initialization

F1CDH : TCAMPRM (7) : - (R/W) : 4.2 BIOS (TCAM)
Start condition is stored at BIOS TCAM open. This area includes protocol, communication parameter ... etc.
See BIOS TCAM for details.

F1D4H : TDFLTCNT (1) : 03H (R/W) : 4.2 BIOS (TCAM)
Specifies the number of retries when Filink protocol is specified at BIOS TCAM and the host protocol does not coincide.

F1D5H : T1STTIME (2) : 001EH (R/W) : 4.2 BIOS (TCAM)
Specifies timeout time until initial data is received at BIOS TCAM (unit : second).

F1D7H : T2NDTIME (2) : 0003H (R/W) : 4.2 BIOS (TCAM)
Specifies timeout time until subsequent data data is received at BIOS TCAM (unit : second).

F1D9H : T1STFLG (1) : 00H (R/O) : 4.2 BIOS (TCAM)
This flag identifies BIOS TCAM "initial time", "subsequent time".
=00H : Initial time
=01H : Subsequent time
=FFH : For future receive

F1DAH : For future use.

F1DBH : DRCVCNT (2) : 0080H (R/W) : 11.2 Communication Protocol Expansion
Indicates area size for data storage at DLL, DL/UL.

F1DDH : DRCVBUF (2) : F85AH (R/W) : 11.2 Communication Protocol Expansion
Indicates the area of the leading address for data storage at DLL, DL/UL.

F1DFH : DSKNUM (1) : 01H (R/W) : 9.2 System Menu
Disk numbers for DL/UL, (DLL).

F1E0H : RNDFIG (1) : FFH (R/W) : 9.2.2 CONFIG Parameter Setting
Specifies the display mode after the decimal point of the electronic calculator.
=FFH : Float mode
≠FFH : Number of display columns after the decimalpoint.

F1E1H : For future use

F1E2H : ICSRACD (1) : 3AH (-) : 11.3 IC card Protocol Expansion
SRA code (the 1st byte during command transmission)

F1E3H : ICFLCLS (1) : 00H (-) : 11.3 IC card Protocol Expansion
File class code. This is the class code when using file-related commands.

F1E4H : ICCLASS (1) : 20H (-) : 11.3 IC card Protocol Expansion
System class code.

F1E5H : ICRECSZ (1) : 40H (-) : 11.3 IC card Protocol Expansion
Specifies the number of bytes of 1 record.

F1E6H : ICHOKDT1 (1) : 00H (R/W) : 11.3 IC card Protocol Expansion
Specifies whether execute the hook processing or not.

```
     Bit = 1 : Jump to the hook.
         = 0 : Do not jump (Default)

 7   6   5   4   3   2   1   0
+---+---+---+---+---+---+---+---+
|   |   |   |   |   | 0 | 0 | 0 |
+---+---+---+---+---+---+---+---+
  |   |   |   |   |
  |   |   |   |   +-------------> ICDHK5 (1 record read)
  |   |   |   +-----------------> ICDHK6 (1 record write)
  |   |   +---------------------> ICDHK7 (IC card mount check)
  |   +-------------------------> ICDHK8 (IC card format)
  +-----------------------------> ICDHK1 (Answer to reset)
```

F1E7H : ICWAIT1 (1) : 32H (R/W) : 11.3 IC card Protocol Expansion
Specifies the interval (in ms units) for awaiting the start of Reset response reception after the Reset status is canceled. Default value = 50 ms(32H)

F1E8H : ICWAIT2 (1) : 64H (R/W) : 11.3 IC card Protocol Expansion
Specifies the interval (in ms units) for awaiting the start of command transmission after the Reset response is received. Default value = 100 ms(64H)

F1E9H : ICTSDT (1) : 3BH (R/W) : 11.3 IC card Protocol Expansion
TS byte collation data at IC card reset response. The initial value (3BH) is LSB First, even parity, Z=1. Collate this data and receive data at reset response, if same, move to the receive process after T0 byte. If different, an error occurs.

F1EAH - F1FFH : For future use.

(3) System work Area III (RSYSAR3)

Variables in this area are initialized when system initialization, reset, restart or the WBOOT process is executed.

Address : Variable (Number of bytes) : Initial value (Type) : Reference

F200H : CNTFG (1) : FFH (R/O) : (2.3) System Start and End
This flag indicates the restart/continue power-off mode. It is set at power off and is checked at power on.
=00H : Continue mode
=FFH : Restart mode

F201H : FRCECNTN (1) : FFH (R/W) : (2.3) System Start and End
Indicates the power-off mode during application execution.
=00H : Restart mode
=FFH : Continue mode

F202H : ICCPWCNT (1) : 3CH (R/O) : (11.3) IC Card Protocol Expansion
This timer counts time after IC card access till power supply stops automatically.

F203H : ICCPWFLG (1) : 00H (R/O) : (1.3) IC Card Protocol Expansion
This flag indicates the power supply state after IC card access.
=01H : ON
=00H : OFF
=FFH : Time for OFF (still ON)

F204H : ELCTFLG (1) : 00H (R/O) : 2.3.5 Sleep Function and Auto Power Off Function
This is used in the system as EL back light.

F205H : TMFUNC (1) : 00H (R/W) : (8.4) 7508 Interrupt Process
Indicates valid/invalid for the user 1 sec timer function.
=00H : Invalid
≠00H : Valid

F206H : TMFLAG (1) : 00H (R/W) : (8.4) 7508 Interrupt Process
This flag indicates whether specified time has elapsed at user 1 sec timer function specification.
=00H : Specified time not elapsed
=FFH : Specified time elapsed
This flag decrements TMSEC (F3C6H) by one using a 1 second interrupt, when it becomes 0000H, it is set to FFH.

F207H : ROMEXQ (1) : 00H (R/O) : (3.4) ROM Execution Type Program Creation
This flag is used in the system to start ROM execution type program.
=00H : Not ROM execution type
≠00H : ROM execution type

F208H : DBGFLG (1) : 00H (R/O) : (10.3) System Jump Table
This flag indicates the normal mode/development cartridge mode for the current cartridge mode.
=00H : Normal mode
=FFH : Development cartridge mode

F209H : COMPCOL (1) : 00H (-) : (5) BDOS
F20AH : STRCOL (1) : 00H (-)
F20BH : COLMN (1) : 00H (-)
This work area is used for character display by BDOS.

F20CH : LISTCP (1) : 00H (R/W) : (5) BDOS
This flag indicates whether LST: output is performed at CON: output by BDOS ^P(10H).
=00H : No LST: output (normal mode)
=01H : LST: output (^P mode)

F20DH : KPCHAR (1) : 00H (R/O) : (5) BDOS
CON: input buffer in BDOS
=00H : No input
≠00H : Input (key code)

F20EH : USRCODE (1) : 00H (R/O) : (5) BDOS
Current user number

F20FH : CURDSK (1) : 00H (R/O) : (5) BDOS
Current disk number

F210H : EFCB (1) : E5H (-) : (5) BDOS
This work area is used to search the empty directory section area at BDOS make file function execution.

F211H : RODSK (2) : 0000H (R/W) : (5) BDOS
This read only vector is controlled by BDOS.

```
 7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0
|0|0|0|0|0|0|0|0|0|0|0|E|D|C|B|A|
 |               |             |   =0 : R/W
 <----F212H----><----F211H---->    =1 : R/O
```

F213H : DLOG (2) : 0000H (R/O) : (5) BDOS
Indicates the disk in login.

```
 7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0
|0|0|0|0|0|0|0|0|0|0|0|E|D|C|B|A|
 |               |             |   =0 : Not log in
 <----F214H----><----F213H---->    =1 : Log in
```

F215H : DMAAD (2) : 0000H (R/O) : (5) BDOS
DMA address used in BDOS. (Normally set to SYSDMA F851H).

F217H : USRDMA (2) : 0000H (R/W) : (5) BDOS
Indicates user DMA address which is set by BDOS SETDMA function.

F219H : For future use

F21AH : KEYF (1) : 00H (R/O) : 4.3 Key Input
Indicates the input state of the console buffer (KEYD:F21BH).
=00H : No input data
=FFH : Input data

F21BH : KEYD (1) : 00H (R/O) : 4.3 Key Input
Console input buffer in which key function code is set.

F21CH : KEYS (1) : 00H (R/O) : 4.3 Key Input
Console input buffer in which key location code is set.

F21DH : YPFCMFLG (1) : 00H (R/W) : 4.3 Key Input
Function key check mode flag
=00H : Function E0 to FEH is executed.
=FFH : Function E0 to FEH is not executed and function code is returned.

F21EH : SKEYFLG (2) : 0000H (R/W) : 4.3 Key Input
This flag makes all functions invalid for function code F0H to FFH.

F220H : CSTOPFLG (1) : 00H (R/W) : 4.3 Key Input
I/O forced break flag by function code F6H.
=00H : Function code F6H key not pressed.
=01H : Function code F6H key pressed.

F221H : KB_FLG (1) : 00H (-) : (4.3) Key Input
This flag indicates the key input mode modification (normal, alphanumerics, kana, electronic calculator) in the key interrupt process.

=00H : No input mode modification
=FFH : Input mode modification

F222H : KALMMOD (1) : 00H (-) : (4.3) Key Input
This flag indicates conversion of Roman/Kana characters in the EHT-10/2 Kana mode.
=00H : conversion
=FFH : No conversion

F223H : SMKFLG (1) : 00H (R/O) : (4.3) Key Input
This flag indicates lowercase letter input in the EHT-10 Kana mode.
=00H : Normally kana input mode
=01H : Lowercase kana input mode

F224H : RM_SIINFLG (1) : 00H (-) : (4.3) Key Input
This flag indicates the number of the consonant inputs in the EHT-10/2 kana mode.

F225H : RM_HATSUFLG (1) : 00H (-) : (4.3) Key Input
Indicates syllabic nasal generation in the EHT-10/2 kana mode.
=00H : No syllabic nasal generated
=FFH : Syllabic nasal generated

F226H : RM_SOKUFLG (1) : 00H (-) : (4.3) Key Input
This flag indicates double consonant generation in the EHT-10/2 kana mode.
=00H : No double consonant generated
=FFH : Double consonant generated.

F227H : KMODE (2) : FB00H (R/O) : (4.3) Key Input
Indicates the current key input mode.

Address	Initial value	Meaning
F227H	00H	User/system =00H : User mode =01H : System mode
F228H	FBH	Key mode =FEH : Electronic calculator mode =FDH : Alphanumeric (alphabet) mode =FCH : Kana mode =FBH : Normal mode

F229H : YPFKCNT (1) : 00H (R/O) : 4.3 Key Input
Indicates the number of characters which are not returned to a user at character string input (electronic calculator or '000' key input).

F22AH : RM_FSIIN (1) : 00H (-) : (4.3) Key Input
Area to save consonant codes which is first input in the EHT-10/2 kana input mode.

F22BH : RM_BIONCD (1) : 00H (-) : (4.3) Key Input
Area to save vowel code in the EHT-10/2 kana input mode.

F22CH : RM_KEYPTR (1) : 00H (-) : (4.3) Key Input
Area to save consonant code in the EHT-10/2 kana input mode.

F22DH : PUT_KEYPTR (2) : F7E9H (R/O) : 4.3 Key Input
Key input buffer (KEY_BUFFER) put pointer.

F22FH : GET_KEYPTR (2) : F7E9H (R/O) : 4.3 Key Input
Key input buffer (KEY_BUFFER) get pointer.

F231H : PFKPTR (2) : F434H (R/W) : 4.3 Key Input
Indicates the leading address of the input character string at character string input.

F233H : KTABPTR (2) : - (R/W) : (4.3) Key Input
Indicates the head of the pointer table of the key table.

Address		Meaning
(KTABPTR)	+0	Normal table
	+2	Kana table
	+4	Alphanumeric table
	+6	Electronic calculator table

F235H : PTR_KEYTBL (2) : - (R/W) : (4.3) Key Input
Indicates the head of the key table in the current key input mode.

F237H : RM_BUFPTR (2) : - (R/O) : (4.3) Key Input
Indicates the head of the area of converted roman/kana character data storage in the EHT-10/2 kana input mode.

F239H : USERKTBL (2) : - (R/O) : (4.3)Key Input
Indicates the head of user normal key table.

F23BH : BPINTEBL (1) : 01H (R/W) : 4.2 BIOS (BEEP)
This flag disables interrupts during BEEP.

F23CH : INHCOPY (1) : 00H (R/O) : (4.2) BIOS (SCRNDUMP)

This flag indicates that screen dump is being executed in EHT-10/2.

=00H : Screen dump not executing

≠00H : Screen dump executing

F23DH : For future use.

F23EH : PRTFLG (1) : 00H (R/W) : 4.5 Printer

This flag indicates whether the destination process for the printer has been executed.

=00H : Future output

≠00H : Output

The destination process must be executed to the printer after I/O byte modification and display character set modification, even when PRTFLG ≠ 00H.

F23FH : DISBRK (1) : 00H (R/O) : (4.3) Key input

This flag makes the I/O process forced break invalid temporarily using function key (F6H).

=00H : I/O process forced break valid

≠00H : I/O process forced break invalid.

F240H : For future use.

F241H : SRSMODE (1) : FFH (R/W) : 7.2 Serial Interface

This flag indicates current serial use status.

=FFH : For future use

=00H : IC card (DISK) in use

=01H : Being used by a user

=02H : Being used by FDD

F242H : SCRLT3 (158) : - (-) : 4.4 LCD display

Work area related to display. See 4.4.7 Work Area Related to Display for further information.

F2E1H : MEMK (1) : 00H (R/O) : -

Indicates the index number of the electronic calculator memory data (radix is 10).

F2E2H : MEMO (1) : 00H (-) : -

Work area used for operation of electronic calculator memory.

F2E3H : MEMD (10) : - (R/O) : -

Electronic calculator memory data mantissa.

F2EDH - F2FFH : For future use.

(4) System Work Area IV (RSYSAR4)

Variables in this area are generally not initialized. However, some variables set an initial value in the system.

Address : Variable (Number of bytes) : Type : Reference

F300H : XUSERBIOE (3) : R/O : (4.6) User BIOS
Instruction to jump to the leading address in the user BIOS area is stored.

F303H : CNTNSP (2) : R/O : (2.3.4) Continue Mode
Area to save the current stack pointer at power off in the continue mode.

F305H : CNTNILVL (2) : R/O : (2.3.4) Continue Mode
Area to save the current 7508 interrupt level at power off in the continue mode.

F307H : YPWSWST (1) : R/O : (2.3) System Start and End
This area stores the current power switch state.
=00H : Power switch off
=01H : Power switch on

F308H : YMAINST (1) : R/O : (2.3) System Start and End
This flag checks the main CPU power supply state at address 0 start (power-on, reset, system initialization).
=00H : Address 0 started from power-off state.
=01H : Address 0 started from power-on state.
This flag is normally set to 01H, when power-off command is output to 7508 CPU, it becomes 00H.

F309H : ZSTARTFG (1) : R/O : (2.3) System Start and End
This flag distinguishes system address 0 start.
=00H : WBOOT
=01H : Power ON
=02H : Alarm (power off)
=03H : Wake (power off)
=05H : Reset
=06H : System initialize

F30AH - F30CH : For future use.

F30DH : EXESTRNG (34) : R/W : 9.2.2 CONFIG Parameter Setting
Area to set execute filename and start parameter at forced selection specification. First byte is the length of the character string.

F32FH : ALRMMSG (34) : R/W : (2.4) Alarm/Wake
Area for alarm message/wake string setting. First byte is the length of the character string.

F351H : STIMEBUF (24) : - : -
Work area used for time setting and time display in CONFIG.

F369H : TIMEWK (32) : - : -
Work area used for time setting and time display in CONFIG.

F389H : SRSTMODE (1) : R/W : 9.4 Process Expansion at Power On
Area to store the main power supply state (YMAINST : F308H) at address 0 start. The system uses this as the flag which indicates whether RAM check must be performed in the reset process. It is also used to check whether power is turned on/off in a user program.
=00H : Power is not turned on/off
≠00H : Power is turned on/off

F38AH : MENUMOD (45) : - : -
45 byte work area used in the MENU process.

F3B7H : ELOFFEND (2) : R/W : 2.3.5 Sleep Function and Auto Power Off\
Function
Time for auto back light off is set to this area by the auto back light off function.

F389H : BUZZLNG (2) : - : (4.3) Key Input
This counter counts key input sound length, the value of KDFLTBZ (F09511) is copied first.

F3BBH : RSINTST (1) : R/W : (8.5) ART Input
This flag indicates data has been received from serial I/F.
=00H : No data received
=01H : Data received

F3BCH : BCINTST (1) : R/W : (8.6) ICF Interrupt
This flag indicates bar code is read.
=00H : No bar code read
=01H : Bar code read

F3BDH : PRNPWCNT (1) : R/W : (4.5) Printer
This counter counts the time of the power-down mode in printer units. The value of PRNPWTM (F025H) is copied first and decrementation is performed by 1 second interrupt.

F3BEH : PRNPWFLG (1) : R/W : (4.5) Printer
This flag indicates printer unit power supply state.
=01H : Normal mode
=00H : Power-down mode
=FFH : Time for power-down (power-down has not been performed)

F3BFH : TBUZ_FLG (1) : - : (4.3) Key Input
This work area is used for key input sound or buzzing in the BIOS BEEP process.

F3C0H : ENTSINT (2) : R/O : (8) Interrupt process
Area to save the current stack in the interrupt process.

F3C2H : INTFG (1) : R/W : (8.4) 7508 Interrupt
This flag indicates the type processed in the 7508 interrupt process.

F3C3H : STS7508 (1) : R/O : (8.4) 7508 Interrupt
Area to store status read from 7508 by OS at 7508 interrupt generation.

F3C4H : FG7508 (1) : R/W : (8.4) 7508 Interrupt
Indicates the process execution status of 1 second interrupt and alarm interrupt.

7	6	5	4	3	2	1	0
-		-	-		-	-	-

→ Initial alarm interrupt (bit 3)
→ 1 second interrupt (bit 6)

(All bytes indicate the process is performed at 1.)

F3C5H : ALRMCT (1) : R/O : (8.4) 7508 Interrupt
This flag ignores alarm interrupt within 10 seconds after initial alarm interrupt generation.

F3C6H : TMSEC (2) : R/W : (8.4) 7508 Interrupt
This area specifies user timer execution time in seconds. Timer starts at TMFUNC (F205H).

F3C8H : PWSWONFLG (1) : R/W : (8.4) 7508 Interrupt
This flag indicates power switch on interrupt generation.
=00H : Not generated
=FFH : Generated

F3C9H : PWSWOFFG (1) : R/W : (8.4) 7508 Interrupt
This flag indicates power switch off interrupt generation.
=00H : Not generated
=01H : Generated

F3CAH : AFTER3 (1) : - : (8.4) 7508 Interrupt
Area used to decide time for alarm display end.

F3CBH - F3CCH : For future use.

F3CDH : BDOLT4 (66) : R/O : 5. BDOS
Used as a BDOS work area.

F40FH : USRSBD (2) : R/O : (5) BDOS
Area to save the user stack pointer in the resident BDOS (RBDOS) process.

F411H : USRFCB (2) : R/O : (5) BDOS
Area to store user specified FCB address in the resident BDOS (RBDOS) process.

F413H : BDOSFN (1) : R/O : (5) BDOS
Area to store user specified BDOS function number in the resident BDOS (RBDOS) process.

F414H : BDSBNK (1) : R/O : (5) BDOS
Area to store user BDOS call bank data in the resident BDOS (RBDOS) process.

F415H : RIOBYTE (1) : R/W : (2.6.5) I/O Byte
Area to store the contents of the current I/O byte (address 0003H) when the resident BDOS (RBDOS) and BIOS (RBIOS) are called. RIOBYTE configuration is the same as I/O byte.

F416H : ERRFLG (1) : R/O : (5.3) BDOS Error
This flag indicates the current BDOS error process mode.
=00H : RSTERR mode (normal mode)
=FFH : SETERR mode

F417H : BIOSERROR (1) : R/O : 4.2 BIOS (READ) (WRITE)
See 4.2 BIOS Function READ for BIOS error code configuration at disk read/write.

F418H : OLDBNK (1) : R/O : (4.1.2) BIOS Process
Area to store the user BIOS call bank data in the resident BIOS (RBIOS) process.

F419H : SVOLDBNK (1) : - : -
Area to save OLDBNK contents in the system.

F41AH : For future use.

F41BH : DISBNK (1) : R/W : 4.2 BIOS (CALLX) (JUMPX) (LDIRX)
This bank data indicates the destination of BIOS CALLX, destination of JUMPX, jump target and LDIRX data transfer target. (This is user set.)

F41CH : CALBNK (1) : - : (4.2) BIOS (CALLX)
Area to store bank data returned at BIOS CALLX.

F41DH : SRCBNK (1) : R/W : 4.2 BIOS (LDIRX)
Area to store bank data which has transferring source data at BIOS LDIRX. (This is user set.)

F41EH : RETBNK (1) : - : -
Area to store bank data returned after process at BIOS LDIRX.

F41FH : SBNKDT (2) : 2 : -
This data changed the transferring source bank data to I/O register output at BIOS LDIRX.

F421H : DBNKDT (2) : - : -
This data changed the transferring source bank data to I/O register output at BIOS LDIRX.

F423H : CURBNK (1) : R/O : (1.4) Memory Map
Area to store current bank data.

F424H : USRSBI (2) : R/O : (4.1.2) BIOS Process
Area to save the user stack pointer when BIOS call is performed in the resident BIOS (RBIOS) process.

F426H : BIOSFN (2) : R/O : (4.1.2) BIOS Process
Area to store called BIOS function numbers (EBxxH) in resident BIOS (RBIOS) process.

F428H : ROMEXQON (1) : - : (3.4) ROM Execute Type Program Creation
This flag is for ROM execute decision when ROM execute type program starts in the system.
=00H : ROM executed
=01H : ROM not executed

F429H : ROMEXBNK (1) : - : (3.4) ROM Execute Type Program Creation
Bank data when moving control of ROM execute type program.

F42AH : ROMEXADD (2) : - : (3.4) ROM Execute Type Program Creation
Address when moving control to ROM execute type program.

F42CH : RZBANKR (1) : R/W : Hardware 2. IO Register Description
Area to store the output state of bank register (P05H). The configuration is the same as BANKR (P05H).

F42DH : RZSBBNKR (1) : R/W : Hardware 2. IO Register Description
Area to store the output state of Sub-bank Register (P04H). The configuration is the same as SBBANKR (P22H).

F42EH : RZIER (1) : R/W : Hardware 2. IO Description
Area to store the output state of interrupt enable register (P04H). The configuration is the same as IER (P04H).

F42FH : CRGDEV (1) : R/W : 11.4 Cartridge Device Expansion
Area to store the device number of the optional unit connected to cartridge I/F. It is set to 00H when not installed.

F430H : SAVEIX (2) : R/O : (4.1.2) BIOS Process
Area to save IX register when BIOS is called.

F432H : SAVEIY (2) : R/O : (4.1.2) BIOS Process
Area to save IY register when BIOS is called.

F434H : YPFKBUF (32) : R/W : 4.3 Key Input
This buffer stores the character string at character string input in the alphanumeric mode or kana mode of the electronic calculator and EHT-10.

F454H : RM_BUF (6) : - : (4.3) Key Input
This buffer stores converted kana code in EHT-10/2 kana mode.

F45AH : FUNC_TBL (96) : R/W : (4.3) Key Input
Function bank and address for function code E0H to FEH is set in a 3 byte configuration.

F4BAH : SVKMODE (2) : - : (4.3) Key Input
Area to save the current key input mode KMODE (F227H) in the system.

F4BCH : SVPFMOD (1) : - : (4.3) Key Input
Area to save YPFCMFLG (F21DH) in the system.

F4BDH : PPUTPTR (2) : R/O : 4.5 Printer
PUT pointer to buffer PRNBUF (FB2AH) for printer unit output.

F4BFH : PGETPTR (2) : R/O : 4.5 Printer
GET pointer from PRNBUF (FB2AH) for printer unit output.

F4C1H : SEKDSK (1) : R/O : (4.2) BIOS (SELDSK)
Physical number of the disk selected by BIOS SELDSK is set.

F4C2H : SEKTRK (2) : R/O : (4.2) BIOS (SETTRK)
Track number which is set by BIOS SETTRK is stored.

F4C4H : SEKSEC (1) : R/O : (4.2) BIOS (SETSEC)
Sector number which is set by BIOS SETSEC is stored.

F4C5H - F4D4H : For future use.

F4D5H : DMAADR (2) : R/O : (6) Disk System Description
DMA buffer address used by BIOS at disk read/write is set. It points to SYSDMA (F851H) normally.

F4D7H : DIRBUF (128) : R/W : (5) BDOS
DMA buffer used by BDOS at directory read/write.

F557H : ALV0 (29) : R/O : (5) BDOS
Allocation area for RAM disk (A:)

F574H : CSV0 (0) : - : (5) BDOS
Checksum area for RAM disk (A:) (Only label is defined.)

F574H : ALV1 (16) : R/O : (5) BDOS
Allocation area for ROM disk (B:)

F584H : CSV1 (0) : - : (5) BDOS
Checksum area for ROM disk (B:) (Only label is defined.)

F584H : ALV2 (8) : R/O : (5) BDOS
Allocation area for IC card (C:)

F58CH : CSV2 (16) : R/O : (5) BDOS
Checksum area for IC card.

F59CH : ALV3 (18) : R/O : (5) BDOS
Allocation area for FDD (D:)

F5AEH : CSV3 (16) : R/O : (5) BDOS
Checksum area for FDD (D:)

F5BEH : ALV4 (18) : R/O : (5) BDOS
Allocation area for FDD (E:)

F5D0H : CSV4 (16) : R/O : (5) BDOS
Checksum area for FDD (E:)

F5E0H - F5E1H : For future use.

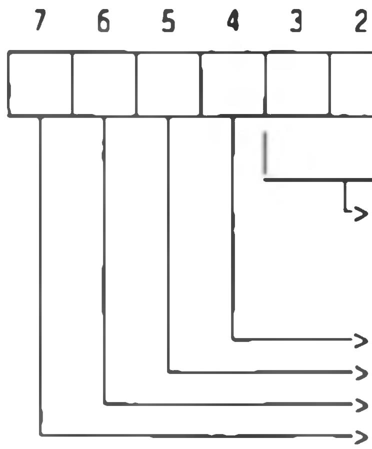
F5E2H : DISK_BNK (1) : - : (6) Disk System Description
Indicate the bank where the terget sector is located at read/write to RAM disk or ROM disk.

F5E3H : HCX (1) : - : (4.2) BIOS (SCRNDUMP)
F5E4H : HCY (1) : - :
F5E5H : HCN (1) : - :
F5E6H : HCDATA (8) : - :
Work area used for screen dump in EHT-10/2.

F5EEH : PKXOFF (1) : - : (4.2) BIOS (KANJI)
F5EFH : PKXLEN (1) : - :
F5F0H : PKFOFF (1) : - :
Work area used in the system when kanji print is performed using BIOS KANJI.

F5F1H : LSTERR (1) : R/W : 4.2 BIOS (LIST)
LST: Return parameter at output.
=00H : Normal end
≠00H : Abnormal end

F5F2H : YKCOUNTRY (1) : R/O : -
This area has the following data.



> Destination is determined by the state of dip switch 1.
> =7H : Japan (on)
> =FH : ASCII (off)
> State of dip switch 2
> State of dip switch 3
> State of dip switch 4
> Distinction of EHT-10 and EHT-10/2
> =0 : EHT-10
> =1 : EHT-10/2

F5F3H : YLDFLTC (1) : R/W : 4.4 LCD Display
Default value of display character set. Low order 4 bits of YKCOUNTRY (F5F2H) is copied at reset.

F5F4H : YLCOUNTRY (1) : R/W : 4.4 LCD Display
Current value of display character set. YLDFLTC (F5F3H) is copied at WBOOT.

F5F5H - F5F8H : For future use.

F5F9H : TIMEEND (2) : R/W : 2.3.5 Sleep Function and Auto Power Off Function
Area to store auto power off time.

F5FBH : SRBADR (9) : R/O : 4.2 BIOS (PUNCH)
Parameter packet to open RS-232C when using RS-232C with I/O device (CON:, RDR:, PUN:, LST:). The configuration is the same as BIOS RSIOX open parameter. When an I/O device is used, copy the 9 bytes that follow SRSADR (F010H) of default data.

F604H - F623H: BIOS RSIOX work area. See RSIOX in 4.2 BIOS Function Description for details.

F624H -F625H : For future use

F626H : EPWKTP (5) : R/W : 6.6.4 Protocol
Area to store data of FMT, DID, SID, FNC, SIZ at EPSP data is send.

F62BH : EPACKC (1) : R/W : 6.6.4 Protocol
AREA to store ACK or NAK sent to the terminal side (FDD) at EPSP data send/receive.

F62CH - F62EH : For future use.

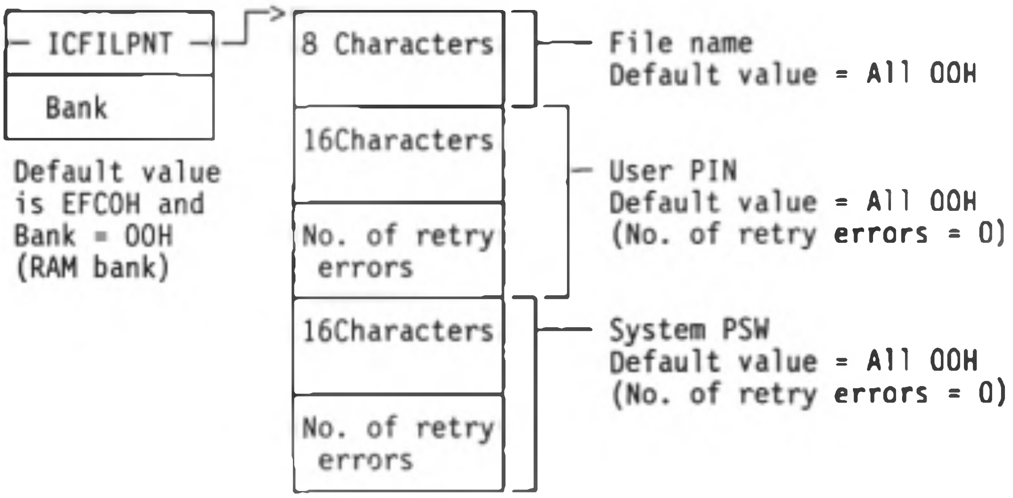
F62FH : RO (18) : - : 6.6.4 Protocol
Work area used in the system at EPSP data send/receive.

F641H : For future use.

F642H : BEEPBASE (1) : - : (4.2) BIOS (BEEP)
Work area used to measure BIOS BEEP time.

F643H : ICDIDPNT (4) : R/W : 11.3 IC card protocol Expansion
Represents the storage area for ID data during registration of the card ID. (Valid only in DISK Mode)

F647H : ICFILPNT (3) : R/W : 11.3 IC card protocol Expansion
Represents the storage area of the File Name data during file creation. All of the above data are valid only in DISK Mode and represent the address of the data set by the ISET function.



F64AH : ICFILSZ (1) : - : 11.3 IC card Protocol Expansion
Specifies data area size of the IC card as a disk. Unit is 128bytes and default value is 38H (7 kbytes).
This variable is set by IDSK function.

F64BH : ICDIRNO (1) : - : 11.3 IC card protocol Expansion
Specifies the number of directories of the IC card as a disk. Default value is 08H. This variable is set by IDSK function.

F64CH : ICOPNFLG (1) : - : 11.3 IC card Protocol Expansion
Flag for distinguish whether the class code of the command data is File class code or System class code.

F64DH : ICDIRBOT (2) : - : 11.3 IC card protocol Expansion
Points the bottom address of the directory area of the IC card as a disk. Default value is 100H. This variable is set by IDSK function.

F64FH : ICDATTOP (2) : - : 11.3 IC card protocol Expansion
Points the logical top address of the data area of the IC card as a disk. Default value is 400H.

F651H : ICBLKSZ (2) : - : 11.3 IC card protocol Expansion
Specifies the block size of the IC card. Default value is 100H (256bytes)

F653H : ICWAIT4 (1) : - : 11.3 IC card protocol Expansion
Specifies the waiting time between to supply the power to the IC card and to supply the clock to the IC card. Unit is mSec and default value is C1H(1mSec).

F654H : ICSTATUS (3)
F657H : ICOFFMD (1) — For future use.
F658H : RZICCTL1 (1)
F659H : RZICCTL2 (1)

F65AH - F65BH : For future use.

F65CH : ICMAXREC (2) : R/O : 11.3 IC card protocol Expansion
F65DH : QTICCARD (1) : R/O : 11.3 IC card protocol Expansion
F65FH : TPICCARD (1) : R/O : 11.3 IC card protocol Expansion
Indicates data for the IC card capacity.

- ICMAXREC -- Indicates maximum number of physical records of IC card. Default is 40H (64records)
- QTICCARD -- Indicates physical capacity of IC card. Unit is specified by ICBLKSZ. Default is 20H (32 x 256bytes).
- TPICCARD -- Indicates the number of records used for directory. Default is 08H (8 records).

F660H : ICONFLG (1) : R/O : 11.3 IC card protocol Expansion
IC card power supply state is indicated.

- =00H : No power supply (closed)
- =01H : During power supply (open)
- =FFH : No access for the proper time, power supply is temporarily stopped.

F661H : ICCVFLG (1) : R/O : 11.3 IC card protocol Expansion
Indicates the rear cover state while the IC card is in use.

- =00H : IC card rear cover closed
- =01H : IC card rear cover open

F662H : ICUSEDV (1) : R/O : 11.3 IC card protocol Expansion
Indicates the IC card current use status.

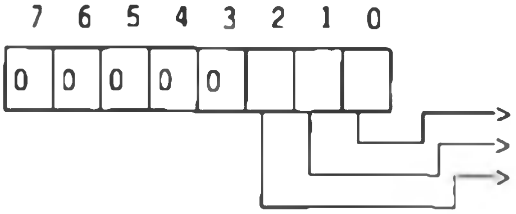
- =FFH (-1) : Command through mode uses IC card.
- =00H : For future use
- =01H : Being used by BIOS ICCARD
- =02H : Being used as a disk

F663H : ICRSTDT (10) : R/O : 11.3 IC card protocol Expansion
Area to store reset response data of the IC card, pointed to by ICRSTPNT (F1AEH).

F66DH : For future use.

F66EH : ICWAIT3 (2) : R/W : 11.3 IC card protocol Expansion
Specifies the interval (in 10-ms units) for a time out error for the response from the IC Card after command transmission. Default value = 00FFH (approx. 2.5 Sec)

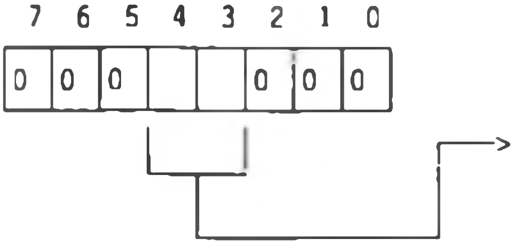
F670H : ICOPNSTS (1) : R/W : 11.3 IC card protocol Expansion
Specifies the processing during File OPEN status (after the IC Card power is ON). (Valid only in DISK Mode)



Default value = 07H

ID collation (1: ON)
User PIN collation (1: ON)
System PSW collation (1: ON)

F671H : ICFMTSTS (1) : R/W : 11.3 IC card protocol Expansion
Specifies the processing during formatting. (Valid only in DISK Mode)



Default value = 10H

= 00 : Do nothing.
= 01 : Delete the specified file only.
= 1X : Create a file and format after deleted the specified file.

F672H : For future use.

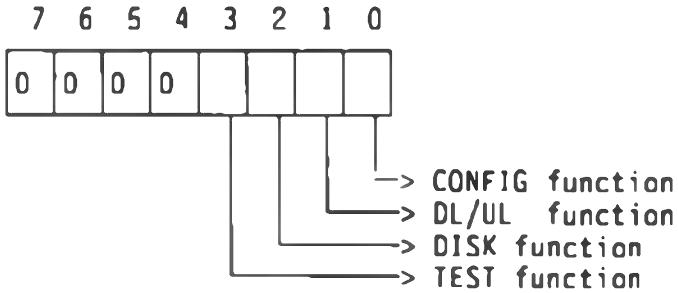
F673H - F6B6H : Work area related to display
See 4.4.7 Work Area Detail Related to Display for further information.

F6B7H : ERRSEC (2) : R/W : 2.5 Self Test
Area to store error sector number when an error occurs at RAM disk checksum. Consecutive numbers are used from track 0 or sector 0 for sector numbers

F6B9H : FNBLOS (1) : - : -
Work area used when BIOS is used in the system.

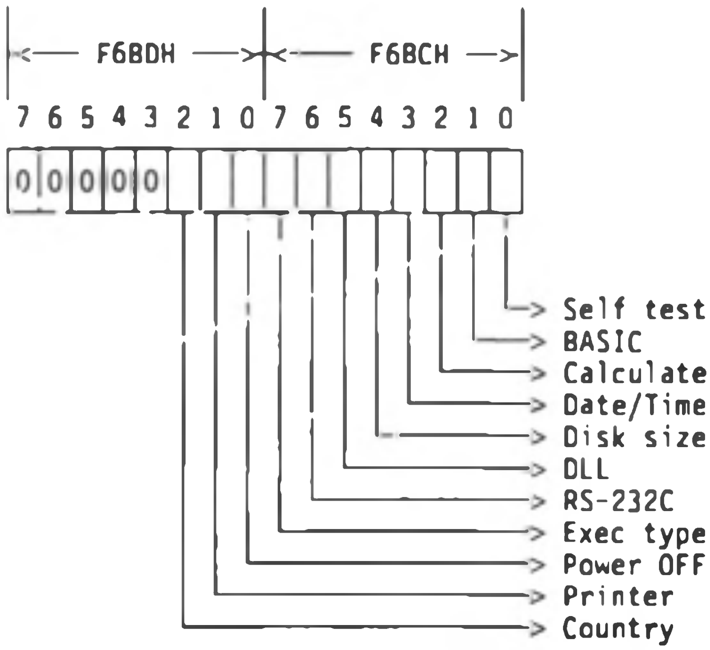
F6BAH : SFUNCCD (1) : - : -
Work area used to select functions from the system menu initial screen.

F6BBH : SYSMMOD (1) : R/W : 9.2.1 System Menu Control
This flag disables the system menu functions.



Bits are disabled by 0 and enabled by 1.

F6BCH : CONFMOD (2) : R/W : 9.2.1 System Menu Control
This flag disables the functions of CONFIG.



Bits are disabled by 0 and enabled by 1.

F6BEH : SVSEKDSK (1) : - : -
Area to save the disk number currently set when disk access is performed using the system menu function.

F6BFH : SVSEKTRK (2) : - : -
Area to save the track number currently set when disk access is performed using system menu function.

F6C1H : SVSEKSEC (1) : - : -
Area to save the sector number currently set when disk access is performed using system menu function.

F6C2H : SVERRFLG (1) : - : -
Area to save the flag which indicates the current BDOS error process mode when disk access is performed using the system menu functions.

F6C3H : TSPSAVE (2) : R/O : 11.2 Communication Protocol Expansion
Area to save the stack pointer when BIOS TCAM starts.

F6C5H : TCNTDT (2) : - : 11.2 Communication Protocol Expansion
Area to check the number of retries when Filink protocol is specified.

F6C7H : TERRTIME (2) : - : 11.2 Communication Protocol Expansion
Work area to check timeout time by BIOS TCAM.

F6C9H : TCAMAREA (12) : R/W : 11.2 Communication Protocol Expansion
Work area to pass parameter in BIOS TCAM application.

It has the following meaning at Filink protocol.
At file send ... Perform connect process after storing filename (11 bytes) in this area.

At file receive ... After connect process, the filename received in this area is stored and returned.

At no sequence
It is not used at TCAM but used for filename memory in the DL/UL process or DLL process. (To equalize the interface to Filink.)

At expansion protocol
Using identical interface with Filink is preferable.

F6D5H : TCAMIF (1) : R/W : 11.2 Communication Protocol Expansion
Area to store whether connect is performed in the send/receive mode when Filink protocol is specified in BIOS TCAM.
=00H : Send mode
=01H : Receive mode

F6D6H : DLLSVSP (2) : R/O : 11.2 Communication Protocol Expansion
Work area to save the stack pointer when the DLL process starts.

F6D8H : LOADAD (2) : R/O : 11.2 Communication Protocol Expansion
This work area stores the next data loading address when loading a file to the DLL process TPA area.

F6DAH : DLLRVS (1) : - : 11.2 Communication Protocol Expansion
This flag indicates message reverse state of "Down Loading" and "Up Loading" in the DLL and DL/UL processes.

F6DBH : DLLFILE (1) : R/O : 11.2 Communication Protocol Expansion
Work area to store type of received file in the DLL process.
=00H : COM file received
=01H : BAS file received
=FFH : For future receive
=FEH : File receive other than COM/BAS (During open)

F6DCH : ERRADR (2) : R/W : 11.2 Communication Protocol Expansion
Indicates error the process execute address when a error occurs during the DLL or DL/UL process.

F6DEH : ULDLSVSP (2) : - : 11.2 Communication Protocol Expansion
Work area to save the stack pointer when the UL/DL process starts.

F6E0H : ULDLTYPE (1) : R/O : 11.2 Communication Protocol Expansion
This flag indicates whether either the UL/DL process is being executed.
=00H : Down Loading
=01H : Up Loading

F6E1H : RENFLG (1) : - : 11.2 Communication Protocol Expansion
This flag indicates whether the rename process is executed at the disk write process in the UL/DL process.

F6E2H : STESTWK (10) : - : -
Work area used in the system menu TEST process.

F6ECH : BCDCB (60) : R/W : 11.5 Barcode Decoder Addition
Work area used in barcode read routine. See 11.5.3 Barcode Reader Work Area Details for further information.

F728H : SGRAPHWK (128) : - : (4.2) BIOS (GRAPHICS)
Work area used by BIOS GRAPHICS and KANJI.

F7A8H : DBGWORK (18) : - : (10.3) System Jump Table
Work area used by EHT-10/2 debugger (WAD).

F7BAH - F7DFH : Work area used by EHT-10/EHT-10/2 debugger (WAD). However, when the development cartridge is not connected, it can be used by a user anytime.

(5) System Work Area V (RSYSAR5)

This area has a buffer and stack area which do not need initialization, and has a system hook table, system jump table and interrupt vector which are initialized at reset.

Address : Variable (Number of bytes) : Reference

F7E0H : KEY_BUFFER (66) : 4.3 Key Input
Key input buffer

F822H - F82CH : For future use

F82DH : SYSFCB (36) : (5) BDOS Description
The FCB data specified by a user is copied to this area while BDOS is in use.

F851H : SYSDMA (128) : (4.2) BIOS (READ) (WRITE)
Data exchange with disk is performed through this area while BDOS and BIOS are in use.

F8D1H : PKT_TOP (9) : (6.5.4) Protocol, 11.3 IC card Protocol Expansion
This EPSP data send/receive area is used in the system at EPSP data send/receive. This is also used as work area for the IC card command and data send/receive.

F8DAH : SCRCH_BUF (128) : (6) Disk System Description
Scratch buffer at disk access.

F95AH : COMBUF (256) : (7.2) Serial Interface
Receive buffer which uses RS-232C with an I/O device (CON:, RDR:, PUN:, LST).

FA5AH : BCDBUF (208) : (11.5) Bar Code Decoder Addition
Buffer to store data read from the barcode reader.

FB2AH : PRNBUF (128) : 4.5 Printer
Output buffer to printer unit.

FBAAH : SYSWORK (261) : -
This is used as a common work area in the system.

FCB0H : STRTSP (96) : (2.3) System Start and End
Stack area for the system address 0 start (STARTER) process.

FD10H : SPEXT (32) : (8) Interrupt Process
Stack area for EXT interrupt process

FD30H : SPICF (48) : (8) Interrupt Process
Stack area for ICF interrupt process

FD60H : SPOVF (56) : (8) Interrupt Process
Stack area for OVF interrupt process

FD98H : SP8251 (48) : (8) Interrupt Process
Stack area for EXT interrupt process

FDC8H : SP7508 (120) : (8) Interrupt Process
Stack area for 7508 interrupt process

FE40H : BIOSSK (176) : (4.1.2) BIOS Process
Stack area for BIOS process

FEF0H : BDOSSK (64) : (5) BDOS
Stack area for the BDOS process

FF60H : SYSHOKTOP (48) : 10.2.4 system Hook (II) Description
System Hook Table (II)

FF90H : RSJMPTOP (48) : 10.3.3 Resident Jump Table Description
Resident Jump Table

FFC0H : RSHOKTOP (48) : 10.2.3 System Hook (I) Description
System Hook Table (I)

FFF0H : VECTBLTOP (16) : 8.2 Interrupt Vector
Interrupt Vector Table

(6) Label Table

Address	Label name	Address	Label name	Address	Label name
F07A	ADDLB1	F1CD	ADDLB2	F6B9	ADDLB4
F42C	ADDLNK	F079	ADDLT1	F1CA	ADDLT2
F6B7	ADDLT4	F0E1	ADSKOPN	F065	AD_RAM_IN
F072	AD_RAM_OLD	F063	AD_ROM_CP1	F3CA	AFTER3
F0E0	AHSTWRT	F3DF	ALLOCA	FFC0	ALMHK0
FFC3	ALMHK1	FFC6	ALMHK2	FFC9	ALMHK3
FFCC	ALMHK4	F029	ALRMAD	F3C5	ALRMCT
F027	ALRMDS	F02C	ALRMFG	F32F	ALRMMSG
F01E	ALRMSOND	F02B	ALRMST	F1CA	ALRMTIME
F028	ALRMTP	F557	ALV0	F574	ALV1
F584	ALV2	F59C	ALV3	F5BE	ALV4
F407	AREC1	F405	ARECORD	F3D1	ARET
F020	ATSHUTOFF	F021	ATSOTIME	F723	BARBIT
BC3A	BARCDBOT	1020	BARCDSIZE	AC1A	BARCDTOP
F6F8	BARFLG	F71C	BARTHR	FFEA	BASHOOK
F07F	BASICB	F07C	BASICF	F07D	BASICR
FA5A	BCDBUF	F6EC	BCDCB	F70C	BCGETP
F3BC	BCINTST	F710	BCLOC	F711	BCLOF
F6FA	BCOPTN	F70E	BCPUTP	F714	BCTLOC
F715	BCTLOF	F712	BCTPTP	F6F9	BCTYPE
F034	BDOLB1	F0D6	BDOLB2	F219	BDOLB3
F40F	BDOLB4	F034	BDOLT1	F0D6	BDOLT2
F209	BDOLT3	F3CD	BDOLT4	25A1	BDOSBOT
F413	BDOSFN	0EFE	BDOSSIZE	FF5F	BDOSSK
0070	BDOSSKSZ	16A3	BDOSTOP	F414	BDSBNK
F005	BDSLAD	F642	BEEPBASE	F4D7	BEGDAT
F007	BI1LAD	F63B	BIAS	F074	BIOLB1
F1B8	BIOLB2	F242	BIOLB3	F673	BIOLB4
F05C	BIOLT1	F0D6	BIOLT2	F219	BIOLT3
F42C	BIOLT4	38DB	BIOS1BOT	12E0	BIOS1SIZE
25FB	BIOS1TOP	46CE	BIOS2BOT	0DF3	BIOS2SIZE
38DB	BIOS2TOP	552C	BIOS3BOT	0E5E	BIOS3SIZE
46CE	BIOS3TOP	DC9D	BIOS4BOT	0F60	BIOS4SIZE
CD3D	BIOS4TOP	F417	BIOSERROR	F426	BIOSFN
FFE7	BIOSHK	0038	BIOSNUM	FEEF	BIOSSK
00B0	BIOSSKSZ	F639	BITMAP	F722	BITPTN
F3E4	BLKMSK	F3E3	BLKSHF	F076	BLNKCNT
F077	BLNKRVRS	F075	BLNKSTAT	F078	BLNKTIME
F034	BNKDTBL	F23B	BPINTEBL	0100	BRBDOSD
F079	BTRYALM	F6B5	BTRYDSP	F0B3	BTRYFG
FCAF	BUFBOT	F3D9	BUFFA	0100	BUFSIZE
F097	BUZZHZ	F3B9	BUZZLNG	F094	BUZ_FLG
F41C	CALBNK	AC1A	CALCBOT	0C46	CALCSIZE
9FD4	CALCTOP	DFF5	CCPBOT	F003	CCPLAD
0358	CCPSIZE	DC9D	CCPTOP	F3D3	CDRMAXA
F3DD	CHECKA	F3EC	CHKSIZ	F30B	CHKSUM
F30A	CHKSUMFG	F618	CHRMSK	F067	CKSMSIZE
F200	CNTNFG	F305	CNTNILVL	F01C	CNTNSOND
F303	CNTNSP	F20B	COLUMN	F95A	COMBUF
F034	COMLB1	F0D6	COMLB2	F209	COMLB3
F3CD	COMLB4	F000	COMLT1	F090	COMLT2
F200	COMLT3	F300	COMLT4	F209	COMPOOL

Address	Label name	Address	Label name	Address	Label name
8D0F	CONFIGBOT	22BF	CONFIGSIZE	6A50	CONFIGTOP
F6BC	CONFMOD	F0C8	CONSCRN1	F0D0	CONSCRN2
F090	CPMSIZ	F42F	CRGDEV	FF6C	CRGHOOK
F220	CSTOPFLG	F1CB	CSTOPPRT	F574	CSV0
F584	CSV1	F58C	CSV2	F5AE	CSV3
F5D0	CSV4	F067	CS_RAM_IN	F423	CURBNK
F20F	CURDSK	F3D7	CURRECA	F3D5	CURTRKA
F633	DA	010B	DATSIZ	F208	DBGFLG
F7A8	DBGWORK	F421	DBNKDT	F40A	DCNT
F3EA	DIRBLK	F4D7	DIRBUF	F724	DIRECF
F3F4	DIRLOC	F3E8	DIRMAX	F062	DIRSIZE
F41B	DISBNK	F23F	DISBRK	F0E9	DISKROV
F0E4	DISKTBL	96D4	DISKUTYBOT	054B	DISKUTYSIZE
9189	DISKUTYTOP	F5E2	DISK_BNK	FF66	DLHK1
67D6	DLLBOT	F851	DLLBUF	F82D	DLLFCB
F6DB	DLLFILE	FF63	DLLHK1	F6DA	DLLRVS
0676	DLLSIZE	F6D6	DLLSVSP	6160	DLLTOP
F1DA	DLLTYPE	F213	DLOG	F215	DMAAD
F4D5	DMAADR	F3F7	DMINX	F159	DPB0
F168	DPB1	F177	DPB2	F186	DPB3
F186	DPB4	F3DB	DPBADDR	F109	DPBASE
F109	DPE0	F119	DPE1	F129	DPE2
F139	DPE3	F149	DPE4	F409	DPTR
F1DD	DRCVBUF	F1DB	DRCVCNT	F40C	DREC
F062	DR_RAM_IN	F061	DR_ROM_CP1	F0E2	DSKDID
F1DF	DSKNUM	F0E3	DSKSID	F07A	DSKTFLG
F06D	DSK_R_W	F074	DSPFLAG	F2E0	DSPTPSV
F2DF	DSPTYPE	F00F	DUPLEX	0000	DXLT0
0000	DXLT1	0000	DXLT2	0000	DXLT3
0000	DXLT4	F210	EFCB	F204	ELCTLFLG
F3B7	ELOFFEND	F023	ELOFTIME	F5E2	ENDDAT
F3C0	ENTSINT	F3CD	ENTSP	F1AB	EP1BRTO
F62B	EPACKC	F1A8	EPATMO	F62C	EPBLCN
F627	EPDDEV	F62D	EPERMD	F1AA	EPETRL
F1A7	EPETMO	F626	EPFMT	F629	EPFNC
F1A9	EPMODE	F628	EPSDEV	F62A	EPSIZ
F1A6	EPTIMO	F1A5	EPTRCN	F626	EPWKTP
F4D1	ERFLAG	F6DC	ERRADR	F416	ERRFLG
F6B7	ERRSEC	F019	EXECTYPE	F30D	EXESTRNG
FFDB	EXTHOOK	F3E5	EXTMSK	F402	EXTVAL
F3F2	FCBsCOPIED	F400	FCBDSK	F3C4	FG7508
FBAA	FILENAME	0088	FLA1SIZE	015A	FLA2SIZE
00ED	FLA3SIZE	04BA	FLA4SIZE	F637	FLGCNT
F6B9	FNBIOS	F05E	FORMATRAM	F702	FRCBUF
F201	FRCECNTN	F704	FRCSIZ	F0EB	FTSTAB
F45A	FUNC_TBL	F22F	GET_KEYPTR	CD3D	GRAPHBOT
1103	GRAPHSIZE	BC3A	GRAPHTOP	F5E6	HCDATA
F5E5	HCN	F5E3	HCX	F5E4	HCY
FFD2	HK8251	0010	HOOKNUM	F1A3	HSARTCR
F1A1	HSARTMR	F1A0	HSCTLR1	F1A4	HSSOUT
F1A2	HSSWR	F4CA	HSTACT	F4C5	HSTDSK
F4C8	HSTSEC	F4C6	HSTTRK	F4CB	HSTWRT
F651	ICBLKSZ	F1B3	ICCDPRM	F1B1	ICCDTIME
F1E4	ICCLASS	F202	ICCPWCNT	F203	ICCPWFLG
F026	ICCPWTM	F661	ICCVFLG	F64F	ICDATTOP
FF6F	ICDHK1	FF72	ICDHK2	FF75	ICDHK3

Address	Label name	Address	Label name	Address	Label name
FF78	ICDHK4	FF7B	ICDHK5	FF7E	ICDHK6
FF81	ICDHK7	FF84	ICDHK8	F643	ICDIDPNT
F64D	ICDIRBOT	F64B	ICDIRNO	F8D1	ICDWORK1
F8D5	ICDWORK2	F6EC	ICFDSP	FFD5	ICFHOOK
F647	ICFILPNT	F64A	ICFILSZ	F1E3	ICFLCLS
F671	ICFMTSTS	F1E6	ICHOKDT1	F65C	ICMAXREC
F657	ICOFFMD	F660	ICONFLG	F64C	ICOPNFLG
F670	ICOPNSTS	F1E5	ICRECSZ	F66D	ICRETRY
F665	ICRSTA1	F666	ICRSTB1	F667	ICRSTC1
F1AD	ICRSTCNT	F668	ICRSTD1	F663	ICRSTDT
F1AE	ICRSTPNT	F664	ICRSTTO	F663	ICRSTTS
F1E2	ICSRACD	F654	ICSTATUS	F672	ICTSBYT
F1E9	ICTSDT	F662	ICUSEDV	F1E7	ICWAIT1
F1E8	ICWAIT2	F66E	ICWAIT3	F653	ICWAIT4
F65A	ICYOBI	F1B0	IDFLTCNT	0100	IFOSRBOT
0100	IFOSRSIZE	0000	IFOSRTOP	6160	IFSYSRBOT
0160	IFSYSRSIZE	6000	IFSYSRTOP	F3CF	INFO
F23C	INHCOPY	F3CC	INPSTOP	F091	INTBIOS
F3C2	INTFG	F092	INTLEVEL	0F18	INTROMBOT
07B0	INTROMSIZE	0768	INTROMTOP	F099	INTS7508
F09B	INTS8251	F0A1	INTSEXT	F09D	INTSICF
F09F	INTSOVF	F093	INTTYPE	F098	INTVLFG
F031	ISTS7508	FFAB	JCALLXX	FFAE	JINTOPR
FFA8	JJUMPXX	FF9F	JLDAXX	FFA5	JLDIRXX
FFB7	JNOP14	FFBA	JNOP15	FFBD	JNOP16
FF93	JPREBIOS	FF96	JPSTBIOS	FF90	JRBDOS
FFB4	JRDBGIOX	FF99	JSCALLX	FFB1	JSDBGIOX
FF9C	JSELBNK	FFA2	JSTAXX	F222	KALMMOD
F222	KANAFLG	F820	KBUFEND	F221	KB_FLG
F095	KDFLTBZ	F21B	KEYD	F21A	KEYF
F21C	KEYS	F7E0	KEY_BUFFER	F227	KMODE
F20D	KPCHAR	E91A	KSYSTBL	F233	KTABPTR
E8D6	KUSER1TBL	E8F8	KUSER2TBL	F7BA	LAB4END
F7BA	LABEND	F1C7	LALPHAYX	F25D	LATRCHK
00D0	LBCDBUFS	F7E0	LBUFTOP	0100	LCBUFS
F673	LCHRFONT	F269	LCRWAIT	F267	LCURATR
F25F	LCURSOR	F242	LDSPMOD	F6FE	LEDCNT
F6FD	LEDTIM	F26F	LESCCNT	F26E	LESCFLG
F270	LESCPRM	F24B	LESCTB1	F24D	LESCTB2
F26C	LFKADDR	F26B	LFKSTAT	F724	LFLAG
F3F6	LINFO	F20C	LISTCP	0002	LKBFES
0040	LKBUFS	F1C8	LKCODEYX	F244	LLCDMOD
F255	LLNKFLG	F26A	LLNKWAIT	F6D8	LOADAD
F06A	LOG_ADRS	F268	LOLDATR	0009	LPACHS
0080	LPRNBUFS	F24F	LSCADDR	F260	LSCCPOSX
F261	LSCCPOSY	F2DD	LSCMDSV	F2DC	LSCMODE
0080	LSCRBS	F25E	LSCROLMD	F249	LSCRTB1
F253	LSCRVRAM	F251	LSCSIZE	F257	LSCSIZEX
F258	LSCSIZEY	0080	LSDMAS	0024	LSFCBS
F11A	LSORBNK	F5F1	LSTERR	0105	LSYSWKS
000B	LTBUFS	F1C4	LTOUCHAD	F245	LUWDMOD
F25B	LUWDPOSY	F25C	LUWDSZY	F683	LVRAMADR
F685	LVRAMDT	F266	LVRAMYOF	F2DE	LVSVFLG
F264	LWDCPOSX	F265	LWDCPOSY	F259	LWDPOSY
F25A	LWDSIZEY	F262	LWDXMIN	F263	LWDYMIN
F24F	LWORKBF0	F27E	LWORKBF1	F2AD	LWORKBF2

Address	Label name	Address	Label name	Address	Label name
F6A2	LW_BPOS	F6A0	LW_DADDR	F6A4	LW_LAL
F69E	LW_SADDR	F3E6	MAXALL	FC18	MAXFILE
F38D	MCURFILE	F38B	MCURLINE	F38C	MCURPAGE
F0B9	MDRV	F397	MDSPBUF	F391	MDSPTOP
F2E3	MEMD	F2E1	MEMK	F2E2	MEMO
15C7	MENUBOT	F38A	MENUMOD	06AF	MENUSIZE
0F18	MENUTOP	F393	MFATTR	F395	MFILEBUF
F38E	MFILENUM	F390	MFSTSCRH	F0BE	MFTYP
F38F	MLINEMAX	F00A	MODEFG	F726	MODULS
F38F	MOLDDRV	F39F	MTIMEBUF	F71E	NRWBAR
F718	NRWSPC	F3EE	OFFSET	F418	OLDBNK
F3FF	OLDDSK	F641	OLDDVMD	F088	OS2LB1
F1EA	OS2LB2	F2ED	OS2LB3	F7BA	OS2LB4
F07A	OS2LT1	F1CD	OS2LT2	F2E1	OS2LT3
F6B9	OS2LT4	8000	OSROMBOT	FFD8	OVFHOOK
F706	OVTIME	FBAA	PBUFEND	F4BF	PGETPTR
F23D	PKANJFLG	F5F0	PKFOFF	F8D2	PKT_DID
F8D6	PKT_DRV	F8D1	PKT_FMT	F8D4	PKT_FNC
F8D6	PKT_RDT	F956	PKT_RSTS	F8D8	PKT_SEC
F8D3	PKT_SID	F8D5	PKT_SIZ	F8D6	PKT_STS
F8D1	PKT_TOP	F8D7	PKT_TRK	F8DA	PKT_WDT
F8D9	PKT_WTP	F5EF	PKXLEN	F5EE	PKXOFF
F700	PLSCNT	F6FF	PLSFRQ	F4BD	PPUTPTR
25B5	PREBIOSBOT	0014	PREBIOSSIZE	25A1	PREBIOSTOP
F92A	PRNBUF	F3BD	PRNPWCNT	F3BE	PRNPWFLG
F025	PRNPWTM	F23E	PRTFLG	F1CC	PRTINIT
25FB	PSTBIOSBOT	0046	PSTBIOSSIZE	25B5	PSTBIOSTOP
F235	PTR_KEYTBL	F22D	PUT_KEYPTR	F01A	PWONSOND
F701	PWRCNT	F3C9	PWSWOFFG	F3C8	PWSWONFG
F65E	QTICCARD	F060	QT_RAM_IN	F071	QT_RAM_OLD
F05F	QT_ROM_CP1	F62F	R0	F630	R0H
F62F	R0L	F631	R1	5E00	R1BDOSTOP
5F00	R1BIOSTOP	5600	R1CCPDTOP	016A	R1EXCHRSIZE
E896	R1EXCHRTOP	F632	R1H	008C	R1KTBLSIZE
E80A	R1KTBLTOP	F631	R1L	002A	R1LNKSIZE
E7E0	R1LNKTOP	6000	R1RAMDSKTOP	0400	R1SCR1SIZE
DC00	R1SCR1TOP	0150	R1SCR2SIZE	E690	R1SCR2TOP
DC00	R1USERTOP	0690	R1VRAMSIZE	E000	R1VRAMTOP
F633	R2	6000	R2BDOSTOP	6100	R2BIOSTOP
5800	R2CCPDTOP	00C4	R2EXCHRSIZE	E93C	R2EXCHRTOP
F634	R2H	0066	R2KTBLSIZE	E8D6	R2KTBLTOP
F633	R2L	0036	R2LNKSIZE	E8A0	R2LNKTOP
6200	R2RAMDSKTOP	0200	R2SCR1SIZE	DE00	R2SCR1TOP
0200	R2SCR2SIZE	E000	R2SCR2TOP	00A0	R2SCR3SIZE
E800	R2SCR3TOP	DE00	R2USERTOP	0200	R2VRAM1SIZE
E200	R2VRAM1TOP	0200	R2VRAM2SIZE	E400	R2VRAM2TOP
0200	R2VRAM3SIZE	E600	R2VRAM3TOP	F635	R3
F636	R3H	F635	R3L	F637	R4
F638	R4H	F637	R4L	F639	R5
F63A	R5H	F639	R5L	F63B	R6
F63C	R6H	F63B	R6L	F63D	R7
F63E	R7H	F63D	R7L	F63F	R8
F640	R8H	F63F	R8L	F5E2	RAMD_BNK
F068	RAMD_SIZE	F5F5	RAMSIZE	F07B	RAMTFLG
F069	RAM_SET	78A2	RBDOSDBOT	009F	RBDOSDSIZE
7803	RBDOSDTOP	EA00	RBDOSLAD	0100	RBDOSLSIZE

Address	Label name	Address	Label name	Address	Label name
0100	RBIOS1SIZE	F001	RBITOP	F401	RCOUNT
F4D3	READOP	F0F5	READTAB	F62E	REGA
16A3	RELOCBOT	00DC	RELOCSIZE	15C7	RELOCTOP
F6E1	RENFLG	F3FE	RESEL	7FE8	RESERVEBOT
0033	RESERVESZ	7FB5	RESERVETOP	F009	RESENQ
F000	RETADD	F41E	RETBNK	F725	RFLAG
F415	RIOBYTE	F1BE	RLCGENG	F1C1	RLCGENK
F1BB	RLCGENN	F246	RLCGENU	F1B8	RLCGENX
F3F3	RMF	F22B	RM_BOINCD	F454	RM_BUF
F237	RM_BUFPTR	F22A	RM_FSIIN	F225	RM_HATUFLG
F22C	RM_SIINCD	F224	RM_SIINFLG	F226	RM_SOKUFLG
F1E0	RNDFIG	F211	RODSK	F42A	ROMEXADD
F429	ROMEXBNK	F207	ROMEXQ	F428	ROMEXQON
8000	ROMIDBOT	0018	ROMIDSIZE	7FE8	ROMIDTOP
EFBB	RRSYSPRBOT	EB00	RRSYSPRTOP	F19E	RS2ARTCR
F19C	RS2ARTMR	F19B	RS2CTLR1	F19F	RS2SOUT
F19D	RS2SWR	F625	RSBYTE	F195	RSBYTED
F624	RSCALSW	F240	RSCLSF	F622	RSFDEV
F4D2	RSFLAG	FFC0	RSHOKTOP	F3BB	RSINTST
FF90	RSJMPTOP	F623	RSODEV	F61A	RSPAKAD
F60E	RSPBITL	F60D	RSPBITR	F60F	RSPPAR
F609	RSPRBAD	F605	RSPRBGP	F607	RSPRBPP
F60B	RSPRBSZ	F611	RSPSPP	F610	RSPSTOPB
F604	RSPSTS	F612	RSRBEAD	F61C	RSRDL
F619	RSSSPP	F621	RSXMODE	F00E	RSXOFF
F616	RSXOFSZ	F00D	RSXON	F614	RSXONSZ
F090	RSYSAR1BOT	0090	RSYSAR1SIZE	F000	RSYSAR1TOP
F200	RSYSAR2BOT	0170	RSYSAR2SIZE	F090	RSYSAR2TOP
F300	RSYSAR3BOT	0100	RSYSAR3SIZE	F200	RSYSAR3TOP
F7D0	RSYSAR4BOT	04D0	RSYSAR4SIZE	F300	RSYSAR4TOP
FF5F	RSYSAR5BOT	0780	RSYSAR5SIZE	F7E0	RSYSAR5TOP
7CB5	RSYSPRBOT	0413	RSYSPRSIZE	78A2	RSYSPRTOP
F1E1	RTNTYP	F05C	RXXLB1	F0D6	RXXLB2
F219	RXXLB3	F42C	RXXLB4	F034	RXXLT1
F0D6	RXXLT2	F219	RXXLT3	F40F	RXXLT4
F0DA	RZARTCR	F0D9	RZARTMR	F42C	RZBANKR
F0D7	RZCMDR	F0D6	RZCTLR1	F0D8	RZCTLR2
F0DE	RZCTLR3	F219	RZCTRL2	F658	RZICCTL1
F659	RZICCTL2	F0DB	RZICCTLR	F42E	RZIER
F0DD	RZIOCTLR	F42D	RZSBBNKR	F0DC	RZSWR
F070	R_W_RETRY	F635	SA	F430	SAVEIX
F432	SAVEIY	F41F	SBNKDT	F6FC	SCNERR
F084	SCONFMOD	F8DA	SCRCH_BUF	7803	SCREENBOT
22D7	SCREENSIZE	552C	SCREENTOP	F079	SCRLB1
F1CA	SCRLB2	F2E1	SCRLB3	F6B7	SCRLB4
F69D	SCRLFG	F074	SCRLT1	F1B8	SCRLT2
F242	SCRLT3	F673	SCRLT4	F3F9	SEASRCHA
F3F8	SEARSCHL	F3E1	SECTPT	F4C1	SEKDSK
F4C9	SEKHST	F4C4	SEKSEC	F4C2	SEKTRK
FC19	SELPOS	F3F5	SEQIO	F6BA	SFUNCCD
F728	SGRAPHWK	0010	SHOKNUM	F3FD	SINGLE
F61E	SISOCNT	F631	SIZE	001F	SIZERAM
0000	SIZEUSER	F00B	SIZRAM	F21E	SKEYFLG
F0B8	SMENUFLG	F223	SMKFLG	FE3F	SP7508
0078	SP7508SZ	FDC7	SP8251	0030	SP8251SZ
F722	SPCBIT	F716	SPCTHR	FD2F	SPENT

Address	Label name	Address	Label name	Address	Label name
0020	SPEXTSZ	FD5F	SPICF	0030	SPICFSZ
FD97	SPOVF	0038	SPOVFSZ	F5FB	SRBADR
F5FF	SRBR	F5FD	SRBSIZ	F41D	SRCBNK
F600	SRDCHR	F241	SRMODE	F601	SRPRT
F010	SRSADR	F602	SRSB	F014	SRSPAK
F603	SRSPARA	F389	SRSTMODE	F196	SRTABL
F369	SSEDITWK	F620	SSXMODE	F082	SSYSMMOD
0768	STARTBOT	0668	STARTSIZE	0100	STARTTOP
F6E2	STESTWK	F351	STIMEBUF	FCAF	STKBOT
F20A	STRTCOL	FD0F	STRTSP	0060	STRTSPSZ
F3C3	STS7508	F726	SUMCHK	F6A7	SVCRSR
F6C2	SVERRFLG	F4BA	SVKMODE	F419	SVOLDBNK
F4BC	SVPFMOD	F6BE	SVSEKDSK	F6C1	SVSEKSEC
F6BF	SVSEKTRK	7D45	SYSAR1BOT	0090	SYSAR1SIZE
7CB5	SYSAR1TOP	7EB5	SYSAR2BOT	0170	SYSAR2SIZE
7D45	SYSAR2TOP	7FB5	SYSAR3BOT	0100	SYSAR3SIZE
7EB5	SYSAR3TOP	F199	SYSARTCR	F197	SYSARTMR
F196	SYSCTLR1	F851	SYSDMA	F82D	SYSFCB
FF60	SYSHOKTOP	F081	SYSINFLG	FF5F	SYSLB5
F000	SYSLT1	F090	SYSLT2	F200	SYSLT3
F300	SYSLT4	6A50	SYSMENUBOT	027A	SYSMENUSIZE
67D6	SYSMENUTOP	F6BB	SYSMMOD	E000	SYSROMBOT
2009	SYSRSVBOT	000B	SYSRSVSIZE	DFF5	SYSRSVTOP
F19A	SYSSOUT	F198	SYSSWR	9FD4	SYSTESTBOT
0900	SYSTESTSIZE	96D4	SYSTESTTOP	FBAA	SYSWORK
7C00	SZRAM	0000	SZUSER	F06E	S_CHK_SUM
F1D9	T1STFLG	F1D5	T1STTIME	F1D7	T2NDTIME
F0A3	TBL7508	F851	TBUF	F3BF	TBUZ_FLG
F6C9	TCAMAREA	F6D5	TCAMIF	F1CD	TCAMPRM
F6C5	TCNTDT	F1D4	TDFLTCNT	F6C7	TERRTIME
FF60	TEXHK1	F6B3	THATRAD	F6AF	THCNTXY
F6AD	THPOSXY	F6B1	THSVSP	F1C9	THSYSFLG
F822	TIMBUF	F708	TIMCNT	F5F9	TIMEEND
F02D	TIMER0	F02F	TIMER1	F032	TIMER1M
F369	TIMEWK	F70A	TIMLOC	F3FB	TINFO
FFDE	TMDT83	FFE1	TMDT85	FFE4	TMDT86
F206	TMFLAG	F205	TMFUNC	FFCF	TMHOOK
F3C6	TMSEC	F05C	TOPRAM	F65F	TPICCARD
F3F0	TRANV	F6FB	TRMCHR	F6C3	TSPSAVE
E850	TSYSTBL	E80A	TUSERTBL	F086	UCONFMOD
9189	ULDLBOT	047A	ULDLSIZE	F6DE	ULDLSVSP
8D0F	ULDLTOP	F6E0	ULDLTYPE	FF69	ULHK1
F4CC	UNACNT	F4CD	UNADSK	F4D0	UNASEC
F4CE	UNATRK	F716	UPCBUF	F00C	USERBIOS
F0DF	USERCRG	F239	USERKTBL	F20E	USRCCDE
F217	USRDMA	F411	USRFCB	F40F	USRSBD
F424	USRSBI	F083	USYSMMOD	FFF0	VECTBLTOP
F403	VRECORD	F3CB	WFUNCFLG	F720	WIDBAR
F71A	WIDSPC	F0FF	WRITAB	F4D4	WRTYPE
F243	WTONLY	F300	XUSRBIOE	F0B6	YALMDS
F0B7	YALMST	F5F2	YKCOUNTRY	F5F4	YLCOUNTRY
F5F3	YLDFLTC	F308	YMAINST	F40E	YOLDDSK
F21D	YPFCMFLG	F434	YPFKBUF	F229	YPFKCNT
F231	YPFKPTR	F0B4	YPOFDS	F0B5	YPOFST
F307	YPWSWST	F5F7	YSIZERAM	F309	ZSTARTFG

5. BIOS FUNCTION LIST

- The meanings of symbols used in the columns of EHT-10 and EHT-10/2 are as follows:

0 : This function can be used in the same way as in HX-40 and PX-4.

o : This function can be used in the same way as in HX-40 and PX-4, but the parameter range is different from that in HX-40 and PX-4.

@ : This function can be used in the same way as in HX-40 and PX-4, but the parameter is different from that in HX-40 and PX-4.

x : This function is not supported in EHT-10/EHT-10/2.

$: This function has been newly added to EHT-10/EHT-10/2.

Entry address		Name	Function	Modification for HX-40/PX-4	EHT	
Offset from WBOOT	Absolute address				10	10/2
-03H	EB00H	BOOT	Cold-boots CP/M.		0	0
+00H	03H	WBOOT	Warm-boots CP/M.		0	0
+03H	06H	CONST	Checks the CON: input status.		0	0
+06H	09H	CONIN	Inputs one character from the CON: device.	Position code information is returned.	o	o
+09H	0CH	CONOUT	Outputs one character to the CON: device.	Function added and ESC sequence parameter partially modified	@	@
+0CH	0FH	LIST	Outputs one character to the LST: device.		0	0
+0FH	12H	PUNCH	Outputs one character to the PUNCH: device.		0	0
+12H	15H	READER	Inputs one character from the READER: device		0	0
+15H	18H	HOME	Sets the disk seek track to 0.		0	0
+18H	1BH	SELDSK	Specifies a drive.	Only drives A to E are supported.	o	o

Entry address		Name	Function	Modification for HX-40/PX-4	EHT	
Offset from WBOOT	Absolute address				10	10/2
+1BH	EB1EH	SETTRK	Specifies a track for read/write operations.	Depends on the drive.	o	o
+1EH	21H	SETSEC	Specifies a sector for read/write operations.	Depends on the drive.	o	o
+21H	24H	SETDMA	Specifies the DMA address for read/write operations.		0	0
+24H	27H	READ	Reads 128-byte data.		0	0
+27H	2AH	WRITE	Writes 128-byte data.		0	0
+2AH	2DH	LISTST	Checks the LST: device status.		0	0
+2DH	30H	SECTRN	Translates a logical sector to a physical sector.		0	0
+30H	33H	PSET	Performs a logical operation for VRAM data.	Covered by GRAPHICS (WBOOT+90H)	X	X
+33H	36H	SCRNDUMP	Dumps the VRAM contents to the LST: device.	No operation is performed in EHT-10.	X	0
+36H	39H	BEEP	Sounds the buzzer.	The hardware beep function has been added.	@	@
+39H +3CH +3FH +42H +45H +48H	3CH 3FH 42H 45H 48H 4BH		No function			
+4BH	4EH	TIMDAT	Sets and reads time, enables and disables the alarm/wake function and sets and reads the alarm/wake time.		0	0

Entry address		Name	Function	Modification for HX-40/PX-4	EHT	
Offset from WBOOT	Absolute address				10	10 / 2
+4EH	EB51H	MEMORY	Reads the current bank information.	The subbank value is returned as the return information.	@	@
+51H	54H	RSIOX	Performs serial communications.	The DSR/DTR line control functions have been added.	o	o
+54H	57H		No function			
+57H	5AH	MASKI	Sets and resets the interrupt mask, allows and prohibits interrupts by the 7508 CPU, and checks the current mask status.	The external interrupt control function has been added.	o	o
+5AH	5DH	LOADX	Reads one-byte data from the specified address in the specified bank.	The bank specification values have been modified because new subbanks have been added.	@	@
+5DH	60H	STORX	Writes one-byte data at the specified address in the specified bank.		@	@
+60H	63H	LDIRX	Transfers the specified length of bank data to the specified another bank.		@	@
+63H	66H	JUMPX	Jumps to the specified bank address.		@	@
+66H	69H	CALLX	Calls the specified bank address through a subroutine.		@	@
+69H	6CH	GETPFK	Reads the key code currently set.	The structure differs from that in PX-4/ HX-40.	@	@
+6CH	6FH	PUTPFK	Registers a key code in the user table.		@	@
+6FH	72H	READSW	Reads the states of switches.		0	0

Entry address		Name	Function	Modification for HX-40/PX-4	EHT	
Offset from WBOOT	Absolute address				10	10 / 2
+72H	EB75H		No function			
+75H	78H	RDVRAM	Reads one character from the screen.	Only one-byte screen buffer data is read.	@	@
+78H	7BH	MCMTX	Processes a microcassette.	No MCT is mounted	X	X
+7BH	7EH	POWEROFF	Turns the system power off.		0	0
+7EH	81H	USERBIOS	Registers a user-created BIOS function.		0	0
+81H	84H	AUTOST	Specifies an automatic start string.	The automatic start string function is not supported.	X	X
+84H	87H	RESIDENT	Sets and resets the RESIDENT function.	The RESIDENT function is not supported.	X	X
+87H	8AH	CONTINUE	Sets and resets the CONTINUE flag.	Not discriminated by the keyboard type (standard or item).	0	0
+8AH	8DH	BARCODE	Supports the barcode reader.	Newly aded functions.	$	$
+8DH	90H	TCAM	Sends and receives data through a public line.		$	$
+90H	93H	GRAPHICS	Supports graphic functions.		$	$
+93H	96H	TOUCH	Sets the indication for a touch-panel key block		$	
+96H	99H	ICCARD	Exchanges data with an IC card.		$	$
+99H	9CH	KEYIN	Initiates the system input functions and exchanges data.		$	$

Entry address		Name	Function	Modification for HX-40/PX-4	EHT	
Offset from WBOOT	Absolute address				10	10/2
+9CH	9FH	KANJI	Displays and prints kanji characters.	Newly added functions.	$	$
+9FH	A2H	BACK LIGHT	Controls the backlight software.			$
+A2H	A5H	INFORM	Checks the addresses of the work areas and jump tables used by the system.		$	$

5. DISPLAY CONTROL FUNCTIONS

(1) List of display control functions

- The meanings of the symbols used in the columns of EHT-10 (window and fixed areas) and EHT-10/2 are as follows:

0: This function can be used in the same way as in HX-40 and PX-4, but the parameter range is different from that in HX-40 and PX-4.

@: This function can be used in the same way as in HX-40 and PX-4, but the parameter is different from that in HX-40 and PX-4.

X: This function is not supported in EHT-10/EHT-10/2.

$: This function has been newly added to EHT-10/EHT-10/2.

Code	Function	Remarks	EHT-10		EHT-10/2
			Window area	Fixed area	
02H	SCREEN LEFT	Function deleted	X	X	X
05H	ERASE END OF LINE		0	0	0
06H	SCREEN RIGHT	Function deleted	X	X	X
07H	BELL		0	0	0
08H	BACK SPACE		0	0	0
09H	TAB		0	0	0
0AH	LINE FEED		0	0	0
0BH	HOME		0	0	0
0CH	CLEAR SCREEN & HOME		0	0	0
0DH	CARRIAGE RETURN		0	0	0
10H	SCREEN UP		0	X	0
11H	SCREEN DOWN		0	X	0
1AH	ERASE END OF SCREEN		0	0	0
1BH	ESCAPE	ESC sequence entry	0	0	0
1CH	CURSOR RIGHT		0	0	0
1DH	CURSOR LEFT		0	0	0
1EH	CURSOR UP		0	0	0

Code	Function	Remarks	EHT-10		EHT-10/2
			Window area	Fixed area	
1FH	CURSOR DOWN		0	0	0
ESC '%'	ACCESS CGROM DIRECTLY	Attribute added	@	@	@
ESC '('	BLOCK REVERSE		0	0	0
ESC '*'	CLEAR SCREEN & HOME	Same as code 0CH	0	0	0
ESC '0'	REVERSE ON		0	0	0
ESC '1'	REVERSE OFF		0	0	0
ESC '2'	CURSOR OFF		0	0	0
ESC '3'	CURSOR ON		0	0	0
ESC '='	SET CURSOR POSITION		0	0	0
ESC 'C'	SET CHARACTER SET TABLE		0	0	0
ESC 'P'	SCREEN DUMP	Supported only in EHT-10/2	X	X	0
ESC 'T'	ERASE END OF LINE		0	0	0
ESC 'Y'	ERASE END OF SCREEN		0	0	0
ESC 7BH	SECRET		0	0	0
ESC 7DH	NON SECRET		0	0	0
ESC 90H	PARTIAL SCROLL UP		0	0	0
ESC 91H	PARTIAL SCROLL DOWN		0	0	0
ESC 92H	SCROLL RIGHT N CHARACTERS	Function deleted	X	X	X
ESC 93H	SCROLL LEFT N CHARACTER	Function deleted	X	X	X
ESC 94H	SET SCROLL STEP	Function deleted	X	X	X
ESC 95H	SET SCROLL MODE		0	X	0
ESC 96H	SCROLL UP 1 LINE		0	X	0
ESC 97H	SCROLL DOWN 1 LINE		0	X	0
ESC 98H	SET SCROLL MERGIN	Function deleted	X	X	X

Code	Function	Remarks	EHT-10		EHT-10/2
			Window area	Fixed area	
ESC A0H ESC A1H ESC A2H ESC A3H ESC A4H ESC A5H	KANA LED ON KANA LED OFF ALPH LED ON ALPH LED OFF CALC LED ON CALC LED OFF	Only in EHT-10/2 EHT-10 has not LED	X	X	0
ESC B0H	FUNCTION KEY CHECK MODE ON		0	0	0
ESC B1H	FUNCTION KEY CHECK MODE OFF		0	0	0
ESC D0H	SET SCREEN SIZE	The number of horizontal columns cannot be specified	@	X	@
ESC D1H	CHANGE ACTIVE SCREEN	Newly added	$	$	$
ESC D2H	DIRECT DISPLAY	Attribute added	@	@	@
ESC D4H	LOCATE TOP OF SCREEN		0	X	0
ESC D5H	LOCATE BOTTOM OF SCREEN		0	X	0
ESC D6H	SELECT CURSOR KIND		0	0	0
ESC D7H	FIND CURSOR		0	X	0
ESC D8H	SET WINDOW	Newly added	$	X	X
ESC D9H	SET ATTRIBUTE	Newly added	$	$	X
ESC DAH	SET DISPLAY TYPE	Newly added	$	$	X
ESC E0H	SET DOWNLOAD CHARACTER	Font size is different.	@	@	0
ESC F0H	KEYBOARD REPEAT ON/OFF		0	0	0
ESC F1H	SET KEYBOARD REPEAT START TIME	Function deleted	X	X	X
ESC F2H	SET KEYBOARD REPAET INTERVAL TIME	Function deleted	X	X	X
ESC F3H	SET ARROW KEY CODE	Function deleted	X	X	X
ESC F4H	SET SCROLL KEY CODE	Function deleted	X	X	X
ESC F5H	SET CONTROL KEY CODE	Function deleted	X	X	X

Code	Function	Remarks	EHT-10		EHT-10/2
			Window area	Fixed area	
ESC F6H	CLEAR KEY BUFFER		0	0	0
ESC F7H	SET KEY SHIFT	Function deleted	X	X	X

(2) Details on display control functions

The details on the display control functions listed in the above table are explained. These display control functions are mainly used by BIOS CONOUT and BASIC Print. For how to use these functions for BIOS CONOUT, see the explanation of "CONOUT" in Chapter 4 of the Software Part. For the BASIC Print statement, refer to the BASIC manual.

Code : Function name

05H : ERASE END OF LINE
Clears the cursor and subsequent columns of the line to blanks.

07H : BELL
Sounds the 880-Hz buzzer for one second.

08H : BACK SPACE
Moves the cursor one column to the left on the screen. If the cursor is positioned at the first column, this command moves the cursor at the last column of the previous line. If the cursor is positioned at the Home position on the screen, this command performs no operation.
Notes:
1. If the cursor is to be moved outside the window screen, this command conforms to the follow or unfollow mode.
2. If the cursor is to be positioned inside the fixed screen which is hidden by the window screen, the cursor cannot be viewed on the LCD screen.

09H : TAB
Sets the cursor at the first tab position to the right of the current cursor position. If no tab position is detected on the current cursor line, this command sets the cursor at the first tab position on the next line. If the cursor is to be positioned outside the screen, this command sets the cursor at the first tab position on the last line on the screen.

Tab position = Column (1 + 8n)
n = 0, 1, 2, ...

See Notes 1 and 2 on Back Space (08H).

0AH : LINE FEED
Moves the cursor one line downward on the screen. If the cursor is positioned on the last line of the window screen, the window screen is scrolled down one line. If the cursor is positioned on the last line of the screen, the screen is scrolled up one line.
Note:
If the cursor is positioned on the last line of the fixed screen, this command performs no operation.

0BH : HOME
Moves the cursor at the home position on the screen. See Notes 1 and 2 on Back Space (08H).

0CH : CLEAR SCREEN & HOME
Clears all the screen contents to blanks and performs home(0BH) processing.

0DH : CARRIAGE RETURN

Moves the cursor at the first column of the line.

Notes:

1. If the cursor is to be positioned outside the window screen, this command conforms to the follow or unfollow mode.
2. If this command is entered immediately after one character is displayed at the last column of the screen, the cursor is moved at the first column of the previous line (the line on which the last one character was displayed).

10H : SCREEN UP

Shifts up the window screen by one screen. If part of the shifted window screen is positioned over the home position, only the part of the window screen on the home and subsequent lines is displayed.

Note:

The cursor remains at the original position on the screen.

11H : SCREEN DOWN

Shifts down the window screen by one screen. If the shifted window screen contents are to overflow the screen, the screen is displayed so that the last line of the screen matches the last line of the window screen. See Note 1 on Screen Up (10H).

1AH : ERASE END OF SCREEN

Clears all screen contents starting from the cursor position to blanks.

1BH : ESCAPE

Enables ESC sequence acceptance.

1CH : CURSOR RIGHT

Moves the cursor one column to the right on the screen. If the cursor is positioned at the last column of the line, this command moves the cursor at the first column of the next screen. If the cursor is positioned at the last column of the screen, this command performs no operation. See Notes 1 and 2 on Back Space (08H).

1DH : CURSOR LEFT

Same as Back Space (08H)

1EH : CURSOR UP

Moves the cursor one line upward on the screen. If the cursor is positioned on the first line, this command performs no operation. See Notes 1 and 2 on Back Space (08H).

1FH : CURSOR DOWN

Moves the cursor position one line downward on the screen. If the cursor is positioned on the last line of the screen, this command performs no operation. See Notes 1 and 2 on Back Space (08H).

ESC '%' : ACCESS CG ROM DIRECTLY

Reads the character corresponding to the specified code from the character generator and displays it at the current cursor position on the screen. The cursor then moves at the next column.

1	ESC	n: Code ($0 \leq n \leq 255$)
2	'%'	
3	n	
4	m	m: Attributes (*1)

*1 The attribute-byte configuration is as follows:

7	6	5	4	3	2	1	0
0	0			*	*	*	*

-> Horizontal ruler line (top)
-> Horizontal ruler line (bottom)
-> Vertical ruler line
-> Comma
-> Reverse
-> Secret
(1:Specified   0:Not specified)

* : Can be specified only in EHT-10 vertical display mode

In EHT-10 (horizontal display mode) and EHT-10/2, only the reverse and secret attributes are made effective. See Section 4.4 for details.

ESC '(' : BLOCK REVERSE

Inverts and displays the specified length of the window screen contents starting from the specified position.

1	ESC	
2	'('	
3	Y	
4	X	
5	n(H)	
6	N(L)	

	EHT-10(Vertical)	EHT-10(Horizontal)	EHT-10/2
Y: Y axis	1 to 14	1 to 10	1 to 4
X: X axis	1 to 12	1 to 25	1 to 20
n:No. of reversed characters	1 to 168	1 to 250	1 to 80

Note:

If the screen is scrolled and the inverted data overflows the window screen in EHT-10/2, inversion is released.

ESC '*' : CLEAR SCREEN & HOME

Same as Clear Screen & Home (0CH)

ESC '0' : REVERSE ON

Turns the reverse display mode on, causing subsequent characters to be inverted and displayed. See Note 1 on Block Reverse.

ESC '1' : REVERSE OFF

Turns the reverse display mode off.

ESC '2' : CURSOR OFF

Does not display the cursor.

ESC '3' : CURSOR ON

Displays the cursor.

ESC '=' : SET CURSOR POSITION

Moves the cursor at the specified position on the screen.

Note:

If the cursor is to be positioned outside the window screen, this command performs different processing according to the follow or unfollow mode.

Follow mode: When the cursor moves upward, this command moves the cursor on the first line of the window. When the cursor moves downward, this command moves the cursor on the last line of the window.

Unfollow mode: The window remains at the same position.

1	ESC
2	'='
3	m+1FH
4	N+1FH

Vertical direction: $1 \le m \le$ maximum screen line
Horizontal direction: $1 \le n \le$ maximum screen column

ESC 'C' : SET CHARACTER SET TABLE

Specifies the character set of the specified country.

1	ESC
2	'C'
3	ID

ID: Country identification character

J: Japan	D: Denmark
U: USA (ASCII)	W: Sweden
F: France	I: Italy
G: Germany	S: Spain
E: Great Britain	N: Norway

Notes:
1. Default value Japan or USA (ASCII) is specified by the DIP switch.

2. The characters that have already been output remain unchanged even if the corresponding codes indicate different characters in the newly-specified character set.

ESC 'P' : SCREEN DUMP
Outputs the VRAM data being displayed to the printer.

Note:
In EHT-10, this command performs no operation.

ESC 'T' : ERASE END OF LINE
Same as Erase End of Line (05H)

ESC 'Y' : ERASE END OF SCREEN
Same as Erase End of Screen (1AH)

ESC 78H : SECRET
Sets the character output mode to secret.
Notes:
1. In EHT-10, characters are output in secret mode.

2. In EHT-10/2, spaces are displayed. If the screen is scrolled and the secret data overflows the window screen and then returns inside the window screen, the character data stored in the screen buffer is displayed.

ESC 7DH : NON SECRET
Releases the secret mode.

ESC 90H : PARTIAL SCROLL UP
Scrolls up the m lines starting from line n by one line. Line (n + m -1) becomes a blank line.

1	ESC
2	90H
3	n-1
4	m

n: Scroll start line
$1 \leq n \leq$ maximum number of screen lines
m: Scroll width
$1 \leq m \leq$ maximum number of screen lines
Notes:
1. If value (n + m) exceeds the maximum number of screen lines, value m is automatically adjusted so that value (n + m -1) becomes the maximum number of screen lines.

2. The cursor remains at the original position on the screen.

ESC 91H : PARTIAL SCROLL DOWN

Scrolls down m lines starting from line n by one line on the screen. Line n becomes a blank line.

1	ESC
2	91H
3	n-1
4	m

n: Scroll start line

$1 \le n \le$ maximum number of screen lines

m: Scroll width

$1 \le m \le$ maximum number of screen lines

See Notes 1 and 2 on Partial Scroll Up.

ESC 95H : SET SCROLL MODE

Specifies whether automatic scrolling is to be performed.

1	ESC
2	95H
3	m

m: Mode

=0: Follow mode (default)

=1: Unfollow mode

Note:

In follow mode, the screen is automatically scrolled according to the cursor movement. In unfollow mode, the screen is not automatically scrolled according to the cursor movement.

ESC 96H : SCROLL UP 1 LINE

Scrolls up the window screen one line.

Notes:

1. If part of the window screen is to be positioned outside the screen, this command performs no operation.

2. The cursor remains at the original position on the screen.

ESC 97H : SCROLL DOWN 1 LINE

Scrolls down the window screen one line. See Notes 1 and 2 on Scroll Up 1 Line.

ESC A0H : KANA LED ON

Turns kana LEDs on (only effective in EHT-10/2).

ESC A1H : KANA LED OFF

Turns kana LEDs off (only effective in EHT-10/2).

ESC A2H : CALC LED ON
Turns the calculator (CALC) LEDs on (only effective in EHT-10/2).

ESC A3H : CALC LED OFF
Turns the calculator (CALC) LEDs off (only effective in EHT-10/2).

ESC A4H : ALPH LED ON
Turns the alphabetic (ALPH) LEDs on (only effective in EHT-10/2).

ESC A5H : ALPH LED OFF
Turns the alphabetic (ALPH) LEDs off (only effective in EHT-10/2).

ESC B0H : FUNCTION KEY CHECK MODE ON
Sets a mode in which, if a function key is pressed, the function assigned to the key is not executed but the code discrete to the key is returned. (YPFCMFLG = FFH)
Note:
See the explanation of CONIN for the information obtained when a key is pressed in this mode.

ESC B1H : FUNCTION KEY CHECK MODE OFF
Releases the function key check mode. (YPFCMFLG = 00H)

ESC C0H : SET SCREEN SIZE
Sets the virtual screen size.

1	ESC
2	D0H
3	n

n :	EHT-10 (Vertical)	EHT-10 (Horizontal)	EHT-10/2
Lines n of Screen	14 to 42	10 to 20	4 to 25

If the virtual screen size is specified, an area of the specified size is allocated and Clear Screen & Home (0CH) processing is performed.
Notes:
1. If the virtual screen size does not exceed 28 lines in EHT-10 (vertical display mode), the fixed screen area is allocated, enabling the fixed screen to be used.
2. Set Screen Size cannot be executed on the fixed screen.
3. Even if the virtual screen size is modified, the CP/M size remains unchanged.

ESC D1H : CHANGE ACTIVE SCREEN
Specifies an effective user screen type.

1	ESC
2	D1H
3	n

n : Screen type

	EHT-10 (Vertical)	EHT-10/2
n=0	Scroll screen	User screen 1
n=1	Fixed screen	User screen 2

Notes:

1. In EHT-10 (horizontal display mode), Change Active Screen is made ineffective. If a virtual screen (scroll screen) size not exceeding 28 lines is specified in Set Screen Size and the fixed screen area has not been allocated in EHT-10 (vertical display mode), Change Active Screen is made ineffective.

2. If Change Active Screen is executed, cursor movement and displaying of one character on the specified screen are made possible. If the user screen is changed in EHT-10/2, the old screen contents are saved, making each CONOUT code effective on the new screen. (Two screens can be used independently. See Section 4.4 for details.)

ESC D2H : DIRECT DISPLAY

Outputs a character at the specified position in VRAM. The character can be directly output at any position on the LCD screen.

			EHT-10(Vertical)	EHT-10(Horizontal)	EHT-10/2
1	ESC	n			
2	D2H	Y: Vertical position	1 to 14	1 to 10	1 to 4
3	Y	X: Horizontal position	1 to 12	1 to 25	1 to 20
4	X				
5	n	n: Character code $00H \le n \le FFH$			
6	m	m: Attributes			

Notes:

1. A character code corresponding to the character generator must be specified.

2. See Note 1 on Access CGROM Directly (ESC+"%") for attributes.

ESC D4H : LOCATE TOP OF SCREEN

Moves the window screen at the top of the screen. The cursor remains at the original position.

ESC D5H : LOCATE END OF SCREEN

Moves window screen at the bottom of the screen. The cursor remains at the original position.

ESC D6H : SELECT CURSOR KIND

Selects the cursor type.

1	ESC
2	D6H
3	n

n: Cursor type
- = 0: Block & blink (default)
- = 1: Block & nonblink
- = 2: Underline & blink
- = 3: Underline & nonblink

ESC D7H : FIND CURSOR

Moves the window screen at the cursor position so that the cursor is positioned on the first window screen line. If the cursor is positioned on the LCD screen, this command performs no operation.

ESC D8H : SET WINDOW

Specifies the position and size of the window screen in EHT-10 (vertical display mode). The cursor remains at the original position on the screen.

1	ESC
2	D8H
3	m
4	n

m: Window start line ($1 \leq m \leq 14$)
n: Window end line ($1 \leq n \leq 14$)

Notes:
1. Value m must not be greater than value n.

2. If the window screen is modified, graphic data is lost.

3. Set Window is only effective in EHT-10 (vertical display mode). This command can change the fixed screen display area (screen part that can be viewed on the LCD screen) by changing the position and size of the window screen.

ESC D9H : SET ATTRIBUTE

Sets character attributes. If this command is entered, the subsequent one character is displayed with the specified attributed.

1	ESC
2	D9H
3	n

n: Attributes (0: Not specified 1: Specified)

7	6	5	4	3	2	1	0
0	0			*	*	*	*

- bit 0 —> Horizontal ruler line (top)
- bit 1 —> Horizontal ruler line (bottom)
- bit 2 —> Vertical ruler line
- bit 3 —> Comma
- bit 4 —> Reverse
- bit 5 —> Secret

Notes:

1. In EHT-10 (horizontal display mode) and EHT-10/2, only reverse and secret attributes are made effective. If the data displayed with these attributes overflows the window screen in EHT-10/2, the specified attributes of the overflowed part of the data are made ineffective.

2. The default value for n is 0.

ESC DAH : SET DISPLAY TYPE

Specifies the character display type in EHT-10. After the character display type is changed, Clear Screen & Home (0CH) must be executed.

1	ESC
2	DAH
3	n

n: Display type
=0: Vertical display (default)
=1: Horizontal display

ESC E0H : SET DOWNLOAD CHARACTER

Defines an external character from E0H to FFH.

EHT-10 (vertical)

1	ESC
2	E0H
3	n
4	p(1)
5	p(2)
6	p(3)
7	p(4)
8	p(5)
9	p(6)
10	p(7)
11	p(8)
12	p(9)
13	p(10)
14	p(11)

EHT-10 (horizontal) or EHT-10/2

1	ESC
2	E0H
3	n
4	p(1)
5	p(2)
6	p(3)
7	p(4)
8	p(5)
9	p(6)
10	p(7)
11	p(8)

n: Character code (E0H ≤ n ≤ FFH)

p(1) to p(11) and p(1) to p(8): Font patterns

Note 1:

An external character pattern of 7 x 11 dots must be specified for EHT-10 (vertical display mode) or that of 6 x 8 dots must be specified for EHT-10 (horizontal display mode) or EHT-10/2.

7 x 11 dot pattern

	7	6	5	4	3	2	1	0
p(1)	0							
p(2)	0							
p(3)	0							
:	:							
p(10)	0							
p(11)	0							

6 x 8 dot pattern

	7	6	5	4	3	2	1	0
p(1)	0	0						
p(2)	0	0						
p(3)	0	0						
:	:	:						
p(7)	0	0						
p(8)	0	0						

Note 2:

For the Japan version, default font patterns have been defined for external characters E0H to E3H.

Example: To register the following pattern in E4H, perform as shown below.

	7	6	5	4	3	2	1	0	
p(1)	0								00H
p(2)	0	*	*	*	*	*	*	*	7FH
p(3)	0								00H
p(4)	0	*	*	*	*	*	*	*	7FH
p(5)	0				*				08H
p(6)	0				*				08H
p(7)	0				*				08H
p(8)	0				*				08H
p(9)	0				*				08H
p(10)	0				*				08H
p(11)	0				*				08H

7 x 11 Pattern

	Data
1	ESC
2	E0H
3	E4H
4	00H
5	7FH
6	00H
7	7FH
8	08H
9	08H
10	08H
11	08H
12	08H
13	08H
14	08H

	7	6	5	4	3	2	1	0	
p(1)	0	0							00H
p(2)	0	0		*	*	*	*	*	1FH
p(3)	0	0							00H
p(4)	0	0		*	*	*	*	*	1FH
p(5)	0	0				*			04H
p(6)	0	0				*			04H
p(7)	0	0				*			04H
p(8)	0	0				*			04H

6 x 8 Pattern

	Data
1	ESC
2	E0H
3	E4H
4	00H
5	1FH
6	00H
7	1FH
8	04H
9	04H
10	04H
11	04H

ESC F0H : KEYBOARD REPEAT ON/OFF

Controls the repeat function of keyboard keys (including the touch-panel keys).

1	ESC
2	F0H
3	n

n: Switch

=0: Repeat function ON

=1: Repeat function OFF (default)

ESC F6H : CLEAR KEY BUFFER

Clears the key input data buffer contents and deletes the pre-hit keys.

7. BDOS FUNCTION LIST

- The interface for calling BDOS in EHT-10/EHT-10/2 is the same as that for calling BDOS by CP/M. See Chapter 5 in Software Part for details.

- This table provides brief explanation of each BDOS function. Refer to the relevant CP/M manual for more details.

Number	Function Name	Input	Output
0	System Reset	C : 00H	None
1	Console Input	C : 01H	A : Input char
2	Console Output	C : 02H E : Output char	None
3	Reader Input	C : 03H	A : Input char
4	Punch Output	C : 04H E : Output char	None
5	List Output	C : 05H E : Outpur char	None
6	Direct Console I/O	C : 06H E : 0FFH (input) : Output char (output)	A : Input char (input) : None
7	Get IOBYTE	C : 07H	A : IOBYTE
8	Set IOBYTE	C : 08H E : IOBYTE	None
9	Print String	C : 09H DE : Address at which the string is stored.	None
10	Read Console Buffer	C : 0AH DE : Buffer address	Loads the buffer with entry from the console.
11	Get Console Status	C : 0BH	A : Console Status
12	Get Version Number	C : 0CH	HL : Version Number
13	Reset Disk System	C : 0DH	None
14	Select Disk	C : 0EH E : Disk number	None
15	Open File	C : 0FH DE : FCB address	A : Directory code

Number	Function Name	Input	Output
16	Close File	C : 10H DE : FCB address	A : Directory code
17	Search for First	C : 11H DE : FCB address	A : Directory code
18	Search for Next	C : 12H	A : Directory code
19	Delete File	C : 13H DE : FCB address	A : Directory code
20	Read Sequential	C : 14H DE : FCB address	A : Return code
21	Write Sequential	C : 15H DE : FCB address	A : Return code
22	Create File	C : 16H DE : FCB address	A : Directory code
23	Rename File	C : 17H DE : FCB address	A : Directory code
24	Get Login Vector	C : 18H	HL : Login vector
25	Get Disk Number	C : 19H	A : Disk Number
26	Set DMA Address	C : 1AH DE : DMA address	None
27	Get Allocation Address	C : 1BH	HL : Allocation address
28	Write Protect Disk	C : 1CH	None
29	Get R/O Vector	C : 1DH	HL : R/O vector
30	Set File Attributes	C : 1EH DE : FCB address	A : Directory code
31	Get DPB Address	C : 1FH	HL : DPB address
32	Set/get User Code	C : 20H E : 0FFH (Get) : User code (Set)	A : None (Set) : User code (Get)
33	Read Random	C : 21H DE : FCB address	A : Return code
34	Write Random	C : 22H DE : FCB address	A : Return code
35	Compute File Size	C : 23H DE : FCB address	FCB r0, r1, r2
36	Set Random Number	C : 24H DE : FCB address	FCB r0, r1, r2

Number	Function Name	Input	Output
37	Reset Disk Drive	C : 25H DE : Drive vector	None.
38			
39			
40	Random Write with Zero File	C : 28H DE : FCB address	A : Return code
251	Verify File	C : 0FBH DE : T～FCB	
252	Remove Tape	C : 0FCH	
253	Mount Tape	C : 0FDH	
254	Read Tape ID	C : 0FEH	
255	Create Tape Directory	C : 0FFH DE : T～FCB	

8. FILINK PROTOCOL

Establish a Link

Sender		Receiver
(1)-Send "R".	→	-Send "S" if "R" is received. -Send received data if data other than "R" is received
(≠"S")	(="S") ←	
(2)-Send "G" if "S" is received. Proceed to "Send File Name." -Go to (1) if other than "S" is received.	→	-Proceed to "receive File Name" if "G" is received. -Wait if other than "G" is received.

Send/Receive File Name

Sender		Receiver
(3)- Send 04H.	→	-Send 08H if 04H is received. -Terminate FILINK if 13H is received. -Otherwise, send "X" and wait (also display data).
(≠08H)	←	(≤1FH)
(4)-Go to (5) if 08H is received. -Go to (3) if other than 08H is received.		
(5)-Send file name and extension, one character at a time.	→	-Send received character. -If received character is 1FH or smaller, send "X" and return to step (3) for receiver.
(6)-Compare received character and send character transmitted in (5). If match→Go to (5) to send next character. If no match→Display "?" and go to (3).	←	

Repeat steps (5) and (6) 11 times to send/receive file name and extension.

(7)-Send 05H.

-Send 09H and proceed to "Send/Receive Data" if 05H is received.
-Send "X" and return to step (3) for receiver if other than 05H is received.

(8)-Proceed to "Send/Receive Data" if 09H is received.
-Go to (3) if other than 09H is received.

Send/Receive Data

Sender	Receiver
(9) -Read 1 record (128 bytes). -If not EOF, send 02H. -If EOF, send 03H. If all files have been sent, send 13H to terminate FILINK. Return to (3) if there is a file to be sent.	-Send "P" and go to (11) if 02H is received. -Go to step (3) for receiver if 03H is received. -Otherwise, send "N" and wait.
(10) -Go to (11) if "P" is received. -Otherwise, go to (9).	
(11) -Send 1 record (128 bytes) and check byte.	-Receive 1 record (128 bytes) and check byte.
(12)	-Compare check bytes. If match→Send "G", writes received record into a file, and return to (9). -If no match→Send "B" and return to (9).
(13) -Read next record and return to (9) if "G" is received. -Return to (9) for a retry if "B" is received. -Otherwise, wait here.	

9. SAMPLE PROGRAM LISTS

(1) SAMPLE 1. BIOS CALL

```
                      ;      ****************************************************
                      ;              BIOS CALL SAMPLE PROGRAM
                      ;      ****************************************************
                      ;
                      ;      NOTE:
                      ;              This sample program is consist of BIOS calling routine.
                      ;              So,this program doesn't run by itself.
                      ;
                      ;      <> assemble condition     <>
                      ;
                             .Z80
                      ;
                      ;      <> loading address        <>
                      ;
                             .PHASE 100h
                      ;
                      ;      <> constant values        <>
                      ;
0000                  BBIOS1         EQU     000000H
E803                  BBIOS2         EQU     0E803H
                      ;
                      ;      ****************************************************
                      ;              BIOS CALL ROUTINE
                      ;      ****************************************************
                      ;
                      ;      NOTE:
                      ;              This routine is used for calling BIOS.
                      ;
                      ;      <> entry parameter        <>
                      ;
                      ;              A:BIOS offset address from WBOOT
                      ;              Depending on each BIOS function.
                      ;
                      ;      <> return parameter       <>
                      ;
                      ;              Depending on each BIOS function.
                      ;
                      ;      <> preserved parameters <>
                      ;
                      ;              IY is used by calling address.
                      ;
                      ;      CAUTION:
                      ;              If your program is a 80H based program, you must use
                      ;              BBIOS2.
                      ;
0100                  BIOS:
0100    E5                   PUSH    HL                   ;Save registers.
0101    D5                   PUSH    DE                   ;
0102    2A 0001              LD      HL,(BBIOS1+1)        ;Get WBOOT entry address.
0105    5F                   LD      E,A                  ;Function code.
0106    16 00                LD      D,00H                ;Get target BIOS function entry address.
0108    19                   ADD     HL,DE                ;
0109    E5                   PUSH    HL                   ;Set target address to IY register.
010A    FD E1                POP     IY                   ;
010C    D1                   POP     DE                   ;Restore registers.
010D    E1                   POP     HL                   ;
010E    FD E9                JP      (IY)                 ;Jump to the target BIOS function
                      ;
                      ;
                             END
```

(2) SAMPLE 2. CONIN CONOUT

```
                                ;       ***********************************************************
                                ;               BIOS CONIN CONOUT CONST
                                ;                       READSW TOUCH KEYIN SAMPLE PROGRAM
                                ;       ***********************************************************
                                ;
                                ;       NOTE:
                                ;               This sample program uses CONIN CONOUT CONST
                                ;               KEYIN READSW and displays 1 character on the LCD.
                                ;
                                ;       <> assemble condition   <>
                                ;
                                        .Z80
                                ;
                                ;       <> loading address      <>
                                ;
                                        .PHASE  100H
                                ;
                                ;       <> constant values      <>
                                ;
EB03                            WBOOT           EQU     0EB03H
EB06                            CONST           EQU     0EB06H
EB09                            CONIN           EQU     0EB09H
EB0C                            CONOUT          EQU     0EB0CH
EB72                            READSW          EQU     0EB72H
EB96                            TOUCH           EQU     0EB96H
EB9C                            KEYIN           EQU     0EB9CH
                                ;
1000                            MAINSP          EQU     01000H
                                ;
                                ;       ***********************************************************
                                ;               MAIN PROGRAM
                                ;       ***********************************************************
                                ;
0100                            START:
0100    31 1000                         LD      SP,MAINSP       ; Set stack pointer
0103    0E 02                           LD      C,02H           ; Read dip SW of EHT-10,10/2
0105    CD EB72                         CALL    READSW          ; Read
0108    E6 80                           AND     10000000B       ; Check the bit 7
010A    FE 80                           CP      10000000B       ; If bit 7 = 1 then the machine is
                                                                ; EHT-10/2
010C    28 1A                           JR      Z,KEYBOD        ; Jump to EHT-10/2 routine
                                ;
010E                            TCHKEY:                         ; else EHT-10
010E    21 013E                         LD      HL,INPDAT       ; Make input-data area for EHT-10
0111    06 01                           LD      B,01H           ; Set the Alphabet input mode
0113    CD EB9C                         CALL    KEYIN           ; Input the Alphabet
0116    21 013E                         LD      HL,INPDAT       ;
0119    4E                              LD      C,(HL)          ;
011A    CD EB0C                         CALL    CONOUT          ; display the inputted data (1 character)
011D    0E 01                           LD      C,01H           ; Make 1 TOUCH key block
011F    11 015E                         LD      DE,KEYBLK       ; Set the key block data
0122    CD EB96                         CALL    TOUCH           ; Display a key block
0125    C3 0134                         JP      NEXT            ;
                                ;
0128                            KEYBOD:                         ; EHT-10/2
0128    06 01                           LD      B,01H           ; Change the keyboard mode to Alphabet
012A    CD EB9C                         CALL    KEYIN           ;
012D    CD EB09                         CALL    CONIN           ; Read the data from CON:
0130    4F                              LD      C,A             ;
0131    CD EB0C                         CALL    CONOUT          ; Display the data
                                ;
0134                            NEXT:
0134    CD EB06                         CALL    CONST           ; Check another key input
0137    FE 00                           CP      00H             ; If any key is pressed, WBOOT
0139    28 F9                           JR      Z,NEXT          ; else waiting
013B    CD EB03                         CALL    WBOOT           ;
                                ;
                                ;       ***********************************************************
                                ;               DATA BUFFER FOR EHT-10     32bytes
                                ;       ***********************************************************
                                ;
013E                            INPDAT:
013E    00 00 00 00                     DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

0142    00 00 00 00
0146    00 00 00 00
014A    00 00 00 00
014E    00 00 00 00                     DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0152    00 00 00 00
0156    00 00 00 00
015A    00 00 00 00
                                ;
                                ;       ***********************************************************
                                ;               DATA OF KEY BLOCK DESCRIPTOR
                                ;       ***********************************************************
                                ;
015E                            KEYBLK:
015E    04 0D 02 01                     DB      4,13,2,1,20H,07H,0,'END'
0162    20 07 00 45
0166    4E 44
                                ;
                                ;
                                        END
```

(3) SAMPLE 3. PUNCH READER LIST

```
                        ;       ***************************************************************
                        ;               BIOS LIST PUNCH READER INFORM SAMPLE PROGRAM
                        ;       ***************************************************************
                        ;
                        ;       NOTE:
                        ;               This program receives 1 character data from RS-232C
                        ;               (READER) , PUNCH the data on the LCD, and LIST the data
                        ;               to the printer unit.
                        ;
                        ;       <> assemble condition <>
                        ;
                                .Z80
                        ;
                        ;       <> loading address <>
                        ;
                                .PHASE 100H
                        ;
                        ;       <> constant values <>
                        ;
EB03                    WBOOT           EQU     0EB03H
EB0C                    CONOUT          EQU     0EB0CH
EB0F                    LIST            EQU     0EB0FH
EB12                    PUNCH           EQU     0EB12H
EB15                    READER          EQU     0EB15H
EB2D                    LISTST          EQU     0EB2DH
EBA5                    INFORM          EQU     0EBA5H
                        ;
0003                    IOBYTE          EQU     00003H
1000                    MAINSP          EQU     01000H
                        ;
                        ;       ***************************************************************
                        ;               MAIN PROGRAM
                        ;       ***************************************************************
                        ;
0100                    START:
0100    31 1000                 LD      SP,MAINSP               ; Set the stack pointer
0103    21 0003                 LD      HL,IOBYTE               ;
0106    36 59                   LD      (HL),59H                ; Change the PUN: device to the LCD
0108    0E 05                   LD      C,05H                   ; Set the parameter of INFORM
010A    CD EBA5                 CALL    INFORM                  ; Look the LSTERR address
010D    22 0150                 LD      (LSTERRAD),HL           ; Save LSTERR address
                        ;
0110                    LOOP:
0110    CD EB15                 CALL    READER                  ; Read 1 character from RS-232C
0113    4F                      LD      C,A                     ; Set the PUNCH parameter
0114    F5                      PUSH    AF                      ; Preserve the register
0115    CD EB12                 CALL    PUNCH                   ; Display the read character on the LCD
0118    CD EB2D                 CALL    LISTST                  ; Look the printer status
011B    2A 0150                 LD      HL,(LSTERRAD)           ; Load LSTERR address
011E    7E                      LD      A,(HL)                  ; Load the content of LSTERR
011F    FE 00                   CP      0                       ; Printer connected ?
0121    20 08                   JR      NZ,DSPMSG               ; write error message if printer not
                                                                ; connected
                        ;
0123    F1                      POP     AF                      ;
0124    4F                      LD      C,A                     ; Set the LIST parameter
0125    CD EB0F                 CALL    LIST                    ; Print the read character
0128    C3 0110                 JP      LOOP                    ;
                        ;
012B                    DSPMSG:
012B    21 0140                 LD      HL,ERRMSG               ; Display the error message
                        ;
012E                    DSP10:
012E    4E                      LD      C,(HL)                  ;
012F    79                      LD      A,C                     ; If data is 0
0130    FE 00                   CP      0                       ; Then the end of data
0132    28 09                   JR      Z,ENDDAT                ;
0134    E5                      PUSH    HL                      ; Preserve register
0135    CD EB0C                 CALL    CONOUT                  ; Display 1 character of message
0138    E1                      POP     HL                      ;
0139    23                      INC     HL                      ; Next data
013A    C3 012E                 JP      DSP10                   ;
                        ;
013D                    ENDDAT:
013D    C3 EB03                 JP      WBOOT                   ;
                        ;
                        ;       ***************************************************************
                        ;               ERROR MESSAGE
                        ;       ***************************************************************
                        ;
0140                    ERRMSG:
0140    50 72 69 6E             DB      'Printer ERROR',0DH,0AH
0144    74 65 72 20
0148    45 52 52 4F
014C    52 0D 0A
014F    00                      DB      00H
```

```
                    ;
                    ;       ************************************************************
                    ;               LSTERR address
                    ;       ************************************************************
                    ;
0150                LSTERRAD:
0150                        DS      2

                            END
```

(4) SAMPLE 4. DISK ACCESS

```
                         ;        ****************************************************
                         ;                BIOS SAMPLE PROGRAM:
                         ;                        SELDSK,SETTRK,SETSEC,SETDMA,READ,WRITE
                         ;        ****************************************************
                         ;
                         ;        NOTE:
                         ;                This sample program is executed only on EHT-10.
                         ;                This is diskcopy program from D: to E:.
                         ;
                         ;        <> assemble condition    <>
                         ;
                                  .Z80
                         ;
                         ;        <> loading address       <>
                         ;
                                  .PHASE  100H
                         ;
                         ;        <> constant values       <>
                         ;
EB03                     WBOOT            EQU     0EB03H
EB09                     CONIN            EQU     0EB09H
EB0C                     CONOUT           EQU     0EB0CH
EB1B                     SELDSK           EQU     0EB1BH
EB1E                     SETTRK           EQU     0EB1EH
EB21                     SETSEC           EQU     0EB21H
EB24                     SETDMA           EQU     0EB24H
EB27                     READ             EQU     0EB27H
EB2A                     WRITE            EQU     0EB2AH
EB96                     TOUCH            EQU     0EB96H
                         ;
1000                     MAINSP           EQU     01000H
0028                     TRKMAX           EQU     40         ; Floppy disk
0040                     SECMAX           EQU     64         ;
                         ;
                         ;        ****************************************************
                         ;                MAIN PROGRAM
                         ;        ****************************************************
                         ;
0100                     START:
0100     31 1000                  LD      SP,MAINSP          ; Set stack pointer
                         ;
0103     0E 1B                    LD      C,1BH              ; Cursor off
0105     CD EB0C                  CALL    CONOUT             ; ESC + '2'
0108     0E 32                    LD      C,'2'              ;
010A     CD EB0C                  CALL    CONOUT             ;
                         ;
010D     21 0000                  LD      HL,00H             ; A=0
0110     22 022B                  LD      (TRKNUM),HL        ; Set first track number
0113     22 022D                  LD      (SECNUM),HL        ; Set first sector number
0116     3E 03                    LD      A,03H              ; Set drive number (D:)
0118     32 022A                  LD      (DSKNUM),A         ;
                         ;
011B                     LOOP:
011B     3A 022A                  LD      A,(DSKNUM)         ; Select disk
011E     4F                       LD      C,A                ;
011F     1E 00                    LD      E,00B              ; First disk access
0121     CD EB1B                  CALL    SELDSK             ;
0124     7C                       LD      A,H                ; Error check
0125     B5                       OR      L                  ; If HL = 0 then error
0126     3E F9                    LD      A,0F9H             ; Set error code
0128     28 53                    JR      Z,DSKERR           ;
                         ;
012A     ED 4B 022B               LD      BC,(TRKNUM)        ; Set current track number
012E     CD EB1E                  CALL    SETTRK             ; No.  then read this track
                         ;
0131     ED 4B 022D               LD      BC,(SECNUM)        ; Set current sector number
0135     CD EB21                  CALL    SETSEC             ; Set sector
                         ;
0138     01 022F                  LD      BC,DMA             ; Set DMA address
013B     CD EB24                  CALL    SETDMA             ;
                         ;
013E     3A 022A                  LD      A,(DSKNUM)         ; Check read /write
0141     FE 03                    CP      03H                ; If DSKNUM=3 then READ
0143     20 05                    JR      NZ,WRITES          ;             else WRITE
0145     CD EB27                  CALL    READ               ; Read 128byte
0148     38 05                    JR      C,ERRCHK           ;
014A                     WRITES:
014A     0E 01                    LD      C,01H              ; No blocking write
014C     CD EB2A                  CALL    WRITE              ; Write 128 byte
014F                     ERRCHK:
014F     B7                       OR      A                  ; Check return parameter
0150     20 38                    JR      NZ,DSKERR          ; If A<>0 then error
                         ;
0152     3A 022A                  LD      A,(DSKNUM)         ; Check read / write
0155     EE 07                    XOR     00000111B          ; If DSKNUM=3 then DSKNUM=4
0157     32 022A                  LD      (DSKNUM),A         ;             else DSKNUM=3
015A     FE 04                    CP      04H                ; Write cycle ?
015C     28 BD                    JR      Z,LOOP             ; Yes. then write
015E     2A 022D                  LD      HL,(SECNUM)        ; Update sector number
0161     2C                       INC     L                  ;
0162     7D                       LD      A,L                ;
0163     FE 40                    CP      SECMAX             ; Check sector number
```

```
0165      20 10                       JR      NZ,NEXT         ;
0167      2A 022B                     LD      HL,(TRKNUM)     ; Update track number
016A      2C                          INC     L
016B      7D                          LD      A,L             ;
016C      FE 29                       CP      TRKMAX          ; Check track number
016E      CA EB03                     JP      Z,WBOOT         ; If over track max ,then WBOOT
0171      22 022B                     LD      (TRKNUM),HL     ; Set new track number
0174      21 0000                     LD      HL,00H          ; A=0  for sector number
0177                        NEXT:
0177      22 022D                     LD      (SECNUM),HL     ; Set new sector number
017A      C3 0118                     JP      LOOP            ; Read / write again
                            ;
017D                        DSKERR:
017D      D6 F9                       SUB     0F9H            ; Change error code 0,1,2,3,4,5,6
017F      5F                          LD      E,A             ; Set error code to DE
0180      CB 23                       SLA     E               ; E = E * 2
0182      16 00                       LD      D,00H           ;
0184      21 01B3                     LD      HL,ERRTBL       ; Set error message table
0187      19                          ADD     HL,DE           ; Get error table address
0188      5E                          LD      E,(HL)          ; Get low address
0189      23                          INC     HL              ;
018A      56                          LD      D,(HL)          ; Get high address
018B      EB                          EX      DE,HL           ; HL = message address
018C      CD 01A7                     CALL    DSPMSG
                                                              ;
018F      0E 01                       LD      C,01H           ; Make 1 touch key
0191      11 0220                     LD      DE,KEYBLK       ; Set keyblock
0194      CD EB96                     CALL    TOUCH           ; Make 'END' key
0197      CD EB09                     CALL    CONIN           ; Wait key in
                            ;
019A      0E 1B                       LD      C,1BH           ; Cursor on
019C      CD EB0C                     CALL    CONOUT          ; ESC + '3'
019F      0E 33                       LD      C,'3'           ;
01A1      CD EB0C                     CALL    CONOUT          ;
01A4      C3 EB03                     JP      WBOOT           ;
                            ;
                            ;         ***********************************************************
                            ;                 DISPLAY MESSAGE STRING
                            ;         ***********************************************************
                            ;
                            ;         NOTE:
                            ;
                            ;         <> entry parameter        <>
                            ;                 HL : Top address of message string
                            ;         <> return parameter       <>
                            ;                 NONE
                            ;         <> preserved registers    <>
                            ;                 NONE
                            ;
01A7                        DSPMSG:
01A7      7E                          LD      A,(HL)          ; Get a data
01A8      B7                          OR      A               ; Check end mark 00H
01A9      C8                          RET     Z               ; If data is 00H then return
                            ;
01AA      E5                          PUSH    HL              ; Save pointer
01AB      4F                          LD      C,A             ; Set data
01AC      CD EB0C                     CALL    CONOUT          ; Display a character
01AF      E1                          POP     HL              ; Restore pointer
01B0      23                          INC     HL              ; Update pointer
01B1      18 F4                       JR      DSPMSG          ;
                            ;
                            ;         ***********************************************************
                            ;                 DATA AREA
                            ;         ***********************************************************
                            ;
01B3                        ERRTBL:
01B3      01C1                        DW      DISKERR
01B5      01D0                        DW      READERR
01B7      01DB                        DW      WRITERR
01B9      01E7                        DW      SLCTERR
01BB      01F4                        DW      BOWPERR
01BD      0214                        DW      ROWPERR
01BF      020F                        DW      ETSTERR
                            ;
01C1                        DISKERR:
01C1      44 69 73 6B                 DB      'Disk not mount',00H
01C5      20 6E 6F 74
01C9      20 6D 6F 75
01CD      6E 74 00
01D0                        READERR:
01D0      52 65 61 64                 DB      'Read error',00H
01D4      20 65 72 72
01D8      6F 72 00
01DB                        WRITERR:
01DB      57 72 69 74                 DB      'Write error',00H
01DF      65 20 65 72
01E3      72 6F 72 00
01E7                        SLCTERR:
01E7      53 65 6C 65                 DB      'Select error',00H
01EB      63 74 20 65
01EF      72 72 6F 72
01F3      00
01F4                        BOWPERR:
01F4      42 2F 4F 20                 DB      'B/O or write protect error',00H
```

```
01F8     6F 72 20 77
01FC     72 69 74 65
0200     20 70 72 6F
0204     74 65 63 74
0208     20 65 72 72
020C     6F 72 00
020F                             ETSTERR:
020F     52 65 61 64                      DB      'Read/Write error',00H
0213     2F 57 72 69
0217     74 65 20 65
021B     72 72 6F 72
021F     00
                                 ;
0220                             KEYBLK:
0220     04 0E 02 01                      DB      4,14,2,1,0DH,07H,00H,'END'     ; End key
0224     0D 07 00 45
0228     4E 44
                                 ;
                                 ;
                                 ;        ****************************************************
                                 ;                WORK AREA
                                 ;        ****************************************************
                                 ;
022A                             DSKNUM:
022A                                      DS      1
022B                             TRKNUM:
022B                                      DS      2
022D                             SECNUM:
022D                                      DS      2
022F                             DMA:
022F                                      DS      128
                                 ;
                                          END
Macros:

Symbols:
EB09    CONIN          EB0C    CONOUT         01C1    DISKERR
022F    DMA            0170    DSKERR         022A    DSKNUM
01A7    DSPMSG         014F    ERRCHK         01B3    ERRTBL
020F    ETSTERR        0220    KEYBLK         011B    LOOP
100C    MAINSP         0177    NEXT           EB27    READ
01D0    READERR        01F4    ROWPERR        0040    SECMAX
022D    SECNUM         EB1B    SELDSK         EB24    SETDMA
EB21    SETSEC         EB1E    SETTRK         0127    SLCTERR
0100    START          EB96    TOUCH          0028    TRKMAX
022B    TRKNUM         EB03    WBOOT          EB2A    WRITE
01DB    WRITERR        014A    WRITES


No Fatal error(s)
```

(5) SAMPLE 5. BEEP

```
                         ;          ************************************************
                         ;                  BIOS BEEP SAMPLE PROGRAM
                         ;          ************************************************
                         ;
                         ;          NOTE:
                         ;                  This sample program plays music
                         ;                  by using beep.
                         ;
                         ;          <> assemble condition            <>
                         ;
                                    .Z80
                         ;
                         ;          <> loading address               <>
                         ;
                                    .PHASE 100h
                         ;
                         ;          <> constant values               <>
                         ;
F23B                     BPINTTBL           EQU     0F23BH
                         ;
EB03                     WBOOT              EQU     0EB03H
EB39                     BEEP               EQU     0EB39H
                         ;
1000                     MAINSP             EQU     01000H
                         ;
                         ;          ************************************************
                         ;                  MAIN PROGRAM
                         ;          ************************************************
                         ;
0100                     START:
0100    31 1000                     LD      SP,MAINSP       ; Set stack pointer
                         ;
0103    3A F23B                     LD      A,(BPINTTBL)    ; Get beep interrupt table.
0106    F6 80                       OR      10000000B       ; Disable 1 sec interrupt table.
0108    32 F23B                     LD      (BPINTTBL),A    ; Set new beep interrupt during beep.
                         ;
010B    21 011E                     LD      HL,SONG         ; Set the top address of song data
                         ;
010E                     LOOP:
010E    46                          LD      B,(HL)          ; Sound type
010F    23                          INC     HL              ; Next pointer
0110    4E                          LD      C,(HL)          ; Sound length
0111    23                          INC     HL              ; Next pointer
0112    79                          LD      A,C             ; If sound length is 0,
0113    B7                          OR      A               ; then end of data.
0114    CA EB03                     JP      Z,WBOOT         ; End of data then WBOOT
                         ;
0117    E5                          PUSH    HL              ; Save song table pointer
0118    CD EB39                     CALL    BEEP            ; Sound
011B    E1                          POP     HL              ; Restore song table pointer
011C    18 F0                       JR      LOOP            ; Loop
                         ;
                         ;          ************************************************
                         ;                  SONG DATA
                         ;          ************************************************
                         ;
011E                     SONG:
011E    11 02 11 02                 DB      17,2,17,2,17,2,17,2,17,2,17,2,17,9
0122    11 02 11 02
0126    11 02 11 02
012A    11 09
012C    11 03 14 05                 DB      17,3,20,5,17,2,15,2,17,9,17,3,17,5
0130    11 02 0F 02
0134    11 09 11 03
0138    11 05
013A    14 02 14 02                 DB      20,2,20,2,20,6,20,2,22,2,20,2,18,6
013E    14 06 14 02
0142    16 02 14 02
0146    12 06
0148    12 02 12 02                 DB      15,2,15,2,15,2,15,2,20,7
014C    12 02 12 26
0150    00 07
0152    0F 02 0F 02                 DB      15,2,15,2,15,2,15,2,15,2,15,2,15,5
0156    0F 02 0F 02
015A    0F 02 0F 02
015E    0F 09
0160    0F 02 0F 02                 DB      15,2,15,2,17,2,17,5,17,2,17,2,17,9
0164    11 02 11 06
0168    11 02 11 02
016C    11 09
016E    11 03 11 08                 DB      17,3,17,9,15,2,17,2,15,6,13,2,15,2
0172    0F 02 11 02
0176    0F 06 0D 02
017A    0F 02
017C    0D 02 0D 12                 DB      13,2,13,18,00,6
0180    00 06
0182    19 02 19 02                 DB      25,2,25,2,25,2,25,2,25,2,25,2,25,8
0186    19 02 19 02
018A    19 02 19 02
018E    19 08
0190    18 02 19 02                 DB      24,2,25,2,24,3,22,9,22,9,22,3,24,2
0194    18 03 16 09
```

```
0198      16 09 16 03
019C      18 02
019E      18 02 18 02                DB     24,2,24,2,24,2,24,2,24,2,24,9,22,2
01A2      18 02 18 02
01A6      18 02 18 09
01AA      16 02
01AC      18 02 16 09                DB     24,2,22,9,20,2,17,2,20,12
01B0      14 02 11 02
01B4      14 0C
01B6      19 02 19 02                DB     25,2,25,2,25,2,25,2,25,2,25,2,25,9
01BA      19 02 19 02
01BE      19 02 19 02
01C2      19 09
01C4      18 02 19 02                DB     24,2,25,2,24,3,22,6,24,3,22,9,22,2
01C8      18 03 16 06
01CC      18 03 16 09
01D0      16 02
01D2      16 02 18 03                DB     22,2,24,3,24,8,22,2,20,2,22,8,20,2
01D6      18 08 16 02
01DA      14 02 16 08
01DE      14 02
01E0      16 02 14 02                DB     22,2,20,2,20,16
01E4      14 10
01E6      00 00                      DB     00,0
                                     END
```

(6) SAMPLE 6. TIMDAT

```
                        ;       ****************************************************************
                        ;               BIOS TIMDAT SAMPLE PROGRAM
                        ;       ****************************************************************
                        ;
                        ;       NOTE:
                        ;               This sample program reads the clock,
                        ;                and displays the time.
                        ;
                        ;       <> assemble condition    <>
                        ;
                                .Z80
                        ;
                        ;       <> loading address       <>
                        ;
                                .PHASE  100H
                        ;
                        ;       <> constant values       <>
                        ;
EB03                    WBOOT           EQU     0EB03H
EB06                    CONST           EQU     0EB06H
EB09                    CONIN           EQU     0EB09H
EB0C                    CONOUT          EQU     0EB0CH
EB4E                    TIMDAT          EQU     0EB4EH
EB72                    READSW          EQU     0EB72H  ; Read switch
EB96                    TOUCH           EQU     0EB96H
EB9C                    KEYIN           EQU     0EB9CH
                        ;
1000                    MAINSP          EQU     01000H  ; Stack pointer
                        ;
                        ;       ****************************************************************
                        ;               MAIN PROGRAM
                        ;       ****************************************************************
                        ;
                        ;       NOTE:
                        ;               Display time until RETURN key is pressed.
                        ;
0100                    START:
0100    31 1000                 LD      SP,MAINSP       ; Set stack pointer
                        ;
0103    21 021A                 LD      HL,CUSROFF      ; Cursor off data
0106    CD 01A9                 CALL    DSPMSG          ; Cursor off
                        ;
0109    21 0220                 LD      HL,DATMSG       ; Date message
010C    CD 01A9                 CALL    DSPMSG          ; Display 'DATE'
                        ;
010F    0E 02                   LD      C,02H           ; Read DIP SW
0111    CD EB72                 CALL    READSW          ; Read
0114    E6 80                   AND     10000000B       ; Check bit 7
0116    B7                      OR      A               ; If A = 0
0117    28 17                   JR      Z,TCHKEY        ;   Then the machine is EHT-10
                        ;
0119                    KEYBOD:                         ;   Else EHT-10/2
0119    11 0102                 LD      DE,0102H        ; Set cursor (1,2)
011C    CD 01FD                 CALL    SETCUR          ;
011F    21 0227                 LD      HL,TIMMSG       ; Time message
0122    CD 01A9                 CALL    DSPMSG          ; Display 'TIME'
0125    06 03                   LD      B,03H           ; Set normal keyboard mode
0127    CD EB9C                 CALL    KEYIN           ;
012A    AF                      XOR     A               ;
012B    32 0257                 LD      (M_TYPE),A      ; M_TYPE flag = 0 ,then EHT-10/2
012E    18 19                   JR      LOOP            ;
                        ;
0130                    TCHKEY:                         ;  This is EHT-10
0130    11 0103                 LD      DE,0103H        ; Set cursor (1,3)
0133    CD 01FD                 CALL    SETCUR          ;
0136    21 0227                 LD      HL,TIMMSG       ; Time message
0139    CD 01A9                 CALL    DSPMSG          ; Display 'TIME'
013C    0E 01                   LD      C,01H           ; Set the key block
013E    11 022F                 LD      DE,KEYBLK       ; Set key block descriptor
0141    CD EB96                 CALL    TOUCH           ; Make 'END' key
0144    3E FF                   LD      A,0FFH          ;
0146    32 0257                 LD      (M_TYPE),A      ; M_TYPE <> 0 ,then EHT-10
                        ;
0149                    LOOP:                           ;
0149    CD EB06                 CALL    CONST           ; Key in check
014C    3C                      INC     A               ; Is any key pressed?
014D    20 07                   JR      NZ,SKIP         ; No.
014F    CD EB09                 CALL    CONIN           ; Get inputted key.
0152    FE 0D                   CP      0DH             ; Is the RETURN key pressed?
0154    28 4A                   JR      Z,TIMEEND       ; Yes , then end
                        ;
0156                    SKIP:                           ;
0156    11 0249                 LD      DE,NTIME        ; Time descriptor
0159    0E 00                   LD      C,00H           ; Read time function
015B    CD EB4E                 CALL    TIMDAT          ; Read time.
                        ;
015E    CD 0185                 CALL    TIMECMP         ; compare the new & old time
0161    28 E6                   JR      Z,LOOP          ; If these are the same ,then loop
                        ;
0163    CD 01C5                 CALL    TIMESET         ; Set new time data
                        ;
0166    3A 0257                 LD      A,(M_TYPE)      ; Check machine type
0169    B7                      OR      A               ; If A = 0 ,
```

```
016A    28 1A                   JR      Z,KEY2          ;  Then EHT-10/2
                        ;
                                                        ; This is EHT-10
016C    11 0402                 LD      DE,0402H        ; Set cursor (4,2)
016F    CD 01FD                 CALL    SETCUR          ;
0172    21 022D                 LD      HL,DATE         ; Date data
0175    CD 01A9                 CALL    DSPMSG          ; Display the date
0178    11 0404                 LD      DE,0404H        ; Set cursor (4,4)
017B    CD 01FD                 CALL    SETCUR          ;
017E    21 0238                 LD      HL,TIME         ; Time data
0181    CD 01A9                 CALL    DSPMSG          ; Display the time
0184    18 C3                   JR      LOOP            ;
                        ;
0186                    KEY2:                           ; This is EHT-10/2
0186    11 0801                 LD      DE,0801H        ;
0189    CD 01FD                 CALL    SETCUR          ; Set cursor (8,1)
018C    21 022D                 LD      HL,DATE         ; Date data
018F    CD 01A9                 CALL    DSPMSG          ; Display the date
0192    11 0802                 LD      DE,0802H        ;
0195    CD 01FD                 CALL    SETCUR          ; Set cursor (8,2)
0198    21 0238                 LD      HL,TIME         ; Time data
019B    CD 01A9                 CALL    DSPMSG          ; Display the time
019E    18 A8                   JR      LOOP            ;
                        ;
01A0                    TIMEEND:                        ;
01A0    21 021D                 LD      HL,CUSRON       ; Cursor on data
01A3    CD 01A9                 CALL    DSPMSG          ; Cursor on
01A6    C3 E803                 JP      WBOOT           ; Jump WBOOT
                        ;
                        ;       ****************************************************
                        ;               DISPLAY MESSAGE UNTIL FIND 00H
                        ;       ****************************************************
                        ;
                        ;       NOTE:
                        ;
                        ;       <> entry parameter      <>
                        ;               HL : The address of message string
                        ;       <> return prameter      <>
                        ;               NONE
                        ;       <> preserved registers  <>
                        ;               NONE
                        ;
01A9                    DSPMSG:
01A9    7E                      LD      A,(HL)          ; Get message data
01AA    B7                      OR      A               ; End mark ?
01AB    C8                      RET     Z               ; Yes, then return
                        ;
01AC    4F                      LD      C,A             ; Set display data
01AD    E5                      PUSH    HL              ; Save pointer to message
01AE    CD E80C                 CALL    CONOUT          ; Display message
01B1    E1                      POP     HL              ; Restore message pointer
01B2    23                      INC     HL              ; Pointer update.
01B3    18 F4                   JR      DSPMSG          ; Loop until find 00H
                        ;
                        ;       ****************************************************
                        ;               CHECK OLD & NEW TIME
                        ;       ****************************************************
                        ;
                        ;       NOTE:
                        ;
                        ;       <> entry parameter      <>
                        ;               NONE
                        ;       <> return parameter     <>
                        ;               ZF : Return information
                        ;                       = 1 : New time and old time are the same.
                        ;                       = 0 : New time is different from old time.
                        ;       <> preserverd registers <>
                        ;               NONE
                        ;
01B5                    TIMECHK:
01B5    21 0249                 LD      HL,NTIME        ; New time data
01B8    11 0250                 LD      DE,OTIME        ; Old time data
01BB    06 06                   LD      B,06H           ; Data counter
                        ;
01BD                    TLOOP:                          ;
01BD    1A                      LD      A,(DE)          ; Get old time data
01BE    BE                      CP      (HL)            ; Compare with new time
01BF    C0                      RET     NZ              ; If those are different, then return.
01C0    13                      INC     DE              ; Pointer update
01C1    23                      INC     HL              ;
01C2    10 F8                   DJNZ    TLOOP           ; Loop 6 time
01C4    C9                      RET                     ;
                        ;
                        ;       ****************************************************
                        ;               SET TIME DATA
                        ;       ****************************************************
                        ;
                        ;       NOTE:
                        ;
                        ;       <> entry parameter      <>
                        ;               NONE
                        ;       <> return parameter     <>
                        ;               NONE
                        ;       <> preserved registers  <>
```

```
                        ;               NONE
                        ;
                        ;
01C5                    TIMESET:
01C5    21 0249                 LD      HL,NTIME        ; Set the time data to the time area
01C8    11 0250                 LD      DE,OTIME        ;
01CB    01 0006                 LD      BC,6            ; Year/month/date/hour/minute/second
01CE    ED B0                   LDIR                    ; Move new data to the old area
                        ;
01D0    21 0249                 LD      HL,NTIME        ; Set BCD data to the message area (ASCII
                                                        ; code)
01D3    11 022D                 LD      DE,DATE         ; HL is source , DE is destination
01D6    06 03                   LD      B,03H           ; B is counter
01D8                    SET10:                          ;
01D8    CD 01EC                 CALL    SETASCII        ; Convert BCD to ASCII
01DB    23                      INC     HL              ; Pointer update
01DC    13                      INC     DE              ;
01DD    10 F9                   DJNZ    SET10           ; Loop 3 times.
                        ;
01DF    11 0236                 LD      DE,TIME         ; Time data setting area
01E2    06 03                   LD      B,03H           ;
01E4                    SET20:
01E4    CD 01EC                 CALL    SETASCII        ; Convert BCD to ASCII
01E7    23                      INC     HL              ; Pointers update
01E8    13                      INC     DE              ;
01E9    10 F9                   DJNZ    SET20           ; Loop 3 times
01EB    C9                      RET                     ;
                        ;
                        ;
                        ;       ***************************************************
                        ;               SET ASCII DATA FROM BCD DATA
                        ;       ***************************************************
                        ;
                        ;       NOTE:
                        ;
                        ;       () entry parameter      ()
                        ;               HL : BCD data address
                        ;               DE : ASCII data setting address
                        ;       () return parameter     ()
                        ;               NONE
                        ;       () preserved registers  ()
                        ;               HL
                        ;
01EC                    SETASCII:
01EC    7E                      LD      A,(HL)          ; Get the BCD data
01ED    F5                      PUSH    AF              ; Save the BCD data
01EE    0F                      RRCA                    ; Move the MSB 4 bits to the LSB 4 bits
01EF    0F                      RRCA                    ;
01F0    0F                      RRCA                    ;
01F1    0F                      RRCA                    ;
01F2    CD 01F6                 CALL    NEXT            ; Set the ASCII data
01F5    F1                      POP     AF              ; Restore the BCD data
01F6                    NEXT:
01F6    E6 0F                   AND     0FH             ; Check the LSB 4 bits
01F8    C6 30                   ADD     A,30H           ; Change to the ASCII data
01FA    12                      LD      (DE),A          ; Set the ASCII data
01FB    13                      INC     DE              ; Setting pointer update
01FC    C9                      RET                     ;
                        ;
                        ;
                        ;       ***************************************************
                        ;               SET CURSOR POSITION
                        ;       ***************************************************
                        ;
                        ;       NOTE:
                        ;
                        ;       () entry parameter      ()
                        ;               D : X direction
                        ;               E : Y direction
                        ;       () return parameter     ()
                        ;               NONE
                        ;       () preserved registers  ()
                        ;               NONE
                        ;
01FD                    SETCUR:
01FD    D5                      PUSH    DE              ; Save cursor position
01FE    D5                      PUSH    DE              ;
01FF    0E 1B                   LD      C,1BH           ; ESC
0201    CD EB0C                 CALL    CONOUT          ;
0204    0E 3D                   LD      C,'='           ; '='
0206    CD EB0C                 CALL    CONOUT          ;
0209    D1                      POP     DE              ;       Restore Y value
020A    7B                      LD      A,E             ;
020B    C6 1F                   ADD     A,01FH          ;
020D    4F                      LD      C,A             ;       Set Y value to Creg.
020E    CD EB0C                 CALL    CONOUT          ;
0211    D1                      POP     DE              ;       Restore X value
0212    7A                      LD      A,D             ;
0213    C6 1F                   ADD     A,01FH          ;
0215    4F                      LD      C,A             ;       Set X value to Creg.
0216    CD EB0C                 CALL    CONOUT          ;
0219    C9                      RET                     ;
```

```
                                  ;
                                  ;
021A                              CUSROFF:
021A    1B 32 00                          DB      1BH,'2',00H            ; Cursor off data
021D                              CUSRON:
021D    1B 33 00                          DB      1BH,'3',00H            ; Cursor on data
                                  ;
0220                              DATMSG:
0220    0C                                DB      0CH                    ; Clear screen
0221    44 41 54 45                       DB      'DATE:'                ;
0225    3A
0226    00                                DB      00H                    ; End mark
0227                              TIMMSG:
0227    54 49 4D 45                       DB      'TIME:'                ;
022B    3A
022C    00                                DB      00H                    ; End mark
022D                              DATE:
022D    30 30 2F 30                       DB      '00/00/00'             ; Date message area
0231    30 2F 30 30
0235    00                                DB      00H                    ;
0236                              TIME:
0236    30 30 3A 30                       DB      '00:00:00'             ; Time message area
023A    30 3A 30 30
023E    00                                DB      00H                    ;
                                  ;
023F                              KEYBLK:
023F    04 0E 02 01                       DB      4,14,2,1,0DH,07H,00,'END'
0243    0D 07 00 45
0247    4E 44

0249                              NTIME:
0249                                      DS      7
0250                              OTIME:
0250                                      DS      7
0257                              M_TYPE:
0257                                      DS      1
                                  ;
                                  END

Macros:

Symbols:
EB09    CONIN              EB0C    CONOUT             EB06    CONST
021A    CUSROFF            021D    CUSRON             022D    DATE
0220    DATMSG             01A9    DSPMSG             0186    KEY2
023F    KEYBLK             0119    KEYBOD             EB9C    KEYIN
0149    LOOP               1000    MAINSP             0257    M_TYPE
01F6    NEXT               0249    NTIME              0250    OTIME
EB7[illegible]    READSW             01D8    SET10              01E4    SET20
01EC    SETASCII           01FD    SETCUR             0156    SKIP
0100    START              0130    TCHKEY             EB4E    TIMDAT
0236    TIME               0195    TIMECHK            01A0    TIMEEND
01C5    TIMESET            0227    TIMMSG             019D    TLOOP
EB88    TOUCH              EB03    WBOOT


No Fatal error(s)
```

(7) SAMPLE 7. RSIOX

```
                                  ;       **************************************************
                                  ;               BIOS RSIOX SAMPLE PROGRAM
                                  ;       **************************************************
                                  ;
                                  ;       NOTE:
                                  ;                This sample program displays the received data,
                                  ;               and sends the inputted data.
                                  ;
                                  ;       <> assemble condition   <>
                                  ;
                                          .Z80
                                  ;
                                  ;       <> loading address      <>
                                  ;
                                          .PHASE  100H
                                  ;
                                  ;       <> constant values      <>
                                  ;
EB03                              WBOOT           EQU     0EB03H
EB06                              CONST           EQU     0EB06H
EB09                              CONIN           EQU     0EB09H
EB0C                              CONOUT          EQU     0EB0CH
EB54                              RSIOX           EQU     0EB54H
EB69                              CALLX           EQU     0EB69H
EB72                              READSW          EQU     0EB72H
EB96                              TOUCH           EQU     0EB96H
EB9C                              KEYIN           EQU     0EB9CH
EBA5                              INFORM          EQU     0EBA5H
                                  ;
0010                              RSOPN           EQU     10H     ; RS232C open function
0020                              RSCLS           EQU     20H     ; Close function
0030                              RSIST           EQU     30H     ; Input status function
0040                              RSOST           EQU     40H     ; Output status function
0050                              RSGET           EQU     50H     ; Get function
0060                              RSPUT           EQU     60H     ; Put function
0090                              RSERR           EQU     90H     ; Error status function
                                  ;
                                  ;
000D                              CR              EQU     0DH     ; Carriage return code
000A                              LF              EQU     0AH     ; Line feed code
001B                              ESC             EQU     1BH     ; Escape code
                                  ;
F010                              SRSADR          EQU     0F010H
                                  ;
003C                              XUSRSCRN        EQU     003CH   ; Change to user screen
003F                              XSYSSCRN        EQU     003FH   ; Change to system screen
1000                              MAINSP          EQU     01000H
                                  ;
                                  ;       **************************************************
                                  ;               MAIN PROGRAM
                                  ;       **************************************************
                                  ;
0100                              START:
0100        31 1000                       LD      SP,MAINSP       ; Set stack pointer
                                  ;
0103        0E 03                         LD      C,03H           ; Get DISBNK address
0105        CD EBA5                       CALL    INFORM          ;
0108        22 01F4                       LD      (DISBNK),HL     ; Save DISBNK address
                                  ;
010B        21 F010                       LD      HL,SRSADR       ; Copy open parameter from system area.
010E        11 01EB                       LD      DE,OPNPRM       ; Application parameter area.
0111        01 0009                       LD      BC,9            ; Parameter number
0114        ED B0                         LDIR                    ; Copy
                                  ;
                                  ;       LD      B,01H           ;
                                  ;       CALL    KEYIN           ; Change the keyboard mode to Alphabet
                                  ;
0116        21 01EB                       LD      HL,OPNPRM       ; Open parameter
0119        06 10                         LD      B,RSOPN         ; RS232C open function
011B        CD EB54                       CALL    RSIOX           ; OPEN
011E        B7                            OR      A               ; Error return?
011F        C2 EB03                       JP      NZ,WBOOT        ; Yes , then WBOOT
                                  ;
0122                              KEYCHK:
0122        CD EB06                       CALL    CONST           ; Get key inputted status
0125        3C                            INC     A               ; Input any key?
0126        CC 013F                       CALL    Z,PUT           ; Yes, then put the data
                                  ;
0129        21 01EB                       LD      HL,OPNPRM       ; Get input status
012C        06 30                         LD      B,RSIST         ; Input status function
012E        CD EB54                       CALL    RSIOX           ; Get input status
0131        3C                            INC     A               ; If there is receiving data,
0132        CC 0162                       CALL    Z,GET           ;  then get the data
0135        18 EB                         JR      KEYCHK          ; Loop
                                  ;
0137                              PEND:
0137        06 20                         LD      B,RSCLS         ; Close RSIOX
0139        CD EB54                       CALL    RSIOX           ;
013C        C3 EB03                       JP      WBOOT           ; Program end.
```

```
                        ;
                        ;       ************************************************************
                        ;               PUT INPUTTED DATA TO RS232C
                        ;       ************************************************************
                        ;
                        ;       NOTE:
                        ;
                        ;       <> entry parameter          <>
                        ;               NONE
                        ;       <> return parameter         <>
                        ;               NONE
                        ;       <> preserved parameter      <>
                        ;               NONE
                        ;
013F                    PUT:
013F    CD EB09                 CALL    CONIN           ; Get inputted data
0142    4F                      LD      C,A             ;
0143    FE 03                   CP      03H             ; If inputted key is 03H,
0145    CA EB03                 JP      Z,WBOOT         ;  then end of program
                        ;
0148    C5                      PUSH    BC              ; Save input key code
0149    C5                      PUSH    BC              ; Save input key code
014A    FE 0D                   CP      CR              ; If inputted key is RETURN,
014C    0E 0A                   LD      C,LF            ;  then LF console out
014E    CC EB0C                 CALL    Z,CONOUT        ;
0151    C1                      POP     BC              ; Restore input key code
0152    CD EB0C                 CALL    CONOUT          ; Console out inputting data
                        ;
0155    C1                      POP     BC              ; Restore input key code
0156    21 01E8                 LD      HL,OPNPRM       ; Put inputting data to RS232C
0159    06 60                   LD      B,RSPUT         ; Put function code
015B    CD EB54                 CALL    RSIOX           ; Put data
015E    C4 01A8                 CALL    NZ,ERRDSP       ; If error is returned,then display error
0161    C9                      RET                     ;
                        ;
                        ;       ************************************************************
                        ;               GET RECEIVED DATA
                        ;       ************************************************************
                        ;
                        ;       NOTE:
                        ;
                        ;       <> entry parameter          <>
                        ;               NONE
                        ;       <> return parameter         <>
                        ;               NONE
                        ;       <> preserved registers      <>
                        ;               NONE
                        ;
0162                    GET:
0162    06 90                   LD      B,RSERR         ; Check error status
0164    CD EB54                 CALL    RSIOX           ; Get error status
0167    E6 74                   AND     01110100B       ; Error happened ?
0169    C4 01A5                 CALL    NZ,ERRDSP       ; Yes, then display error
                        ;
016C    21 01E8                 LD      HL,OPNPRM       ; Get received data
016F    06 50                   LD      B,RSGET         ; Get function
0171    CD EB54                 CALL    RSIOX           ; Get
0174    C4 01A5                 CALL    NZ,ERRDSP       ; If error,then display error
                        ;
0177    4F                      LD      C,A             ; Console out received data
0178    CD 0182                 CALL    RVSON           ; Reverse on
017B    CD EB0C                 CALL    CONOUT          ; Display received data
017E    CD 015F                 CALL    RVSOFF          ; Reverse off
0181    C9                      RET                     ;
                        ;
                        ;       ************************************************************
                        ;               REVERSE MODE ON
                        ;       ************************************************************
                        ;
                        ;       NOTE:
                        ;
                        ;       <> entry parameter          <>
                        ;               NONE
                        ;       <> return parameter         <>
                        ;               NONE
                        ;       <> preserved registers      <>
                        ;               BC
                        ;
0182                    RVSON:
0182    C5                      PUSH    BC              ; Save BC register
0183    0E 1B                   LD      C,ESC           ; Reverse on command
0185    CD EB0C                 CALL    CONOUT          ; ESC + '0'
0188    0E 30                   LD      C,'0'           ;
018A    CD EB0C                 CALL    CONOUT          ;
018D    C1                      POP     BC              ; Restore BC register
018E    C9                      RET                     ;
```

```
                              ;
                              ;       ********************************************************
                              ;               REVERSE MODE OFF
                              ;       ********************************************************
                              ;
                              ;       NOTE:
                              ;
                              ;       <> entry parameter       <>
                              ;               NONE
                              ;       <> return parameter      <>
                              ;               NONE
                              ;       <> preserved registers   <>
                              ;               BC
                              ;
018F                          RVSOFF:
018F    C5                            PUSH    BC              ; Save BC register
0190    0E 1B                         LD      C,ESC           ; Reverse off command
0192    CD EB0C                       CALL    CONOUT          ; ESC + '1'
0195    0E 31                         LD      C,'1'           ;
0197    CD EB0C                       CALL    CONOUT          ;
019A    C1                            POP     BC              ; Restore BC register
019B    C9                            RET                     ;
                                                              ;
                              ;
                              ;       ********************************************************
                              ;               DISPLAY MESSAGE UNTIL FIND 00H
                              ;       ********************************************************
                              ;
                              ;       NOTE:
                              ;
                              ;       <> entry parameter       <>
                              ;               HL : Message data top address
                              ;       <> return parameter      <>
                              ;               NONE
                              ;       <> preserved registers   <>
                              ;               NONE
                              ;
019C                          DSPMSG:
019C    7E                            LD      A,(HL)          ; Get display data
019D    B7                            OR      A               ; Check end code
019E    C8                            RET     Z               ; Yes , then return
                              ;
019F    4F                            LD      C,A             ;
01A0    E5                            PUSH    HL              ; Save data pointer
01A1    CD EB0C                       CALL    CONOUT          ; Console out the data
01A4    E1                            POP     HL              ; Restore data pointer
01A5    23                            INC     HL              ; Pointer update
01A6    18 F4                         JR      DSPMSG          ; Loop
                              ;
                              ;
                              ;       ********************************************************
                              ;               DISPLAY ERROR MESSAGE
                              ;       ********************************************************
                              ;
                              ;       NOTE:
                              ;
                              ;       <> entry parameter       <>
                              ;               A : Error status
                              ;       <> return parameter      <>
                              ;               NONE
                              ;       <> preserved registers   <>
                              ;               ALL registers
                              ;
01A8                          ERRDSP:
01A8    F5                            PUSH    AF              ; Save all registers
01A9    C5                            PUSH    BC              ;
01AA    D5                            PUSH    DE              ;
01AB    E5                            PUSH    HL              ;
                              ;
01AC    F5                            PUSH    AF              ; Save error status
                              ;
                                      LD      IX,XSYSSCRN     ; Change to system screen
                                      LD      A,0FFH          ; Set system bank
                                      LD      (DISBNK),A      ;
                                      CALL    CALLX           ; Call OS jump table
                              ;
                                      LD      C,0CH           ; Clear screen & home
                                      CALL    CONOUT          ;
                              ;
01AD    21 01CA                       LD      HL,ERRMSG       ; Set error message address
01B0    CD 019C                       CALL    DSPMSG          ; Display 'ERROR occured'
                              ;
01B3    F1                            POP     AF              ; Restore error status
01B4    06 08                         LD      B,8             ; Set loop counter
01B6                          WTAREC:
01B6    0E 30                         LD      C,30H           ; Creg = '0'
01B8    07                            RLCA                    ; Shift left
01B9    30 01                         JR      NC,WTBIT        ;
01BB    0C                            INC     C               ; Creg = '1'
01BC                          WTBIT:
01BC    F5                            PUSH    AF              ; Save Areg
01BD    C5                            PUSH    BC              ; Save Breg
01BE    CD EB0C                       CALL    CONOUT          ; Display 1 bit
```

```
01C1    C1                          POP     BC              ; Restore Breg
01C2    F1                          POP     AF              ; Restore Areg
01C3    10 F1                       DJNZ    WTAREG          ; Loop 9 time
                        ;
                        ;           LD      C,02H           ; Read DIP SW
                        ;           CALL    READSW          ; Read
                        ;           AND     10000000B       ; Check bit 7
                        ;           OR      A               ; If bit 7 = 1 then this is EHT-10/2
                        ;           JR      NZ,KEYBOD       ; Jump to EHT-10/2 routine
                        ;
                                                            ; EHT-10 routine
                        ;           LD      C,01H           ; Make 1 key block
                        ;           LD      DE,KEYBLK       ; Set key block data
                        ;           CALL    TOUCH           ; Display a key block
                        ;           JR      NEXT            ;
                        ;
01C5                    KEYBOD:
                        ;           LD      B,01H           ; Change the keyboard mode to Alphabet
                        ;           CALL    KEYIN           ;
                        ;
01C5                    NEXT:
                        ;           CALL    CONIN           ; Wait for key in
                        ;
                        ;           LD      IX,XUSRSCRN     ; Change to user screen
                        ;           LD      A,0FFH          ; OS bank
                        ;           LD      (DISBNK),A      ;
                        ;           CALL    CALLX           ;
                        ;
01C5    E1                          POP     HL              ; Restore registers
01C6    D1                          POP     DE              ;
01C7    C1                          POP     BC              ;
01C8    F1                          POP     AF              ;
01C9    C9                          RET
                        ;
                        ;
                        ;           ************************************************************
                        ;                   DATA AREA
                        ;           ************************************************************
                        ;
01CA                    ERRMSC:
01CA    45 52 52 4F                 DB      'ERROR OCCURED',0DH,0AH,00H
01CE    52 20 4F 43
01D2    43 55 52 45
01D6    44 0D 0A 00
01DA    41 20 3D 20                 DB      'A = ',00H
01DE    00
                        ;
                        ;
01DF                    KEYBLK:
01DF    03 0E 03 01                 DB      3,14,3,1,0DH,07H,00H,'Press'
01E3    0D 07 00 50
01E7    72 65 73 73
                        ;
                        ;           ************************************************************
                        ;                   WORK AREA
                        ;           ************************************************************
                        ;
01EB                    OPNPRM:
01EB                                DS      9               ; RSIOX open parameter area
                        ;
01F4                    DISBNK:
01F4                                DS      2               ; Destination bank area
                        ;
                                    END
Macros:

Symbols:
E569    CALLX           EB09    CONIN           EB0C    CONOUT
EB06    CONST           000D    CR              01F4    DISBNK
015C    DSPMSC          0135    ERRESP          01CA    ERRMSC
001B    ESC             0162    GET             EBA5    INFORM
01DF    KEYBLK          01C5    KEYBOD          0122    KEYCHK
EB5C    KEYIN           000A    LF              0100    MAINST
01C5    NEXT            01EB    OPNPRM          0137    PEND
0137    PUT             EB72    READSW          0020    RSCLS
0090    RSERR           0050    RSGET           EB54    RSIOX
0030    RSTST           0010    RSOPN           0040    RSOST
0060    RSPUT           015F    RVSOFF          0152    RVSON
0019    SRSADR          01CC    START           ?896    TOUCH
EB03    WBOOT           0196    WTAREG          01BC    WTBIT
003F    XSYSSCRN        003C    XUSRSCRN


No Fatal error(s)
```

(8) SAMPLE 8. GETPFK

```
                            ;       ************************************************
                            ;                   BIOS GETPFK SAMPLE PROGRAM
                            ;       ************************************************
                            ;
                            ;       NOTE:
                            ;               This sample program is displaying present
                            ;               defined key code in normal mode
                            ;
                            ;       <> assemble condition    <>
                                    .Z80
                            ;
                            ;       <> loading address       <>
                            ;
                                    .PHASE  100H
                            ;
                            ;       <> constant values       <>
                            ;
EB03                        WBOOT           EQU     0EB03H          ; WBOOT entry address.
EB09                        CONIN           EQU     0EB09H          ; CONIN entry address.
EB0C                        CONOUT          EQU     0EB0CH          ; CONOUT entry address.
EB6C                        GETPFK          EQU     0EB6CH          ; GETPFK entry address.
EB72                        READSW          EQU     0EB72H          ; READSW entry address.
EB96                        TOUCH           EQU     0EB96H          ; TOUCH entry address.
                            ;
1000                        MAINSP          EQU     1000H           ; Stack pointer.
                            ;
0003                        BREAK           EQU     03H             ; STOP code.
                            ;
                            ;       ************************************************
                            ;                   MAIN PROGRAM
                            ;       ************************************************
                            ;
                            ;       NOTE:
                            ;
0100                        START:
0100    31 1000                     LD      SP,MAINSP               ; Set stack pointer.
                            ;
0103    0E 0C                       LD      C,0CH                   ; Clear screen.
0105    CD EB0C                     CALL    CONOUT                  ;
0108    0E 02                       LD      C,2                     ; READSW parameter.
010A    CD EB72                     CALL    READSW                  ; Read DIP swich.
                                                                    ;
010D    07                          RLCA                            ; Check EHT-10 or EHT-10/2
010E    38 08                       JR      C,SKIPTCH               ; If target machine is EHT-10/2
                                                                    ; then
                                                                    ; skip key block set.
                                                                    ;
0110    0E 02                       LD      C,2                     ; The number of key block.
0112    11 0171                     LD      DE,TOUCHDT              ; Key block discriptor.
0115    CD EB96                     CALL    TOUCH                   ; Set key block.
0118                        SKIPTCH:                                ;
                            ;
                            ;       ************************************************
                            ;                   GET KEY CODE TABLE DATA
                            ;       ************************************************
                            ;
                            ;       NOTE :
                            ;               This procedure is getting all key code table data.
                            ;
0118    0E 01                       LD      C,1                     ; Normal mode.
011A    06 FF                       LD      B,0FFH                  ; Read all key code.
011C    21 0155                     LD      HL,PFKBUF               ; Key code reading area address.
011F    CD EB6C                     CALL    GETPFK                  ; Get key code data.
                                                                    ;
                            ;       ************************************************
                            ;                   DISPLAY KEY CODE BUFFER DATA
                            ;       ************************************************
                            ;
                            ;       NOTE:
                            ;               This procedure is displaying key code table data.
                            ;
0122    0E 02                       LD      C,2                     ; READSW parameter.
0124    CD EB72                     CALL    READSW                  ; Read DIP swich.
                                                                    ;
0127    07                          RLCA                            ; Check EHT-10 or EHT-10/2
0128    38 04                       JR      C,TYPEII                ;
012A                        TYPEI:                                  ;
012A    06 46                       LD      B,70                    ; Set loop counter for EHT-10
012C    18 05                       JR      LOOP1                   ; (EHT-10 has 70 key codes)
012E                        TYPEII:                                 ;
012E    06 22                       LD      B,34                    ; Set loop counter for EHT-10/2
                                                                    ; (EHT-10/2 has 34 key codes)
0130    21 0185                     LD      HL,PFKBUF               ; Set display data buffer address.
                                                                    ;
0133                        LOOP1:                                  ;
0133    E5                          PUSH    HL                      ; Save pointer.
0134    C5                          PUSH    BC                      ; Save loop counter.
                                                                    ;
0135    7E                          LD      A,(HL)                  ; Load display data.
0136    CD 0149                     CALL    DSPDAT                  ; Display data.
```

```
                                                    ;
0139    CD EB09                 CALL    CONIN       ; Get any inputted key.
013C    FE 03                   CP      BREAK       ; STOP code?
013E    CA EB03                 JP      Z,WBOOT     ; Yes, then end.
                                                    ;
0141    C1                      POP     BC          ; Restore loop counter.
0142    E1                      POP     HL          ; Restore pointer.
0143    23                      INC     HL          ; Pointer update.
                                                    ;
0144    10 ED                   DJNZ    LOOP1       ; Loop.
                                                    ;
0146    C3 EB03                 JP      WBOOT       ; End.
                                                    ;
                                                    ;
                    ;*************************************************
                    ;           DISPLAY DATA
                    ;*************************************************
                    ;
                    ;   NOTE:
                    ;           Display data in A register in hexa image.
                    ;
0149                DSPDAT:
0149    F5                      PUSH    AF          ; Save data.
014A    CB 3F                   SRL     A           ;
                                                    ;
014C    CB 3F                   SRL     A           ;
014E    CB 3F                   SRL     A           ;
0150    CB 3F                   SRL     A           ;
0152    CD 0164                 CALL    TBSDAT      ; Translate upper 4 bit to ASCII
                                                    ; code.
0155    4F                      LD      C,A         ; Load display data to C register.
                                                    ;
0156    CD EB0C                 CALL    CONOUT      ; Display ASCII data.
                                                    ;
0159    F1                      POP     AF          ; Restore data
015A    E6 0F                   AND     0FH         ;
015C    CD 0164                 CALL    TBSDAT      ; Translate lower 4bit to ASCII
                                                    ; code.
015F    4F                      LD      C,A         ; Load display data to C register.
                                                    ;
0160    CD EB0C                 CALL    CONOUT      ; Display ASCII data
0163    C9                      RET                 ;
                                                    ;
                    ;*************************************************
                    ;           TRANSLATE BINARY DATA TO ASCII DATA
                    ;*************************************************
                    ;
                    ;   NOTE :
                    ;           Translate binary data to ASCII code.
                    ;
                    ;           <> entry parameter      <>
                    ;                   A : binary data.
                    ;           <> return parameter     <>
                    ;                   A : ASCII data.
0164                TBSDAT:
0164    FE 0A                   CP      0AH         ; Data < 0AH?
0166    30 03                   JR      NC,TBSDAT5  ; No.
                                                    ;
0168    F6 30                   OR      30H         ; Data < 0AH
016A    C9                      RET                 ;
016B                TBSDAT5:                        ;
016B    0E 09                   LD      C,09H       ; Data >= 0AH
016D    91                      SUB     C           ;
016E    F6 40                   OR      40H         ;
0170    C9                      RET                 ;
                                                    ;
                                                    ;
                    ;*************************************************
                    ;           TOUCH KEY DISCRIPTOR
                    ;*************************************************
                    ;
                    ;
0171                TCHKDT:
0171    02 0E 02 01             DB      2,14,2,1,0DH,7,0,'CNT'
0175    0D 07 00 43
0179    4E 54
017B    04 0E 02 01             DB      4,14,2,1,03H,7,0,'END'
017F    03 07 00 45
0183    4E 44
                    ;
                    ;*************************************************
                    ;           KEY CODE READ BUFFER
                    ;*************************************************
                    ;
0185                PFKBUF:
0185                            DS      70          ; Key code data reading area.

                                END
```

(9) SAMPLE 9. PUTPFK

```
                        ;       ****************************************************
                        ;               BIOS PUTPFK SAMPLE PROGRAM
                        ;       ****************************************************
                        ;
                        ;       NOTE:
                        ;               This sample program sets the data to the
                        ;               user key table in normal mode,
                        ;               and gets the data for displaying it.
                        ;               (for EHT-10/2)
                        ;
                        ;       <> assemble condition   <>
                                .Z80
                        ;
                        ;       <> loading address      <>
                        ;
                                .PHASE  100H
                        ;
                        ;       <> constant values      <>
                        ;
EB03                    WBOOT           EQU     0EB03H          ; WBOOT entry address.
EB0C                    CONOUT          EQU     0EB0CH          ; CONOUT entry address.
EB09                    CONIN           EQU     0EB09H          ; CONIN entry address.
EB6F                    PUTPFK          EQU     0EB6FH          ; PUTPFK entry address.
EB6C                    GETPFK          EQU     0EB6CH          ; GETPFK entry address.
                        ;
1000                    MAINSP          EQU     1000H           ; Stack pointer.
                        ;
                        ;       ****************************************************
                        ;               MAIN PROGRAM
                        ;       ****************************************************
                        ;
                        ;       NOTE:
                        ;
0100                    START:
0100    31 0010                 LD      SP,MAINSP               ; Set stack pointer.
                                                                ;
0103    0E 0C                   LD      C,0CH                   ; Clear screen.
0105    CD EB0C                 CALL    CONOUT                  ;
                                                                ;
                        ;       ****************************************************
                        ;               INITIATE KEY CODE TABLE
                        ;       ****************************************************
                        ;
                        ;       NOTE:
                        ;               This procedure is initiating key code table.
                        ;
                                                                ;
0108    0E 00                   LD      C,0                     ; Initiate key table function.
010A    CD EB6F                 CALL    PUTPFK                  ; Initiate key table.
                                                                ;
                        ;       ****************************************************
                        ;               SET KEY TABLE
                        ;       ****************************************************
                        ;
                        ;       NOTE:
                        ;               This procedure is setting key code table.
                        ;
010D    06 01                   LD      B,1                     ; Key 1 to key 34.
010F    0E 01                   LD      C,1                     ; Normal mode.
0111    3E 01                   LD      A,1                     ; Setting data 1 to 34.
0113                    LOOP1:                                  ;
0113    F5                      PUSH    AF                      ; Save setting data.
0114    C5                      PUSH    BC                      ; Save key position no.
                                                                ;
0115    CD EB6F                 CALL    PUTPFK                  ; Set key code table.
                                                                ;
0118    C1                      POP     BC                      ; Restore key position no.
0119    F1                      POP     AF                      ; Restore setting data.
                                                                ;
011A    3C                      INC     A                       ; Update setting data.
011B    04                      INC     B                       ; Update key position no.
011C    FE 23                   CP      35                      ; Loop 34 times.
011E    38 F3                   JR      C,LOOP1                 ;
                                                                ;
                        ;       ****************************************************
                        ;               DISPLAY SETTING KEY CODE DATA
                        ;       ****************************************************
                        ;
                        ;       NOTE :
                        ;               This procedure is getting key code table data,
                        ;               and displaying it.
                        ;
0120    0E 01                   LD      C,1                     ; Normal mode.
0122    06 01                   LD      B,1                     ; Key 1 to 34.
0124                    LOOP2:                                  ;
0124    C5                      PUSH    BC                      ; Save key position no.
0125    CD EB6C                 CALL    GETPFK                  ; Get key code.
0128    CD 0138                 CALL    DSPDAT                  ; Display key code.
012B    C1                      POP     BC                      ; Restore key position no.
012C    04                      INC     B                       ; Update key position no.
```

```
012D    78                    LD      A,B                ; Loop 34 times.
012E    FE 23                 CP      35                 ;
0130    38 F2                 JR      C,LOOP2            ;
                                                         ;
0132    CD EB09               CALL    CONIN              ; Waiting for inputting any key.
                                                         ;
0135    C3 EB03               JP      WBOOT              ; End.
                                                         ;
                      ;       ************************************************
                      ;               DISPLAY DATA
                      ;       ************************************************
                      ;
                      ;       NOTE:
                      ;               Displaying data in A register in hexa image.
                      ;
0138                  DSPDAT:                            ;
0138    F5                    PUSH    AF                 ; Save data.
0139    CB 3F                 SRL     A                  ;
013B    CB 3F                 SRL     A                  ;
013D    CB 3F                 SRL     A                  ;
013F    CB 3F                 SRL     A                  ;
0141    CD 0153               CALL    TRSDAT             ; Translate upper 4bit to ASCII
                                                         code.
0144    4F                    LD      C,A                ; Load display data to C register.
                                                         ;
0145    CD EB0C               CALL    CONOUT             ; Display ASCII data.
                                                         ;
0148    F1                    POP     AF                 ; Restore data
0149    E6 0F                 AND     0FH                ;
014B    CD 0153               CALL    TRSDAT             ; Translate lower 4bit to ASCII
                                                         code.
014E    4F                    LD      C,A                ; Load display data to C register.
014F    CD EB0C               CALL    CONOUT             ; Display ASCII data
0152    C9                    RET                        ;
                                                         ;
                      ;       ************************************************
                      ;               TRANSLATE BINARY DATA TO ASCII DATA
                      ;       ************************************************
                      ;
                      ;       NOTE :
                      ;               Translate binary data to ASCII code.
                      ;
                      ;               <> entry parameter         <>
                      ;                       A : binary data.
                      ;               <> return parameter        <>
                      ;                       A : ASCII data.
0153                  TRSDAT:                            ;
0153    FE 0A                 CP      0AH                ; Data < 0AH?
0155    30 03                 JR      NC,TRSDAT5         ; No.
                                                         ;
0157    F6 30                 OR      30H                ; Data < 0AH
0159    C9                    RET                        ;
015A                  TRSDAT5:                           ;
015A    0E 09                 LD      C,09H              ; Data >= 0AH
015C    91                    SUB     C                  ;
015D    F6 40                 OR      40H                ;
015F    C9                    RET                        ;
                                                         ;
                              END
```

(10) SAMPLE 10. POWEROFF BACKLIGHT

```
                                ;       ********************************************************
                                ;               AUTO POWER OFF & AUTO BACKLIGHT OFF
                                ;       ********************************************************
                                ;
                                ;       NOTE:
                                ;                This sample program sets auto power off time and auto
                                ;               backlight off time and turns them off when the time is up
                                ;               unless any key is pressed.
                                ;
                                ;       <> assemble condition   <>
                                ;
                                        .Z80
                                ;
                                ;       <> loading address      <>
                                ;
                                        .PHASE  100H
                                ;
                                ;       <> constant values      <>
                                ;
EB06                            CONST           EQU     0EB06H  ; CONST entry address
EB09                            CONIN           EQU     0EB09H  ; CONIN entry address
EB7E                            POWEROFF        EQU     0EB7EH  ; POWEROFF entry address
EBA2                            BACKLIGHT       EQU     0EBA2H  ; BACKLIGHT entry address
                                ;
F021                            ATSOTIME        EQU     0F021H  ;
F02D                            TIMER0          EQU     0F02DH  ;
F5F9                            TIMEEND         EQU     0F5F9H  ;
F023                            ELOFTIME        EQU     0F023H  ;
F3B7                            ELOFFEND        EQU     0F3B7H  ;
                                ;
1000                            MAINSP          EQU     01000H  ; Stack pointer
                                ;
                                ;       ********************************************************
                                ;               MAIN PROGRAM
                                ;       ********************************************************
                                ;
0100                            START:
0100    31 1000                         LD      SP,MAINSP       ; Set stack pointer
                                ;
                                                                ; Set auto power off time
0103    2A F021                         LD      HL,(ATSOTIME)   ; Get auto power off time
0106    ED 5B F02D                      LD      DE,(TIMER0)     ; Get 1 sec counter
010A    19                              ADD     HL,DE           ;
010B    22 F5F9                         LD      (TIMEEND),HL    ; Set auto power off time
                                ;
                                                                ; Set auto backlight off time
010E    2A F023                         LD      HL,(ELOFTIME)   ; Get auto backlight off time
0111    ED 5B F02D                      LD      DE,(TIMER0)     ; Get 1 sec counter
0115    19                              ADD     HL,DE           ;
0116    22 F3B7                         LD      (ELOFFEND),HL   ; Set auto backlight off time
                                ;
0119                            LOOP:
0119    CD EB06                         CALL    CONST           ; Check key in ?
011C    3C                              INC     A               ; If Areg = 0FFH
011D    28 26                           JR      Z,KEY_IN        ;  Then key in.
                                ;
                                ;       Application inserts the process in this part
                                ;        which needs to disable AUTO POWER OFF or AUTO BACKLIGHT OFF
                                ;        with CONST.
                                ;
                                                                ; Else not key in.
011F    2A F5F9                         LD      HL,(TIMEEND)    ; Get auto power off time
0122    ED 5B F02D                      LD      DE,(TIMER0)     ; Get 1 sec counter
0126    B7                              OR      A               ; Reset carry flag
0127    ED 52                           SBC     HL,DE           ; (TIMEEND)-(TIMER0)
0129    30 05                           JR      NC,BKLICHK      ;  > 0 then not power off
                                ;
012B    0E 00                           LD      C,00H           ; Set continue mode power off
012D    CD EB7E                         CALL    POWEROFF        ; Power off
                                ;
0130                            BKLICHK:
0130    2A F3B7                         LD      HL,(ELOFFEND)   ; Get auto backlight off time
0133    ED 5B F02D                      LD      DE,(TIMER0)     ; Get 1 sec counter
0137    B7                              OR      A               ; Reset carry flag
0138    ED 52                           SBC     HL,DE           ; (ELOFFEND)-(TIMER0)
013A    30 05                           JR      NC,SLEEP        ; > 0 then not backlight off
                                ;
013C    0E 00                           LD      C,00H           ; Set backlight off
013E    CD EBA2                         CALL    BACKLIGHT       ; Backlight off
                                ;
0141                            SLEEP:
0141    76                              HALT                    ;
0142    C3 0119                         JP      LOOP            ;
                                ;
0145                            KEY_IN:
0145    CD EB09                         CALL    CONIN           ;
0148    0E 01                           LD      C,01H           ;
014A    CD EBA2                         CALL    BACKLIGHT       ; BACKLIGHT ON
014D    C9                              RET
                                END
```

(11) SAMPLE 11. CONTINUE

```
                            ;       ***************************************************
                            ;               BIOS CONTINUE SAMPLE PROGRAM
                            ;       ***************************************************
                            ;
                            ;       NOTE: This program determines the continue mode or the
                            ;             restart mode.
                            ;
                            ;       <> assemble condition   <>
                            ;
                                    .Z80
                            ;
                            ;       <> loading address      <>
                            ;
                                    .PHASE  100H
                            ;
                            ;       <> constant values      <>
                            ;
EB03                        WBOOT           EQU     0EB03H
EB09                        CONIN           EQU     0EB09H
EB0C                        CONOUT          EQU     0EB0CH
EB8A                        CONTINUE        EQU     0EB8AH
E896                        TOUCH           EQU     0E896H
                            ;
1000                        MAINSP          EQU     01000H
                            ;
                            ;       ***************************************************
                            ;               MAIN PROGRAM
                            ;       ***************************************************
                            ;
0100                        START:
0100    31 1000                     LD      SP,MAINSP       ; Set stack pointer
                            ;
0103    0E 0C                       LD      C,0CH           ; Clear screen
0105    CD EB0C                     CALL    CONOUT          ;
                            ;
0108    21 014E                     LD      HL,CONTMSG      ; Display 'CONTINUE'
010B    CD 0142                     CALL    DSPMSG          ;
                            ;
010E    0E 01                       LD      C,01H           ; Make 1 touch key
0110    11 0158                     LD      DE,KEYBLK1      ; 'END' key
0113    CD E896                     CALL    TOUCH           ;
0116                        LOOP1:
0116    0E 02                       LD      C,02H           ; Make 2 touch keys
0118    11 0163                     LD      DE,KEYBLK2      ; 'ON','OFF' key
011B    CD E896                     CALL    TOUCH           ;
                            ;
011E    CD EB09                     CALL    CONIN           ; Wait key in
0121    FE 0D                       CP      0DH             ; 'END' key ?
0123    28 1A                       JR      Z,ENDCONT       ; Yes,then end
0125    11 1707                     LD      DE,01707H       ; Initial attribute
0128    0E 01                       LD      C,1             ;
012A    FE 01                       CP      01H             ; Check touched key
012C    28 04                       JR      Z,SETATT        ; If 'ON' ,then jump
012E    11 0717                     LD      DE,00717H       ; Else 'OFF' touched
0131    0D                          DEC     C               ; Set Creg=0
0132                        SETATT:
0132    7A                          LD      A,D             ;
0133    32 0168                     LD      (KEYBLK2+5),A   ; Set 'ON' key attribute
0136    7B                          LD      A,E             ;
0137    32 0172                     LD      (KEYBLK2+15),A  ; Set 'OFF' key attribute
                            ;
013A    CD EB8A                     CALL    CONTINUE        ; Set continue flag  or reset
013D    18 D7                       JR      LOOP1           ; Rewrite key
                            ;
013F                        ENDCONT:
013F    C3 EB03                     JP      WBOOT           ; Jump WBOOT
                            ;
                            ;
                            ;       ***************************************************
                            ;               DISPLAY STRING MESSAGE UNTIL FIND 00H
                            ;       ***************************************************
                            ;
                            ;       NOTE:
                            ;
                            ;       <> entry parameter      <>
                            ;               HL : top address of string message
                            ;       <> return parameter     <>
                            ;               NONE
                            ;       <> preserved registers  <>
                            ;               NONE
                            ;
0142                        DSPMSG:
0142    7E                          LD      A,(HL)          ; Get a data of message
0143    B7                          OR      A               ; Check end mark
0144    C8                          RET     Z               ; If find 00H then return
                            ;
0145    E5                          PUSH    HL              ; Save pointer to message
0146    4F                          LD      C,A             ; Set data to Creg
0147    CD EB0C                     CALL    CONOUT          ; Display data
014A    E1                          POP     HL              ; Restore pointer
014B    23                          INC     HL              ; Pointer update
014C    18 F4                       JR      DSPMSG          ;
```

```
                                        ;
                                        ;       ***********************************************************
                                        ;               DATA AREA
                                        ;       ***********************************************************
                                        ;
014E                                    CONTMSG:
014E        1B 32 43 4F                         DB      1BH,'2','CONTINUE',00H          ; Cursor off &
0152        4E 54 49 4E
0156        55 45 00
                                                                                        ;   'CONTINUE'
0158                                    KEYBLK1:
0159        04 0E 02 01                         DB      4,14,2,1,00H,07H,00H,'END'
015D        0D 07 00 45
0161        4E 44
                                        ;
0163                                    KEYBLK2:
0163        03 03 02 01                         DB      3,3,2,1,01H,07H,00H,' ON'
0167        01 07 00 20
016B        4F 4E
016D        04 03 02 01                         DB      4,3,2,1,02H,07H,00H,'OFF'
0171        02 07 00 4F
0175        46 46
                                        ;
                                                END
```

(12) SAMPLE 12. BARCODE

```
                        ;       ************************************************************
                        ;               BIOS BARCODE SAMPLE PROGRAM
                        ;       ************************************************************
                        ;
                        ;       NOTE:
                        ;               This program reads the data from the barcode reader
                        ;               and displays the data.
                        ;
                        ;       <> assemble condition <>
                        ;
                                .Z80
                        ;
                        ;       <> loading address <>
                        ;
                                .PHASE 100H
                        ;
                        ;       <> constant values <>
                        ;
EB03                    WBOOT           EQU     0EB03H
EB06                    CONST           EQU     0EB06H
EB09                    CONIN           EQU     0EB09H
EB0C                    CONOUT          EQU     0EB0CH
EB72                    READSW          EQU     0EB72H
EB8D                    BARCODE         EQU     0EB8DH
EB96                    TOUCH           EQU     0EB96H
EB9C                    KEYIN           EQU     0EB9CH
                        ;
1000                    MAINSP          EQU     01000H
                                ************************************************************
                        ;               MAIN PROGRAM
                                ************************************************************
                        ;
0100                    START:
0100    31 1000                 LD      SP,MAINSP       ; Set the stack pointer
0103    0E 1B                   LD      C,1BH           ;
0105    CD EB0C                 CALL    CONOUT          ; CONOUT ESC + '2'   cursor OFF
0108    0E 32                   LD      C,32H           ;
010A    CD EB0C                 CALL    CONOUT          ;
                        ;
010D    0E 02                   LD      C,02H           ; Read dip SW of EHT-10,10/2
010F    CD EB72                 CALL    READSW          ;
0112    E6 80                   AND     10000000B       ; Check the bit 7
0114    B7                      OR      A               ; If bit 7 = 1 then the machine is
                                                        EHT-10/2
0115    20 0A                   JR      NZ,KEYBOD       ; Jump to the EHT-10/2 routine
                        ;
0117                    TCHKEY:                         ; EHT-10 Touch key routine
0117    0E 01                   LD      C,01H           ; Make 1 touch key block
0119    11 01C3                 LD      DE,KEYBLK       ; Set the key block descriptor address
011C    CD EB96                 CALL    TOUCH           ; Display the key block
011F    18 1F                   JR      BAR1            ;

0121                    KEYBOD:                         ; EHT-10/2
0121    06 03                   LD      B,03H           ; Change the keyboard to the alphabet mode
0123    CD EB9C                 CALL    KEYIN           ;
0126    0E 1B                   LD      C,1BH           ; Set cursor
0128    CD EB0C                 CALL    CONOUT          ; ESC
012B    0E 3D                   LD      C,'='           ; =
012D    CD EB0C                 CALL    CONOUT          ; '='
0130    0E 23                   LD      C,4+01FH        ; +
0132    CD EB0C                 CALL    CONOUT          ; 4
0135    0E 20                   LD      C,1+01FH        ; +
0137    CD EB0C                 CALL    CONOUT          ; 1
013A    21 0160                 LD      HL,ENDMSG       ; Display 'press return key to exit'
013D    CD 0187                 CALL    DSPMSG          ;
                        ;
0140                    BAR1:                           ;
0140    CD 0175                 CALL    BAROPN          ; Open the barcode reader
0143                    BAR2:                           ;
0143    0E 03                   LD      C,03H           ; Read the barcode status
0145    CD EB8D                 CALL    BARCODE         ;
0148    78                      LD      A,B             ; Check BCreg.
0149    B1                      OR      C               ; BC is the number of data
014A    C4 0157                 CALL    NZ,DSPDAT       ; If any data exists,then display it
                        ;
014D    CD EB06                 CALL    CONST           ; Check key in
0150    B7                      OR      A               ; If Areg. = 0
0151    28 F0                   JR      Z,BAR2          ; Then not key in
                        ;
0153    CD EB09                 CALL    CONIN           ; Get a key in data
0156    FE 0D                   CP      0DH             ; If code is 0DH
0158    20 E9                   JR      NZ,BAR2         ; Then END
015A    CD 0181                 CALL    BARCLS          ; Close the barcode reader
015D    CD EB03                 CALL    WBOOT           ;
                        ;
                        ;
0160                    ENDMSG:
0160    50 72 65 73             DB      'Press RETURN to exit',00H
0164    73 20 52 45
0168    54 55 52 4E
016C    20 74 6F 20
0170    65 78 69 74
```

```
0174      00
                                  ;
                                  ;
                                  ;
                                  ;       ****************************************************************
                                  ;               OPEN THE BARCODE READER
                                  ;       ****************************************************************
                                  ;
                                  ;       NOTE:
                                  ;               This subroutine opens the barcode reader.
                                  ;
                                  ;               CODE TYPE  : JAN / EAN / UPC-A / UPC-E
                                  ;               OPTION     : BUZZER        ON
                                  ;                            DELIMITER     ON
                                  ;                            LED CTRL      ON
                                  ;                            ZERO ADD      OFF   (only UPC-E)
                                  ;
                                  ;       <> entry parameter <>
                                  ;
                                  ;               none
                                  ;
                                  ;       <> return parameter <>
                                  ;
                                  ;               none
                                  ;
                                  ;       <> preserved registers  <>
                                  ;
                                  ;               none
                                  ;
0175                              BAROPN:
0175      0E 00                           LD      C,00H           ;
0177      3E 01                           LD      A,01H           ; Code type  JAN / EAN / UPC-A / UPC-E
0179      16 00                           LD      D,00H           ; Set Option
017B      1E 0D                           LD      E,0DH           ; Delimiter is 0DH
017D      CD EB8D                         CALL    BARCODE         ; Open
0180      C9                              RET                     ;

                                  ;       ****************************************************************
                                  ;               CLOSE THE BARCODE READER
                                  ;       ****************************************************************
                                  ;
                                  ;       NOTE:
                                  ;               This subroutine closes the barcode reader.
                                  ;
                                  ;       <> entry parameter <>
                                  ;
                                  ;               none
                                  ;
                                  ;       <> return parameter <>
                                  ;
                                  ;               none
                                  ;
                                  ;       <> preserved registers <>
                                  ;
                                  ;               none
                                  ;
0181                              BARCLS:
0181      0E 01                           LD      C,01H           ; Close function number
0183      CD EB5D                         CALL    BARCODE         ; Close
0186      C9                              RET                     ;


                                  ;       ****************************************************************
                                  ;               DISPLAY THE READ DATA
                                  ;       ****************************************************************
                                  ;
                                  ;       NOTE:
                                  ;               This subroutine reads the data from the buffer,
                                  ;               and displays the data on the LCD.
                                  ;
                                  ;       <> entry parameter <>
                                  ;
                                  ;               none
                                  ;
                                  ;       <> return parameter <>
                                  ;
                                  ;               none
                                  ;
                                  ;       <> preserved registers <>
                                  ;
                                  ;               ALL
                                  ;
0187                              DSPDAT:
0187      F5                              PUSH    AF              ; Save registers
0188      C5                              PUSH    BC              ;
0189      D5                              PUSH    DE              ;
018A      E5                              PUSH    HL              ;
018B      0E 0B                           LD      C,0BH           ; Set cursor  (1,1)
018D      CD EB0C                         CALL    CONOUT          ;
0190      06 14                           LD      B,20            ; Number of characters to display
                                  ;
0192                              LOOP:                           ;
```

```
0192    0E 02                       LD      C,02H           ; Read function number
0194    05                          DEC     B               ;
0195    C5                          PUSH    BC              ; Save counter Breg.
0196    CD EB8D                     CALL    BARCODE         ; Read the data
0199    C1                          POP     BC              ; Restore Breg.
019A    28 16                       JR      Z,ENDDSP        ; If ZF = 1, then no data in the buffer
019C    FE 00                       CP      00H             ; Check the delimiter
019E    28 08                       JR      Z,DSPSP         ;
01A0    4F                          LD      C,A             ; Display data
01A1    C5                          PUSH    BC              ; Save counter
01A2    CD EB0C                     CALL    CONOUT          ;
01A5    C1                          POP     BC
01A6    18 EA                       JR      LOOP            ;
                            ;
01A8                        DSPSP:                          ;
01A8    0E 20                       LD      C,20H           ; Display ' ' to clear old data
01AA    C5                          PUSH    BC              ; Save counter
01AB    CD EB0C                     CALL    CONOUT          ;
01AE    C1                          POP     BC              ; Restore Breg.
01AF    05                          DEC     B               ;
01B0    20 F6                       JR      NZ,DSPSP        ;
                            ;
01B2                        ENDDSP:                         ;
01B2    E1                          POP     HL              ; Restore registers
01B3    D1                          POP     DE              ;
01B4    C1                          POP     BC              ;
01B5    F1                          POP     AF              ;
01B6    C9                          RET                     ;
                            ;
                            ;
                            ;       **********************************************************
                            ;                   DISPLAY STRING
                            ;       **********************************************************
                            ;
                            ;       NOTE:
                            ;                   Display string data to the CON: until find 00H.
                            ;
                            ;       <> entry parameter <>
                            ;
                            ;                   HL = Top address of string data.
                            ;
                            ;       <> return parameter <>
                            ;
                            ;                   none
                            ;
                            ;       <> preserved register <>
                            ;
                            ;                   none
                            ;
01B7                        DSPMSG:
01B7    7E                          LD      A,(HL)          ; Get data
01B8    B7                          OR      A               ; Check terminater
01B9    C8                          RET     Z               ; If find terminater then return
                            ;
01BA    E5                          PUSH    HL              ;
01BB    4F                          LD      C,A             ;
01BC    CD EB0C                     CALL    CONOUT          ; Display data
01BF    E1                          POP     HL              ;
01C0    23                          INC     HL              ; Update pointer
01C1    18 F4                       JR      DSPMSG          ;


                            ;       **********************************************************
                            ;                   DATA OF KEY BLOCK DESCRIPTOR
                            ;       **********************************************************
                            ;
01C3                        KEYBLK:
01C3    04 0E 02 01                 DB      04,14,2,1,0DH,07H,00,'END'
01C7    0D 07 00 45
01CB    4E 44
                            ;
                            ;
                            ;
                            END
```

(13) SAMPLE 13. TCAM

```
                          ;       ***************************************************
                          ;               BIOS TCAM SAMPLE PROGRAM
                          ;       ***************************************************
                          ;
                          ;       NOTE:
                          ;                 This sample program is executed only on EHT-10.
                          ;                 This sample program receives a file with FILINK.
                          ;                 If same named program already exists in RAM disk,
                          ;               then old file is erased.
                          ;
                          ;       <> assemble condition    <>
                          ;
                                  .Z80
                          ;
                          ;       <> loading address       <>
                          ;
                                  .PHASE  100H
                          ;
                          ;       <> constant values       <>
                          ;
EB03                      WBOOT           EQU     0EB03H
EB09                      CONIN           EQU     0EB09H
EB0C                      CONOUT          EQU     0EB0CH
EB90                      TCAM            EQU     0EB90H
EB96                      TOUCH           EQU     0EB96H
EBA5                      INFORM          EQU     0EBA5H
                          ;
0005                      BDOS            EQU     00005H
1000                      MAINSP          EQU     01000H
                          ;
                          ;       ***************************************************
                          ;               MAIN PROGRAM
                          ;       ***************************************************
                          ;
0100                      START:
0100    31 1000                   LD      SP,MAINSP          ; Set stack pointer
                          ;
0103    0E 1B                     LD      C,1BH              ; Clear screen
0105    CD EB0C                   CALL    CONOUT             ; ESC + '2'
0108    0E 32                     LD      C,'2'              ;
010A    CD EB0C                   CALL    CONOUT             ;
                          ;
010D    21 0293                   LD      HL,SENDMSG         ; Display 'SEND'
0110    CD 01D1                   CALL    DSPMSG             ;
                          ;
0113    0E 09                     LD      C,09H              ; Get address of TCAMAREA
0115    CD EBA5                   CALL    INFORM             ; HL = TCAMAREA address
0118    22 02B9                   LD      (TCAMADR),HL       ; Save TCAMAREA address
                          ;
011B    3E 01                     LD      A,01H              ; Function connect
011D    06 01                     LD      B,01H              ; Receive file
011F    CD EB90                   CALL    TCAM               ; Connect
0122    38 14                     JR      C,TCAMERR          ; If CY=1 then error
                          ;
0124    CD 015D                   CALL    RCVFILE            ; Receive file to RAM disk
0127    38 0F                     JR      C,TCAMERR          ; If CY=1 then error occured in RCVFILE
                          ;
0129    3E 04                     LD      A,04H              ; Function disconnect
012B    CD EB90                   CALL    TCAM               ; Disconnect
012E    38 08                     JR      C,TCAMERR          ; If CY=1 then error
                          ;
0130    21 027E                   LD      HL,ENDMSG          ; Display 'Finished'
0133    CD 01D1                   CALL    DSPMSG             ;
0136    18 17                     JR      ENDPRO             ;
                          ;
0138                      TCAMERR:
0138    F5                        PUSH    AF                 ; Save error code
0139    0E 0C                     LD      C,0CH              ; Clear screen
013B    CD EB0C                   CALL    CONOUT             ;
013E    F1                        POP     AF                 ; Restore error code
                          ;
013F    21 01DD                   LD      HL,ERRTBL          ; Set error table address
0142    5F                        LD      E,A                ; Get error code
0143    16 00                     LD      D,00H              ; Dreg = 0
0145    CB 23                     SLA     E                  ; Shift left  E=E x 2
0147    19                        ADD     HL,DE              ; Make error message address
0148    5E                        LD      E,(HL)             ; Save error message address
0149    23                        INC     HL                 ;
014A    56                        LD      D,(HL)             ;
014B    EB                        EX      DE,HL              ; HL is point to error message
014C    CD 01D1                   CALL    DSPMSG             ; Display message
                          ;
014F                      ENDPRO:
014F    0E 01                     LD      C,01H              ; Make 1 touch key block
0151    11 02AB                   LD      DE,KEYBLK          ; Set key block
0154    CD EB96                   CALL    TOUCH              ; Make 'Press' key
0157    CD EB09                   CALL    CONIN              ; Wait for key in
015A    C3 EB03                   JP      WBOOT              ; End program
```

```
                          ;
                          ;       ****************************************************
                          ;               RECEIVE DATA & MAKE FILE
                          ;       ****************************************************
                          ;
                          ;       NOTE:
                          ;               Receive data and write it to file.
                          ;
                          ;       <> entry parameter        <>
                          ;               NONE
                          ;       <> return parameter       <>
                          ;               CY = 1 : error occured in this routine
                          ;                    A : error code
                          ;               CY = 0 : no error
                          ;       <> preserved registers    <>
                          ;               NONE
                          ;
015D                      RCVFILE:
015D    21 0298                   LD      HL,RCVMSG       ; Display 'Receiving'
0160    CD 01D1                   CALL    DSPMSG          ;

0163    3E 01                     LD      A,01H           ; Set A:
0165    32 02B8                   LD      (FCB),A         ; Set drive code
                          ;
0168    01 000B                   LD      BC,11           ; Number of filename character
016B    2A 0298                   LD      HL,(TCAMADR)    ; Received file name area
016E    11 02BC                   LD      DE,FCB+1        ; FCB file name area
0171    ED B0                     LDIR                    ; Transmit file name
0173    AF                        XOR     A               ; A=0
0174    32 02C7                   LD      (FCB+12),A      ;
0177    32 02D8                   LD      (FCB+32),A      ; Set 'cr' 0
                          ;
017A    0E 1A                     LD      C,1AH           ; Function number
017C    11 02E0                   LD      DE,DMA          ;
017F    CD 0005                   CALL    BDOS            ; Set DMA address
                          ;
0182    0E 11                     LD      C,11H           ; Function number
0184    11 02B8                   LD      DE,FCB          ; Set FCB address
0187    CD 0005                   CALL    BDOS            ; File directory search
018A    3C                        INC     A               ; Find?
018B    28 08                     JR      Z,ENDDEL        ; No. Then not file delete
                          ;
018D    0E 13                     LD      C,13H           ; Delete file
018F    11 02B8                   LD      DE,FCB          ; Set FCB address
0192    CD 0005                   CALL    BDOS            ; Delete
                          ;
0195                      ENDDEL:
0195    0E 16                     LD      C,16H           ; Make new file
0197    11 02B8                   LD      DE,FCB          ; Set FCB address
019A    CD 0005                   CALL    BDOS            ; Make
019D    3C                        INC     A               ; Make success ?
019E    20 04                     JR      NZ,STARTW       ; Yes. Then continue
01A0    37                        SCF                     ; Set carry flag for error
01A1    3E 09                     LD      A,09H           ; Set error code
01A3    C9                        RET                     ;
                          ;
01A4                      STARTW:
01A4    0E 2E                     LD      C,'.'           ; Display '.' by 128 received
01A6    CD EB0C                   CALL    CONOUT          ;
                          ;
01A9    3E 03                     LD      A,03H           ; Function receive
01AB    21 02E0                   LD      HL,DMA          ; Set buffer address
01AE    CD EB90                   CALL    TCAM            ; Receive
01B1    D8                        RET     C               ; If CY=1 then error occured
01B2    78                        LD      A,B             ; Check BC zero.
01B3    B1                        OR      C               ; If BCreg=0
01B4    28 0B                     JR      Z,CLSFL         ;   then receive end
                          ;
01B6    0E 15                     LD      C,15H           ; Function sequential write
01B8    11 02B8                   LD      DE,FCB          ; Set FCB address
01BB    CD 0005                   CALL    BDOS            ; Write 1 sector
01BE    B7                        OR      A               ; Check Areg.
01BF    28 E3                     JR      Z,STARTW        ; Loop
                          ;
01C1                      CLSFL:
01C1    F5                        PUSH    AF              ; If Areg<>0 then error
01C2    0E 10                     LD      C,10H           ; Function file close
01C4    11 02B8                   LD      DE,FCB          ; Set FCB address
01C7    CD 0005                   CALL    BDOS            ; Close file
01CA    F1                        POP     AF              ;
01CB    B7                        OR      A               ; Check Areg.
01CC    C8                        RET     Z               ; A=0 then no error
01CD    3E 0A                     LD      A,10            ; Error code
01CF    37                        SCF                     ; Set carry flag
01D0    C9                        RET                     ;
```

```
                        ;       ******************************************************
                        ;               DISPLAY MESSAGE
                        ;       ******************************************************
                        ;
                        ;       NOTE:
                        ;               Display message string until find 00H.
                        ;
                        ;       <> entry parameter      <>
                        ;               HL : Top address of message
                        ;       <> return parameter     <>
                        ;               NONE
                        ;       <> preserved registers  <>
                        ;               NONE
                        ;
01D1                    DSPMSG:
01D1    7E                      LD      A,(HL)          ; Get a data
01D2    B7                      OR      A               ; Check end mark 00H
01D3    C8                      RET     Z               ; If 00H then return
                        ;
01D4    E5                      PUSH    HL              ; Save pointer for address
01D5    4F                      LD      C,A             ; Set a data
01D6    CD EB0C                 CALL    CONOUT          ; Display a message
01D9    E1                      POP     HL              ; Restore pointer
01DA    23                      INC     HL              ; Update pointer
01DB    18 F4                   JR      DSPMSG          ;
                        ;
                        ;
                        ;       ******************************************************
                        ;               DATA AREA
                        ;       ******************************************************
                        ;
01DD                    ERRTBL:
01DD    01F3                    DW      PARAERR         ; Dummy
01DF    01F3                    DW      PARAERR         ; Parameter error
01E1    0203                    DW      OPENERR         ; Already open error
01E3    0210                    DW      NOPNERR         ; Not open error
01E5    0219                    DW      FENDERR         ; Forced end
01E7    0224                    DW      BOVFERR         ; Buffer over flow error
01E9    0235                    DW      TIMEERR         ; Time out error
01EB    023E                    DW      PROTERR         ; Protcol error
01ED    024C                    DW      COMMERR         ; Communication error
01EF    0260                    DW      FLMKERR         ; File make error
01F1    0272                    DW      FLWTERR         ; File write error
                        ;
01F3                    PARAERR:
01F3    50 61 72 61             DB      'Parameter error',00H
01F7    6D 65 74 65
01FB    72 20 65 72
01FF    72 6F 72 00
0203                    OPENERR:
0203    41 6C 72 65             DB      'Already open',00H
0207    61 64 79 20
020B    6F 70 65 6E
020F    00
0210                    NOPNERR:
0210    4E 6F 74 20             DB      'Not open',00H
0214    6F 70 65 6E
0218    00
0219                    FENDERR:
0219    46 6F 72 63             DB      'Forced end',00H
021D    65 64 20 65
0221    6E 64 00
0224                    BOVFERR:
0224    42 75 66 66             DB      'Buffer over flow',00H
0228    65 72 20 6F
022C    76 65 72 20
0230    66 6C 6F 77
0234    00
0235                    TIMEERR:
0235    54 69 6D 65             DB      'Time out',00H
0239    20 6F 75 74
023D    00
023E                    PROTERR:
023E    50 72 6F 74             DB      'Protcol error',00H
0242    63 6F 6C 20
0246    65 72 72 6F
024A    72 00
024C                    COMMERR:
024C    43 6F 6D 6D             DB      'Communication error',00H
0250    75 6E 69 63
0254    61 74 69 6F
0258    6E 20 65 72
025C    72 6F 72 00
0260                    FLMKERR:
0260    46 69 6C 65             DB      'File can not make',00H
0264    20 63 61 6E
0268    20 6E 6F 74
026C    20 6D 61 6B
0270    65 00
0272                    FLWTERR:
0272    57 72 69 74             DB      'Write error',00H
0276    65 20 65 72
027A    72 6F 72 00
                        ;
```

```
027E                        ENDMSG:
027E     0C 46 69 6E                DB      0CH,'Finished',00H
0282     69 73 68 65
0286     64 00
0288                        RCVMSG:
0288     0C 52 65 63                DB      0CH,'Receiving',00H
028C     65 69 76 69
0290     6E 67 00
                            ;
0293                        SENDMSG:
0293     0C 53 65 6E                DB      0CH,'Send file    from HOST',00H
0297     64 20 66 69
029B     6C 65 20 20
029F     20 20 66 72
02A3     6F 6D 20 48
02A7     4F 53 54 00
02AB                        KEYBLK:
02AB     03 03 03 01                DB      3,3,3,1,00H,07H,00H,'Press'
02AF     00 07 00 50
02B3     72 65 73 73
                            ;
02B7                        ERMGADR:
02B7                                DS      2               ; Error message address save area
02B9                        TCAMADR:
02B9                                DS      2               ; TCAMAREA address area
                            ;
02BB                        FCB:
02BB                                DS      37              ; FCB area
02E0                        DMA:
02E0                                DS      128             ; DMA area
                            ;
                                    END

Macros:

Symbols:
0005    BDOS            0224    BOVFERR         01C1    CLSFL
024C    CONFERR         EB09    CONIN           EB0C    CONOUT
02E0    DMA             01D1    DSPMSG          0195    ENDDBL
027E    ENDMSG          014F    ENDPRO          02B7    ERMGADR
010D    ERRTBL          02BB    FCB             0219    FENDERR
0260    FLMXERR         0272    FLWTERR         EBA5    INFORM
02AB    KEYBLK          1000    MAINSP          0210    NOPNERR
0203    OPENERR         01F3    PARAERR         023E    PROTERR
015D    RCVFILE         0288    RCVMSG          0293    SENDMSG
0100    START           01A4    STARTW          EB90    TCAM
02B9    TCAMADR         0138    TCAMERR         0235    TIMERR
EB96    TOUCH           EB03    WBOOT


No Fatal error(s)
```

(14) SAMPLE 14. GRAPHICS (LINE)

```
                                ;       ************************************************************
                                ;               BIOS GRAPHICS(LINE) SAMPLE PROGRAM
                                ;       ************************************************************
                                ;
                                ;       NOTE:
                                ;               This sample program is executed only on EHT-10.
                                ;               This program displays the line pattern until "END"
                                ;              key is pressed.
                                ;
                                ;       <> assemble condition   <>
                                ;
                                        .Z80
                                ;
                                ;       <> loading address      <>
                                ;
                                        .PHASE  100H
                                ;
                                ;       <> constant values      <>
                                ;
EB03                            WBOOT           EQU     0EB03H
EB06                            CONST           EQU     0EB06H
EB09                            CONIN           EQU     0EB09H
EB0C                            CONOUT          EQU     0EB0CH
EB93                            GRAPHICS        EQU     0EB93H
EB96                            TOUCH           EQU     0EB96H
                                ;
1000                            MAINSP          EQU     01000H
                                ;
                                ;       ************************************************************
                                ;               MAIN PROGRAM
                                ;       ************************************************************
                                ;
0100                            START:
0100    31 1000                         LD      SP,MAINSP       ; Set stack pointer
                                ;
0103    0E 1B                           LD      C,1BH           ; Cursor off
0105    CD EB0C                         CALL    CONOUT          ; ESC + '2'
0108    0E 32                           LD      C,'2'           ;
010A    CD EB0C                         CALL    CONOUT          ;
                                ;
010D    0E 00                           LD      C,00H           ; Set init function
010F    CD EB93                         CALL    GRAPHICS        ; Package initialize
                                ;
0112    06 0C                           LD      B,12            ; Display 12 lines
0114    0E 05                           LD      C,05H           ; Line function
0116    21 0148                         LD      HL,PACK1        ; Set package address
0119                            LOOP:
0119    C5                              PUSH    BC              ; Save counter
011A    E5                              PUSH    HL              ; Save package address
011B    CD EB93                         CALL    GRAPHICS        ; Display line
011E    E1                              POP     HL              ; Restore package address
011F    C1                              POP     BC              ; Restore counter
0120    11 000C                         LD      DE,12           ;
0123    19                              ADD     HL,DE           ; Make next package
0124    10 F3                           DJNZ    LOOP
                                ;
0126    0E 01                           LD      C,01H           ; Make 1 touch key
0128    11 013E                         LD      DE,KEYBLK       ; Make 'END' key
012B    CD EB96                         CALL    TOUCH           ;
012E    CD EB09                         CALL    CONIN           ; Wait for key in
                                ;
0131    0E 1B                           LD      C,1BH           ; Cursor on
0133    CD EB0C                         CALL    CONOUT          ; ESC + '3'
0136    0E 33                           LD      C,'3'           ;
0138    CD EB0C                         CALL    CONOUT          ;
013B    C3 EB03                         JP      WBOOT           ;
                                ;
                                ;       ************************************************************
                                ;               DATA AREA
                                ;       ************************************************************
                                ;
013E                            KEYBLK:
013E    04 0E 02 01                     DB      4,14,2,1,00H,07H,00H,'END'
0142    00 07 00 45
0146    4E 44
                                ;
0148                            PACK1:
0148    000A 0064                       DW      10,100,50,100
014C    0032 0064
0150    01 00                           DB      1,0
0152    FFFF                            DW      0FFFFH
                                ;
0154    000A 003C                       DW      10,60,50,60
0158    0032 003C
015C    01 00                           DB      1,0
015E    FFFF                            DW      0FFFFH
                                ;
0160    000A 0064                       DW      10,100,10,60
0164    000A 003C
0168    01 00                           DB      1,0
016A    FFFF                            DW      0FFFFH
                                ;
```

```
016C     0032 0064          DW      50,100,50,60
0170     0032 003C
0174     01 00              DB      1,0
0176     FFFF               DW      0FFFFH

0178     001E 002D          DW      30,45,65,45
017C     0041 002D
0180     01 00              DB      1,0
0182     FFFF               DW      0FFFFH

0184     0041 0055          DW      65,85,65,45
0188     0041 002D
018C     01 00              DB      1,0
018E     FFFF               DW      0FFFFH

0190     0041 0055          DW      65,85,50,100
0194     0032 0064
0198     01 00              DB      1,0
019A     FFFF               DW      0FFFFH

019C     0041 002D          DW      65,45,50,60
01A0     0032 003C
01A4     01 00              DB      1,0
01A6     FFFF               DW      0FFFFH

01A8     001E 002D          DW      30,45,10,60
01AC     000A 003C
01B0     01 00              DB      1,0
01B2     FFFF               DW      0FFFFH

01B4     001E 002D          DW      30,45,30,85
01B8     001E 0055
01BC     01 00              DB      1,0
01BE     AAAA               DW      0AAAAH

01C0     001E 0055          DW      30,85,65,85
01C4     0041 0055
01C8     01 00              DB      1,0
01CA     AAAA               DW      0AAAAH

01CC     001E 0055          DW      30,85,10,100
01D0     000A 0064
01D4     01 00              DB      1,0
01D6     AAAA               DW      0AAAAH

                            END
```

(15) SAMPLE 15. GRAPHICS (CIRCLE)

```
                                ;
                                ;        ************************************************
                                ;                BIOS GRAPHICS(CIRCLE) SAMPLE PROGRAM
                                ;        ************************************************
                                ;
                                ;        NOTE:
                                ;                This sample program is executed only on EHT-10.
                                ;                This program displays the circle pattern until "END"
                                ;                key is pressed.
                                ;
                                ;        <> assemble condition   <>
                                ;
                                         .Z80
                                ;
                                ;        <> loading address      <>
                                ;
                                         .PHASE  100H
                                ;
                                ;        <> constant values      <>
                                ;
EB03                            WBOOT           EQU     0EB03H
EB06                            CONST           EQU     0EB06H
EB09                            CONIN           EQU     0EB09H
EB0C                            CONOUT          EQU     0EB0CH
EB93                            GRAPHICS        EQU     0EB93H
EB96                            TOUCH           EQU     0EB96H
                                ;
1000                            MAINSP          EQU     01000H
                                ;
                                ;        ***************************************************
                                ;                MAIN PROGRAM
                                ;        ***************************************************
                                ;
0100                            START:
0100    31 1000                          LD      SP,MAINSP          ; Set stack pointer
                                ;
0103    0E 1B                            LD      C,1BH              ; Cursor off
0105    CD EB0C                          CALL    CONOUT             ; ESC + '2'
0108    0E 32                            LD      C,'2'              ;
010A    CD EB0C                          CALL    CONOUT             ;
                                ;
010D    0E 0C                            LD      C,0CH              ; Clear screen
010F    CD EB0C                          CALL    CONOUT             ;
                                ;
0112    06 24                            LD      B,36               ; Display counter
0114    0E 06                            LD      C,06H              ; Circle function number
0116                            LOOP:
0116    C5                               PUSH    BC                 ; Save counter
0117    21 0147                          LD      HL,PACK            ; Set package address
011A    CD EB93                          CALL    GRAPHICS           ; Display circles
011D    C1                               POP     BC                 ; Restore counter
011E    2A 014B                          LD      HL,(PACK+4)        ; Get X direction radius
0121    2B                               DEC     HL                 ; Decrement X radius
0122    22 014B                          LD      (PACK+4),HL        ; Set new radius
0125    2A 014D                          LD      HL,(PACK+6)        ; Get Y direction radius
0128    2B                               DEC     HL                 ;
0129    2B                               DEC     HL                 ; Decrement Y radius
012A    22 014D                          LD      (PACK+6),HL        ; Set new radius
012D    10 E7                            DJNZ    LOOP               ; Until Y radius < 1
                                ;
012F    0E 01                            LD      C,01H              ; Make 1 touch key for END
0131    11 0156                          LD      DE,KEYTBL          ; Set a key block data
0134    CD EB96                          CALL    TOUCH              ; Display a key block
0137    CD EB09                          CALL    CONIN              ; Wait for key in
                                ;
013A    0E 1B                            LD      C,1BH              ; Cursor on
013C    CD EB0C                          CALL    CONOUT             ; ESC + '3'
013F    0E 33                            LD      C,'3'              ;
0141    CD EB0C                          CALL    CONOUT             ;
0144    C3 EB03                          JP      WBOOT              ; Jump WBOOT
                                ;
                                ;        ***************************************************
                                ;                DATA AREA
                                ;        ***************************************************
                                ;
0147                            PACK:
0147    002A 004A                        DW      42,74              ; Center coordinate
014B    0028 0048                        DW      40,72              ; X , Y radius
014F    01 00 00                         DB      1,0,0              ; Attribute & flag
0152    0000 0000                        DW      0,0H               ; Start & end angle
                                ;
0156                            KEYTBL:
0156    04 0E 02 01                      4,14,2,1,00H,07H,00H,'END'        ; END key
015A    00 07 00 45
015E    4E 44
                                ;
                                         END
```

(16) SAMPLE 16. GRAPHICS (TILE)

```
                                ;       **********************************************************
                                ;                 BIOS GRAPHICS(TILE) SAMPLE PROGRAM
                                ;       **********************************************************
                                ;
                                ;       NOTE:
                                ;                 This sample program is executed only on EHT-10.
                                ;                 This program displays the tile pattern until "END"
                                ;               key is pressed.
                                ;
                                ;       <> assemble condition    <>
                                ;
                                        .Z80
                                ;
                                ;       <> loading address       <>
                                ;
                                        .PHASE  100H
                                ;
                                ;       <> constant values       <>
                                ;
EB03                            WBOOT           EQU     0EB03H
EB09                            CONIN           EQU     0EB09H
EB0C                            CONOUT          EQU     0EB0CH
EB93                            GRAPHICS        EQU     0EB93H
EB96                            TOUCH           EQU     0EB96H
                                ;
1000                            MAINSP          EQU     01000H
                                ;
                                ;       **********************************************************
                                ;                 MAIN PROGRAM
                                ;       **********************************************************
                                ;
0100                            START:
0100    31 1000                         LD      SP,MAINSP       ; Set stack pointer
                                ;
0103    0E 1B                           LD      C,1BH           ; Cursor off
0105    CD EB0C                         CALL    CONOUT          ; ESC + '2'
0108    0E 32                           LD      C,'2'           ;
010A    CD EB0C                         CALL    CONOUT          ;
                                ;
010D    0E 0C                           LD      C,0CH           ; Clear screen
010F    CD EB0C                         CALL    CONOUT          ;
                                ;
0112    06 04                           LD      B,04H           ; Number of circle
0114    0E 06                           LD      C,06H           ; Circle function
0116                            LOOP1:
0116    C5                              PUSH    BC              ; Save counter
0117    21 016B                         LD      HL,PACK1        ; Set package address
011A    CD EB93                         CALL    GRAPHICS        ; Display circle
011D    C1                              POP     BC              ; Restore counter
011E    11 000A                         LD      DE,10           ;
0121    2A 016F                         LD      HL,(PACK1+4)    ; Get X direction radius
0124    B7                              OR      A               ; Reset carry flag
0125    ED 52                           SBC     HL,DE           ; HL - 10
0127    22 016F                         LD      (PACK1+4),HL    ; Set new X radius
012A    22 0171                         LD      (PACK1+6),HL    ; Set new Y radius
012D    10 E7                           DJNZ    LOOP1           ; Display 4 circle
                                ;
012F    06 05                           LD      B,05H           ; Display 5 times
0131    0E 08                           LD      C,08H           ; Tile function
0133    21 0196                         LD      HL,TILEPTN      ; Set tile pattern address
0136                            LOOP2:
0136    E5                              PUSH    HL              ; Save tile pattern address
0137    C5                              PUSH    BC              ; Save counter
0138    21 017A                         LD      HL,PACK2        ; Set package address
013B    CD EB93                         CALL    GRAPHICS        ;
013E    C1                              POP     BC              ; Restore counter
013F    2A 017A                         LD      HL,(PACK2)      ; Get start address
0142    11 000A                         LD      DE,10           ;
0145    19                              ADD     HL,DE           ; Make next package
0146    22 017A                         LD      (PACK2),HL      ; Set new start X cordinate
                                ;
0149    E1                              POP     HL              ; Restore tile pattern address
014A    11 0004                         LD      DE,04H          ; Skip 4 byte
014D    19                              ADD     HL,DE           ; Make next tile pattern address
014E    22 017F                         LD      (PACK2+5),HL    ; Set new tile pattern address
0151    10 E3                           DJNZ    LOOP2           ;
                                ;
0153    0E 01                           LD      C,01H           ; Make 1 touch key
0155    11 018E                         LD      DE,KEYBLK       ; Set the key block number
0158    CD EB96                         CALL    TOUCH           ; Display a key block
                                ;
015B    CD EB09                         CALL    CONIN           ; Wait for touch
015E    0E 1B                           LD      C,1BH           ; Cursor on
0160    CD EB0C                         CALL    CONOUT          ; ESC + '3'
0163    0E 33                           LD      C,'3'           ;
0165    CD EB0C                         CALL    CONOUT          ;
                                ;
0168    C3 EB03                         JP      WBOOT           ; Jump WBOOT
                                ;
```

```
                                   ;          ************************************************************
                                   ;                  DATA AREA
                                   ;          ************************************************************
                                   ;
016B                               PACK1:
016B      002A 004A                          DW      42,74              ; Center
016F      0028 0028                          DW      40,40              ; X , Y direction radius
0173      01 00 00                           DB      1,0,0              ; Attribute & flag
0176      0000 0000                          DW      0,0                ; Start & end angle
                                   ;
017A                               PACK2:
017A      002B 004A                          DW      43,74              ; Start cordinate
017E      04                                 DB      4                  ; Tile pattern length
017F      0188 019C                          DW      TILEPTN,TILEB      ; Tile pattern address
0183      01                                 DB      1                  ; Attribute
0184      01A8 0080                          DW      WORK,80H           ; Work area
                                   ;
0188                               TILEPTN:                             ; Tile pattern
0188      AA55 AA55                          DW      0AA55H,0AA55H      ;
018C      CCCC 3333                          DW      0CCCCH,03333H      ;
0190      FF99 99FF                          DW      0FF99H,099FFH      ;
0194      0303 3030                          DW      00303H,03030H      ;
0198      1144 1144                          DW      01144H,01144H      ;
                                   ;
019C                               TILEB:
019C      0000                               DW      00000H             ; Back ground tile pattern
                                   ;
019E                               KEYBLE:
019E      04 0E 02 01                        DB      4,14,2,1,0DH,07H,00H,'END'
01A2      0D 07 00 45
01A6      4E 44
                                   ;
01A8                               WORK:
01A8                                         DS      80H                ; Work area
                                   ;
                                             END
```

17) SAMPLE 17. KANJI

```
                        ;
                        ;       ****************************************************
                        ;                BIOS KANJI SAMPLE PROGRAM
                        ;       ****************************************************
                        ;
                        ;       NOTE:
                        ;                This sample program displays and prints out kanji.
                        ;                (code 0400H - 04FEH)
                        ;
                        ;       <> assemble condition <>
                        ;
                                .Z80
                        ;
                        ;       <> loading address <>
                        ;
                                .PHASE  100H
                        ;
                        ;       <> constant values <>
                        ;
EB03                    WBOOT           EQU     0EB03H
EB0C                    CONOUT          EQU     0EB0CH
EB9F                    KANJI           EQU     0EB9FH
                        ;
1000                    MAINSP          EQU     01000H
                        ;
                        ;       ****************************************************
                        ;                MAIN PROGRAM
                        ;       ****************************************************
                        ;
0100                    START:
0100    31 1000                 LD      SP,MAINSP       ; Set the stack pointer
0103    0E 1B                   LD      C,1BH           ;
0105    CD EB0C                 CALL    CONOUT          ; CONOUT ESC + '2'  cursor OFF
0108    0E 32                   LD      C,32H           ;
010A    CD EB0C                 CALL    CONOUT          ;
                        ;
010D    3E 00                   LD      A,00H           ; Set the code(low) of kanji
010F    11 0141                 LD      DE,KANDAT       ;
0112    13                      INC     DE              ;
0113    13                      INC     DE              ; Calculate the low code setting address
0114    13                      INC     DE              ;
                        ;
0115                    LOOP:
0115    12                      LD      (DE),A          ; Set the low code to the data package
0116    21 0141                 LD      HL,KANDAT       ; Set the address of kanji data
0119    06 01                   LD      B,01H           ; Set the parameter to display on the LCD
011B    F5                      PUSH    AF              ; Preserve registers
011C    D5                      PUSH    DE              ;
011D    CD EB9F                 CALL    KANJI           ; Display on the LCD
                        ;
0120    21 0141                 LD      HL,KANDAT       ; Set the address of kanji data
0123    06 02                   LD      B,02H           ; Set the parameter to print out
0125    CD EB9F                 CALL    KANJI           ; Print out
0128    0E 0C                   LD      C,0CH           ;
012A    CD EB0C                 CALL    CONOUT          ; CONOUT 0CH Clear screen & home
012D    D1                      POP     DE              ;
012E    F1                      POP     AF              ;
012F    3C                      INC     A               ;
0130    FE 00                   CP      0               ; Is all data displayed ?
0132    20 E1                   JR      NZ,LOOP         ; Yes then WBOOT else LOOP
                        ;
0134    0E 1B                   LD      C,1BH           ; CONOUT ESC + '3' cursor ON
0136    CD EB0C                 CALL    CONOUT          ;
0139    0E 33                   LD      C,33H           ;
013B    CD EB0C                 CALL    CONOUT          ;
013E    C3 EB03                 CALL    WBOOT           ;
                        ;
                        ;       ****************************************************
                        ;                KANJI DATA
                        ;       ****************************************************
                        ;
0141                    KANDAT:
0141    00 00 01 00             DB      0,0,1,0,4
0145    04
                        ;
                                END
```

(18) SAMPLE 18. MASKI

```
                                ;
                                ;       ****************************************
                                ;               BIOS MASKI SAMPLE PROGRAM
                                ;       ****************************************
                                ;
                                ;       NOTE :
                                ;               This sample program makes 1sec. interrupt and OVF
                                ;               interrupt disable.
                                ;
                                ;       <>assemble condition    <>
                                ;
                                                .Z80
                                ;
                                ;       <> loading address      <>
                                ;
                                                .PHASE          100H
                                ;
                                ;       <> constant value       <>
                                ;
EB03                            WBOOT           EQU     0EB03H          ; WBOOT entry address.
EB5A                            MASKI           EQU     0EB5AH          ; MASKI entry address.
1000                            STACKAD         EQU     1000H           ; Stack area.

00F7                            BROVF           EQU     11110111B       ; OVF interrupt bit masking data.
00F7                            BR1SEC          EQU     11110111B       ; 1sec. interrupt bit masking
                                                                        ; data.

0100                            START:
0100    31 1000                         LD      SP,STACKAD      ; Set stack pointer.

0103    06 50                           LD      B,50H           ; Read current interrupt mode.
0105    CD EB5A                         CALL    MASKI           ;
0108    ED 43 011A                      LD      (OLDINTST),BC   ; Save current interrupt mode.

010C    78                              LD      A,B             ;
010D    E6 F7                           AND     BROVF           ; Reset OVF interrupt bit (disable)
010F    47                              LD      B,A             ;

0110    79                              LD      A,C             ;
0111    E6 F7                           AND     BR1SEC          ; Reset 1sec. interrupt bit.(disable)
0113    4F                              LD      C,A             ;

0114    CD EB5A                         CALL    MASKI           ; Set new interrupt mode.

0117    C3 EB03                         JP      WBOOT           ; Jump WBOOT

011A                            OLDINTST:       DS      2       ; Old interrupt mode save area.

                                        END
```

(19) SAMPLE 19. STORX LDIRX

```
                                  ;
                                  ;       **************************************************
                                  ;               BIOS STORX,LDIRX SAMPLE PROGRAM
                                  ;       **************************************************
                                  ;
                                  ;       NOTE :
                                  ;                This sample program initializes the directory
                                  ;               area of RAM disk.
                                  ;
                                  ;
                                  ;       <>assemble condition      <>
                                  ;
                                                .Z80
                                  ;
                                  ;       <> loading address        <>
                                  ;
                                                .PHASE          100H
                                  ;
                                  ;       <> constant value         <>
                                  ;
EB03                              WBOOT         EQU     0EB03H          ; WBOOT entry address.
EB60                              STORX         EQU     0EB60H          ; STARX entry address.
EB63                              LDIRX         EQU     0EB63H          ; LDIRX entry address.
EBA5                              INFORM        EQU     0EBA5H          ; INFORM entry address.

F065                              AD_RAM_IN     EQU     0F065H          ; Top address of RAM disk
                                                                        ; (main RAM)
F068                              RAMD_SIZE     EQU     0F068H          ; Extend RAM size.(for RAM disk)
F062                              DIRSIZE       EQU     0F062H          ; directory area size (/128byte)

0003                              ADDISBNK      EQU     3               ; INFORM parameter (DISBNK)
0007                              ADSBCBNK      EQU     7               ; INFORM parameter (SBCBNK)

00E5                              FMTPTN        EQU     0E5H            ; Format pattern.

1000                              STACKAD       EQU     1000H           ; Stack area.


0100                              START:
0100      31 1000                         LD      SP,STACKAD      ; Set stack pointer.

0103      CD 014F                         CALL    CALDRTOP        ; Calculate directry area top address
                                                                  ; & bank.

0106      3E E5                           LD      A,FMTPTN        ; Format pattern (0E5h)
0108      ED 4B 0169                      LD      BC,(DIRBNK)     ; Creg. : bank of directory area.
010C      2A 0167                         LD      HL,(DIRADR)     ; HLreg.: top address of directory area.
010F      CD EB60                         CALL    STORX           ; Set format pattern.

0112      0E 03                           LD      C,ADDISBNK      ; Get DISCBNK address.
0114      CD EBA5                         CALL    INFORM          ;
0117      E5                              PUSH    HL              ; Save.

0118      0E 07                           LD      C,ADSBCBNK      ; Get SBCBNK address.
011A      CD EBA5                         CALL    INFORM          ;
011D      3A 0169                         LD      A,(DIRBNK)      ; Get bank data of directory area.
0120      77                              LD      (HL),A          ; Set to SBCBNK
0121      E1                              POP     HL              ; Restore DISBNK address.
0122      77                              LD      (HL),A          ; Set bank data.

0123      0E 00                           LD      C,0             ;
0125      3A F062                         LD      A,(DIRSIZE)     ; Directory area size. (/128b)
0128      CB 3F                           SRL     A               ;
012A      47                              LD      B,A             ; Make number of transmitting data.
012B      CB 19                           RR      C               ;
012D      0B                              DEC     BC              ;

012E      2A 0167                         LD      HL,(DIRADR)     ; HLreg.: top address of directory area.
0131      54                              LD      D,H             ; DEreg.: next address.
0132      5D                              LD      E,L             ;
0133      13                              INC     DE              ;
0134      CD EB63                         CALL    LDIRX           ; Format directory area with 0E5H.

0137      AF                              XOR     A               ; Areg.=00H
0138      EB                              EX      DE,HL           ; HLreg.: top address of checksum area.
0139      ED 4B 0169                      LD      BC,(DIRBNK)     ; Creg.  : bank data of checksum area.
013D      CD EB60                         CALL    STORX           ; Set 00H to checksum area top.h

0140      3A F062                         LD      A,(DIRSIZE)     ; (directory area size) / 128byte
0143      3D                              DEC     A               ;  --> checksum data number of directory
                                                                  ;        area.
0144      4F                              LD      C,A             ;
0145      06 00                           LD      B,0             ;
0147      13                              INC     DE              ;
0148      13                              INC     DE              ; DEreg.:(HLreg.+1)
0149      CD EB63                         CALL    LDIRX           ; Initialize checksum data of directory
                                                                  ; area.

014C      C3 EB03                         JP      WBOOT           ; Jump WBOOT
```

```
                         ;       ******************************************************************
                         ;               Calculate top address & bank of  Directory area
                         ;       ******************************************************************

014F                     CALDRTOP:
014F    2A F065                  LD      HL,(AD_RAM_IN)  ; If extend RAM not mounted, then
0152    11 6000                  LD      DE,6000H        ; (AD_RAM_IN) is top address.
0155    3A F068                  LD      A,(RAMD_SIZE)   ; else 6000H.
0158    B7                       OR      A               ;
0159    28 01                    JR      Z,CALDRT10      ;
015B    EB                       EX      DE,HL           ;

015C                     CALDRT10:
015C    22 0167                  LD      (DIRADR),HL     ; Set top address of directory area.
015F    07                       RLCA                    ;
0160    07                       RLCA                    ; Make bank data
0161    07                       RLCA                    ;
0162    07                       RLCA                    ;
0163    32 0169                  LD      (DIRBNK),A      ; Set  bank data.
0166    C9                       RET                     ;

0167                     DIRADR:         DS      2       ; Top address of directory area.
0169                     DIRBNK:         DS      1       ; bank data of directory area.

                                 END
```

(20) SAMPLE 20. Serial switch

```
                                        .Z80
                                        .PHASE  0000H

                              ;      *************************************
                              ;             SELECT SERIAL MODE
                              ;      *************************************
                              ;
                              ;      NOTE :
                              ;        Check current serial mode and if it is different from
                              ;        selected mode
                              ;       then set new mode parameters to ART.
                              ;        If same mode is setting then return immediately.
                              ;       Control register are ...
                              ;             CTLR1    (P00H)
                              ;             ARTMR    (P15H)
                              ;             SWR      (P18H)
                              ;             ARTCR    (P16H)
                              ;
                              ;      <> entry parameter <>
                              ;             Creg.   : Select mode code.
                              ;                          00H : SYSTEM.
                              ;                          01H : USER.
                              ;                          02H : EPSP.
                              ;
                              ;      <> return parameter <>
                              ;             Areg.   : Select information.
                              ;                          00H : Same device select.
                              ;                          01H : Device changed.
                              ;             Zflag.  : Depend on A register.
                              ;
                              ;      <> preserved registers <>
                              ;              DE
                              ;
                              ;      <> Constant value <>
                              ;
                              ;      < port address >
0000                          ZCTLR1        EQU     000H
0015                          ZARTMR        EQU     015H
0015                          ZARTSR        EQU     015H
0016                          ZARTCR        EQU     016H
0018                          ZSWR          EQU     018H
                              ;      < bit address of port data >
0004                          ZTXEMPTY      EQU     00000100B

0010                          ZBRG0         EQU     00010000B
0020                          ZBRG1         EQU     00100000B
0040                          ZBRG2         EQU     01000000B
0080                          ZBRG3         EQU     10000000B
0004                          ZSSW0         EQU     00000100B
0008                          ZSSW1         EQU     00001000B

0001                          ZTXE          EQU     00000001B
0002                          ZRDTR         EQU     00000010B
0004                          ZRXE          EQU     00000100B
0020                          ZRRTS         EQU     00100000B
                              ;      <> system work area define <>
F241                          SRMODE        EQU     0F241H
F196                          SRTABL        EQU     0F196H
F0D6                          BZCTLR1       EQU     0F0D6H
F0D9                          BZARTMR       EQU     0F0D9H
F0DA                          BZARTCR       EQU     0F0DAH
F0DC                          BZSWR         EQU     0F0DCH


0000                          SELSER:
0000    3A F241                      LD      A,(SRMODE)          ; Current serial mode.
0003    91                           SUB     C                   ; SRMODE = Selected mode?
0004    C8                           RET     Z                   ; Yes. (No operation is done.)

0005                          FSELSER:                           ; Forced SELSER.
0005    DB 15                        IN      A,(ZARTSR)          ;
0007    E6 04                        AND     ZTXEMPTY            ; Transmit empty ?
0009    28 FA                        JR      Z,FSELSER           ; No. (wait)

000B    79                           LD      A,C                 ;
000C    32 F241                      LD      (SRMODE),A          ; Set new serial mode.

000F    07                           RLCA                        ;
0010    07                           RLCA                        ;
0011    81                           ADD     A,C                 ;
0012    4F                           LD      C,A                 ;
0013    06 00                        LD      B,0                 ; BC : Table offset.
0015    21 F196                      LD      HL,SRTABL           ; HL : Serial parameter table top.

0018    09                           ADD     HL,BC               ; HL : New parameter table
                                                                 ; address.
```

```
                                      ;       < Set TXE and RXE to 'Low' >

0019    3A F0DA                               LD      A,(RZARTCR)              ;
001C    E6 FA                                 AND     0FFH-ZTXE-ZRXE           ;
001E    32 F0DA                               LD      (RZARTCR),A              ; Update system area.
0021    D3 16                                 OUT     (ZARTCR),A               ; Output I/O register.(CTLR1)

                                      ;       < For CTLR1 ( Set Baudrate ) >

0023    7E                                    LD      A,(HL)                             ;
0024    E6 F0                                 AND     ZBRC3+ZBRC2+ZBRC1+ZBRC0            ; Clear baudrate data.
0026    77                                    LD      (HL),A
0027    3A F0D6                               LD      A,(RZCTLR1)                        ; Get new CTLR1 data.
002A    E6 0F                                 AND     0FFH-ZBRC3-ZBRC2-ZBRC1-ZBRC0       ; Mask out baudrate data.
002C    B6                                    OR      (HL)                               ; Merge
002D    32 F0D6                               LD      (RZCTLR1),A                        ; Update system area.
0030    D3 00                                 OUT     (ZCTLR1),A                         ; Output I/O register.

                                      ;       < Pre-read ARTMR >

0032    23                                    INC     HL                       ;
0033    46                                    LD      B,(HL)                   ; There are don't care bits.

                                      ;       < For SWR ( Set serial switch ) >

0034    23                                    INC     HL                       ;
0035    7E                                    LD      A,(HL)                   ; Get new SWR data.
0036    E6 0C                                 AND     ZSSW1+ZSSW0              ; Mask out serial switch bit.
0038    77                                    LD      (HL),A                   ; Save.
0039    3A F0DC                               LD      A,(RZSWR)                ; Get current SWR data.
003C    E6 F3                                 AND     0FFH-ZSSW1-ZSSW0         ; Clear serial switch.
003E    B6                                    OR      (HL)                     ; merge.
003F    32 F0DC                               LD      (RZSWR),A                ; Update system area.
0042    D3 19                                 OUT     (ZSWR),A                 ; Output I/O register.

                                      ;       < Set ARTCR >

0044    23                                    INC     HL                       ;
0045    3A F0DC                               LD      A,(RZSWR)                ; Get current SWR
0048    E6 08                                 AND     ZSSW1                    ; If current mode is RS-232C
004A    7E                                    LD      A,(HL)                   ;
004B    20 09                                 JR      NZ,SETARTIO              ; then set HL pointed data.

004D    E6 DD                                 AND     0FFH-ZRDTR-ZRRTS         ; Clear control line.(DTR,RTS)
004F    4F                                    LD      C,A                      ; Save.
0050    3A F0DA                               LD      A,(RZARTCR)              ; Get current ARTCR
0053    E6 22                                 AND     ZRDTR+ZRRTS              ; Clear other current parameters.
0055    B1                                    OR      C                        ; New mode and old control line.

0056                                  SETARTIO:
0056    32 F0DA                               LD      (RZARTCR),A              ; Update system area.
0059    D3 16                                 OUT     (ZARTCR),A               ; Output I/O register.

                                      ;       < Set ARTMR >

005B    78                                    LD      A,B                      ; Saved ARTMR data.
005C    32 F0D9                               LD      (RZARTMR),A              ;
005F    D3 15                                 OUT     (ZARTMR),A               ;

                                      ;       < Set return parameter >

0061    AF                                    XOR     A              ; Areg = 1 ( Zflag OFF )
0062    3C                                    INC     A              ;
0063    C9                                    RET                    ;


                                              END
Macros:

Symbols:
0005    FSELSER         F0DA    RZARTCR         F0D9    RZARTMR
F0D6    RZCTLR1         F0DC    RZSWR           0000    SELSER
0056    SETARTIO        F241    SRMODE          F196    SRTABL
0016    ZARTCR          0015    ZARTMR          0014    ZARTSR
0010    ZBRC0           0020    ZBRC1           0040    ZBRC2
0080    ZBRC3           0000    ZCTLR1          0002    ZRDTR
0020    ZRRTS           0004    ZRXE            0004    ZSSW0
0008    ZSSW1           0019    ZSWR            0001    ZTXE
0004    ZTXEMPTY


No Fatal error(s)
```

(21) SAMPLE 21. CALLX

```
                        ;       ****************************************
                        ;               BIOS CALLX SAMPLE PROGRAM
                        ;       ****************************************
                        ;
                        ;       NOTE :
                        ;               This sample program plays the music.
                        ;               This program uses MELODY routine of system jump table.
                        ;
                        ;       <Assemble condition     >
                        ;
                                        .Z80
                        ;
                        ;       < loading address       >
                        ;
                                        .PHASE          100H
                        ;
                        ;       < constant value        >
                        ;
EB03                    WBOOT           EQU     0EB03H          ; WBOOT entry address.
EB69                    CALLX           EQU     0EB69H          ; CALLX entry address.
EBA5                    INFORM          EQU     0EBA5H          ; INFORM entry address.
0025                    MELODY          EQU     00025H          ; MELODY routine in system jump
                                                                ; table.
0003                    ADDISBNK        EQU     3               ; INFORM parameter (DISBNK)
8000                    MUSICTBL        EQU     8000H           ; MELODY data address.
0001                    ADDPINT         EQU     1               ; INFORM parameter (BPINTEBL)
00FF                    SYSBNK          EQU     0FFH            ;
1000                    STACKAD         EQU     1000H           ; Stack area.


0100                    START:
0100    31 1000                 LD      SP,STACKAD      ; Set stack pointer.

0103    0E 01                   LD      C,ADDPINT       ; Disable 1sec. interrupt in MELODY.
0105    CD EBA5                 CALL    INFORM          ;
0108    CB FE                   SET     7,(HL)          ;
010A    0E 03                   LD      C,ADDISBNK      ; Get DISBNK address.
010C    CD EBA5                 CALL    INFORM          ;
010F    3E FF                   LD      A,SYSBNK        ; Areg. : SYSTEM bank data.
0111    77                      LD      (HL),A          ; Set to DISBNK

0112    21 012E                 LD      HL,MUSICDT      ; Transmit music data to upper 8000H
0115    11 8000                 LD      DE,MUSICTBL     ;
0118    01 012C                 LD      BC,MUSICSZ      ;
011B    ED B0                   LDIR                    ;

011D    21 8000                 LD      HL,MUSICTBL     ; Music table.
0120    DD 21 0025              LD      IX,MELODY       ;
0124    0E 02                   LD      C,2             ; Loop counter.
0126    CD EB69                 CALL    CALLX           ; CALL MELODY.

0129    C3 EB03                 JP      WBOOT           ; Jump WBOOT


012C    0029            MUSICSZ:        DW      41
012E                    MUSICDT:
012E    0A 00                           DB      10,0
0130    04 19 04 1B                     DB      4,25,   4,27,   5,29,   4,25,   4,27,   5,29
0134    05 1D 04 19
0138    04 1B 05 1D
013C    04 20 04 1D                     DB      4,32,   4,29,   4,27,   4,25,   4,27,   4,29,
                                                5,27
0140    04 1B 04 19
0144    04 1B 04 1D
0148    05 1B
014A    04 19 04 1B                     DB      4,25,   4,27,   5,29,   4,25,   4,27,   5,25
014E    05 1D 04 19
0152    04 1B 05 19
0156    04 20 04 1D                     DB      4,32,   4,29,   4,27,   4,25,   4,27,   4,29,
                                                5,25
015A    04 1B 04 19
015E    04 1B 04 1D
0162    05 19
0164    04 20 04 20                     DB      4,32,   4,32,   4,29,   4,32,   4,34,   4,34,
                                                8,32
0168    04 1D 04 20
016C    04 22 04 22
0170    08 20
0172    04 1D 04 1D                     DB      4,29,   4,29,   4,27,   4,27,   16,25
0176    04 1B 04 1B
017A    10 19
017C    00                              DB      0

                                END
```

(22) SAMPLE 22. RDVRAM

```
                            ;       ***************************************************
                            ;               BIOS RDVRAM SAMPLE PROGRAM
                            ;       ***************************************************
                            ;
                            ;       NOTE:  This sample program reads the screen buffer
                            ;             data and displays the data.
                            ;
                            ;       <> assemble condition   <>
                            ;
                                    .Z80
                            ;
                            ;       <> loading address      <>
                            ;
                                    .PHASE  100H
                            ;
                            ;       <> constant values      <>
                            ;
                            ;               BIOS entries
                            ;
EB03                        WBOOT           EQU     0EB03H
EB09                        CONIN           EQU     0EB09H
EB0C                        CONOUT          EQU     0EB0CH
EB6F                        PUTPFK          EQU     0EB6FH
EB78                        RDVRAM          EQU     0EB78H
                            ;
                            ;               System work addresses
                            ;
F257                        LSCSIZEX        EQU     0F257H
F258                        LSCSIZEY        EQU     0F258H
F5F2                        YKCOUNTRY       EQU     0F5F2H
                            ;
                            ;               Other constants
                            ;
1000                        MAINSP          EQU     01000H
                            ;
000A                        LF              EQU     0AH
000D                        CR              EQU     0DH
                            ;
                            ;       ***************************************************
                            ;               MAIN PROGRAM
                            ;       ***************************************************
                            ;
0100                        START:
0100      31 1000                   LD      SP,MAINSP       ; Set stack pointer
0103      3A F5F2                   LD      A,(YKCOUNTRY)   ; If ENT-10, then define the key-code
0106      07                        RLCA                    ; at the top of the screen.
0107      01 0101                   LD      BC,0101H        ;
010A      3E 01                     LD      A,01H           ;  (Key code)
010C      D4 EB6F                   CALL    NC,PUTPFK       ;  (Key define)
010F      01 0101                   LD      BC,0101H        ; Initial position
0112                        LOOP10:
0112      79                        LD      A,C             ; Display line number
0113      CD 0142                   CALL    DSP10           ;
0116      CD 0156                   CALL    CRLF            ;
                            ;
0119                        LOOP20:
0119      C5                        PUSH    BC              ;
011A      CD EB78                   CALL    RDVRAM          ; Read the screen buffer data
011D      C1                        POP     BC              ;
011E      B7                        OR      A               ; Error?
011F      20 16                     JR      NZ,STOP         ; Yes
                            ;
0121      CD 013D                   CALL    DSPHL           ; Display attribute & character code
0124      04                        INC     B               ; Next column position
0125      3A F257                   LD      A,(LSCSIZEX)    ;
0128      B8                        CP      B               ; Within screen?
0129      30 EE                     JR      NC,LOOP20       ; Yes
                            ;
012B      CD 0156                   CALL    CRLF            ; Move cursor to next line
012E      0C                        INC     C               ; Next line position
012F      06 01                     LD      B,01H           ; Top of column
0131      3A F258                   LD      A,(LSCSIZEY)    ;
0134      B9                        CP      C               ; Within screen?
0135      30 DB                     JR      NC,LOOP10       ; Yes
                            ;
0137                        STOP:
0137      CD EB09                   CALL    CONIN           ; Input any key
013A      C3 EB03                   JP      WBOOT           ; End of the program
                            ;
                            ;       ***************************************************
                            ;               DISPLAY THE HL DATA BY HEX
                            ;       ***************************************************
                            ;
                            ;       NOTE:
                            ;               Display HL register data by changing hex code.
                            ;
                            ;       <> entry parameter      <>
                            ;               HL : Displayed data
                            ;       <> return parameter     <>
                            ;               NONE
                            ;       <> preserved registers  <>
                            ;               BC,DE,HL
```

```
                                 ;
013D                             DSPHL:
013D     7E                               LD      A,M           ; Display first byte by hex
013E     CD 0142                          CALL    DSP10         ;
0141     7D                               LD      A,L           ; Display second byte by hex
                                 ;
0142                             DSP10:
0142     F5                               PUSH    AF            ; Save entry data
0143     0F                               RRCA                  ; Move MSB 4 bit to LSB 4 bit
0144     0F                               RRCA                  ;
0145     0F                               RRCA                  ;
0146     0F                               RRCA                  ;
0147     CD 014B                          CALL    DSP20         ; Convert binary to hex and display it
014A     F1                               POP     AF            ; Restore entry data (Use LSB 4 bits)
                                 ;
014B                             DSP20:
014B     E6 0F                            AND     0FH           ; Neglect MSB 4 bits
014D     C6 90                            ADD     A,90H         ; 0000B -- 1111B convert 0 -- F
014F     27                               DAA                   ;
0150     CE 40                            ADC     A,40H         ;
0152     27                               DAA                   ;
0153     C3 015D                          JP      SCONOUT       ; Display it
                                 ;
                                 ;        ***********************************************************
                                 ;                CONSOLE OUT CR & LF CODE
                                 ;        ***********************************************************
                                 ;
                                 ;        NOTE:
                                 ;                Move the cursor position to the top of next line.
                                 ;
                                 ;        <> entry parameter       <>
                                 ;                NONE
                                 ;        <> return parameter      <>
                                 ;                NONE
                                 ;        <> preserved registers   <>
                                 ;                BC,DE,HL
                                 ;
0156                             CRLF:
0156     3E 0D                            LD      A,CR          ; Carriage return
0158     CD 015D                          CALL    SCONOUT       ;
015B     3E 0A                            LD      A,LF          ; Line feed
                                 ;
015D                             SCONOUT:
015D     C5                               PUSH    BC            ; Save registers
015E     D5                               PUSH    DE            ;
015F     E5                               PUSH    HL            ;
0160     4F                               LD      C,A           ;
0161     CD EB0C                          CALL    CONOUT        ;
0164     E1                               POP     HL            ; Restore registers
0165     D1                               POP     DE            ;
0166     C1                               POP     BC            ;
0167     C9                               RET                   ;

                                          END
```

(23) SAMPLE 23. ICCARD

```
                                    ;       ************************************************************
                                    ;               BIOS ICCARD SAMPLE PROGRAM
                                    ;       ************************************************************
                                    ;
                                    ;       NOTE:
                                    ;               This sample program reads the data of IC-card.
                                    ;
                                    ;       <> assemble condition     <>
                                    ;
                                            .Z80
                                    ;
                                    ;       <> loading address        <>
                                    ;
                                            .PHASE  100H
                                    ;
                                    ;       <> constant values        <>
                                    ;
                                    ;               BIOS entries
                                    ;
EB03                                WBOOT           EQU     0EB03H
EB06                                CONST           EQU     0EB06H
EB09                                CONIN           EQU     0EB09H
EB0C                                CONOUT          EQU     0EB0CH
EB6F                                PUTPFK          EQU     0EB6FH
EB99                                ICCARD          EQU     0EB99H
                                    ;
                                    ;               System work addresses
                                    ;
F183                                ICCDPRM         EQU     0F183H
F12B                                ICTSDT          EQU     0F12BH
F604                                BSPSTS          EQU     0F604H
F5F2                                YBCOUNTRY       EQU     0F5F2H
                                    ;
                                    ;               Other constants
                                    ;
1000                                MAINSP          EQU     01000H

000A                                LF              EQU     0AH
000D                                CR              EQU     0DH

0000                                PKON            EQU     000H
00FB                                TSKON           EQU     0FBH
                                    ;
                                    ;       ************************************************************
                                    ;               MAIN PROGRAM
                                    ;       ************************************************************
                                    ;
0100                                START:
0100        31 1000                         LD      SP,MAINSP       ; Set stack pointer
0103        CD 0164                         CALL    KEYSET          ; Set default key table.
0106        CD 02BF                         CALL    ICCLOSE         ; Close IC-card
0109        CD 02B8                         CALL    ICOPEN          ; Open IC-card
                                    ;
010C                                LOOP:
010C        CD 0138                         CALL    DTWRITE         ; Send command or data
010F        38 21                           JR      C,STOP          ; Send error
0111        01 0000                         LD      BC,0000H        ; Wait for some time
0114                                LOOP00:
0114        0B                              DEC     BC              ;
0115        78                              LD      A,B             ;
0116        B1                              OR      C               ;
0117        20 FB                           JR      NZ,LOOP00       ;
0119                                LOOP10:
0119        CD EB06                         CALL    CONST           ; Check inputted-key status
011C        3C                              INC     A               ;
011D        28 13                           JR      Z,STOP          ; Any key in
011F        CD 02FA                         CALL    ICSTATUS        ; Check received status
0122        3C                              INC     A               ; Any data received?
0123        20 F4                           JR      NZ,LOOP10       ;
0125                                LOOP20:
0125        CD 0154                         CALL    DTREAD          ; Yes. Display the received data.
0128        38 08                           JR      C,STOP          ; No error
012A        CD 02FA                         CALL    ICSTATUS        ; Check received status
012D        3C                              INC     A               ; Any data received?
012E        28 F5                           JR      Z,LOOP20        ; Yes
0130        18 DA                           JR      LOOP            ;
                                    ;
0132                                STOP:
0132        CD EB09                         CALL    CONIN           ; Input any key.
0135        C3 EB03                         JP      WBOOT           ;
```

```
                    ;
                    ;       *************************************************************
                    ;               SEND DATA TO IC-CARD
                    ;       *************************************************************
                    ;
                    ;       NOTE:
                    ;               Send command to IC-card
                    ;
                    ;       <> entry parameter      <>
                    ;               (CMDPNT) -- Sending data pointer
                    ;       <> return parameter     <>
                    ;               (CMDPNT) -- Next pointer
                    ;               CY : Return information
                    ;                       =0 -- Normal end
                    ;                       =1 -- Error or end of data
                    ;       <> preserved registers  <>
                    ;               NONE
0138                DTWRITE:
0138    2A 022C             LD      HL,(CMDPNT)     ; Command data pointer
013B    4E                  LD      C,(HL)          ; Get data counter
013C    79                  LD      A,C             ; Check end of data
013D    B7                  OR      A               ;
013E    37                  SCF                     ;
013F    C8                  RET     Z               ; End of data
                    ;
0140    E5                  PUSH    HL
0141    23                  INC     HL              ; Data top address
0142    E5                  PUSH    HL              ;
0143    06 00               LD      B,00H           ; Set next data pointer
0145    09                  ADD     HL,BC           ;
0146    22 022C             LD      (CMDPNT),HL
0149    E1                  POP     HL              ;
014A    CD 02EA             CALL    ICWRITE         ; Send the data
014D    E1                  POP     HL
014E    F5                  PUSH    AF              ; Save return parameter
014F    CD 0171             CALL    DSPDT           ; Display the data
0152    F1                  POP     AF
0153    C9                  RET                     ;
                    ;
                    ;       *************************************************************
                    ;               RECEIVE DATA FROM IC-CARD
                    ;       *************************************************************
                    ;
                    ;       NOTE:
                    ;               Receive data or status from IC-card
                    ;
                    ;       <> entry parameter      <>
                    ;               NONE
                    ;       <> return parameter     <>
                    ;               (DATAPNT) -- Received data
                    ;               CY : Return information
                    ;                       =0 -- Normal end
                    ;                       =1 -- Error
                    ;       <> preserved registers  <>
                    ;               NONE
0154                DTREAD:
0154    2A 01AA             LD      HL,(DATAPNT)    ; Set received data storing area
0157    23                  INC     HL              ;
0158    CD 03C3             CALL    ICREAD          ; Read received data
015B    D8                  RET     C               ; Error
015C    2A 01AA             LD      HL,(DATAPNT)    ; Set received data count
015F    71                  LD      (HL),C          ;
0160    CD 0171             CALL    DSPDT           ; Display the data
0163    C9                  RET                     ;
                    ;
                    ;       *************************************************************
                    ;               SET DEFAULT KEY TABLE
                    ;       *************************************************************
                    ;
                    ;       NOTE:
                    ;               Set default key table for EHT-10
                    ;
                    ;       <> entry parameter      <>
                    ;               NONE
                    ;       <> return parameter     <>
                    ;               NONE
                    ;       <> preserved registers  <>
                    ;               NONE
0164                KEYSET:
0164    3A FEF2             LD      A,(YKCOUNTRY)   ; If EHT-10, then define the key-code
0167    07                  RLCA                    ; at the top of the screen.
0168    01 0101             LD      BC,0101H        ;
016B    3E 01               LD      A,01H           ;  (Key code)
016D    D4 EB6F             CALL    NC,PUTPFK       ;  (Key define)
0170    C9                  RET                     ;
```

```
                        ;
                        ;       **********************************************************
                        ;               DISPLAY THE DATA
                        ;       **********************************************************
                        ;
                        ;       NOTE:
                        ;               Display the data by hex code
                        ;
                        ;       <> entry parameter      <>
                        ;               HL : Data address
                        ;                    First byte is data length
                        ;       <> return parameter     <>
                        ;               NONE
                        ;       <> preserved registers  <>
                        ;               BC,DE,HL
                        ;
0171                    DSPDT:
0171    E5                      PUSH    HL              ; Save registers
0172    D5                      PUSH    DE              ;
0173    C5                      PUSH    BC              ;
0174    46                      LD      B,(HL)          ; Display data counter
0175    23                      INC     HL              ; Data top address
                        ;
0176                    DSPDT10:
0176    7E                      LD      A,(HL)          ; Get display data
0177    CD 0184                 CALL    DSPHEX          ; Convert to hex code & display it
017A    23                      INC     HL              ; Next data pointer
017B    10 F9                   DJNZ    DSPDT10         ;
                        ;
017D    CD 0198                 CALL    CRLF            ; Carriage return & line feed
0180    C1                      POP     BC              ; Restore registers
0181    D1                      POP     DE              ;
0182    E1                      POP     HL              ;
0183    C9                      RET                     ;
                        ;
                        ;       **********************************************************
                        ;               DISPLAY THE DATA BY HEX CODE
                        ;       **********************************************************
                        ;
                        ;       NOTE:
                        ;               Display HL register data by changing hex code.
                        ;
                        ;       <> entry parameter      <>
                        ;               A  : Displayed data
                        ;       <> return parameter     <>
                        ;               NONE
                        ;       <> preserved registers  <>
                        ;               BC,DE,HL
                        ;
0184                    DSPHEX:
0184    F5                      PUSH    AF              ; Save entry data
0185    0F                      RRCA                    ; Move MSB 4 bit to LSB 4 bit
0186    0F                      RRCA                    ;
0187    0F                      RRCA                    ;
0188    0F                      RRCA                    ;
0189    CD 018D                 CALL    DSP20           ; Convert binary to hex and display it
018C    F1                      POP     AF              ; Restore entry data (Use LSB 4 bits)
                        ;
018D                    DSP20:
018D    E6 0F                   AND     0FH             ; Neglect MSB 4 bits
018F    C6 90                   ADD     A,90H           ; 0000B -- 1111B convert 0 -- F
0191    27                      DAA                     ;
0192    CE 40                   ADC     A,40H           ;
0194    27                      DAA                     ;
0195    C3 019F                 JP      SCONOUT         ; Display it
                        ;
                        ;       **********************************************************
                        ;               CONSOLE OUT CR & LF CODE
                        ;       **********************************************************
                        ;
                        ;       NOTE:
                        ;               Move the cursor position to the top of next line.
                        ;
                        ;       <> entry parameter      <>
                        ;               NONE
                        ;       <> return parameter     <>
                        ;               NONE
                        ;       <> preserved registers  <>
                        ;               BC,DE,HL
                        ;
0198                    CRLF:
0198    3E 0D                   LD      A,CR            ; Carriage return
019A    CD 019F                 CALL    SCONOUT         ;
019D    3E 0A                   LD      A,LF            ; Line feed
                        ;
019F                    SCONOUT:
019F    C5                      PUSH    BC              ; Save registers
01A0    D5                      PUSH    DE              ;
01A1    E5                      PUSH    HL              ;
01A2    4F                      LD      C,A             ;
01A3    CD EB0C                 CALL    CONOUT          ;
01A6    E1                      POP     HL              ; Restore registers
01A7    D1                      POP     DE              ;
01A8    C1                      POP     BC              ;
01A9    C9                      RET                     ;
```

```
                                  ;
01AA    01AC                      DATAPNT:      DW      BUFF
01AC                              BUFF:         DS      128
022C    022E                      CMDPNT:       DW      ICCMDTB
                                  ;
                                  ;       ***************************************************
                                  ;               IC-CARD COMMAND TABLE
                                  ;       ***************************************************
                                  ;
                                  ;       Following data is commands for IC-card.
                                  ;
022E                              ICCMDTB:
                                  .LIST
                                  ;
                                  ;       ***************************************************
                                  ;               OPEN IC-CARD
                                  ;       ***************************************************
                                  ;
                                  ;       NOTE:
                                  ;               This routine opens the IC-card.
                                  ;               Parameters for communication to IC-card are
                                  ;              9600 bps, 8 bits, even parity, 1 stop bit in
                                  ;              default value. So if communication parameters
                                  ;              are unmatch to IC-card, you change ICCDPRM as
                                  ;              you like.
                                  ;               And if reset process of system is unmatch to
                                  ;              your IC-card, you extend it by using IC-card's
                                  ;              hook. For example, the cause of waiting too
                                  ;              short, answer-to-reset being unmatch, or etc.
                                  ;               This sample routine changes to non parity.
                                  ;
                                  ;       <> entry parameter          <>
                                  ;               NONE
                                  ;       <> return parameter         <>
                                  ;               HL : answer-to-reset data setting area top addr
                                  ;               CY : Return information
                                  ;                       =0 -- Normally opened
                                  ;                       =1 -- Error
                                  ;                         A reg. is error code.
                                  ;                               =1 : Parameter error
                                  ;                                2 : Already opened
                                  ;                                3 : Not opened
                                  ;                                4 : Force stop
                                  ;                                5 : Receive buffer overflow
                                  ;                                6 : Time out
                                  ;                                7 : Retry error
                                  ;                                8 : Communication error
                                  ;                                9 : Power off stop
                                  ;                                A : IC-card's case is opened
                                  ;                                B : Serial already used
                                  ;                                C : IC-card used by disk
                                  ;       <> preserved registers      <>
                                  ;               NONE
                                  ;
02BB                              ICOPEN:
                                  ;
                                  ;       If you want to execute by non parity, you must insert
                                  ;      next four statements in this program.
                                  ;       (Default is even parity.)
                                  ;*      LD      A,PNON          ; Set parity to non parity
                                  ;*      LD      (ICCDPRM+2),A   ;
                                  ;*      LD      A,TSNON         ; Change TS-byte checking data
                                  ;*      LD      (ICTSET),A      ;
02BB    CD 0308                           CALL    IOPEN           ; Open IC-card
02BE    C9                                RET                     ;
                                  ;
                                  ;       ***************************************************
                                  ;               CLOSE IC-CARD
                                  ;       ***************************************************
                                  ;
                                  ;       NOTE:
                                  ;               This routine closes the IC-card.
                                  ;
                                  ;       <> entry parameter          <>
                                  ;               NONE
                                  ;       <> return parameter         <>
                                  ;               NONE
                                  ;       <> preserved registers      <>
                                  ;               NONE
                                  ;
02BF                              ICCLOSE:
02BF    CD 030C                           CALL    ICLOSE          ; Close IC-card
02C2    C9                                RET                     ;
```

```
                        ;
                        ;       ****************************************************
                        ;               READ FROM IC-CARD
                        ;       ****************************************************
                        ;
                        ;       NOTE:
                        ;               This routine reads data from the IC-card.
                        ;               This routine takes received data and set them
                        ;             to appointed area. It judges the end of data
                        ;             by time out (3 seconds).
                        ;               If communication error is happened, you must
                        ;             be close the IC-card and start from initial
                        ;             step. But when you continue the operation, you
                        ;             must reset error-flag because ICCARD's READ
                        ;             routine reads no data during error-flag being on.
                        ;
                        ;       <> entry parameter       <>
                        ;               HL : Received data storing area top address
                        ;       <> return parameter      <>
                        ;               CY : Return information
                        ;                       =0 -- Normally opened
                        ;                         BC reg. is received data count.
                        ;                       =1 -- Error
                        ;                         A reg. is error code.
                        ;                               =1 : Parameter error
                        ;                                2 : Already opened
                        ;                                3 : Not opened
                        ;                                4 : Force stop
                        ;                                5 : Receive buffer overflow
                        ;                                6 : Time out
                        ;                                7 : Retry error
                        ;                                8 : Communication error
                        ;                                9 : Power off stop
                        ;                                A : IC-card's case is opened
                        ;                                B : Serial already used
                        ;                                C : IC-card used by disk
                        ;       <  preserved registers  <>
                        ;               NONE
                        ;
02C3                    ICREAD:
02C3    01 0000                 LD      BC,0000H        ; Data counter initialize
                        ;
02C6                    ICRD10:
02C6    E5                      PUSH    HL              ; Save data pointer
02C7    C5                      PUSH    BC              ; Save data counter
02C8    CD 0310                 CALL    IBRAD           ; Read data by 1 byte
02CB    C1                      POP     BC              ;
02CC    E1                      POP     HL              ;
02CD    38 05                   JR      C,ICRD20        ; Read error
02CF    77                      LD      (HL),A          ; Store the read data
02D0    23                      INC     HL              ; Pointer up
02D1    03                      INC     BC              ; Counter up
02D2    18 F2                   JR      ICRD10          ;
                        ;
02D4                    ICRD20:
02D4    F5                      PUSH    AF              ; Save error status
02D5    3A F604                 LD      A,(RSPSTS)      ; Reset error flag
02D8    E6 9F                   AND     10001111B       ; (Reset parity, flaming, over-run)
02DA    32 F604                 LD      (RSPSTS),A      ;
02DD    F1                      POP     AF              ; Restore error flag
                        ;
02DE    FE 06                   CP      06H             ; Check error status
02E0    37                      SCF                     ;
02E1    C0                      RET     NZ              ; Not time out error
02E2    78                      LD      A,B             ; Check read data counter
02E3    B1                      OR      C               ;
02E4    3E 06                   LD      A,06H           ;
02E6    37                      SCF                     ;
02E7    C8                      RET     Z               ; No data received
02E8    AF                      XOR     A               ;
02E9    C9                      RET                     ; Normal return
```

```
                        ;
                        ;       ****************************************************
                        ;               WRITE THE DATA INTO IC-CARD
                        ;       ****************************************************
                        ;
                        ;       NOTE:
                        ;               This routine writes the data into IC-card.
                        ;
                        ;       <> entry parameter      <>
                        ;               HL : Data top address for writing
                        ;               BC : Data counter for writing
                        ;       <> return parameter     <>
                        ;               CY : Return information
                        ;                       =0 -- Normally opened
                        ;                       =1 -- Error
                        ;                         A reg. is error code.
                        ;                               =1 : Parameter error
                        ;                                2 : Already opened
                        ;                                3 : Not opened
                        ;                                4 : Force stop
                        ;                                5 : Receive buffer overflow
                        ;                                6 : Time out
                        ;                                7 : Retry error
                        ;                                8 : Communication error
                        ;                                9 : Power off stop
                        ;                                A : IC-card's case is opened
                        ;                                B : Serial already used
                        ;                                C : IC-card used by disk
                        ;       <> preserved registers  <>
                        ;               NONE
                        ;
02EA                    ICWRITE:
02EA    E5                      PUSH    HL              ; Save data pointer
02EB    C5                      PUSH    BC              ; Save data counter
02EC    4E                      LD      C,(HL)          ; Get writing data
02ED    CD 0314                 CALL    IWRITE          ; Write it to IC-card
02F0    C1                      POP     BC              ;
02F1    E1                      POP     HL              ;
02F2    D8                      RET     C               ; Error

02F3    23                      INC     HL              ; Data pointer up
02F4    0B                      DEC     BC              ; Data counter down
02F5    78                      LD      A,B             ; Check end of data
02F6    B1                      OR      C               ;
02F7    C8                      RET     Z               ; End of data (Normal return)
02F8    18 F0                   JR      ICWRITE         ; Loop until data end
                        ;
                        ;       ****************************************************
                        ;               READ IC-CARD STATUS
                        ;       ****************************************************
                        ;
                        ;       NOTE:
                        ;               This routine reads the received status from
                        ;             IC-card.
                        ;               If error is happened, you must reset error
                        ;             status for next access.
                        ;
                        ;       <> entry parameter      <>
                        ;               NONE
                        ;       <> return parameter     <>
                        ;               CY : Return information
                        ;                       =0 -- Normally opened
                        ;                         A reg. is received data status
                        ;                           =00H -- No data in received buffer
                        ;                           =FFH -- Any data in received buffer
                        ;                         BC reg. is received data count (bytes)
                        ;                           (BC reg. is only active in A=FFH)
                        ;                       =1 -- Error
                        ;                         A reg. is error code.
                        ;                               =1 : Parameter error
                        ;                                2 : Already opened
                        ;                                3 : Not opened
                        ;                                4 : Force stop
                        ;                                5 : Receive buffer overflow
                        ;                                6 : Time out
                        ;                                7 : Retry error
                        ;                                8 : Communication error
                        ;                                9 : Power off stop
                        ;                                A : IC-card's case is opened
                        ;                                B : Serial already used
                        ;                                C : IC-card used by disk
                        ;       <> preserved registers  <>
                        ;               NONE
                        ;
02FA                    ICSTATUS:
02FA    CD 0319                 CALL    ISTATUS         ; Read IC-card's status
02FD    F5                      PUSH    AF              ; Save error status
02FE    3A F604                 LD      A,(RSPSTS)      ; Reset error flag
0301    E6 8F                   AND     10001111B       ; (Reset parity, flaming, over-run)
0303    32 F604                 LD      (RSPSTS),A      ;
0306    F1                      POP     AF              ; Restore error flag
0307    C9                      RET                     ;
```

```
                                        ;
                                        ;       ====================================================
                                        ;               BIOS ICCARD ACCESS ROUTINE
                                        ;       ====================================================
                                        ;
                                        ;       NOTE:
                                        ;               These routines call BIOS ICCARD.
                                        ;
                                        ;       () entry parameter       ()
                                        ;               Depend on each function
                                        ;       () return parameter      ()
                                        ;               CY : Return information
                                        ;                       =0 -- Normally opened
                                        ;                         Other registers depend on each function
                                        ;                       =1 -- Error
                                        ;                         A reg. is error code.
                                        ;                                 =1 : Parameter error
                                        ;                                  2 : Already opened
                                        ;                                  3 : Not opened
                                        ;                                  4 : Force stop
                                        ;                                  5 : Receive buffer overflow
                                        ;                                  6 : Time out
                                        ;                                  7 : Retry error
                                        ;                                  8 : Communication error
                                        ;                                  9 : Power off stop
                                        ;                                  A : IC-card's case is opened
                                        ;                                  B : Serial already used
                                        ;                                  C : IC-card used by disk
                                        ;       () preserved registers   ()
                                        ;               NONE
                                        ;
  0308                                  IOPEN:
  0308      3E 00                               LD      A,00H           ; Open IC-card
  030A      18 0E                               JR      ICCALL          ;
  030C                                  ICLOSE:
  030C      3E 01                               LD      A,01H           ; Close IC-card
  030E      18 0A                               JR      ICCALL          ;
  0310                                  IREAD:
  0310      3E 02                               LD      A,02H           ; Read from IC-card
  0312      18 06                               JR      ICCALL          ;
  0314                                  IWRITE:
  0314      3E 03                               LD      A,03H           ; Write into IC-card
  0316      18 02                               JR      ICCALL          ;
  0318                                  ISTATUS:
  0318      3E 04                               LD      A,04H           ; Read IC-card's status
  031A      CD EB99                     ICCALL: CALL    ICCARD          ;
  031D      C9                                  RET                     ;

                                                END
Macros:

Symbols:
01AC    BUFF            022C    CMDPNT          EB09    CONIN
EB0C    CONOUT          EB06    CONST           000D    CR
0198    CRLF            01AA    DATAPNT         018D    DSP20
0171    DSPDT           0176    DSPDT10         0184    DSPHEX
0154    DTREAD          0136    DTWRITE         031A    ICCALL
EB99    ICCARD          F1B3    ICCOPEN         02BF    ICCLOSE
022E    ICCMDTR         02CC    ICLOSE          02B8    ICOPEN
02C8    ICRD10          02D4    ICRD20          02C3    ICREAD
02FA    ICSTATUS        F1E9    ICTSDT          02EA    ICWRITE
0308    IOPEN           0310    IREAD           0318    ISTATUS
0314    IWRITE          0164    KEYSET          000A    LF
010C    LOOP            0114    LOOP00          0119    LOOP10
0125    LOOP20          1000    MAINSP          0000    PNON
2B6F    PUTPFE          F604    RSPSTS          019F    SCONOUT
0100    START           0132    STOP            00FB    TSNON
EB03    WBOOT           F5F2    YKCOUNTER


No Fatal error(s)
```

(24) SAMPLE 24. User BIOS Area

```
                                ;
                                ;       ******************************************
                                ;               USERBIOS SAMPLE PROGRAM
                                ;       ******************************************
                                ;
                                ;       NOTE :
                                ;               This sample program shows how to make USERBIOS-AREA.
                                ;
                                ;       <> assemble condition   <>
                                ;
                                        .Z80
                                ;
                                ;       <> loading address      <>
                                ;
                                        .PHASE          100H
                                ;
                                ;       <> constant value       <>
                                ;
                                ;       < BIOS entry point >
EB03                            WBOOT           EQU     0EB03H          ; WBOOT entry address.
EB0C                            CONOUT          EQU     0EB0CH          ; CONOUT entry address.
EB39                            BEEP            EQU     0EB39H          ; BEEP entry address.
EB69                            CALLX           EQU     0EB69H          ; CALLX entry address.
                                ;
                                ;       < system service routines >
0013                            BIOSRTLD        EQU     0013H           ; SBIOS loader.
0016                            SETRAMAD        EQU     0016H           ; Set system parameter.
0019                            CHGRAMD         EQU     0019H           ; Change RAM disk address.
                                ;
                                ;       < system area define >
F008                            STZRAM          EQU     0F008H          ; Standard RAM size of RAM disk.
F00C                            USERBIOS        EQU     0F00CH          ; USERBIOS area size.
F060                            ST_RAM_IN       EQU     0F060H          ; RAM disk size.
F065                            AD_RAM_IN       EQU     0F065H          ; Start address of RAM
                                                                        ; disk.(current)
F072                            AD_RAM_OLD      EQU     0F072H          ; Start address of RAM disk.(old)
F06B                            RAMD_SIZE       EQU     0F06BH          ; Extend RAM size
F41B                            DISBNK          EQU     0F41BH          ; Object bank for CALLX.
                                ;
                                ;       < USERBIOS area define >
DC00                            UBBOTTOM        EQU     0DC00H          ; USERBIOS area bottom
                                                                        ; address.
DBF0                            HEADTOP         EQU     UBBOTTOM-10H    ; Header top address.
DBF0                            ID              EQU     HEADTOP         ;(2) ID
DBF2                            NAME            EQU     HEADTOP+02H     ;(8) Program name.
DBFA                            SIZE            EQU     HEADTOP+0AH     ;(1) Program size.
DBFB                            OVWRT           EQU     HEADTOP+0BH     ;(1) Overwrite flag.
DBFC                            RELSAD          EQU     HEADTOP+0CH     ;(2) Release program
                                                                        ;    address.
DBFE                            UNUSE           EQU     HEADTOP+0EH     ;(1) unused byte.
DBFF                            CHKSUM          EQU     HEADTOP+0FH     ;(1) check sum
0010                            UBSSIZE         EQU     10              ; USERBIOS area size = 4K
                                                                        ; (Please change if
                                                                        ; neccesary)
DB00                            RELSTOP         EQU     UBBOTTOM-100H   ; Release progam address.
CC00                            STRTOP          EQU     UBBOTTOM-UBSSIZE*256    ; Program start address.
0050                            MAXSZ           EQU     50H             ; max KB
                                ;
                                ;       < dispaly data >
000C                            CLS             EQU     0CH             ; Clear screen & home.
001B                            ESC             EQU     1BH             ; ESC sequence.
000D                            CR              EQU     0DH             ; Carridge return.
000A                            LF              EQU     0AH             ; Line feed.
                                ;
                                ;       ******************************
                                ;               Main Routine
                                ;       ******************************
                                ;
0100                            UBICS:
0100    31 0254                         LD      SP,STACKAD      ; Set stack pointer.

0103    CD 01C0                         CALL    CHK_HEAD        ; Check existence of header.
0106    20 13                           JR      NZ,HEAD40       ; Jump if does not exist.

0108    CD 01D5                         CALL    CHK_USR         ; Check used module .
010B    28 06                           JR      Z,HEAD20        ; Jump if used oneself.

010D    3A DBFB                         LD      A,(OVWRT)       ; Check overwrite flag.
0110    B7                              OR      A
0111    28 27                           JR      Z,ERREND        ; If cannot overwrite then jump.

0113                            HEAD20:
0113    21 0118                         LD      HL,HEAD40       ; Set return address.
```

```
0116    E5                          PUSH    HL                  ;
0117    2A 0BFC                     LD      HL,(RLSAD)          ; Do Jerase process.
011A    E9                          JP      (HL)                ;

011B                        HEAD40:
011B    CD 0142                     CALL    UBMAKE              ; Make USERBIOS area.
011E    30 1A                       JR      NC,ERREND           ; Jump if error.

0120    CD 01A9                     CALL    UBLOAD              ; Load object program.
0123    CD 0255                     CALL    MAKEHEAD            ; Make header.
0126    CD 0354                     CALL    OPNUBIOS            ; Open process for USERBIOS area.
0129    21 01F2                     LD      HL,MSGOK            ; Set O.K. message address.
012C    01 0010                     LD      BC,0010H            ; Wait 1.6sec.

012F                        MAIN_END:
012F    C5                          PUSH    BC                  ; Save.
0130    CD 01E6                     CALL    PRMSG               ; Display message.
0133    C1                          POP     BC                  ; Restore.
0134    CD EB38                     CALL    BEEP                ; Buzzer ON
0137    C3 EB03                     JP      WBOOT               ; LET'S EXIT FROM THIS PROGRAM !

013A                        ERREND:
013A    21 0202                     LD      HL,MSGERR           ; Set error message address.
013D    01 2210                     LD      BC,2210H            ; BEEP (880HZ, 1.6SEC.)
0140    18 ED                       JR      MAIN_END            ;

                            ;       **********************************
                            ;               Make USERBIOS area
                            ;       **********************************
                            ;
                            ;       <> return parameter <>
                            ;               Carry ON : Normal end.
                            ;               Carry OFF: Error end.
                            ;
0142                        UBMAKE:

0142    3A F00C                     LD      A,(USERBIOS)        ; Check size of USERBIOS area size.
0145    FE 10                       CP      RUBSIZE             ;
0147    30 0E                       JR      NC,UBMK10           ; Jump if OK


0149    3A F008                     LD      A,(SIZRAM)          ; Check new size is avalable or not
014C    07                          RLCA                        ; *2 (/1KB --> /512B)
014D    C6 08                       ADD     A,RUBSIZE/2         ;
014F    FE 50                       CP      MAXSZ               ;
0151    D0                          RET     NC                  ; If not available.

0152    3E 10                       LD      A,RUBSIZE           ;  Set new USERBIOS area size.
0154    32 F00C                     LD      (USERBIOS),A        ;

0157                        UBMK10:
0157    DD 21 0016                  LD      IX,SETRAMAD         ; Set new system parameters.
015B    CD 017A                     CALL    JCALLX              ;

015E    3A F086                     LD      A,(RAMD_SIZE)       ; extend RAM size.
0161    47                          LD      B,A                 ;
0162    3A F080                     LD      A,(QT_RAM_IN)       ; Total RAM disk size.
0165    4F                          LD      C,A                 ;
0166    AF                          XOR     A                   ; No format.
0167    DD 21 0019                  LD      IX,CHGRAMD          ; Set new RAM disk size.
016B    CD 017A                     CALL    JCALLX              ;
016E    CD 3154                     CALL    RELOC               ; Relocate RAM disk area.
0171    DD 21 0013                  LD      IX,BIOSJTLD         ; Load BIOS jump table.
0175    CD 017A                     CALL    JCALLX
0178    37                          SCF
0179    C9                          RET

017A                        JCALLX:
017A    F5                          PUSH    AF                  ; Save
017B    3E FF                       LD      A,0FFH              ; Set object bank data. (system bank)
017D    32 F428                     LD      (DISBNK),A          ;
0180    F1                          POP     AF                  ; Restore.
0181    C3 EB69                     JP      CALLX               ;

                            ;       **********************************
                            ;               Relocate RAM disk
                            ;       **********************************
                            ;
                            ;       NOTE :
                            ;                This program's function is to relocate RAM disk contens
                            ;               according to new RAM disk to address.
                            ;
                            ;       <> entry parameter       <>
                            ;               NONE
                            ;       <> return parameter      <>
                            ;               NONE
                            ;
0184                        RELOC:
0184    3A F008                     LD      A,(SIZRAM)          ; Get RAM disk size of standard part.
0187    B7                          OR      A                   ; If size = 00H
0188    C8                          RET     Z                   ; then return.
```

```
0189    07                  RLCA                                 ;
018A    07                  RLCA                                 ; /1Kbyte ==> /1byte
018B    47                  LD      B,A                          ;
018C    0E 00               LD      C,0                          ;

018E    2A F072             LD      HL,(AD_RAM_OLD)              ; Old RAM disk top address.
0191    ED 5B F065          LD      DE,(AD_RAM_IN)               ; New RAM disk top address.

0195    E5                  PUSH    HL                           ; Save.
0196    B7                  OR      A                            ; Clear carry flag.
0197    ED 52               SBC     HL,DE                        ; Check transmit direction.
0199    E1                  POP     HL                           ; Restore
019A    C8                  RET     Z                            ; If does not change then return.

019B    38 03               JR      C,RELOC10                    ; Jump if old top adress < new top
                                                                 ; address.
019D    ED B0               LDIR                                 ; Relocate
019F    C9                  RET                                  ;

01A0                RELOC10:
01A0    0B                  DEC     BC                           ;
01A1    09                  ADD     HL,BC                        ; HLreg.:bottom address of old.
01A2    EB                  EX      DE,HL                        ;
01A3    09                  ADD     HL,BC                        ;
01A4    EB                  EX      DE,HL                        ; DEreg.:bottom address of new.
01A5    03                  INC     BC                           ;
01A6    ED B8               LDDR                                 ; Relocate.
01A8    C9                  RET                                  ;


                    ;
                    ;       ************************************
                    ;               Load object program
                    ;       ************************************
                    ;
                    ;       NOTE :
                    ;               this program is for loading the programs to USER BIOS
area
                    ;
                    ;
                    ;
01A9                UBLOAD:
01A9    21 0284             LD      HL,UBTOP                     ; Load the user program.
01AC    11 CCC0             LD      DE,RUBTOP                    ;
01AF    01 0100             LD      BC,UBSIZE                    ;
01B2    ED B0               LDIR                                 ;

01B4    21 0354             LD      HL,RLSTOP                    ; Load the release-program
01B7    11 DB0C             LD      DE,RRLSTOP                   ;
01BA    01 000B             LD      BC,RLSSIZE                   ;
01BD    ED B0               LDIR
01BF    C9                  RET


                    ;       ****************************************************
                    ;               Check existence of USER-BIOS-HEADER
                    ;       ****************************************************
                    ;
                    ;        << entry parameter >>
                    ;               NONE
                    ;
                    ;        << return parameter >>
                    ;               Zflag ON  : Header exists
                    ;               Zflag OFF : Header does not exist
                    ;
01C0                CHK_HEAD:
01C0    21 0BF0             LD      HL,HEADTOP                   ;
01C3    06 1C               LD      B,1CH                        ; length of USER-BIOS-HEADER
01C5    AF                  XOR     A                            ; Areg.=00H

01C6                CHK_SUM:
01C6    86                  ADD     A,(HL)                       ; Sum checking.
01C7    23                  INC     HL                           ;
01C8    10 FC               DJNZ    CHK_SUM                      ;
01CA    B7                  OR      A                            ; result is 00h ?
01CB    C0                  RET     NZ                           ; No!

01CC    2A 0BF0             LD      HL,(HEADTOP)                 ; Check ID code ('UB')
01CF    11 4255             LD      DE,'BU'                      ;
01D2    ED 52               SBC     HL,DE                        ;
01D4    C9                  RET                                  ;
```

```
                                    ;       ****************************************
                                    ;               Check used module name
                                    ;       ****************************************
                                    ;
                                    ;        <> entry parameter <>
                                    ;               NON
                                    ;
                                    ;       <> return parameter <>
                                    ;               Zflag ON  : Used by oneself.
                                    ;               Zflag OFF : Used by Another user
                                    ;
01D5                                CHK_USB:
01D5    21 0276                             LD      HL,UBNAME       ; New module name.
01D8    11 08F2                             LD      DE,NAME         ; current module name
01DB    06 08                               LD      B,08H           ; length.

01DD                                UB_CHK:
01DD    1A                                  LD      A,(DE)          ; Check module name.
01DE    BE                                  CP      (HL)            ;
01DF    C0                                  RET     NZ              ;
01E0    23                                  INC     HL              ;
01E1    13                                  INC     DE              ;
01E2    10 F9                               DJNZ    UB_CHK          ;
01E4    AF                                  XOR     A               ;
01E5    C9                                  RET

                                    ;       ********************************
                                    ;               Dispaly a message
                                    ;       ********************************
                                    ;
                                    ;       <> entry parameter <>
                                    ;               HLreg. : Start address of a message.
                                    ;                        The message must be terminated by '00H'
                                    ;
01E6                                PRMSG:
01E6    7E                                  LD      A,(HL)          ; Read display data.
01E7    23                                  INC     HL              ;
01E8    B7                                  OR      A               ;
01E9    C8                                  RET     Z               ; End of data.

01EA    4F                                  LD      C,A             ;
01EB    E5                                  PUSH    HL              ;
01EC    CD 880C                             CALL    CONOUT          ;  Display it.
01EF    E1                                  POP     HL              ;
01F0    18 F4                               JR      PRMSG           ;

                                    ;       ***************************
                                    ;               Message area
                                    ;       ***************************

01F2                                MSGCMP:
01F2    0C 0D 0A                            DB      CLS,CR,LF
01F5    1B 32                               DB      ESC,'2'
01F7    20 43 6F 6D                         DB      ' Completed'
01FB    70 6C 65 74
01FF    65 64
0201    00                                  DB      00H

0202                                MSGERR:
0202    0C 0D 0A                            DB      CLS,CR,LF
0205    1B 32                               DB      ESC,'2'
0207    43 61 6E 6E                         DB      'Cannot mount'
020B    6F 74 20 6D
020F    6F 75 6E 74
0213    00                                  DB      00H

                                    ;       < stack area define >

0214                                                DS      40H
0254                                STACKAD         EQU     $
```

```
;*****************************************************************************

                                    ; Caution :
                                    ;       It is enough to change next routines that you want to use
                                    ;       the USER BIOS AREA.
                                    ;

;*****************************************************************************
```

```
                                ;       ****************************************************
                                ;               Open porcess for USERBIOS area
                                ;       ****************************************************

0254                            OPNUBIOS:

                                ;**********************************************************
                                ;*                                                        *
                                ;*  If you will use system hooks ,please rewrite them     *
                                ;*  in this routine.                                      *
                                ;*                                                        *
                                ;**********************************************************
                                ;
0254    C9                              RET




                                ;       ***********************************************
                                ;               Set head_r of USER-BIOS-AREA
                                ;       ***********************************************
                                ;
                                ;       <> entry parameter <>
                                ;               NONE
                                ;       <> return parameter <>
                                ;               NONE

0255                            MAKEHEAD:

0255    21 D800                         LD      HL,RELSTOP      ; Set address of release routine.
0258    22 0280                         LD      (REL_IC),HL     ;

025B    21 0274                         LD      HL,HEAD_INF     ; Calculate check sum data.
025E    06 0F                           LD      B,0FH           ; header size.
0260    AF                              XOR     A               ; Areg.= 00H.

0261                            CAL_SUM:
0261    96                              SUB     (HL)            ;
0262    23                              INC     HL              ; Sum check.
0263    10 FC                           DJNZ    CAL_SUM         ;

0265    32 0283                         LD      (HD_CHK_SUM),A  ; set check sum data.

0268    21 0274                         LD      HL,HEAD_INF     ; Copy new header.
026B    11 D873                         LD      DE,HEADTOP      ;
026E    01 0010                         LD      BC,10H          ;
0271    ED B0                           LDIR
0273    C9                              RET


0274    55 42                   HEAD_INF:       DB      'UB'            ; ID of header.
0276    54 45 53 54             UBNAME:         DB      'TESTPROG'      ; Name of program. (Please
                                                                        ; change to
027A    50 52 4F 47
                                                                        ; own name)
027E    10                                      DB      BUSIZE          ; Size of program. (256 byte
                                                                        ; boundary)
027F    00                                      DB      00H             ; Overwrite flag.
0280                            REL_IC:         DS      2               ; address of release routine.
0282    00                                      DB      00H             ;
0283                            HD_CHK_SUM:     DS      1               ; check sum data.

0284                            PAUSE           EQU     $
                                                .DEPHASE


                                ;       **********************************
                                ;               User program define
                                ;       **********************************



                                                .PHASE          RUBTOP
0284                            UBTOP           EQU     PAUSE


CC00                                            DS      100H            ;(DUMMY)
```

```
                              ;*****************************************************************
                              ;*                                                               *
                              ;*  You will set up your own routine that will be relocated      *
                              ;*  to USERBIOS area in this storage                             *
                              ;*                                                               *
                              ;*****************************************************************

 0100                         UBSIZE          EQU     $-RUBTOP

                                              .DEPHASE

                              ;
                              ;       ******************************
                              ;               Release routine
                              ;       ******************************
                              ;
                              ;       This routine will be mounted on HOOK AREA.
                              ;

                                              .PHASE  RRLSTOP
 0354                         RLSTOP          EQU     PAUSE+UBSIZE

                              ;*****************************************************************
                              ;*                                                               *
                              ;*  If you will use system hooks,you must initialize them        *
                              ;*  in this routine.                                             *
                              ;*                                                               *
                              ;*****************************************************************

                              ;       < Header null clear >

 D800     06 10                       LD      B,10H           ; Counter set.
 D802     21 DBF0                     LD      HL,HEADTOP      ; Header top address.

 D805                         REL_LOOP:
 D805     36 00                       LD      (HL),00H        ; Null clear.
 D807     23                          INC     HL              ;
 D808     10 FB                       DJNZ    REL_LOOP        ;
 D80A     C9                          RET

 000B                         RLSSIZE         EQU     $-RRLSTOP


                                      END
Macros:

Symbols:
F0C5    AD_RAM_IN       F072    AD_RAM_OLD      EB39    BEEP
0013    BIOSJTLD        EB00    CALLX           0261    CAL_SUM
0019    CBCBAND         DBFF    CBRSCR          01C2    CBK_READ
01C6    CHK_SUM         01D5    CHK_CSB         00CC    CLS
EB0C    CONOUT          000D    CR              F419    DISBNK
013A    ERRESD          001B    ESC             C970    EDID
0112    HEAD20          011B    HEAD40          DBF0    HEADTOP
0274    HEAD_INP        017A    JCALLX          000A    LF
012F    MAIN_END        0255    MAKEREAD        005D    MAKSE
0202    MSGERR          01F2    MSGOK           DBF2    NAME
0254    OPNCBIOS        DBFB    OVWRT           0254    PAUSE
0126    PRNSC           F000    QT_RAM_IN       F065    RAMD_SIZE
0154    RELOC           01A0    RELOC10         0220    REL_IC
D805    REL_LOOP        DBFC    RLSAD           000B    RLSSIZE
0354    RLSTOP          D800    RRLSTOP         0010    RUBSIZE
CC00    RUBTOP          0016    SETRAMAD        DBFA    SIZE
F00B    SIZRAM          0254    STACKAD         0C00    UBBOTTOM
0100    UBICS           0149    UBLOAD          0182    UBMAKE
0157    UBMK10          0276    UBNAME          01CC    UBSIZE
0254    UBTOP           0100    UB_CHR          0253    UB_CHK_SUM
DBFE    UNUSE           F00C    USERBIOS        E9C2    42007


No Fatal error(s)
```

(25) SAMPLE 25. AUTO POWEROFF

```
                         ;         ****************************************************
                         ;                   AUTO POWER OFF SAMPLE PROGRAM
                         ;         ****************************************************
                         ;
                         ;         NOTE:
                         ;                This sample program gets the key-in data and
                         ;                checks auto-power-off time.
                         ;
                         ;         <> assemble condition    <>
                         ;
                                   .Z80
                         ;
                         ;         <> loading address       <>
                         ;
                                   .PHASE   100H
                         ;
                         ;         <> constant values       <>
                         ;
                         ;                BIOS entries
                         ;
EB03                     WBOOT          EQU      0EB03H
EB06                     CONST          EQU      0EB06H
EB09                     CONIN          EQU      0EB09H
EB0C                     CONOUT         EQU      0EB0CH
EB6F                     PUTPFK         EQU      0EB6FH
EB7E                     POWEROFF       EQU      0EB7EH
EBA2                     BACKLIGT       EQU      0EBA2H
                         ;
                         ;                System work addresses
                         ;
F020                     ATSHUTOFF      EQU      0F020H
F021                     ATSOTIME       EQU      0F021H
F023                     SLOFTIME       EQU      0F023H
F02C                     TIMER0         EQU      0F02CH
F387                     ?LOFTIME       EQU      0F387H
F5F2                     TXCOUNTRY      EQU      0F5F2H
F5F9                     TIMEEND        EQU      0F5F9H
                         ;
                         ;                Other constants
                         ;
1000                     MAINSP         EQU      01000H
                         ;
001B                     STOP           EQU      001BH
                         ;
                         ;         ****************************************************
                         ;                   MAIN PROGRAM
                         ;         ****************************************************
                         ;
0100                     START:
0100    31 1000                    LD      SP,MAINSP      ; Set stack pointer
0103    CD 0114                    CALL    KEYSET         ; Set default key table.
                         ;
0106    CD 0122          LOOP:     CALL    KEYINP         ; Get key-in data.
0109    4F                         LD      C,A            ;
010A    FE 1B                      CP      STOP           ; Is it stop code?
010C    CA EB03                    JP      Z,WBOOT        ; Yes. (End of program)
010F    CD EB0C                    CALL    CONOUT         ; Display key-in data.
0112    18 F2                      JR      LOOP           ; Loop
                         ;
                         ;         ****************************************************
                         ;                   SET DEFAULT KEY TABLE
                         ;         ****************************************************
                         ;
                         ;         NOTE:
                         ;                Set default key table by each machine.
                         ;
                         ;         <> entry parameter       <>
                         ;                NONE
                         ;         <> return parameter      <>
                         ;                NONE
                         ;         <> preserved registers   <>
                         ;                NONE
                         ;
0114                     KEYSET:
0114    3A F5F2                    LD      A,(TXCOUNTRY)  ; Check machine type
0117    07                         RLCA                   ; Is it ENT-10
0118    01 FF01                    LD      BC,0FF01H      ;
011B    21 0173                    LD      HL,KEYTBL      ; Default key table for ENT-10
011E    D4 EB6F                    CALL    NC,PUTPFK      ; Set all default keys
0121    C9                         RET                    ;
```

```
                        ;
                        ;       ************************************************************
                        ;                       GET KEY-IN DATA
                        ;       ************************************************************
                        ;
                        ;       NOTE:
                        ;               Get key-in data from console.
                        ;               This routine uses BIOS CONST, so you can check
                        ;               other status
                        ;
                        ;       <> entry parameter              <>
                        ;               NONE
                        ;       <> return parameter             <>
                        ;               A  : Key-in data.
                        ;       <> preserved registers          <>
                        ;               NONE
                        ;
0122                    KEYINP:
0122    2A F021                 LD      HL,(ATSOTIME)   ; Set auto power-off time (Step 1)
0125    ED 5B F02D              LD      DE,(TIMERO)     ;
0129    19                      ADD     HL,DE           ;
012A    22 F5F9                 LD      (TIMEEND),HL    ;
                        ;
012D    2A F023                 LD      HL,(BLOFTIME)   ; Set auto backlight-off time (Step 2)
0130    19                      ADD     HL,DE           ;
0131    22 F387                 LD      (BLOFFEND),HL   ;
                        ;
0134                    KEY110:
0134    CD EB06                 CALL    CONST           ; Check console inputted status
0137    3C                      INC     A               ; Inputted any key?
0138    28 2E                   JR      Z,KEY190        ; Yes
                        ;
013A    3A F020                 LD      A,(ATSHUTOFF)   ; Check auto power-off time (Step 3)
013D    B7                      OR      A               ; Disable auto power-off?
013E    28 0F                   JR      Z,KEY120        ; Yes
0140    2A F5F9                 LD      HL,(TIMEEND)    ;
0143    ED 5B F02D              LD      DE,(TIMERO)     ;
0147    B7                      OR      A               ;
0148    ED 52                   SBC     HL,DE           ; Auto power-off time has been reached?
014A    0E 01                   LD      C,01H           ;
014C    FC E87C                 CALL    M,POWEROFF      ; Yes (Power off by continue mode)
                        ;
014F                    KEY120:
014F    2A F023                 LD      HL,(BLOFTIME)   ; Check auto backlight-off time (Step 4)
0152    7C                      LD      A,H             ;
0153    B5                      OR      L               ; Disable auto backlight-off?
0154    28 0F                   JR      Z,KEY130        ; Yes
0156    2A F387                 LD      HL,(BLOFFEND)   ; Check auto backlight-off time (Step 4)
0159    ED 5B F02D              LD      DE,(TIMERO)     ;
015D    B7                      OR      A               ;
015E    ED 52                   SBC     HL,DE           ; Auto backlight-off time has been
                                                        ; reached?
0160    0E 00                   LD      C,00H           ;
0162    FC EBA2                 CALL    M,BACKLIGT      ; Yes (Backlight off)
                        ;
                        ;       You can check another status. For example, check printer-ready-
                        ;       status or receive-interrupt status, etc.
                        ;
0165                    KEY130:
0165    76                      HALT                    ;
0166    18 CC                   JR      KEY110          ;
                        ;
                        ;
0168                    KEY190:
0168    CD EB09                 CALL    CONIN           ; Get key-in data.
016B    F5                      PUSH    AF              ;
016C    0E 01                   LD      C,01H           ;
016E    CD EBA2                 CALL    BACKLIGT        ; Backlight on
0171    F1                      POP     AF              ;
0172    C9                      RET                     ;
                        ;
                        ;       ************************************************************
                        ;                       DEFAULT KEY DATA TABLE
                        ;       ************************************************************
                        ;
                        ;       Default key table for EHT-10
                        ;
0173                    KEYTBL:
0173    01 FF FF FF             DB      001H,   0FFH,   0FFH,   0FFH,   0FFH
0177    FF
0178    FF FF FF FF             DB      0FFH,   0FFH,   0FFH,   0FFH,   0FFH
017C    FF
017D    FF FF FF FF             DB      0FFH,   0FFH,   0FFH,   0FFH,   0FFH
0181    FF
0182    FF FF FF FF             DB      0FFH,   0FFH,   0FFH,   0FFH,   0FFH
0186    FF
0187    FF FF FF FF             DB      0FFH,   0FFH,   0FFH,   0FFH,   0FFH
018B    FF
018C    FF FF FF FF             DB      0FFH,   0FFH,   0FFH,   0FFH,   0FFH
0190    FF
0191    FF FF FF FF             DB      0FFH,   0FFH,   0FFH,   0FFH,   0FFH
0195    FF
0196    FF FF FF FF             DB      0FFH,   0FFH,   0FFH,   0FFH,   0FFH
019A    FF
```

```
019B      FF FF FF FF          DB      0FFH,   0FFH,   0FFH,   0FFH,   0FFH
019F      FF
01A0      FF FF FF FF          DB      0FFH,   0FFH,   0FFH,   0FFH,   0FFH
01A4      FF
01A5      FF FF FF FF          DB      0FFH,   0FFH,   0FFH,   0FFH,   0FFH
01A9      FF
01AA      FF FF FF FF          DB      0FFH,   0FFH,   0FFH,   0FFH,   0FFH
01AE      FF
01AF      FF FF FF FF          DB      0FFH,   0FFH,   0FFH,   0FFH,   0FFH
01B3      FF
01B4      FA FB FC FD          DB      0FAH,   0FBH,   0FCH,   0FDH,   0FEH
01B8      FE

                               END
```

(26) SAMPLE 26. BDOS ERROR a.

```
                                ;       ********************************************
                                ;               SAMPLE PROGRAM FOR BDOS ERROR
                                ;       ********************************************
                                ;
                                ;       NOTE:
                                ;               This sample program shows how to change BDOS error vector.
                                ;
                                ;               Its function is same as SETERR.
                                ;
                                ;       <> assemble condition   <>
                                ;
                                                .Z80
                                ;
                                ;       <> loading address      <>
                                ;
                                                .PHASE 100H
                                ;
                                ;       <> constant values      <>
                                ;
EB03                            WBOOT           EQU     0EB03H          ; WBOOT entry address.
F7A8                            JJUMPXX         EQU     0F7A8H          ; JUMPX address in resident
                                                                        ; jump table.
0010                            GOBACK          EQU     0010H           ; BDOS end process.
                                ;
F418                            DISBNK          EQU     0F418H          ; JUMPX parameter address.
F3D1                            ABRT            EQU     0F3D1H          ; BDOS error code save area.
0005                            BDOSE           EQU     0005H           ; BDOS entry address.
                                ;
                                ;       ********************************************
                                ;               Set BDOS error recovery mode
                                ;       ********************************************
                                ;
0100                            SETVECT:
0100    21 0110                         LD      HL,XERRVTB              ; New BDOS error vector address.
0103    ED 5B 0006                      LD      DE,(BDOSE+1)            ; BDOS1 entry address.
0107    13                              INC     DE                      ;
0108    13                              INC     DE                      ;
0109    13                              INC     DE                      ; DEreg.: Error vector top.
010A    01 0008                         LD      BC,0008H                ; COPY.
010D    ED B0                           LDIR                            ;
010F    C9                              RET                             ;
                                ;
                                ;       <> Error Vector Table <>
                                ;
0110                            XERRVTB:
0110    0119                            DW      XBADSEC         ; Bad sector error.
0112    011D                            DW      XBADSEL         ; Bad select error.
0114    0122                            DW      XRODSK          ; R/O disk error.
0116    0127                            DW      XROFILE         ; R/O file error.
                                ;
0119                            XBADSEC:
                                ;**************************************************
                                ;*  You can set your own BAD SECTOR ERROR recovery *
                                ;*  routine in this storage.                       *
                                ;**************************************************
0119    26 01                           LD      H,1             ; Set error code.
011A    C3 012A                         JP      XERROR          ;
                                ;
011D                            XBADSEL:
                                ;**************************************************
                                ;*  You can set your own BAD SELECT ERROR recovery *
                                ;*  routine in this storage.                       *
                                ;**************************************************
011D    26 02                           LD      H,2             ; Set error code.
011F    C3 012A                         JP      XERROR
                                ;
0122                            XRODSK:
                                ;**************************************************
                                ;*  You can set your own R/O DISK ERROR recovery   *
                                ;*  routine in this storage.                       *
                                ;**************************************************
0122    26 03                           LD      H,3             ; Set error code.
0124    C3 012A                         JP      XERROR          ;
```

```
0127                      XROFILE:
                          ;***********************************************************
                          ;*  You can set your own I/O FILE ERROR recovery           *
                          ;* routine in this storage.                                *
                          ;***********************************************************
0127    21 0004                   LD      HL,4            ; Set error code.


012A                      $ERROR:
012A    2E FF                     LD      L,0FFH          ; Set error code to return parameter.
012C    22 F3D1                   LD      (ABEXT),HL      ;
012F    DD 21 0010                LD      IX,GOBACK       ; BDOS exit process routine.
0133    3E FF                     LD      A,0FFH          ; Set to system bank.
0135    32 F418                   LD      (DISBNK),A      ;
0138    C3 FFA8                   JP      JJUMPXX         ; JUMPX


                                  END
```

```
                        ;       ***********************************************
                        ;               SAMPLE PROGRAM FOR SETERR,RSTERR
                        ;       ***********************************************
                        ;
                        ;       NOTE:
                        ;               This sample program shows how to use SETERR and RSTERR.
                        ;
                        ;
                        ;       <> assemble condition   <>

                                        .Z80

                        ;       <> loading address      <>

                                        .PHASE 100H

                        ;       <> constant values      <>

EB03                    WBOOT           EQU     0EB03H          ; WBOOT entry address.
EB69                    CALLX           EQU     0EB69H          ; CALLX entry address.
EBA5                    INFORM          EQU     0EBA5H          ; INFORM entry address.
000A                    SETERR          EQU     0000AH          ; SETERR address in system
                                                                ; jump table.
000D                    RSTERR          EQU     0000DH          ; RSTERR address in system
                                                                ; jump table.

                        ;       ********************************************
                        ;               Set BDOS error recovery mode
                        ;       ********************************************


0100                    SETMODE:
0100    DD 21 000A              LD      IX,SETERR       ; SETERR address in system jump table.
0104    18 04                   JR      CHGERR          ;

0106                    RSTMODE:
0106    DD 21 000D              LD      IX,RSTERR       ; RSTERR address in system jump table.

010A                    CHGERR:
010A    0E 03                   LD      C,3             ; Get DISBNK address.
010C    CD EBA5                 CALL    INFORM          ;
010F    3E FF                   LD      A,0FFH          ; Set system bank data to DISBNK
0111    77                      LD      (HL),A          ;
0112    CD EB69                 CALL    CALLX           ; Do SETERR or RSTERR.
0115    C9                      RET                     ;



                        ;       ******************************
                        ;               Check BDOS error
                        ;       ******************************


0116                    CHKERR:
0116    B7                      OR      A               ; If normal return then return.
0117    C8                      RET     Z               ;

0118    E5                      PUSH    HL              ; Save return parameters.
0119    D5                      PUSH    DE              ;
011A    C5                      PUSH    BC              ;
011B    F5                      PUSH    AF              ;

011C    11 0130                 LD      DE,ERRVTB       ; Error vector top address.
011F    EB                      EX      DE,HL           ;
0120    19                      ADD     HL,DE           ;
0121    19                      ADD     HL,DE           ; Calculate object error handler address.
0122    7E                      LD      A,(HL)          ;
0123    23                      INC     HL              ;
0124    66                      LD      H,(HL)          ;
0125    6F                      LD      L,A             ; HLreg. : object error handler address.
0126    11 012B                 LD      DE,ERRRET       ; Set return address.
0129    D5                      PUSH    DE              ;
012A    E9                      JP      (HL)            ; CALL error handler.


012B                    ERRRET:
012B    F1                      POP     AF              ;
012C    C1                      POP     BC              ; Restore BDOS return parameter.
012D    D1                      POP     DE              ;
012E    E1                      POP     HL              ;
012F    C9                      RET

                        ;       <> Error Vector Table <>
```

```
0130                    ERRVTR:
0130    0138                    DW      BADSEC          ; Bad sector error.
0132    0139                    DW      BADSEL          ; Bad select error.
0134    013A                    DW      RODSK           ; R/O disk error.
0136    013B                    DW      ROFILE          ; R/O file error.



0138                    BADSEC:
                        ;**************************************************************
                        ;*  You can set your own BAD SECTOR ERROR recovery           *
                        ;* routine in this strage.                                    *
                        ;**************************************************************
0138    C9                      RET
0139                    BADSEL:
                        ;**************************************************************
                        ;*  You can set your own BAD SELECT ERROR recovery           *
                        ;* routine in this strage.                                    *
                        ;**************************************************************
0139    C9                      RET
013A                    RODSK:
                        ;**************************************************************
                        ;*  You can set your own R/O DISK ERROR recovery             *
                        ;* routine in this strage.                                    *
                        ;**************************************************************
013A    C9                      RET
013B                    ROFILE:
                        ;**************************************************************
                        ;*  You can set your own R/O FILE ERROR recovery             *
                        ;* routine in this strage.                                    *
                        ;**************************************************************
013B    C9                      RET


                                END
```

```
                                ;       ****************************************************
                                ;               POWER OFF & ALARM CONTROL SAMPLE PROGRAM
                                ;       ****************************************************
                                ;
                                ;       NOTE :
                                ;                This program shows how to disable and check
                                ;               interrupt. (caused by power off and alarm)
                                ;
                                ;       <> assemble condition   <>

                                                .Z80

                                ;       <> loading address      <>

                                                .PHASE  100H

                                ;       <> constant values      <>

F0B4                            YPOFDS          EQU     0F0B4H          ; Power OFF int. disable flag.
F0B5                            YPOFST          EQU     0F0B5H          ; Power OFF int. status.
F0B6                            YALMDS          EQU     0F0B6H          ; ALARM int. disable flag.
F0B7                            YALMST          EQU     0F0B7H          ; ALARM int. status.

FF90                            JPSTBIOS        EQU     0FF90H          ; Post-BIOS entry (resident
                                                                        ; system)

0100    C3 0137                         JP      START           ; Jump to sample routine.


                                ;       ****************************************
                                ;               Disable Power OFF & Alarm
                                ;       ****************************************
                                ;
                                ;       NOTE :
                                ;               This program's function is disable Power off & ALARM int.
                                ;
                                ;       <> entry parameter      <>
                                ;               NONE
                                ;       <> return parameter     <>
                                ;               NONE
                                ;       <> preserved registers  <>
                                ;               All preserved.

0103                            DISABLE:
0103    E5                              PUSH    HL              ; Save register.
0104    21 F0B4                         LD      HL,YPOFDS       ; Set power OFF disable bit for user.
0107    CB F6                           SET     6,(HL)          ;

0109    21 F0B6                         LD      HL,YALMDS       ;
010C    CB F6                           SET     6,(HL)          ; Set alarm disable bit for user.
010E    E1                              POP     HL              ;
010F    C9                              RET                     ;


                                ;       ****************************************
                                ;               Check interrupt & execute
                                ;       ****************************************
                                ;
                                ;       NOTE :
                                ;                This pergram's function is check and enable Power OFF
                                ;               and ALARM int.
                                ;               If Power OFF or ALARM int has occured, execute power off
                                ;               or display the alarm message.
                                ;
                                ;
                                ;       <> entry parameter      <>
                                ;               NONE
                                ;       <> return parameter     <>
                                ;               NONE
                                ;       <> preserved registers  <>
                                ;               All preserved

0110                            CHKINT:
0110    E5                              PUSH    HL              ; Save registers.
0111    F5                              PUSH    AF              ;

0112    21 F0B4                         LD      HL,YPOFDS       ; Reset power OFF disbale bit for user.
0115    CB B6                           RES     6,(HL)          ;

0117    21 F0B5                         LD      HL,YPOFST       ; Power OFF interrupt status bit.
011A    CD 012E                         CALL    BITCOPY         ; Copy user bit to BIOS.

011D    21 F0B6                         LD      HL,YALMDS       ; Reset alarm disable bit.
0120    CB B6                           RES     6,(HL)          ;
```

```
0122    21 F0B1                 LD      HL,YALMST       ; Alarm interrupt status bit.
0125    CD 012E                 CALL    BITCOPY         ; Copy user bit to BIOS.

0128    F1                      POP     AF              ; restore registers.
0129    E1                      POP     HL              ;
012A    CD FF96                 CALL    JPSTBIOS        ; Call PSTBIOS.
                                                        ; ( If Power OFF or Alarm interrupt
                                                        ; is occured,
                                                        ;    the PSTBIOS executes Power OFF
                                                        ;    or displays
                                                        ;    Alarm message.)
012D    C9                      RET                     ;

012E                    BITCOPY:
012E    7E                      LD      A,(HL)          ; Get interrupt status(Power OFF or Alarm)
012F    E6 40                   AND     01000000B       ; mask out user bit.
0131    07                      RLCA                    ; Copy to BIOS bit.
0132    B6                      OR      (HL)            ; Merge.
0133    E6 BF                   AND     10111111B       ; Reset user bit.
0135    77                      LD      (HL),A          ; Set new status.
0136    C9                      RET                     ;


                        ;       ******************************
                        ;               Sample program
                        ;       ******************************
                        ;
                        ;       NOTE :
                        ;               In this program, ALARM & power OFF int. is disabled
                        ;               while inputting & displaying until the loop counter
                        ;               becomes zero.
                        ;
EB03                    WBOOT           EQU     0EB03H          ; WBOOT entry.
EB06                    CONST           EQU     0EB06H          ; BIOS CONST entry.
EB09                    CONIN           EQU     0EB09H          ; BIOS CONIN entry.
EB0C                    CONOUT          EQU     0EB0CH          ; BIOS CONOUT entry.
EB98                    TOUCH           EQU     0EB98H          ; BIOS TOUCH entry.


0137                    START:
0137    11 0160                 LD      DE,KEYDATA      ;  key define.
013A    0E 02                   LD      C,2             ;
013C    CD EB98                 CALL    TOUCH           ;

013F                    START2:
013F    CD 0103                 CALL    DISABLE         ; Power OFF & alarm int.disable.
0142    06 0A                   LD      B,10            ; loop counter.

0144                    LOOP2:
0144    C5                      PUSH    BC              ; save loop counter.

0145                    LOOP1:
0145    CD EB06                 CALL    CONST           ; Check key input status.
0148    3C                      INC     A               ;
0149    20 FA                   JR      NZ,LOOP1        ; loop of not key in.

014B    CD EB09                 CALL    CONIN           ; Get key.
014E    FE 03                   CP      03H             ; If end key then WBOOT
0150    CA EB03                 JP      Z,WBOOT         ;
0153    4F                      LD      C,A             ;
0154    CD EB0C                 CALL    CONOUT          ; display inputted code.
0157    C1                      POP     BC              ;
0158    10 EA                   DJNZ    LOOP2           ; loop

015A    CD 0113                 CALL    CHKINT          ; Check power off and alarm int.
                                                        ; If int. has already occured,then
                                                        ; execute.
015D    C3 013F                 JP      START2          ; loop.

0160                    KEYDATA:
0160    01 03 05 01             DB      1,3,5,1,41H,7,0,'INPUT KEY'
0164    41 07 00 49
0168    4E 50 55 54
016C    20 4B 45 59
0170    01 07 05 01             DB      1,7,5,1,03H,7,0,'   END   '
0174    03 07 00 20
0178    20 20 45 4E
017C    44 20 20 20

                                END
```

(28) SAMPLE 28. Mount Check of the cartridge device

```
                                ;       ****************************************
                                ;               CBGHOOK SAMPLE PROGRAM
                                ;       ****************************************
                                ;
                                ;       NOTE :
                                ;               This program shows how to use CBGHOOK.
                                ;               This program is constructed of next 3 modules.
                                ;
                                ;                       1. CBGHOOK expand module.(CBGPHOOK)
                                ;                       2. Setup module.(CBGSETUP)
                                ;                       3. Release module.(CBGRLS)
                                ;
                                ;               If you use USER BIOS area for this program, you must
                                ;               execute UBIOS program. (see Chapter 3.6)
                                ;
                                ;
                                ;       <> assemble condition   <>
                                ;
                                                .Z80
                                ;
                                ;       <> loading address      <>
                                ;
                                                .PHASE  100H
                                ;
                                ;       <> constant values      <>
                                ;
FF6C                            CBGHOOK         EQU     0FF6CH          ; CBGHOOK entry address.
F000                            RETADD          EQU     0F000H          ; 'RET' instruction address.
F42F                            CBGDEV          EQU     0F42FH          ; Cartridge device code address
F00F                            USERCBC         EQU     0F00FH          ; User cartridge check flag.

0016                            ZIOSTR          EQU     016H            ; For printer busy
0010                            ZCMSDOR         EQU     010H            ; For output register.
0011                            ZCMSSR          EQU     011H            ; For output register status.
0019                            ZIOCTLR         EQU     019H

0003                            M190            EQU     003H            ; Device code for printer unit.
001B                            ESC             EQU     01BH            ; Printer ESC sequence code.



                                ;       ****************************************
                                ;               Expand module for CBGHOOK
                                ;       ****************************************

0100                            CBGPHOOK:
0100    3A F42F                         LD      A,(CBGDEV)              ; If cartridge device is not
                                                                        ; printer
0103    FE 03                           CP      M190                    ; unit,
0105    C0                              RET     NZ                      ; then return.

0106    21 0121                         LD      HL,PRNDATA              ; Printer initial data address.
0109    46                              LD      B,(HL)                  ; Get data number.

010A                            PRNLOOP:
010A    C5                              PUSH    BC                      ; save loop counter.
010B    23                              INC     HL                      ;
010C    4E                              LD      C,(HL)                  ; Get 1 data
010D    CD 0114                         CALL    PRNOUT                  ; Output data
0110    C1                              POP     BC                      ; recover loop counter.
0111    10 F7                           DJNZ    PRNLOOP                 ; loop
0113    C9                              RET                             ;

0114                            PRNOUT:
0114    DB 16                           IN      A,(ZIOSTR)              ; bit0 is printer ready status.
                                                                        ;          ( 1 is busy )
0116    47                              LD      B,A                     ;
0117    DB 11                           IN      A,(ZCMSSR)              ; bit0 is output register status
                                                                        ;          ( 1 is ready )
0119    B0                              OR      B                       ;
011A    0F                              RRCA                            ; bit0 set Carry
011B    38 F7                           JR      C,PRNOUT                ; Loop if busy.

011D    79                              LD      A,C                     ;
011E    D3 10                           OUT     (ZCMSDOR),A             ; Output.
0120    C9                              RET                             ;

                                ;       < initial data >

0121                            PRNDATA:
0121    22                              DB      022H                    ; data number.
0122    1B 26                           DB      ESC,'&'                 ; Set down load character,
0124    E4 E8                           DB      0E4H,0E8H               ; address 0E4H - 0E8H
0126    4E 20 1F 20                     DB      04EH,020H,01FH,020H,04EH,000H   ; Tuesday
012A    4E 00
012C    24 14 7F 14                     DB      024H,014H,07FH,014H,022H,000H   ; Wednesday
0130    22 00
```

```
0132    24 14 7F 14              DB      024H,014H,07FH,014H,024H,000H   ; Thursday
0136    24 00
0138    54 6A 79 6A              DB      054H,06AH,079H,06AH,054H,000H   ; Friday
013C    54 00
013E    40 44 7F 44              DB      040H,044H,07FH,044H,040H,000H   ; Saturday
0142    40 00

                         ;       **************************************
                         ;               CRCHOOK setup module
                         ;       **************************************

0144                     CRCSETUP:
0144    F3                       DI                                     ; ***** DI *****
0145    3A 0158                  LD      A,(CRCBANK)                    ; Get expand procedure bank.
0148    32 FF6C                  LD      (CRCHOOK),A                    ; Set
014B    2A 0159                  LD      HL,(CRCADDR)                   ; Set expand procedure address.
014E    22 FF6D                  LD      (CRCHOOK+1),HL                 ; Set
0151    3E 02                    LD      A,0150                         ; Set User code.
0153    32 F0DF                  LD      (USERCRC),A                    ;
0156    FB                       EI                                     ; ***** EI *****
0157    C9                       RET                                    ;

0158    00               CRCBANK:        DB      00H                    ; Expand procedure bank (BANK 0=0)
0159    0100             CRCADDR:        DW      CRCPROG                ; Expand procedure address.

                         ;       ********************************************
                         ;               Release expand procedure
                         ;       ********************************************

015B                     CRCREL:
015B    F3                       DI                                     ; ***** DI *****
015C    3E FF                    LD      A,0FFH                         ; Set system bank.
015E    32 FF6C                  LD      (CRCHOOK),A                    ;
0161    21 F000                  LD      HL,RETADD                      ; Set return address.
0164    22 FF6D                  LD      (CRCHOOK+1),HL                 ;
0167    AF                       XOR     A                              ;
0168    32 F0DF                  LD      (USERCRC),A                    ; Reset User flag.
016B    FB                       EI                                     ; ***** EI *****
016C    C9                       RET                                    ;

                                 END
```

(29) SAMPLE 29. ART Interrupt

```
                        ;       ************************************************
                        ;               ART(Rx-Ready) HOOK SAMPLE PROGRAM
                        ;       ************************************************
                        ;
                        ;       NOTE:
                        ;               This sample program does as following.
                        ;                1. Move extend-part to 8000H in RAM.
                        ;                2. Re-write ART(Rx-Ready) hook address to 8000H
                        ;                3. Extend-part is to receive data & set it into
                        ;                   receive-buffer.
                        ;                4. Neglect system's process.
                        ;               Extend part neglects XON/XOFF, DTR/DSR, RTS/CTS,
                        ;             and SI/SO controll.
                        ;               This program doesn't check whether 8000H of RAM is
                        ;             free or not. You must decrease RAM disk to gain enough
                        ;             User-BIOS area.(If you use User-BIOS area, you must
                        ;             change loading address.)
                        ;
                        ;       <> assemble condition   <>
                        ;
                                .Z80
                        ;
                        ;       <> loading address      <>
                        ;
                                .PHASE  100H
                        ;
                        ;       <> constant values      <>
                        ;
                        ;               BIOS entries
                        ;
EB03                    WBOOT           EQU     0EB03H
                        ;
                        ;               System work addresses
                        ;
F388                    RSINTST         EQU     0F388H
F41C                    RZBANKR         EQU     0F41CH
F42D                    RZSBBNKR        EQU     0F42DH
F604                    RSPSTS          EQU     0F604H
F605                    RSPRBGP         EQU     0F605H
F607                    RSPRBPP         EQU     0F607H
F609                    RSPRHAD         EQU     0F609H
F612                    RSRBEAD         EQU     0F612H
F61C                    RSRDL           EQU     0F61CH
                        ;
                        ;               Other constants
                        ;
0005                    ZBANKR          EQU     05H
0014                    ZARTDIR         EQU     14H
0015                    ZARTSR          EQU     15H
0016                    ZIOSTR          EQU     16H
0022                    ZSBBNKR         EQU     22H

0002                    ZRXRDY          EQU     00000010B
0080                    ZRDSR           EQU     10000000B
0010                    ZRCD            EQU     00100000B

1000                    MAINSP          EQU     01000H
8000                    LOADADDR        EQU     08000H
FFD2                    ARTHOOK         EQU     0FFD2H
0042                    RAMBK           EQU     00042H

0080                    RSSDSR          EQU     10000000B
0040                    RSSFRE          EQU     01000000B
0020                    RSSORE          EQU     00100000B
0010                    RSSPRE          EQU     00010000B
0008                    RSSCD           EQU     00001000B
0004                    RSSRBO          EQU     00000100B
0002                    RSSRBF          EQU     00000010B
0007                    RSDSR           EQU     7
0006                    RSFRE           EQU     6
0005                    RSORE           EQU     5
0004                    RSPRE           EQU     4
0003                    RSCD            EQU     3
0002                    RSRBO           EQU     2
0001                    RSRBF           EQU     1

                        ;
                        ;       ************************************************
                        ;               MAIN PROGRAM
                        ;       ************************************************
                        ;
0100                    START:
0100    31 1000                 LD      SP,MAINSP       ; Set stack pointer
0103    F3                      DI                      ; Disable all interrupt
0104    21 011E                 LD      HL,HOOKDATA     ; Load hook data
0107    11 FFD2                 LD      DE,ARTHOOK      ;
010A    01 0003                 LD      BC,0003H        ;
010D    ED B0                   LDIR                    ;
                        ;
010F    21 0121                 LD      HL,EXTEND       ; Load extend program data
0112    11 8000                 LD      DE,LOADADDR     ;
0115    01 007B                 LD      BC,EXTBOT-EXTTOP
```

```
0119     ED B0                    LDIR                      ;
011A     FB                       EI                        ; Restore interrupt status
011B     C3 EB03                  JP      WBOOT             ;
                          ;
                          ;       New hook data
                          ;
011E                      HOOKDATA:
011E     C3 8000                  JP      EXTTOP
                          ;
                          ;       New extend-program data
                          ;
0121                      EXTEND:
                          ;
                          ;       <> loading address      <>
                          ;
                                  .DEPHASE
                                  .PHASE  LOADADDR
8000                      EXTTOP:
                          ;
                          ;       ***************************************************
                          ;               EXTEND PART FOR NEW ART INTERRUPT ROUTINE
                          ;       ***************************************************
                          ;
                          ;       NOTE:
                          ;               This routine is the process for new art-interrupt.
                          ;
                          ;       <> entry parameter      <>
                          ;               NONE
                          ;       <> return parameter     <>
                          ;               NONE
                          ;       <> preserved registers  <>
                          ;               NONE
                          ;
8000                      EXTD00:
8000     21 F604                  LD      HL,RSPSTS         ;
8003     DB 15                    IN      A,(ZARTSR)        ; Check Rx-ready status
8005     47                       LD      B,A               ; Save status register
8006     E6 02                    AND     ZRXRDY            ;
8008     28 56                    JR      Z,EXTD90          ; No data in
800A     78                       LD      A,B               ;
800B     07                       RLCA                      ; Get error status
800C     E6 70                    AND     RSSFRE+RSSORE+RSSPRE
800E     57                       LD      D,A               ; (Framing, Over-run, Parity error)
800F     78                       LD      A,B               ; Get DSR line status
8010     E6 80                    AND     ZRDSR             ;
8012     5F                       LD      E,A               ;
8013     DB 18                    IN      A,(ZIOSTB)        ; Get CD line status
8015     E6 10                    AND     ZRCD              ;
8017     0F                       RRCA                      ;
8018     B2                       OR      D                 ;
8019     B3                       OR      E                 ;
801A     B6                       OR      (HL)              ;
801B     EE 98                    XOR     RSSDSR+RSSCD      ;
801D     77                       LD      (HL),A            ; Set line status
                          ;
801E     DB 14                    IN      A,(ZARTDIR)       ; Read received data
8020     21 F604                  LD      HL,RSPSTS         ; Check receive buffer
8023     CB 4E                    BIT     RSRBF,(HL)        ;
8025     28 04                    JR      Z,EXTD10          ; Buffer isn't full
8027     CB D6                    SET     RSRBO,(HL)        ; Set buffer-overflow bit
8029     18 35                    JR      EXTD90            ;
                          ;
802B                      EXTD10:
802B     21 F2BB                  LD      HL,RSINTST        ; Set interrupt flag
802E     36 01                    LD      (HL),01H          ;
8030     2A F607                  LD      HL,(RSPRBPP)      ; Store receive data into buffer
8033     CD 8062                  CALL    ISSTAX            ;
8036     23                       INC     HL                ; Next buffer address
8037     54                       LD      D,H               ;
8038     5D                       LD      E,L               ;
8039     ED 4B F612               LD      BC,(RSRBEAD)      ; Bottom of buffer?
803D     B7                       OR      A                 ;
803E     ED 42                    SBC     HL,BC             ;
8040     EB                       EX      DE,HL             ;
8041     20 03                    JR      NZ,EXTD20         ; No
8043     2A F609                  LD      HL,(RSPRBAD)      ; Change put-pointer to buffer-top
                          ;
8046                      EXTD20:
8046     22 F607                  LD      (RSPRBPP),HL      ; Set next data put-pointer
8049     ED 4B F605               LD      BC,(RSPRBGP)      ; Check buffer full
804D     B7                       OR      A                 ;
804E     ED 42                    SBC     HL,BC             ;
8050     20 05                    JR      NZ,EXTD30         ; Buffer isn't full
8052     21 F604                  LD      HL,RSPSTS         ;
8055     CB CE                    SET     RSRBF,(HL)        ; Set buffer-full bit
                          ;
8057                      EXTD30:
8057     2A F61C                  LD      HL,(RSRDL)        ; Increase receive-data counter
805A     23                       INC     HL                ;
805B     22 F61C                  LD      (RSRDL),HL        ;
805E     18 A0                    JR      EXTD00            ; Check next data being received
                          ;
                          ;
8060                      EXTD90:
```

```
8060    E1                               POP     HL              ; Neglect OS process
8061    C9                               RET                     ; End of extend process
                         ;
                         ;               ****************************************************
                         ;                       STORE DATA TO RAM AREA
                         ;               ****************************************************
                         ;
                         ;               NOTE:
                         ;                       This routine stores the data into RAM.
                         ;                       You must call this routine in DI status.
                         ;
                         ;               <> entry parameter        <>
                         ;                       HL : Setting address
                         ;                       A  : Setting data
                         ;               <> return parameter       <>
                         ;                       NONE
                         ;               <> preserved registers    <>
                         ;                       ALL
                         ;
8062                     ISSTAX:
8062    C5                               PUSH    BC              ; Save registers
8063    F5                               PUSH    AF              ;
8064    F5                               PUSH    AF              ;
8065    ED 4B F42C                       LD      BC,(RZBANKR)    ; Save current bank data
8069    3E 42                            LD      A,RAMBK         ;
806B    D3 05                            OUT     (ZBANKR),A      ;
806D    AF                               XOR     A               ;
806E    D3 22                            OUT     (ZSBNKR),A      ;
8070    F1                               POP     AF              ;
8071    77                               LD      (HL),A          ; Set data into RAM.
8072    79                               LD      A,C             ; Restore old bank
8073    D3 05                            OUT     (ZBANKR),A      ;
8075    78                               LD      A,B             ;
8076    D3 22                            OUT     (ZSBNKR),A      ;
8078    F1                               POP     AF              ; Restore registers
8079    C1                               POP     BC              ;
807A    C9                               RET                     ;
807B                     EXTBOT:

                                         END

Macros:

Symbols:
7FD2    ARTBOOK         0003    BSCD            0007    BSDSB
0008    BSFRR           0005    BSOBB           0004    BSPBR
00C1    BSRBF           0002    BSRRO           807B    EXTBOT
8030    EXTD00          802B    EXTD13          8046    EXTD20
8057    EXTD30          8060    EXTD90          0121    EXTEND
8000    EXTTOP          0112    BOOKDATA        8062    ISSTAX
8000    LOADADDR        1000    MAINBF          0042    RAMBK
F3B8    SSIRTST         F608    RSFRRAD         F6C5    RSPBBCP
F607    RSPBBF?         F604    RSPSYS          F612    RSBREAD
F61C    RSRDL           0008    RSSCD           0080    RSSDSB
0040    RSSTRR          0020    RSSOBE          0010    RSSPBB
0002    RSSBBF          0004    RSSRRO          F42C    RZBANKR
F423    RZSBNKR         0100    START           EB03    WBOOT
0014    ZARTDIR         0015    ZARTSR          0005    ZBANKR
0018    ZIOSTB          0010    ZBCD            0060    ZBDSB
0002    ZBXBOT          0022    ZSBNKR

No Fatal error(s)
```

(30) SAMPLE 30. EXTHOOK

```
                                ;       ***********************************************
                                ;                 EXT INTERRUPT HOOK SAMPLE PROGRAM
                                ;       ***********************************************
                                ;
                                ;       NOTE :
                                ;               This program shows how to use EXTHOOK.
                                ;               This program is constructed of next 3 modules.
                                ;
                                ;                       1. EXTHOOK expand module.(EXTINT)
                                ;                       2. Setup module.(EXTSETUP)
                                ;                       3. Release module.(EXTRLS)
                                ;
                                ;               If you use USER BIOS area for this program, you must
                                ;               execute UBIOS program. (see Chapter 4.6)
                                ;
                                ;
                                ;       <> Assemble condition   <>
                                ;
                                                .Z80
                                ;
                                ;       <> loading address      <>
                                ;
                                                .PHASE  100H
                                ;
                                ;       <> Constant values      <>
                                ;

FFD8                            EXTHOOK         EQU     0FFD8H          ; EXT hook address.
0DE8                            ENCRGINT        EQU     00DE8H          ; Cartridge interrupt enable
                                                                        ; module.
0DE1                            DSCRGINT        EQU     00DE1H          ; Cartridge interrupt disable
                                                                        ; module.
F000                            RETADD          EQU     0F000H          ; 'RET' instruction address.
EF4F                            RETAD           EQU     0EF4FH          ; EXT interrupt end process.
EFB9                            PSTEXT          EQU     0EFB9H          ; 'JR RETAD' address.
F0D8                            B2ICCTLB        EQU     0F0D8H          ; P19H output data save address

0018                            ZIOSTR          EQU     018H            ; P18H(for CRG & IC card int.
                                                                        ; status)
0023                            ZITSR           EQU     023H            ; P23H(for timer int. status)

00C9                            RETCMD          EQU     0C9H            ; 'RET' instruction.
0018                            JRCMD           EQU     018H            ; 'JR' instruction.

                                ;       **************************************
                                ;                 EXTHOOK extend program
                                ;       **************************************

0100                            EXTINT:
0100    ED 73 0142                      LD      (SVSEXTSP),SP           ; Save system stack pointer.
0104    31 0186                         LD      SP,EXTSP                ; Set new stack pointer.

0107    11 F000                         LD      DE,RETADD               ; Set post procedure.
010A    22 0144                         LD      (SVRETADD),HL           ; ( Cartridge int. control )

010D    21 0125                         LD      HL,RETEXT               ; Set return address.
0110    E5                              PUSH    HL                      ;

0111    DB 23                           IN      A,(ZITSR)               ; If timer int. has occured,
0113    0F                              RRCA                            ;  then call timer process.
0114    DC 0123                         CALL    C,INTTM                 ;

0117    DB 18                           IN      A,(ZIOSTR)              ; If cartridge I/F int. has
                                                                        ; occured,
0119    0F                              RRCA                            ; then call cartridge process.
011A    F5                              PUSH    AF                      ;
011B    D4 0134                         CALL    NC,INTCRG               ;
011E    F1                              POP     AF                      ;
011F    0F                              RRCA                            ;
0120    0F                              RRCA                            ; If IC card I/F int. has occured,
0121    D2 0111                         JP      NC,INTICC               ; Then call IC card process.
0124    C9                              RET                             ;


0125                            RETEXT:
0125    ED 5B 0144                      LD      DE,(SVRETADD)           ; Get cartirge I/F int. control
                                                                        ; ADDR.
0129    ED 7B 0142                      LD      SP,(SVSEXTSP)           ; restore system stack pointer.
012D    21 EF4F                         LD      HL,RETAD                ; Change return address
0130    E3                              EX      (SP),HL                 ; to post process.
0131    D5                              PUSH    DE                      ; Set catridge I/F int. controll
                                                                        ; routine address.
0132    E9                              JP      (HL)                    ; Jump to return address.
```

```
                                        ;         < For Timer interrupt >
0133                                    INTTM:
                                        ;*************************************************************
                                        ;*  If you want to expand the 1msec/8msec interrupt procedure  *
                                        ;* you can insert your own routine in this area.                *
                                        ;*************************************************************
0133      C9                                    RET                             ;


                                        ;         < For Cartridge I/F interrupt >
0134                                    INTCRC:
0134      3A F0D8                               LD      A,(RZICCTLB)            ; Check CRC int. masking status.
0137      E6 40                                 AND     01000000B               ;
0139      C0                                    RET     NZ                      ; Return if disable.

                                        ;*************************************************************
                                        ;*  If you want to expand the interrupt procedure form the     *
                                        ;* cartridge I/F, you can insert your own routine in this area.   *
                                        ;*************************************************************
013A      21 F000                               LD      HL,RETADD               ; You can select post process
013D      22 0144                               LD      (SVRETADD),HL           ; Cartridge int. control  :
0140      C9                                    RET                             ;       1. ENCRCINT..(enable)
                                                                                ;       2. DSCRCINT..(disable)
                                                                                ;       3. RETADD....(nothing)
                                        ;         < For IC card interrupt >
0141                                    INTICC:
                                        ;*************************************************************
                                        ;*  If you want to expand the interrupt procedure form the IC  *
                                        ;* card I/F, you can insert your own routine in this area.      *
                                        ;*************************************************************
0141      C9                                    RET


0142                                    SVSEXTSP:       DS      2               ; system stack pointer save area
0144                                    SVRETADD:       DS      2               ; return method

0188                                    EXTSP           EQU     $+40H           ; stack area.


                                        ;         ************************************
                                        ;                 EXTHOOK setup module
                                        ;         ************************************

0146                                    EXTSETUP:
0146      F3                                    DI                              ; *** DI ***
0147      3E C9                                 LD      A,RETCMD                ; Set RET instruction.
0149      32 EF89                               LD      (PSTEXT),A              ;
014C      21 0100                               LD      HL,EXTINT               ; Set EXT hook jump address.
014F      22 FFD9                               LD      (EXTHOOK+1),HL          ;
0152      FB                                    EI                              ; *** EI ***

0153      C9                                    RET


                                        ;         ***************************************
                                        ;                 EXTHOOK release module
                                        ;         ***************************************

0154                                    EXTRLS:
0154      F3                                    DI                              ; *** DI ***
0155      3E C3                                 LD      A,JPCMD                 ; Set JP instruction.
0157      32 EF89                               LD      (PSTEXT),A              ;
015A      21 F000                               LD      HL,RETADD               ; Reset EXT hook jump address.
015D      22 FFD9                               LD      (EXTHOOK+1),HL          ;
0160      C9                                    RET                             ;


                                                END

Macros:

Symbols:
0DE1      DSCRCINT          0DE8      ENCRCINT          FFD8      EXTHOOK
0100      EXTINT            0154      EXTRLS            0146      EXTSETUP
0188      EXTSP             0134      INTCRC            0141      INTICC
0133      INTTM             00C3      JPCMD             EF89      PSTEXT
EF4F      RETAD             F000      RETADD            00C9      RETCMD
0125      RETEXT            F0D8      RZICCTLB          0144      SVRETADD
0142      SVSEXTSP          0018      ZIOSTB            0023      ZITSR
```

(31) SAMPLE 31. BIOSHOOK

```
                        ;       ****************************************
                        ;               BIOS HOOK SAMPLE PROGRAM
                        ;       ****************************************
                        ;
                        ;       NOTE :
                        ;               This program shows how to extend the BIOS HOOK.
                        ;               This program is constructed of next 3 modules.
                        ;                       1. Extended BIOS module (EXBIOSE)
                        ;                       2. setup module (OPNUBIOS)
                        ;                       3. release module (RLSPROC)
                        ;
                        ;               These modules are  parts of UBIOS porgram (see
                        ;               Chapter 4.6).
                        ;               You must execute the UBIOS program when you use this
                        ;               program.
                        ;
                        ;
                        ;
                        ;       <> assemble condition   <>

                                        .Z80

                        ;       <> constant value       <>

0000                    PAUSE           EQU     00000H          ; dummy label.
FFE7                    BIOSHK          EQU     0FFE7H          ; BIOS HOOK address.
DC00                    UBBOTTOM        EQU     0DC00H          ; user BIOS area bottom
                                                                ; address
0010                    BUBSIZE         EQU     16              ; new user BIOS area size.
CC00                    BUBTOP          EQU     UBBOTTOM-BUBSIZE*256    ; expand procedure top.
0007                    BIOSJPTB        EQU     00007H          ; BIOS jump table top.
000F                    TARGETBIOS      EQU     0000FH          ; L'ST function.

                        ;       ************************************************
                        ;               Setup procedure for OSEXBIOS area
                        ;       ************************************************

0000'                   OPNUBIOS:
0000'   21 CC00                 LD      HL,EXBIOSE              ; Extended BIOS entry address.
0003'   22 FFE8                 LD      (BIOSHK+1),HL           ; Set new BIOS HOOK address.
0006'   C9                      RET

                        ;       **********************************
                        ;               Extend BIOS program
                        ;       **********************************

0000                    EBTOP           EQU     PAUSE

                        ;       <> loading address      <>

                                        .PHASE          BUBTOP

CC00                    EXBIOSE:
CC00    ED 73 CC2F              LD      (SAVESP),SP     ; Save current stack pointer.
CC04    31 CC71                 LD      SP,EXBIOSSP     ; Set new stack pointer.
CC07    E5                      PUSH    HL              ; save entry parameters.
CC08    D5                      PUSH    DE              ;
CC09    F5                      PUSH    AF              ;

CC0A    2A CC2F                 LD      HL,(SAVESP)     ; Get this BIOS function entry address.
CC0D    23                      INC     HL              ;
CC0E    23                      INC     HL              ;
CC0F    5E                      LD      E,(HL)          ;
CC10    23                      INC     HL              ;
CC11    56                      LD      D,(HL)          ;

CC12    2A 0008                 LD      HL,(BIOSJPTB+1) ; BIOS jump table top address.
CC15    EB                      EX      DE,HL           ;
CC16    B7                      OR      A               ;
CC17    ED 52                   SBC     HL,DE           ;
CC19    7D                      LD      A,L             ; Areg. : this BIOS function code.
CC1A    FE 0F                   CP      TARGETBIOS      ; If this BIOS is target function ,
CC1C    CA CC27                 JP      Z,EXBIOS        ; then jump.

CC1F                    EXBIOSR:
CC1F    F1                      POP     AF              ; Recover entry parameters.
CC20    D1                      POP     DE              ;
CC21    E1                      POP     HL              ;
CC22    ED 7B CC2F              LD      SP,(SAVESP)     ; Recover old stack pointer.
CC26    C9                      RET                     ;


CC27                    EXBIOS:
CC27    F1                      POP     AF              ; Recover entry parameters.
CC28    D1                      POP     DE              ;
CC29    E1                      POP     HL              ;
```

```
                                        ;*****************************************************************************
                                        ;*  You can insert your own extended-BIOS routine in this area.             *
                                        ;*****************************************************************************
 CC2A     ED 7B CC2F                           LD       SP,(SAVESP)       ; Recover old entry address.
 CC2E     C9                                   RET

 CC2F                                   SAVESP:         DS       2

 CC31                                                   DS       40H
 CC71                                   EXBIOSSP        EQU      $


 0071                                   UBSIZE          EQU      $-RUBTOP

                                                        .DEPHASE


                                        ;       ******************************
                                        ;               Release routine
                                        ;       ******************************
                                        ;
                                        ;       This routine will be mounted on HOOK AREA.
                                        ;
 0071                                   RLSTOP          EQU      PACSZ+UBSIZE
 D800                                   RRLSTOP         EQU      0D800H
 D8F0                                   HEADTOP         EQU      0D8F0H
 F000                                   RETADD          EQU      0F000H

                                        ;       <> loading address      <>

                                                        .PHASE   RRLSTOP


 D800                                   RLSPROC:

 D800     21 F000                              LD       HL,RETADD         ; Initial address.
 D803     22 FFE8                              LD       (BIOSBK+1),HL     ; BIOS HOOK initialize.

                                        ;       < Header null clear >

 D806     06 10                                LD       B,10H             ; Counter set.
 D808     21 D8F0                              LD       HL,HEADTOP        ; Header top address.

 D80B                                   REL_LOOP:
 D80B     36 C0                                LD       (HL),0C0H         ; Header null clear.
 D80D     23                                   INC      HL                ;
 D80E     10 FB                                DJNZ     REL_LOOP          ;
 D810     C9                                   RET                        ;

 0011                                   RLSSZ           EQU      $-RRLSTOP

                                                        .DEPHASE



                                                END
Macros:

Symbols:
FFE7    BIOSBK          0003    BIOSJPTB        CC27    EXBIOS
CC00    EXBIOSA         CC19    EXBIOSB         CC71    EXBIOSSP
D8F0    HEADTOP         D900    OPNCBIOS        0000    PACSZ
D80B    REL_LOOP        F000    RETADD          D800    RLSPROC
0011    RLSSZ           0071    RLSTOP          D800    RRLSTOP
4F10    RCBCLIST        CC2B    RUBTOP          CC2F    SAVESP
000F    TARGETBIOS      CC00    TOBCTTSN        0071    UBSIZE
0000    CBTOP


No Fatal error(s)
```

(32) SAMPLE 32. Extend communication protocol

```
                        ;       ************************************************************
                        ;                EXTEND COMMUNICATION SAMPLE PROGRAM
                        ;       ************************************************************
                        ;
                        ;       NOTE:
                        ;               This sample program extend communication-method.
                        ;               This uses BTAM-80 partially, so this doesn't work
                        ;               by itself. If you use this, you must modify this and
                        ;               debug it.
                        ;
                        ;       <> assemble condition    <>
                        ;
                                .Z80
                        ;
                        ;       <> loading address       <>
                        ;
                                .PHASE  0605UH
                        ;
                        ;       <> constant values       <>
                        ;
                        ;               BIOS entries
                        ;
E803                    WBOOT           EQU     0EB03H
E806                    CONST           EQU     0EB06H
E809                    CONIN           EQU     0EB09H
E80C                    CONOUT          EQU     0EB0CH

002D                    BTAM            EQU     0002DH
                        ;
                        ;               System work addresses
                        ;
F1CD                    TCAMPBM         EQU     0F1CDH
F1D4                    TOFLICNT        EQU     0F1D4H
F1D5                    T1STTIME        EQU     0F1D5H
F1D7                    T2NDTIME        EQU     0F1D7H
F1D9                    T1STFLG         EQU     0F1D9H
F1DB                    DRCVCNT         EQU     0F1DBH
F1DD                    DRCVBUF         EQU     0F1DDH

F42C                    BZBANER         EQU     0F42CH
F42D                    BZSBBER         EQU     0F42DH

F6C3                    TSPSAVE         EQU     0F6C3H
F6C5                    TCNTDT          EQU     0F6C5H
F6C7                    TEBBTIME        EQU     0F6C7H
F6C7                    TOSEESZ         EQU     TEBBTIME
F6C9                    TCAMAREA        EQU     0F6C9H
F6D5                    TCAMIF          EQU     0F6D5H

F6D8                    LOADAD          EQU     0F6D8H
F6DC                    EERADR          EQU     0F6DCH
F8AA                    FILENAME        EQU     0F8AAH
F000                    RETADR          EQU     0F000H

EFD0                    LCB             EQU     RETADR  -45     ;(22)
EFDC                    DCT             EQU     LCB     +12     ;(20)
EFE4                    TLIST           EQU     DCT     +05     ;(24)
EFF2                    NLIST           EQU     TLIST   +14     ;(40)
EFF9                    STAMP           EQU     NLIST   +06     ;(43)
                        ;
                        ;               Other constants
                        ;
1000                    MAINSP          EQU     01000H
E860                    TESBEL          EQU     0E660H

00E2                    APLBOM          EQU     0E2H
0042                    2ASBK           EQU     042H
0050                    BECSZ           EQU     050H

0005                    ESANER          EQU     05H
00E2                    ES29NEZ         EQU     0E2H

0000                    NOHANM          EQU     00000000B       ; Connection type (Leased line)
0008                    HENKAN          EQU     00001000B       ; Code conversion on
0002                    TOUKA           EQU     00000010B       ; Touka mode
0000                    TEXTS           EQU     00000000B       ; SMT TEXT format (ETB)
0001                    TEXTND          EQU     00000001B       ; SMT TEXT format (ETX)
0000                    STATUS          EQU     0               ; Status.....LCB+1
0005                    TIKCNT          EQU     5               ;
                        ;
                        ;       DCT parameters
                        ;
0005                    SYN             EQU     00000101B       ; SYN count (=5)
0030                    BAUD            EQU     00110000B       ; 9600 BPS
                        ;
                        ;       Line attribute parameters
                        ;
0000                    KAISEN          EQU     00000000B       ; Leased line
0002                    TRACE           EQU     00000010B       ; Trace mode on
0004                    STSION          EQU     00000100B       ; Secondary station
                        ;
0000                    TRX_ETB         EQU     00000000B       ; Check code (Receive ETB)
```

```
0001                    TEX_ETX     EQU     00000001B       ; Check code (Receive ETX)

0003                    ERR3        EQU     03H
0008                    ERR8        EQU     08H
                        ;
                        ;       ***************************************************
                        ;               MAIN PROGRAM
                        ;       ***************************************************
                        ;
6080    00 0F                   DB      000H,00EH           ; ROM-execute ID
6082    00 00 00                DB      000H,000H,000H      ;
                        ;
6085                    START:
6085    31 1000                 LD      SP,MAINSP           ; Set stack pointer
6088    F3                      DI                          ; Disable all interrupt
6089    21 60A8                 LD      HL,HOOKDATA         ; Load hook data
608C    11 F660                 LD      DE,TEXHK1           ;
608F    01 000C                 LD      BC,3*4              ;
6092    ED B0                   LDIR                        ;
6094    21 60B4                 LD      HL,LCBDT            ; Copy initialize data
6097    11 EFD0                 LD      DE,LCB              ; (LCB, DCT)
609A    01 002B                 LD      BC,DTBOT-LCBDT      ;
609D    ED B0                   LDIR                        ;
                        ;
609F    3E 03                   LD      A,03H               ; Set protocol to extend type
60A1    32 F1C5                 LD      (TCAMPBM),A         ;
60A4    FB                      EI                          ; Restore interrupt status
60A5    C3 EB03                 JP      WBOOT               ; End of initialize
                        ;
                        ;       New hook data
                        ;
60A8                    HOOKDATA:
60A8    E2                      DB      APLROM              ; TCAM
60A9    600F                    DW      USTCAM              ;
60AB    E2                      DB      APLROM              ; DLLRK1
60AC    6142                    DW      EXDLL               ;
60AE    E2                      DB      APLROM              ; DLRK1
60AF    6170                    DW      EXDL                ;
60B1    E2                      DB      APLROM              ; ULRK1
60B2    616F                    DW      EXUL                ;
                        ;
                        ;       ***********************************************
                        ;       *                                             *
                        ;       *       WORK AREA                             *
                        ;       *                                             *
                        ;       ***********************************************
                        ;
                        ; *** LCB ***
                        ;
60B4    00              LCBDT:  DB      0                   ; Command
60B5    00                      DB      0                   ; Status
60B6    0000                    DW      0                   ; Buffer size
60B8    0000                    DW      0                   ; Buffer address
60BA    0000                    DW      0                   ; Reserved
60BC    0000                    DW      0                   ; Reserved
60BE    0000                    DW      0                   ; Reserved
                        ;
                        ; *** DCT ***
                        ;
60C0    00                      DB      0                   ; Mode
60C1    00                      DB      0                   ; Extension
60C2    EFE4                    DW      TLIST               ; Timer table address
60C4    EFF2                    DW      NLIST               ; Retry counter table address
60C6    0000                    DW      0                   ; For BSC-1
                        ;
                        ; *** TIMER TABLE ***
                        ;
60C8    0006                    DW      6                   ; T1
60CA    0006                    DW      6                   ; T2
60CC    0032                    DW      50                  ; T3
60CE    000C                    DW      0                   ; T4
60D0    000C                    DW      0                   ; T5
60D2    0000                    DW      0                   ; T6
60D4    0035                    DW      40                  ; T7
                        ;
                        ; *** RETRY COUNTER TABLE ***
                        ;
60D6    07                      DB      7                   ; N1
60D7    07                      DB      7                   ; N2
60D8    07                      DB      7                   ; N3
60D9    07                      DB      7                   ; N4
60DA    00                      DB      0                   ; N5
60DB    00                      DB      0                   ; N6
                        ;
                        ; *** BTAM I/F PARAMETERS ***
                        ;
60DC    00                      DB      0                   ; B1 (Sending mode)
60DD    0080                    DW      RECSZ               ; B2 (User's receive size)
60DF                    DTBOT:
```

```
                                 ;
                                 ;       ****************************************************
                                 ;       NOTE:
                                 ;               Following program is extend part for another
                                 ;               protocol.
                                 ;               This program works under BTAM-60 (by using
                                 ;               BTAM calling). So if BTAM-60 doesn't install,
                                 ;               it doesn't work normally. Please refer to the
                                 ;               manual of BTAM-60 for further information.
                                 ;               In this program, BTAM-60 is modified to BIOS
                                 ;               TCAM, and for both DLL and UL/DL normally
                                 ;               working system ROM is extended by hook.
                                 ;
                                 ;               The protocol of BTAM-60 is used BSC-1 terminal.
                                 ;       ****************************************************
                                 ;
                                 ;       *****   Jump table          *****
                                 ;
60DF                             ESTCAM:
60DF        C3 917C                      JP      TCAM            ; Jump new TCAM program
                                 ;
                                 ;       BIOS TCAM Jump table
                                 ;
60E2        C3 61A5              OPEN:   JP      BOPEN           ; Connect line
60E5        C3 628A              SEND:   JP      BSEND           ; Send 1 record
60E8        C3 61D1              RCV:    JP      BRCV            ; Receive 1 record
60EB        C3 63C1              CLOSE:  JP      BCLOSE          ; Disconnect line
                                 ;
                                 ;       BTAM access macro table
                                 ;       (This table destroyed HL register)
                                 ;
0021                             LDBL    EQU     00100001B       ; LD HL,NNNN instruction
60EE        0E FF                BTINT:  LD      C,-1            ; Initialize BTAM
60F0        21                           DB      LDBL
60F1        0E 00                BTMOD:  LD      C,0             ; Generation BTAM
60F3        21                           DB      LDBL
60F4        0E 01                IOLST:  LD      C,1             ; Define IO list
60F6        21                           DB      LDBL
60F7        0E 02                LOPEN:  LD      C,2             ; Open line
60F9        21                           DB      LDBL
60FA        0E 03                LCLOSE: LD      C,3             ; Close line
60FC        21                           DB      LDBL
60FD        0E 04                RINITL: LD      C,4             ; Read initial
60FF        21                           DB      LDBL
6100        0E 05                RCNTNE: LD      C,5             ; Read continue
6102        21                           DB      LDBL
6103        0E 06                RCONCT: LD      C,6             ; Read connect
6105        21                           DB      LDBL
6106        0E 07                RRVIT:  LD      C,7             ; Read interrupt
6108        21                           DB      LDBL
6109        0E 09                RINIQ:  LD      C,9             ; Read initial inquiry
610B        21                           DB      LDBL
610C        0E 0A                WINITL: LD      C,10            ; Write initial
610E        21                           DB      LDBL
610F        0E 0B                WCNTNE: LD      C,11            ; Write continue
6111        21                           DB      LDBL
6112        0E 0C                WINRST: LD      C,12            ; Write initial and reset
6114        21                           DB      LDBL
6115        0E 0D                WCNRST: LD      C,13            ; Write connect and reset
6117        21                           DB      LDBL
6118        0E 0E                WCONCT: LD      C,14            ; Write connect
611A        21                           DB      LDBL
611B        0E 0F                WINQC:  LD      C,15            ; Write inquiry
611D        21                           DB      LDBL
611E        0E 10                WWBT:   LD      C,16            ; Write wait before transmission
6120        21                           DB      LDBL
6121        0E 11                WTTDL:  LD      C,17            ; Write TTD
6123        21                           DB      LDBL
6124        0E 12                WNAK:   LD      C,18            ; Write NAK
6126        21                           DB      LDBL
6127        0E 13                WRESET: LD      C,19            ; Write reset
6129        21                           DB      LDBL
612A        0E 14                WDISCT: LD      C,20            ; Write disconnect
612C        21                           DB      LDBL
612D        0E 17                CWATCH: LD      C,23            ; Control watch
612F        21                           DB      LDBL
6130        0E 15                CENABL: LD      C,21            ; Control enable
6132        21                           DB      LDBL
6133        0E 19                CDSABL: LD      C,25            ; Control disable
6135        21                           DB      LDBL
6136        0E 1A                BTWAIT: LD      C,26            ; Time wait
6138        21                           DB      LDBL
6139        0E 1B                BTSCOP: LD      C,27            ; Scope
613B        CD 002D                      CALL    BTAM            ; Execute BTAM
613E        47                           LD      B,A             ; Set return code (For error)
613F        4C                           LD      C,H             ;
6140        B7                           OR      A               ;
6141        C9                           RET                     ;
```

```
                        ;
                        ;       **********************************************************
                        ;               EXTEND DLL ROUTINE
                        ;       **********************************************************
                        ;
                        ;       NOTE:
                        ;               This routine extends DLL for new TCAM.
                        ;
                        ;       CAUTION:
                        ;               If TCAM receive size isn't 1 record (128 bytes),
                        ;             you must make new DLL routine.
                        ;               But it is very difficult to extend DLL for this.
                        ;
                        ;       <> entry parameter       <>
                        ;               NONE
                        ;       <> return parameter      <>
                        ;               NONE
                        ;       <> preserved registers   <>
                        ;               NONE
6142                    EXDLL:
6142    21 6159                 LD      HL,FILEBAS      ; Copy new file name
6145    11 F9AA                 LD      DE,FILENAME     ; (Always receive BAS file)
6148    01 000B                 LD      BC,11           ;
614B    ED B0                   LDIR                    ;
614D    C9                      RET                     ;
                        ;
614E                    FILECOM:
614E    54 45 58 54             DB      'TEXT    COM'
6152    20 20 20 20
6156    43 4F 4D
6159                    FILEBAS:
6159    54 45 58 54             DB      'TEXT    BAS'
615D    20 20 20 20
6161    42 41 53
6164                    FILEDAT:
6164    54 45 58 54             DB      'TEXT    DAT'
6168    20 20 20 20
616C    44 41 54
                        ;
                        ;       **********************************************************
                        ;               EXTEND UL ROUTINE
                        ;       **********************************************************
                        ;
                        ;       NOTE:
                        ;               This routine extends UL for new TCAM.
                        ;               But new TCAM routine supports system's
                        ;             require, so in this program only return
                        ;             to system.
                        ;
                        ;       <> entry parameter       <>
                        ;               NONE
                        ;       <> return parameter      <>
                        ;               NONE
                        ;       <> preserved registers   <>
                        ;               NONE
616F                    EXUL:
616F    C9                      RET
                        ;
                        ;       **********************************************************
                        ;               EXTEND DL ROUTINE
                        ;       **********************************************************
                        ;
                        ;       NOTE:
                        ;               This routine extends DL for new TCAM.
                        ;
                        ;       CAUTION:
                        ;               If TCAM receive size isn't 1 record (128 bytes),
                        ;             you must make new DL routine.
                        ;
                        ;       <> entry parameter       <>
                        ;               NONE
                        ;       <> return parameter      <>
                        ;               NONE
                        ;       <> preserved registers   <>
                        ;               NONE
6170                    EXDL:
6170    21 6164                 LD      HL,FILEDAT      ; Copy new file name
6173    11 F9AA                 LD      DE,FILENAME     ; (Always receive DAT file)
6176    01 000B                 LD      BC,11           ;
6179    ED B0                   LDIR                    ;
617B    C9                      RET                     ;
```

```
                            ;
                            ;       ***************************************************
                            ;               NEW TCAM ROUTINE
                            ;       ***************************************************
                            ;
                            ;       NOTE:
                            ;               This routine is new BIOS TCAM routine.
                            ;               When BIOS TCAM is extended, it is very important
                            ;               sending or receiving by 128 bytes unit.
                            ;
                            ;       <> entry parameter         <>
                            ;               A  : Function number
                            ;                       =01 -- Connect
                            ;                        02 -- Send
                            ;                        03 -- Receive
                            ;                        04 -- Disconnect
                            ;               Other registers are depend on each function
                            ;       <> return parameter        <>
                            ;               CY : Return information
                            ;                       =0 -- Normal return
                            ;                             Other registers depend on each
                            ;                             function.
                            ;                       =1 -- Error return
                            ;                               A reg. is error code.
                            ;                               BC reg. is more detail information.
                            ;                                B reg. = BTAM's A reg.
                            ;                                C reg. = BTAM's H reg.
                            ;
                            ;       <> preserved registers  <>
                            ;               NONE
617C                        TCAM:
617C    E5                          PUSH    HL                      ; Set return address
617D    21 6194                     LD      HL,TCAMRET              ;
6180    E3                          EX      (SP),HL                 ;
6181    3D                          DEC     A                       ; Check function code
6182    CA 60E2                     JP      Z,OPEN                  ; Connect function
6185    3D                          DEC     A                       ;
6186    CA 60E5                     JP      Z,SEND                  ; Send function
6189    3D                          DEC     A                       ;
618A    CA 60E8                     JP      Z,RCV                   ; Receive function
618D    3D                          DEC     A                       ;
618E    CA 60EB                     JP      Z,CLOSE                 ;
6191    3E 01                       LD      A,ERR1                  ;
6193    C9                          RET                             ;
                            ;
                            ;       TCAM return process
                            ;               A reg. is return status. If A reg. is equal to zero,
                            ;               TCAM is normally ended, else error return.
                            ;
6194                        TCAMRET:
6194    B7                          OR      A                       ;
6195    C8                          RET     Z                       ; Normal return
                            ;
6196    E5                          PUSH    HL                      ;
6197    D5                          PUSH    DE                      ;
6198    2A EC3                      LD      HL,(SPSAVE)             ; Neglect system error process
619B    11 F000                     LD      DE,RETADR               ;  Change return address
619E    2B                          DEC     HL                      ;
619F    72                          LD      (HL),D                  ;
61A0    2B                          DEC     HL                      ;
61A1    73                          LD      (HL),E                  ;
61A2    D1                          POP     DE                      ;
61A3    E1                          POP     HL                      ;
61A4    3E 05                       LD      A,ERR5                  ; (Communication error code)
61A6    37                          SCF                             ; Error flag
61A7    C9                          RET                             ;
                            ;
                            ;       ***************************************************
                            ;               CONNECT LINE
                            ;       ***************************************************
                            ;
                            ;       NOTE:
                            ;               Connect line.
                            ;               This doesn't send or receive file name.
                            ;
                            ;       <> entry parameter         <>
                            ;               BC : Connection type (Don't care)
                            ;                       B = 00 -- Sending mode
                            ;                           01 -- Receiving mode
                            ;               (TCAMAREA) -- File name if need
                            ;       <> return parameter        <>
                            ;               A  : Return information
                            ;                       = 00H -- Normal return
                            ;                       <>00H -- Error return
                            ;                                BC reg. means more detail information
                            ;
                            ;       <> preserved registers     <>
                            ;               NONE
61A8                        BOPEN:
61A8    CD 60EE                     CALL    BTINT                   ; Initialize BTAM
61AB    3E 35                       LD      A,SYN+BAUD              ; EXT clock, ER-DR on, 9600 bps, SYN=3
61AD    32 EFDC                     LD      (DCT),A                 ; Set DCT mode
61B0    AF                          XOR     A                       ;
61B1    32 EFDD                     LD      (DCT+1),A               ; Set BSC-1 (or BSC-2)
61B4    11 EFDC                     LD      DE,DCT                  ;
61B7    CD 60F1                     CALL    BTMOD                   ; Generation BTAM (Modify BTAM)
                            ;
```

```
61BA     16 09                     LD      D,TIMCNT           ; Timer counter
61BC     1E 06                     LD      E,TRACE+STSICN     ; Leased line, trace on, 2nd STA., 4W
61BE     CD 60F3                   CALL    LOPEN              ; Open line
61C1     C2 63C0                   JP      NZ,FATAL           ; Fatal error
                           ;
61C4     AF                        XOR     A                  ; Initialize parameters
61C5     32 F1D9                   LD      (T1STFLG),A        ; Reset first access flag
61C8     67                        LD      H,A                ;
61C9     6F                        LD      L,A                ;
61CA     22 F1D5                   LD      (T1STTIME),HL      ;
61CD     22 F1D7                   LD      (T2NDTIME),HL      ;
61D0     C9                        RET                        ;
                           ;
                           ;       ***************************************************
                           ;               RECEIVE 1 RECORD
                           ;       ***************************************************
                           ;
                           ;       NOTE:
                           ;               Receive data from host computer by 1 record.
                           ;               This uses following work area.
                           ;                T1STFLG(1)  -- Status flag
                           ;                              Bit 7 : Receive end flag
                           ;                              Bit 1 : First receive flag
                           ;                T1STTIME(2) -- Data counter in system buffer
                           ;                T2NDTIME(2) -- Current data setting address
                           ;                TUSIBSZ (2) -- Free area size in user buffer
                           ;       <> entry parameter        <>
                           ;               HL : Receive buffer top address
                           ;       <> return parameter       <>
                           ;               A  : Return information
                           ;                       = 00H -- Normal return
                           ;                              BC reg. means received data counter
                           ;                              If BC reg. equal to zero, then end of
                           ;                              receiving. (Text end)
                           ;                       <>00H -- Error return
                           ;                              BC reg. means more detail information
                           ;       <> preserved registers    <>
                           ;               NONE
                           ;
61D1                       BRCV:
61D1     3A F1D9                   LD      A,(T1STFLG)        ; Check end of receiving
61D4     E6 80                     AND     10000000B          ; (MSB is receive EOT flag)
61D6     28 08                     JR      Z,RCV000           ; Not end
                           ;
61D8     01 0000                   LD      BC,0000H           ; Set return parameter
61DB     78                        LD      A,B                ;
61DC     32 F1D9                   LD      (T1STFLG),A        ; (Clear receive flag)
61DF     C9                        RET                        ; End of receiving (BC = 0)
                           ;
61E0                       RCV000:
61E0     22 F1D7                   LD      (T2NDTIME),HL      ; Data setting address to user buffer
61E3     2A EFF9                   LD      HL,(BTASIZE+1)     ; Set user area setting size
61E6     22 F6C7                   LD      (TUSRBSZ),HL       ;
61E9     3A F1D9                   LD      A,(T1STFLG)        ; Check first time access
61EC     E6 02                     AND     00000010B          ;
61EE     20 32                     JR      NZ,RCV100          ; Not first time
                           ;
61F0     3E 09                     LD      A,ZENKAN           ; Code conversion on
61F2     32 EFD0                   LD      (LCB),A            ;
61F5     AF                        XOR     A                  ; Clear status
61F6     32 EFD1                   LD      (LCB+1),A          ;
61F9     2A F1DB                   LD      HL,(DRCVCST)       ; Set buffer size
61FC     22 EFD2                   LD      (LCB+2),HL         ;
61FF     2A F1DD                   LD      HL,(DRCVBUF)       ; Set buffer start address
6202     22 EFD4                   LD      (LCB+4),HL         ;
6205     21 0000                   LD      HL,0               ; Clear reserved area
6208     22 EFD6                   LD      (LCB+6),HL         ;
620B     11 EFD0                   LD      DE,LCB             ; Read initial
620E     CD 60FD                   CALL    RINITL             ;
6211     C2 625C                   JP      NZ,ERRCHK0         ; BTAM error
                           ;
6214     3A F1D9                   LD      A,(T1STFLG)        ; Set first receive flag
6217     F6 02                     OR      00000010B          ;
6219     32 F1D9                   LD      (T1STFLG),A        ;
621C     22 F1D5                   LD      (T1STTIME),HL      ; Set receive-data count (First)
621F     CD 6253                   CALL    NORMAL             ; Read status check
                           ;
6222                       RCV100:
6222     ED 4B F1D5                LD      BC,(T1STTIME)      ; Receive data counter in system buffer
6226     78                        LD      A,B                ; Check data counter
6227     B1                        OR      C                  ;
6228     28 35                     JR      Z,RCV200           ; No data received
                           ;
                           ;       Copy received data to user's area
                           ;
622A     ED 5B F1DD                LD      DE,(DRCVBUF)       ; Received data address
622E     2A F1D7                   LD      HL,(T2NDTIME)      ; Data setting address
                           ;
6231                       RCV110:
6231     1A                        LD      A,(DE)             ; Received data
6232     CD 6309                   CALL    ISSTAX             ; Set it to RAM area
6235     23                        INC     HL                 ;
6236     13                        INC     DE                 ;
6237     0B                        DEC     BC                 ;
6238     E5                        PUSH    HL                 ;
```

```
6239    2A F6C7                 LD      HL,(TUSERSZ)    ; Check user buffer size
623C    2B                      DEC     HL              ;
623D    22 F6C7                 LD      (TUSERSZ),HL    ;
6240    7C                      LD      A,H             ; User's buffer full?
6241    B5                      OR      L               ;
6242    E1                      POP     HL              ;
6243    28 09                   JR      Z,RCV150        ; Yes
6245    78                      LD      A,B             ; Data remain in buffer?
6246    B1                      OR      C               ;
6247    20 E8                   JR      NZ,RCV110       ; Yes
                        ;
6249                    RCV120:
6249    22 F1D7                 LD      (T2NDTIME),HL   ; Save data setting address in user buffer

624C    18 14                   JR      RCV200          ; Receive next TEXT
                        ;
                        ;       Move received data in receive-buffer
                        ;
624E                    RCV150:
624E    ED 43 F1D5              LD      (T1STTIME),BC   ; Save remain data count in buffer
6252    78                      LD      A,B             ; Any data remained?
6253    B1                      OR      C               ;
6254    28 06                   JR      Z,RCV160        ; No
6256    2A F1DD                 LD      HL,(DRCVBUF)    ; Receive-buffer top address
6259    EB                      EX      DE,HL           ; HL reg. is remain-data top address
625A    ED B0                   LDIR                    ; Move!
                        ;
625C                    RCV160:
625C    ED 4B EFF9              LD      BC,(BTAMIF+1)   ; Return parameters (Received data size)
6260    AF                      XOR     A               ; (Normal return)
6261    C9                      RET                     ;
                        ;
                        ;       Receive next TEXT
                        ;
6262                    RCV200:
6262    AF                      XOR     A               ; Status clear
6263    32 EFD1                 LD      (LCB+1),A       ;
6266    2A F1DB                 LD      HL,(DRCVCNT)    ; Set new receive buffer size
6269    22 EFD2                 LD      (LCB+2),HL      ;
626C    2A F1DD                 LD      HL,(DRCVBUF)    ; Set new receive buffer address
626F    22 EFD4                 LD      (LCB+4),HL      ;
6272    11 EFD0                 LD      DE,LCB          ; Set LCB address
6275    CD 6100                 CALL    RCNTXE          ;
6278    C2 6298                 JP      NZ,ERRCHK1      ; Receive error
                        ;
627B    22 F1D5                 LD      (T1STTIME),HL   ; Data count in buffer
627E    CD 6283                 CALL    NORMAL          ;
6281    18 BF                   JR      RCV100          ;
                        ;
                        ;       Check receive data status
                        ;
6283                    NORMAL:
6283    3A EFD1                 LD      A,(LCB+1)       ; Read status
6286    E6 01                   AND     TEX_ETX         ; STX...ETX ?
6288    32 F6D5                 LD      (TCAMIF),A      ; Yes (Setting 1)
628B    C9                      RET                     ;
                        ;
                        ;       Error check for read initial
                        ;
628C                    ERRCHK0:
628C    78                      LD      A,B             ; Error code
628D    FE FF                   CP      -1              ; Fatal error?
628F    CA 63C0                 JP      Z,FATAL         ; Yes
6292    FE 01                   CP      1               ; Exception error?
6294    C2 638C                 JP      NZ,LINERR       ; No, line error
6297    C9                      RET                     ; Error return
                        ;
                        ;
                        ;       Error check for read continue
                        ;
6298                    ERRCHK1:
6298    FE 01                   CP      1               ; Exception error?
629A    20 F0                   JR      NZ,ERRCHK0      ; No
629C    79                      LD      A,C             ;
629D    B7                      OR      A               ; Receive EOT?
629E    20 EC                   JR      NZ,ERRCHK0      ; No
                        ;
                        ;       Receive EOT. (Free data-line)
                        ;
62A0    3A F6D5                 LD      A,(TCAMIF)      ; Already received ETX?
62A3    3D                      DEC     A               ;
62A4    C0                      RET     NZ              ; No (Illegal EOT)
                        ;
62A5    3A F1D9                 LD      A,(T1STFLG)     ; Set end of text flag
62A8    F6 80                   OR      10000000B       ;
62AA    32 F1D9                 LD      (T1STFLG),A     ;
62AD    2A EFF9                 LD      HL,(BTAMIF+1)   ; Calculate received data size
62B0    ED 5B F6C7              LD      DE,(TUSERSZ)    ;  (User buffer size) - (Free size)
62B4    ED 52                   SBC     HL,DE           ;
62B6    44                      LD      B,H             ; Set received data size
62B7    4D                      LD      C,L             ;
62B8    AF                      XOR     A               ; Normal return
62B9    C9                      RET                     ;
                        ;
```

```
                              ;         ************************************************
                              ;                 SEND 1 RECORD
                              ;         ************************************************
                              ;
                              ;         NOTE:
                              ;                 Send data to host computer by 1 record.
                              ;                 This uses following work area.
                              ;                  T1STTIME(2) -- Data pointer in user buffer
                              ;                  T2NDTIME(2) -- Data counter in user buffer
                              ;                  TCNTDT  (1) -- NACK counter
                              ;         <> entry parameter          <>
                              ;                 HL : Receive buffer top address
                              ;                 BC : Sending data count (Bytes)
                              ;         <> return parameter         <>
                              ;                 A  : Return information
                              ;                       = 00H -- Normal return
                              ;                                BC reg. means received data counter
                              ;                      <>00H -- Error return
                              ;                                BC reg. means more detail information
                              ;         <> preserved registers    <>
                              ;                 NONE
62BA                          BSEND:
62BA    22 F1D5                         LD      (T1STTIME),HL   ; Data pointer in user buffer
62BD    ED 43 F1D7                      LD      (T2NDTIME),BC   ; Data counter in user buffer
                              ;
                              ;         Copy data from RAM to system area
                              ;
62C1                          SND100:
62C1    ED 5B F1DD                      LD      DE,(DRCVBUF)    ; Sending data address in buffer
62C5    2A F1D5                         LD      HL,(T1STTIME)   ; Data pointer in user buffer
62C8    ED 4B F1DB                      LD      BC,(DRCVCNT)    ;
                              ;
62CC                          SND110:
62CC    CD 63F2                         CALL    ISLDAT          ; Read data
62CF    12                              LD      (DE),A          ; Set it to system buffer
62D0    23                              INC     HL              ;
62D1    13                              INC     DE              ;
62D2    0B                              DEC     BC              ;
62D3    E5                              PUSH    HL              ;
62D4    2A F1D7                         LD      HL,(T2NDTIME)   ; Decrement data counter
62D7    2B                              DEC     HL              ;
62D8    22 F1D7                         LD      (T2NDTIME),HL   ;
62DB    7C                              LD      A,H             ; All data buffering?
62DC    B5                              OR      L               ;
62DD    E1                              POP     HL              ;
62DE    28 04                           JR      Z,SND120        ; Yes
62E0    78                              LD      A,B             ;
62E1    B1                              OR      C               ; System buffer full?
62E2    20 E8                           JR      NZ,SND110       ; No
                              ;
62E4                          SND120:
62E4    22 F1D5                         LD      (T1STTIME),HL   ; Save buffer pointer
62E7    2A F1DB                         LD      HL,(DRCVCNT)    ; Calculate data length in buffer
62EA    ED 42                           SBC     HL,BC           ;  (Buffer size - Buffer space)
                              ;
                              ;         Set LCB
                              ;
62EC    3E 0A                           LD      A,TEXTB+HENKAN+TOUKA
62EE    32 EFD0                         LD      (LCB),A         ; STX...ETX , DENSAN flag on
62F1    AF                              XOR     A               ;
62F2    32 EFD1                         LD      (LCB+1),A       ; Clear status
62F5    22 EFD2                         LD      (LCB+2),HL      ; Set data length
62F8    2A F1DD                         LD      HL,(DRCVBUF)    ;
62FB    22 EFD4                         LD      (LCB+4),HL      ; Set buffer address
62FE    01 0000                         LD      BC,0000H        ;
6301    ED 43 EFD6                      LD      (LCB+6),BC      ; Clear reserve area
                              ;
6305    3A F1D9                         LD      A,(T1STFLG)     ; Check first time access
6308    E6 01                           AND     00000001B       ;
630A    20 17                           JR      NZ,SND200       ; Not first time
                              ;
630C                          SND150:
630C    AF                              XOR     A               ; Initialize NACK counter
630D    32 F6C5                         LD      (TCNTDT),A      ;
6310                          SND160:
6310    11 EFD0                         LD      DE,LCB          ; Write initial
6313    CD 610C                         CALL    WINITL          ;
6316    C2 6362                         JP      NZ,ERRCHK2      ; Write error
6319    3A F1D9                         LD      A,(T1STFLG)     ; Set first write flag
631C    F6 01                           OR      00000001B       ;
631E    32 F1D9                         LD      (T1STFLG),A     ;
6321    18 08                           JR      SND300          ; Go to check of end
                              ;
                              ;         Write continue
                              ;
6323                          SND200:
6323    11 EFD0                         LD      DE,LCB          ; Set LCB
6326    CD 610F                         CALL    WCNTNE          ; Write continue
6329    C2 6362                         JP      NZ,ERRCHK2      ; Write error
                              ;
                              ;         End of sending check
                              ;
632C                          SND300:
632C    2A F1D7                         LD      HL,(T2NDTIME)   ; Check data count in buffer.
```

```
632F    7C                      LD      A,H             ;
6330    B5                      OR      L               ; Any data remain?
6331    20 9E                   JR      NZ,SND100       ; Yes
                        ;
                        ;       Write end
                        ;
6333                    SND400:
6333    3A EFF5                 LD      A,(STAMIF)      ; Check sending mode
6336    E6 01                   AND     00000001B       ;  Continue mode?
6338    C8                      RET     Z               ; Yes
                        ;
6339                    SND410:
6339    3E 0B                   LD      A,TEXTMD+HENKAN+TOUKA
633B    32 EFD0                 LD      (LCB),A         ; STX...ETX, HENKAN on
633E    AF                      XOR     A               ; Clear status
633F    32 EFD1                 LD      (LCB+1),A       ;
6342    21 0000                 LD      HL,0000H        ; Set data length to zero
6345    22 EFD2                 LD      (LCB+2),HL      ;
6348    11 EFD0                 LD      DE,LCB          ; Set LCB
634B    CD 610F                 CALL    WCNTNE          ; Write continue (Send STX...ETX)
634E    C2 6362                 JP      NZ,ERRCHK2      ; Write error
                        ;
6351    3A F1D9                 LD      A,(T1STFLG)     ; Reset write-first flag
6354    E6 FE                   AND     11111110B       ;
6356    32 F1D9                 LD      (T1STFLG),A     ;
6359    CD 63CE                 CALL    WRESETX         ; Write reset
635C    C2 63C0                 JP      NZ,FATAL        ;
635F    3E 00                   LD      A,00H           ; Normal return
6361    C9                      RET                     ;
                        ;
                        ;       Error check for writing initial/continue
                        ;
6362                    ERRCHK2:
6362    FE FF                   CP      -1              ; Fatal error?
6364    CA 63C0                 JP      Z,FATAL         ; Yes
6367    FE 01                   CP      1               ; Exception error?
6369    C2 63BC                 JP      NZ,LINERR       ; No
                        ;
636C    79                      LD      A,C             ;
636D    B7                      OR      A               ; Receive EOT?
636E    C8                      RET     Z               ; Yes
636F    FE 01                   CP      1               ; Receive DISC?
6371    C8                      RET     Z               ; Yes
6372    FE 02                   CP      2               ; Receive RVI?
6374    28 11                   JR      Z,RVI           ; Yes
6376    FE 03                   CP      3               ; Receive WACK?
6378    28 21                   JR      Z,WACK          ; Yes
637A    FE 04                   CP      4               ; Contension?
637C    28 5F                   JR      Z,SND150        ; Yes (Only WINIT)
637E    FE 05                   CP      5               ; ENQ --> Receive WACK ?
6380    28 2D                   JR      Z,WACK00        ; Yes
6382    FE 06                   CP      6               ; ENQ --> Receive EOT ?
6384    28 56                   JR      Z,SND150        ; Yes (Check end)
6386    C9                      RET                     ;
                        ;
                        ;       Receive RVI data
                        ;
6387                    RVI:
6387    3A F1D9                 LD      A,(T1STFLG)     ; Set write-first flag
638A    F6 01                   OR      00000001B       ;
638C    32 F1D9                 LD      (T1STFLG),A     ;
638F    2A F1D7                 LD      HL,(T2NDTIME)   ; Check data count in buffer.
6392    7C                      LD      A,H             ;
6393    B5                      OR      L               ; Finished?
6394    C2 62C1                 JP      NZ,SND100       ; No (Send next data)
                        ;
6397    CD 63CE                 CALL    WRESETX         ; Write reset
639A    C9                      RET                     ;
                        ;
                        ;       Receive WACK data
                        ;
639B                    WACK:
639B    3A F1D9                 LD      A,(T1STFLG)     ; Set write-first flag
639E    F6 01                   OR      00000001B       ;
63A0    32 F1D9                 LD      (T1STFLG),A     ;
63A3    2A F1D7                 LD      HL,(T2NDTIME)   ; Check data count in buffer.
63A6    CD 6119                 CALL    WINQU           ; Write inquire
63A9    C2 6362                 JP      NZ,ERRCHK2      ; Send error
63AC    C3 632C                 JP      SND300          ; End check
                        ;
63AF                    WACK00:
63AF    3A F6C5                 LD      A,(TCNTDT)      ; Check WACK counter
63B2    3C                      INC     A               ;
63B3    32 F6C5                 LD      (TCNTDT),A      ; (WACK count up)
63B6    FE 11                   CP      17              ; WACK count > 16 ?
63B8    DA 6310                 JP      C,SND160        ; Write initial
63BB    C9                      RET                     ;
```

```
                        ;
                        ;       Line error
                        ;               C reg. means error status.
                        ;                       =0 -- Data error          (Write only)
                        ;                        1 -- ACK0/ACK1 error     (Write only)
                        ;                        2 -- Time out error
                        ;                        3 -- Data check error    (Read only)
                        ;                        4 -- ID error
                        ;
63BC                    LINERR:
63BC    CD 63CE                 CALL    WRESETX         ; Write reset
63BF    C9                      RET                     ;
                        ;
                        ;       Fatal error
                        ;               BTAM closes line automatically.
                        ;               C reg. means error status.
                        ;                       =-1 -- Stop by external device
                        ;                         0 -- Illegal macro
                        ;                         1 -- BSC check (Modem not ready)
                        ;                         2 -- Length overrun
                        ;                         3 -- Receive overrun
                        ;                         4 -- Carrier detect
                        ;                         5 -- Not used
                        ;                         6 -- Not opened
                        ;                         7 -- Double open
                        ;
63C0                    FATAL:
63C0    C9                      RET                     ; No operation here
                        ;
                        ;       ****************************************************
                        ;               CLOSE LINE
                        ;       ****************************************************
                        ;
                        ;       NOTE:
                        ;               This routine closes line.
                        ;               If BTAMIF's bit 0 is zero and sending data mode,
                        ;               send STX...ETX block.
                        ;       <> entry parameter      <>
                        ;               NONE
                        ;       <> return parameter     <>
                        ;               NONE
                        ;       <> preserved registers  <>
                        ;               NONE
                        ;
63C1                    BCLOSE:
63C1    3A EFF9                 LD      A,(BTAMIF)      ; Check sending mode
63C4    E6 01                   AND     00000001B       ;  Continue mode?
63C6    C4 5339                 CALL    NZ,SND410       ; Yes (Send STX...ETX)
63C9    CD 60FA                 CALL    LCLOSE          ; Close line.
63CC    AF                      XOR     A               ; Normal return
63CD    C9                      RET                     ;
                        ;
                        ;       Write reset command
                        ;               Only BC reg. is preserved.
                        ;
63CE                    WRESETX:
63CE    C5                      PUSH    BC              ; Save register
63CF    CD 6127                 CALL    WRESET          ; Write reset
63D2    20 03                   JR      NZ,WRES10       ; Fatal error
63D4    C1                      POP     BC              ; Restore register
63D5    79                      LD      A,C             ; Restore error code
63D6    C9                      RET                     ;
63D7                    WRES10:
63D7    D1                      POP     DE              ; Dummy POP
63D8    C9                      RET                     ;
                        ;
                        ;       ****************************************************
                        ;               STORE DATA TO RAM AREA
                        ;       ****************************************************
                        ;
                        ;       NOTE:
                        ;               This routine stores the data into RAM.
                        ;               You must call this routine in DI status.
                        ;       <> entry parameter      <>
                        ;               HL : Setting Address
                        ;               E  : Setting data
                        ;       <> return parameter     <>
                        ;               NONE
                        ;       <> preserved registers  <>
                        ;               ALL
                        ;
63D9                    ISSTAK:
63D9    C5                      PUSH    BC              ; Save registers
63DA    F5                      PUSH    AF              ;
63DB    F5                      PUSH    AF              ;
63DC    ED 4B F42C              LD      BC,(RZBANKR)    ; Save current bank data
63E0    3E 42                   LD      A,RAMBK         ;
63E2    D3 05                   OUT     (ZBANKR),A      ;
63E4    AF                      XOR     A               ;
63E5    D3 22                   OUT     (ZSBBNKR),A     ;
63E7    F1                      POP     AF              ;
63E8    77                      LD      (HL),A          ; Set data into RAM.
63E9    79                      LD      A,C             ; Restore old bank
63EA    D3 05                   OUT     (ZBANKR),A      ;
63EC    78                      LD      A,B             ;
63ED    D3 22                   OUT     (ZSBBNKR),A     ;
63EF    F1                      POP     AF              ; Restore registers
```

```
63F0        C1                              POP     BC                          ;
63F1        C9                              RET                                 ;
                                ;
                                ;           ****************************************************
                                ;                   GET DATA FROM RAM AREA
                                ;           ****************************************************
                                ;
                                ;           NOTE:
                                ;                   This routine gets the data from RAM.
                                ;                   You must call this routine in DI status.
                                ;           <> entry parameter      <>
                                ;                   HL : Getting address
                                ;           <> return parameter     <>
                                ;                   A  : Gotten data
                                ;           <> preserved registers   <>
                                ;                   BC,DE,HL
63F2                            ISLDAX:
63F2        C5                              PUSH    BC              ; Save register
63F3        ED 4B F42C                      LD      BC,(RZBANKR)    ; Save current bank data
63F7        3E 42                           LD      A,BANBK         ;
63F9        D3 05                           OUT     (ZBANKR),A      ;
63FB        3E 00                           LD      A,00H           ;
63FD        D3 22                           OUT     (ZSBBNKR),A     ;
63FF        7E                              LD      A,(HL)          ; Get data from RAM.
6400        F5                              PUSH    AF              ;
6401        79                              LD      A,C             ; Restore old bank
6402        D3 05                           OUT     (ZBANKR),A      ;
6404        78                              LD      A,B             ;
6405        D3 22                           OUT     (ZSBBNKR),A     ;
6407        F1                              POP     AF              ; Restore gotten data
6408        C1                              POP     BC              ; Restore register
6409        C9                              RET                     ;

                                            END
```

Macros:

Symbols:

0082	APLROM	0030	BAUD	63C1	BCLOSE
61A8	BOPEN	6101	BRCV	62BA	BSEND
0020	BTAM	EFF9	BTAMIF	6022	BTIST
60F1	BTMOD	6130	BTSCOP	6130	BTWAIT
6133	CDSABL	6130	CENABL	6028	CLOSE
EB09	CONIN	EB0C	CONOUT	EB06	CONST
612D	CWATCH	EFDC	DCT	F100	DBCVBUF
F138	DBCVCST	600F	DTBOT	0001	ERR1
0008	ERR3	F60C	ERRADR	625C	ERRCHK0
6299	ERRCHK1	6362	ERRCHK2	6170	EXDL
6102	EXDL2	616F	EXUL	63C0	FATAL
6159	FILEBAS	614E	FILECOM	6164	FILEDAT
FBA4	FILENAME	0006	HENKAN	60A6	HOOKDATA
60F4	IOLST	63F2	ISLDAX	63D9	ISSTAX
0000	KAISEN	0000	KOKANM	EFD0	LCB
6CB4	LCBOT	60FA	LCLOSE	0021	LDBL
63BC	LINPRE	F6D9	LOADAD	60F7	LOPEN
1000	MAINSP	EFF2	MLIST	6253	NORMAL
6022	OPEN	0042	BANBK	6100	RCNTNE
6103	RCONCT	6029	RCV	6120	RCV000
6222	RCV100	6231	RCV110	6249	RCV120
624E	RCV1[illegible]0	625C	RCV160	6282	RCV200
0080	RECSZ	F0C0	RETADR	6109	RINIO
60FD	RINITL	6106	RRVIT	6397	RVI
F42C	RZBANKR	F42D	RZSBBKR	6025	SEND
62C1	SND100	62CC	SND110	62E4	SND120
630C	SND150	6310	SND160	6322	SND200
632C	SND300	6333	SND400	6338	SND410
6055	START	0000	STATUS	0004	STSION
00C5	SYN	F2D6	TISTFLC	F1D9	TISTTIME
F1D7	T2NDTIME	617C	TCAN	F6C9	TCAMAREA
F6D3	TCAMIF	F1CD	TCAMPRM	6164	TCAMEET
[illegible]	TCNTDT	F1D4	TDPLTCNT	F6C7	TERRTIME
F66[illegible]	TEXBKL	0000	TEXTB	0001	TEXTMD
000C	TEX_STR	0001	TEX_ETX	00C5	TIBCXT
[illegible]	[illegible]	0032	TOUKA	00C2	TRACE
FEC3	TSPSAVE	F8C7	TUSEBSZ	600F	USTCAN
6398	WACK	63AF	WACKOO	EB03	WBOOT
6115	WCNRST	[illegible]	WCNTNE	6115	WCONCT
[illegible]	WDISCT	610C	WINITL	[illegible]	WINQU
[illegible]	WINRST	6124	WNAK	63D7	WRES10
[illegible]	WRESET	63C[illegible]	WRESETX	6121	WTTDL
[illegible]	WWAIT	0005	ZBANKR	0022	ZSBBNKR

No fatal error(s)

[33] SAMPLE 33. Extend IC card protocol

```
                        ;       ****************************************************
                        ;               SAMPLE PROGRAM FOR EXTENDED IC-CARD
                        ;       ****************************************************
                        ;
                        ;       NOTE:   This sample program extends IC-card executing
                        ;               by another protocol. But, in this program neglected
                        ;               the part of sending or receiving commands.
                        ;                This sample program is divided into two parts.
                        ;                One is initializing system area. Another is extended
                        ;               part for another protocol. Extended part executes in
                        ;               this program must be located in an application ROM
                        ;               (256 Kbit).Please write this program into P-ROM.
                        ;
                        ;       <> assemble condition   <>
                        ;
                                .Z80
                        ;
                        ;       <> loading address      <>
                        ;
                                .PHASE  06050H
                        ;
                        ;       <> constant values      <>
                        ;
                        ;               BIOS entries
                        ;
EB03                    WBOOT           EQU     0EB03H
EB09                    CONIN           EQU     0EB09H
EB0C                    CONOUT          EQU     0EB0CH
                        ;
                        ;               System work addresses
                        ;
F06A                    LOGADR          EQU     0F06AH
F0B5                    YPOFST          EQU     0F0B5H

F0D6                    RZCTLR1         EQU     0F0D6H
F0D9                    RZARTMR         EQU     0F0D9H
F0DA                    RZARTCR         EQU     0F0DAH
F0DB                    RZICCTLR        EQU     0F0DBH
F0DC                    RZSWR           EQU     0F0DCH
F0DD                    RZIOCTLR        EQU     0F0DDH
F0DE                    RZCTLR3         EQU     0F0DEH
F428                    RZIER           EQU     0F428H

F026                    ICCPWTM         EQU     0F026H
F196                    SRTABL          EQU     0F196H
F197                    SYSARTMR        EQU     0F197H
F1A8                    EPIBRTO         EQU     0F1A8H
F1AE                    ICBSTPNT        EQU     0F1AEH
F1E5                    ICBECSZ         EQU     0F1E5H
F1E6                    ICHOMOT1        EQU     0F1E6H
F1E9                    ICTSDT          EQU     0F1E9H
F202                    ICCPWCNT        EQU     0F202H
F203                    ICCPWFLG        EQU     0F203H
F241                    SBMODE          EQU     0F241H

F4C2                    SEKTBL          EQU     0F4C2H
F4C4                    SEKSEC          EQU     0F4C4H
F4D5                    DMAADR          EQU     0F4D5H

F65C                    ICMAXREC        EQU     0F65CH
F65E                    QTICCARD        EQU     0F65EH

F660                    ICONFLG         EQU     0F660H
F661                    ICCWFLG         EQU     0F661H
F662                    ICCSEDV         EQU     0F662H
F663                    ICTSDT          EQU     0F663H

F177                    DPB2            EQU     0F177H
F65?                    TPICCARD        EQU     0F65?H
                        ;
                        ;               Other constants
                        ;
1000                    MAINSP          EQU     01000H
FF6F                    HOOKADDR        EQU     0FF6FH
0050                    WAITTIME        EQU     00050H

0032                    TSBYTE          EQU     032H
0084                    MODE            EQU     084H
0002                    APLROM          EQU     002H
000A                    ANSCNT          EQU     00AH
0020                    RECSIZE         EQU     020H

0001                    RDCMD           EQU     01H
0002                    WTCMD           EQU     02H

0000                    ZCTLR1          EQU     00H
0004                    ZIER            EQU     04H
0014                    ZARTDOR         EQU     14H
0014                    ZARTDIR         EQU     14H
0015                    ZARTMR          EQU     15H
0015                    ZARTSR          EQU     15H
```

```
0016                    ZIOSTB          EQU     16H
0016                    ZARTCB          EQU     16H
0017                    ZICCTLR         EQU     17H
0018                    ZSWR            EQU     18H
0019                    ZIOCTLR         EQU     19H
0023                    ZCTLR3          EQU     23H

0006                    ERR6            EQU     06H
0007                    ERR7            EQU     07H
0009                    ERR9            EQU     09H
000B                    ERR11           EQU     0BH
                        ;
                        ;
                        ;       ***************************************************
                        ;               MAIN PROGRAM
                        ;       ***************************************************
                        ;
                        ;       Next two statements are header for ROM-execute program.
                        ;
6080    DD DE                   DB      0DDH,0DEH       ; Header for ROM-execute
6082    00 00 00                DB      000H,000H,000H  ;

6085                    START:
6085    31 1000                 LD      SP,MAINSP       ; Set stack pointer
                        ;
6088    21 60BD                 LD      HL,HOOKDATA     ; Change HOOK jump address
608B    11 FF6F                 LD      DE,HOOKADDR     ; Jump to 0000H of application ROM
608E    01 0018                 LD      BC,3*8          ;
6091    F3                      DI                      ; Disable all interrupt
6092    ED B0                   LDIR                    ;
6094    FB                      EI                      ;
                        ;
6095    3E F8                   LD      A,11111000B     ; Set hook execute address
6097    32 F126                 LD      (ICSOKDT1),A    ;
                        ;
                        ;       The following parameters are depend on each IC-card.
                        ;       So if you need, please set them as you like.
                        ;
609A    3E 3B                   LD      A,TSBYTE        ; Set TS byte comparing data
609C    32 F129                 LD      (ICTSDT),A      ;
609F    3E 84                   LD      A,90DE          ; Set parity, data length, stop bit.
60A1    32 F187                 LD      (SYSARTMR),A    ;
60A4    3E 20                   LD      A,RECSIZE       ; Set 1 record physical size
60A6    32 F125                 LD      (ICRECSZ),A     ;
60A9    3E 10                   LD      A,16            ; Set IC-card size
60AB    32 F652                 LD      (QTICCARD),A    ;
60AE    21 0078                 LD      HL,16*1024/136  ; Set maximum record number
60B1    22 F65C                 LD      (ICMAXREC),HL   ;
60B4    21 3000                 LD      HL,3000H        ; Set new time-out time
60B7    22 F1AB                 LD      (EPIBBTO),HL    ;

60BA    C3 EB03                 JP      WBOOT           ;

                        ;       ***************************************************
                        ;               HOOK JUMP DATA
                        ;       ***************************************************
                        ;
60BD                    HOOKDATA:
60BD    02                      DB      APLROM          ; For HOOK No. 1
60BE    60D5                    DW      UBDTTOP+00H     ;
60C0    02                      DB      APLROM          ; For HOOK No. 2
60C1    60D8                    DW      UBDTTOP+03H     ;
60C3    02                      DB      APLROM          ; For HOOK No. 3
60C4    60DB                    DW      UBDTTOP+06H     ;
60C6    02                      DB      APLROM          ; For HOOK No. 4
60C7    60DE                    DW      UBDTTOP+09H     ;
60C9    02                      DB      APLROM          ; For HOOK No. 5
60CA    60E1                    DW      UBDTTOP+0CH     ;
60CC    02                      DB      APLROM          ; For HOOK No. 6
60CD    60E4                    DW      UBDTTOP+0FH     ;
60CF    02                      DB      APLROM          ; For HOOK No. 7
60D0    60E7                    DW      UBDTTOP+12H     ;
60D2    02                      DB      APLROM          ; For HOOK No. 8
60D3    60EA                    DW      UBDTTOP+15H     ;
                        ;
                        ;       ***************************************************
                        ;       ***************************************************
                        ;
                        ;       You make another program for extended IC-card protocol.
                        ;       The following program is for it.
                        ;
                        ;       ***************************************************
                        ;       ***************************************************
                        ;
60D5                    UBDATA:
                        ;
                        ;       ***************************************************
                        ;               HOOK JUMP TABLE FOR EXTEND IC-CARD
                        ;       ***************************************************
                        ;
                        ;       NOTE:
                        ;               Jump to this part with all interrupt disable.
                        ;
```

```
60D5                        UBOTTOP:
60D5    C3 60ED                     JP      UICDMK1         ; Reset IC-card
60D8    C3 617F                     JP      UICDMK2         ;
60DB    C3 617F                     JP      UICDMK3         ;
60DE    C3 617F                     JP      UICDMK4         ;
60E1    C3 6169                     JP      UICDMK5         ; Read 1 record
60E4    C3 6191                     JP      UICDMK6         ; Write 1 record
60E7    C3 61D6                     JP      UICDMK7         ; Mount check of IC-card
60EA    C3 6209                     JP      UICDMK8         ; Format IC-card

                            ;
                            ;       ****************************************************
                            ;               RESET IC-CARD AND RECEIVE RESPONCE
                            ;       ****************************************************
                            ;
                            ;       <> entry parameter      <>
                            ;               NONE
                            ;       <> return parameter     <>
                            ;               CY : Return information
                            ;                       =0 : Normally opened
                            ;                       =1 : Open error
                            ;                            A  : Error code
                            ;                                 =06 -- Time out error
                            ;                                 =07 -- Protocol error
                            ;                                 =0A -- Cover is opened
                            ;       <> preserved registers  <>
                            ;               NONE
                            ;
60ED                        UICDMK1:
60ED    F3                          DI                      ;
60EE    DB 16                       IN      A,(ZIOSTR)      ; Check IC-card cover status.
60F0    E6 04                       AND     00000100B       ; Cover is opened?
60F2    37                          SCF                     ;
60F3    3E 0A                       LD      A,ERR11         ; Error return code.
60F5    CA 617F                     JP      Z,HOOKRET       ; Yes
                            ;
60F8    3A FDD8                     LD      A,(RZICCTLR)    ; Enable Cover interrupt
60FB    F6 26                       OR      00100110B       ; Set to reception mode
60FD    D3 17                       OUT     (ZICCTLR),A     ; Supply Vcc to IC-card
60FF    F6 27                       OR      00100111B       ;
6101    D3 17                       OUT     (ZICCTLR),A     ; Supply clock to IC-card
6103    32 FDD8                     LD      (RZICCTLR),A    ;
                            ;
6106    01 0050                     LD      BC,WAITTIME     ; Wait until Vcc and clock stationary
6109    CD 62C6                     CALL    WAIT            ;
                            ;
610C    3A FDDA                     LD      A,(RZARTCR)     ; Reset error of serial line
610F    F6 10                       OR      00010000B       ;
6111    D3 15                       OUT     (ZARTCR),A      ;
6113    3A FDDD                     LD      A,(RZIOCTLR)    ; Set RST to low level
6116    E6 FB                       AND     11111011B       ;
6118    32 FDDD                     LD      (RZIOCTLR),A    ;
611B    D3 16                       OUT     (ZIOCTLR),A     ;
                            ;
611D    DB 16                       IN      A,(ZIOSTR)      ; Check IC-card cover status.
611F    E6 04                       AND     00000100B       ; Cover is opened?
6121    37                          SCF                     ;
6122    3E 0A                       LD      A,ERR11         ; Error return code
6124    28 59                       JR      Z,HOOKRET       ; Yes
6126    3E 01                       LD      A,01H           ; Set IC-card power-on flag
6128    32 F660                     LD      (ICONFLG),A     ;
                            ;
                            ;       Receive answer-to-reset using the subroutine 180COM.
                            ;       If no need of receiving answer-to-reset, please insert
                            ;      Next two statements.
                            ;
612B    B7                          OR      A               ; No need of receiving answer-to-reset
612C    18 51                       JR      HCONRET         ; End of hook
                            ;
612E    2A F1AE                     LD      HL,(ICRSTRNT)   ; Clear receiving area
6131    54                          LD      D,H             ;
6132    5D                          LD      E,L             ;
6133    36 00                       LD      (HL),00H        ;
6135    13                          INC     DE              ;
6136    01 0009                     LD      BC,ANSCNT-1     ;
6139    ED B0                       LDIR                    ;
                            ;
613B    2A F1AE                     LD      HL,(ICRSTRNT)   ; Receive TS byte
613E    CD 625C                     CALL    ICRDDT          ; Receive data and set it to RAM
6141    38 39                       JR      C,UICD150       ; Error (Perhaps time out)
6143    3A F1E9                     LD      A,(ICTSDT)      ; Check TS byte code
6146    BE                          CP      (HL)            ;
6147    3E 07                       LD      A,ERR7          ; Error code
6149    37                          SCF                     ;
614A    20 33                       JR      NZ,HOOKRET      ; Unmatch
                            ;
614C    23                          INC     HL              ; Receive T0 byte
614D    CD 625C                     CALL    ICRDDT          ;
6150    38 2A                       JR      C,UICD190       ; Receive error
6152    23                          INC     HL              ;
6153    4F                          LD      C,A             ;
6154    E6 0F                       AND     00001111B       ;
6156    57                          LD      D,A             ; Save complementary byte
                            ;
                            UICD110:
6157
```

```
6157    79                      LD      A,C             ; Check interface byte
6158    0F                      RRCA                    ;
6159    0F                      RRCA                    ;
615A    0F                      RRCA                    ;
615B    0F                      RRCA                    ;
615C    06 04                   LD      B,04H           ; Loop counter
                        ;
615E                    UICD120:
615E    0F                      RRCA                    ;
615F    F5                      PUSH    AF              ;
6160    DC 628C                 CALL    C,ICRDDT        ; Receive data if interface byte exists
6163    38 16                   JR      C,UICD180       ; Receive error
6165    4F                      LD      C,A             ;
6166    F1                      POP     AF              ;
6167    23                      INC     HL              ;
6168    10 F4                   DJNZ    UICD120         ; Loop 4 times (TAn,TBn,TCn,TDn)
616A    38 E8                   JR      C,UICD110       ; TDn byte exists
                        ;
616C    7A                      LD      A,D             ; Check complementary byte
616D    B7                      OR      A               ;
616E    28 0F                   JR      Z,HOOKRET       ; No complementary byte
6170    47                      LD      B,A             ;
                        ;
6171                    UICD130:
6171    CD 628C                 CALL    ICRDDT          ; Receive complementary byte
6174    38 08                   JR      C,UICD190       ;
6176    23                      INC     HL              ;
6177    10 F8                   DJNZ    UICD130         ;
6179    18 04                   JR      HOOKRET         ; Normal return
                        ;
617B                    UICD180:
617B    F1                      POP     AF              ;
617C                    UICD190:
617C    3E 06                   LD      A,ERR6          ; Error code (Time out)
617E    37                      SCF                     ; Error return
                        ;
                        ;       Following HOOKs are not used
                        ;
617F                    UICDHK2:
617F                    UICDHK3:
617F                    UICDHK4:
                        ;
                        ;       HOOK return
                        ;
617F                    HOOKRET:
617F    F5                      PUSH    AF              ; Save return code
6180    3A F42E                 LD      A,(RZIER)       ; Restore interrupt status
6183    D3 04                   OUT     (ZIER),A        ;
6185    F1                      POP     AF              ;
6186    FB                      EI                      ; Enable interrupt
6187    C9                      RET                     ;
                        ;
                        ;       ***************************************************
                        ;               RECEIVE DATA BY 1 RECORD
                        ;       ***************************************************
                        ;
                        ;       NOTE:
                        ;               Receive data by 1 record from IC-card.
                        ;               Calculate read-start address by SEKTRK & SEKSEC,
                        ;             and read into DMAADR from it.
                        ;
                        ;       << entry parameter >>
                        ;               (SEKTRK)  : Track for reading start address
                        ;               (SEKSEC)  : Sector for reading start address
                        ;               (DMAADR)  : Buffer address for reading into
                        ;       << return parameter >>
                        ;               CY : Return information
                        ;                       =0 : Normally opened
                        ;                       =1 : Open error
                        ;                  A    : Error code
                        ;                         =06 -- Time out error
                        ;                         =07 -- Protocol error
                        ;                         =CA -- Cover is opened
                        ;       << preserved registers >>
                        ;               All without AF
                        ;
6188                    UICDHK5:
6188    F3                      DI                      ;
6189    C5                      PUSH    BC              ; Save registers
618A    D5                      PUSH    DE              ;
618B    E5                      PUSH    HL              ;
618C    CD 6190                 CALL    ICRB00          ; Read 1 record
618F    18 07                   JR      ICRD1WT         ;
```

```
                              ;
                              ;        ****************************************************
                              ;                SEND DATA BY 1 RECORD
                              ;        ****************************************************
                              ;
                              ;        NOTE:
                              ;                Send data by 1 record to IC-card.
                              ;                Calculate write-start address by SENTRK & SENSEC,
                              ;               and write the data of DMAADR.
                              ;
                              ;        <> entry parameter        <>
                              ;                (SENTRK)  : Track for reading start address
                              ;                (SENSEC)  : Sector for reading start address
                              ;                (DMAADR)  : Buffer address for reading into
                              ;        <> return parameter       <>
                              ;                CY : Return information
                              ;                        =0 : Normally opened
                              ;                        =1 : Open error
                              ;                           A   : Error code
                              ;                                 =06 -- Time out error
                              ;                                 =07 -- Protocol error
                              ;                                 =0A -- Cover is opened
                              ;        <> preserved registers    <>
                              ;                All without AF
6191                          UICDHK6:
6191      F3                           DI                          ;
6192      C5                           PUSH    BC                  ; Save registers
6193      D5                           PUSH    DE                  ;
6194      E5                           PUSH    HL                  ;
6195      CD 61BC                      CALL    ICBW00              ; Write 1 record
                              ;
6198                          ICBDENT:
6198      E1                           POP     HL                  ; Restore registers
6199      D1                           POP     DE                  ;
619A      C1                           POP     BC                  ;
619B      18 E2                        JR      BOOSRET             ;
                              ;
                              ;        ****************************************************
                              ;                READ DATA BY 1 RECORD
                              ;        ****************************************************
                              ;
                              ;        NOTE:
                              ;                Same as UICDHK5
                              ;
                              ;        <> entry parameter        <>
                              ;                Same as UICDHK5
                              ;        <> return parameter       <>
                              ;                Same as UICDHK5
                              ;        <> preserved registers    <>
                              ;                NON
619D                          ICBR00:
619D      0E 00                        LD      C,00H               ; Select serial line to IC-card
619F      CD 6348                      CALL    SELSER              ;
61A2      CD 627A                      CALL    FILLNULL            ; Fill null code to DMA buffer
61A5      CD 63AA                      CALL    ICALRCD             ; Calculate record number
61A8      38 10                        JR      C,ICBR20            ; Virtual record
                              ;
61AA      2A F4D5                      LD      HL,(DMAADR)         ; DMA buffer address
                              ;
61AD                          ICBR10:
61AD      C5                           PUSH    BC                  ; Save loop counter
61AE      0E 01                        LD      C,RDCMD             ; Receive 1 physical record
61B0      CD 64A1                      CALL    ICMDPNC             ;
61B3      C1                           POP     BC                  ;
61B4      D8                           RET     C                   ; Error return
61B5      CD 645A                      CALL    IBLKCHK             ; Set next record number
61B8      10 F3                        DJNZ    ICBR10              ;
                              ;
61BA                          ICBR20:
61BA      B7                           OR      A                   ; Normal return
61BB      C9                           RET                         ;
                              ;
                              ;        ****************************************************
                              ;                WRITE DATA BY 1 RECORD
                              ;        ****************************************************
                              ;
                              ;        NOTE:
                              ;                Same as UICDHK6
                              ;
                              ;        <> entry parameter        <>
                              ;                Same as UICDHK6
                              ;        <> return parameter       <>
                              ;                Same as UICDHK6
                              ;        <> preserved registers    <>
                              ;                NON
61BC                          ICBW00:
61BC      0E 00                        LD      C,00H               ; Select serial line to IC-card
61BE      CD 6348                      CALL    SELSER              ;
61C1      CD 63AA                      CALL    ICALRCD             ; Calculate record number
61C4      38 10                        JR      C,ICBW20            ; Virtual record
61C6      2A F4D5                      LD      HL,(DMAADR)         ; DMA buffer address
                              ;
61C9                          ICBW10:
61C9      C5                           PUSH    BC                  ; Save loop counter
```

```
61CA    0E 02                   LD      C,WTCMD         ; Send 1 physical record
61CC    CD 64A1                 CALL    ICMDFNC         ;
61CF    C1                      POP     BC              ;
61D0    D8                      RET     C               ; Error return
61D1    CD [illegible]          CALL    IBLKCNK         ; Set next record number
61D4    10 F3                   DJNZ    ICRW10          ;
                        ;
61D6                    ICRW20:
61D6    B7                      OR      A               ; Normal return
61D7    C9                      RET                     ;
                        ;
                        ;       *****************************************************
                        ;               IC-CARD MOUNT CHECK
                        ;       *****************************************************
                        ;
                        ;       NOTE:
                        ;
                        ;       <> entry parameter      <>
                        ;               NONE
                        ;       <> return parameter     <>
                        ;               CY : Return information
                        ;                       =0 : Normally opened
                        ;                               (QTICCARD) -- IC-card size
                        ;                               (ICMAXREC) -- IC-card max record No
                        ;                               (TPICCARD) -- IC-card directory size
                        ;                               (DPB2)     -- Disk parameter block
                        ;                       =1 : Open error
                        ;                           A : Error code
                        ;                               =06 -- Time out error
                        ;                               =07 -- Protocol error
                        ;                               =0A -- Cover is opened
                        ;       <> preserved registers  <>
                        ;               NONE
                        ;
61D8                    UICDNK7:
61D8    F3                      DI                      ;
                        ;
                        ;       In this part, you must make IC-card software-open status,
                        ;       and determine the size of IC-card and set it to QTICCARD,
                        ;       and set IC-card max record size to ICMAXREC.
                        ;
                        ;       Following part is setting the system area by IC-card's size.
                        ;
F17C                    DSM2L           EQU     DPB2+5
F17E                    DRM2L           EQU     DPB2+7
F182                    CKS2L           EQU     DPB2+11

61D9    3A F65B                 LD      A,(QTICCARD)    ; IC-card size (Per kilo byte)
61DC    B7                      OR      A               ;
61DD    28 01                   JR      Z,CICD700       ; No size
61DF    3D                      DEC     A               ; Max IC-card size - 1
                        ;
61E0                    CICD700:
61E0    32 F17C                 LD      (DSM2L),A       ; Set disk size max.
61E3    01 0208                 LD      BC,2*256+8      ;
61E6    FE 02                   CP      2               ;
61E8    38 0C                   JR      C,CICD720       ; Less than 2 kilo bytes
61EA    CB 00                   RLC     B               ; B reg. is 4, C reg. is 16
61EC    CB 01                   RLC     C               ;
61EE    FE 05                   CP      5               ;
61F0    38 04                   JR      C,CICD720       ; Less than 5 kilo bytes
61F2    CB 00                   RLC     B               ; B reg. is 5, C reg. is 32
61F4    CB 01                   RLC     C               ;
                        ;
61F6                    CICD720:
61F6    78                      LD      A,B             ; Set sector number of directory
61F7    32 F95F                 LD      (TPICCARD),A    ;
61FA    79                      LD      A,C             ; Set directory number
61FB    3D                      DEC     A               ;
61FC    32 F17E                 LD      (DRM2L),A       ;
61FF    79                      LD      A,C             ; Set check sum vector size
6200    0F                      RRCA                    ;
6201    0F                      RRCA                    ;
6202    32 F182                 LD      (CKS2L),A       ;
6205    B7                      OR      A               ; Normal return
6206    C3 [illegible]          JP      HOOKRET         ;
```

```
                                ;
                                ;       ****************************************************
                                ;               FORMAT IC-CARD
                                ;       ****************************************************
                                ;
                                ;       NOTE:
                                ;               This routine is formating IC-card for disk use.
                                ;
                                ;       <> entry parameter          <>
                                ;               NONE
                                ;       <> return parameter         <>
                                ;               CY : Return information
                                ;                       =0 : Normally formatted
                                ;                       =1 : format error
                                ;                    A   : Error code
                                ;                               =01 -- Not mounted
                                ;                               =02 -- Cannot format
                                ;       <> preserved registers      <>
                                ;               NONE
                                ;
6209                            UICDBKS:
6209    3A F66C                         LD      A,(ICONFLG)      ; Check IC-card power-on status
620C    3D                              DEC     A                ;
620D    28 06                           JR      Z,UICD800        ; Already power on
620F    CD 61D9                         CALL    UICDBKT          ; IC-card mount check
6212    DA 617F                         JP      C,MOONRET        ; Mount error
                                ;
6215                            UICD800:
6215    01 0064                         LD      BC,100           ; Wait 100 millisecond (For key-sound)
6218    CD 62C6                         CALL    WAIT             ;
621B    3A F0DE                         LD      A,(BZCTLB3)      ; Stop key sound
621E    E6 3F                           AND     00111111B        ;
6220    32 F0DE                         LD      (BZCTLB3),A      ;
6223    D3 23                           OUT     (BZCTLB3),A      ;
                                ;
6225    CD 627A                         CALL    FILLNULL         ; Fill null code to DMA buffer
6228    CD 6465                         CALL    ICCETMAX         ; Calculate max track & record number
                                ;
622B    2A F4C2                         LD      HL,(SEKTRK)      ; Save current track & sector
622E    E5                              PUSH    HL               ;
622F    3A F4C4                         LD      A,(SEKSEC)       ;
6232    F5                              PUSH    AF               ;
6233    21 0000                         LD      HL,0000H         ; Set new track & sector
6236    22 F4C2                         LD      (SEKTRK),HL      ;
6239    AF                              XOR     A                ;
623A    32 F4C4                         LD      (SEKSEC),A       ;
                                ;
623D                            UICD810:
623D    CD 632A                         CALL    ICCHKMAX         ; Check max track & sector
6240    38 22                           JR      C,UICD820        ; Over max record
6242    CD 6181                         CALL    UICDWNG          ; Write null code into IC-card
6245    38 21                           JR      C,UICD830        ; Write error
                                ;
6247    3A F0B5                         LD      A,(YPOFST)       ; Check power-off interrupt
624A    B7                              OR      A                ;
624B    20 19                           JR      NZ,UICD830       ; Power-off happened
                                ;
624D    3A F4C4                         LD      A,(SEKSEC)       ; Set next sector number
6250    3C                              INC     A                ;
6251    32 F4C4                         LD      (SEKSEC),A       ;
6254    D6 40                           SUB     64               ;
6256    20 E5                           JR      NZ,UICD810       ;
                                ;
6258    32 F4C4                         LD      (SEKSEC),A       ; Sector No. is 0
625B    2A F4C2                         LD      HL,(SEKTRK)      ; Set next track number
625E    23                              INC     HL               ;
625F    22 F4C2                         LD      (SEKTRK),HL      ;
6262    18 D9                           JR      UICD810          ;
                                ;
6264                            UICD820:
6264    F1                              POP     AF               ;
6265    B7                              OR      A                ; Normal return
6266    18 [illegible]                  JR      UICD840          ;
                                ;
6268                            UICD830:
6268    CD 6314                         CALL    ICCLS00          ; Close IC-card
626B    06 02                           LD      B,02H            ; Error code
626D    F1                              POP     AF               ;
626E    37                              SCF                      ; Error return
                                ;
626F                            UICD840:
626F    32 F4C4                         LD      (SEKSEC),A       ; Restore track & sector number
6272    E1                              POP     HL               ;
6273    22 F4C2                         LD      (SEKTRK),HL      ;
6276    78                              LD      A,B              ; Set return code
6277    C3 617F                         JP      MOONRET          ;
```

```
                                ;
                                ;       ****************************************************
                                ;               UTILITY SUBROUTINES
                                ;       ****************************************************
                                ;
                                ;       NOTE:
                                ;               The following routines are utilities.
                                ;
                                ;       ****************************************************
                                ;               FILL NULL CODE TO DMA BUFFER
                                ;       ****************************************************
                                ;
                                ;       NOTE:
                                ;               Fill null code (E5H) to DMA buffer.
                                ;       <> entry parameter      <>
                                ;               NONE
                                ;       <> return parameter     <>
                                ;               NONE
                                ;       <> preserved registers  <>
                                ;               NONE
                                ;
627A                            FILLNULL:
627A    2A F4D5                         LD      HL,(DMAADR)     ; DMA buffer address
627D    06 80                           LD      B,80H           ; 128 bytes
627F    3E E5                           LD      A,0E5H          ; Null code
6281                            FILNUL10:
6281    77                              LD      (HL),A          ; Set null code to DMA buffer
6282    23                              INC     HL              ;
6283    10 FC                           DJNZ    FILNUL10        ;
6285    C9                              RET                     ;
                                ;
                                ;       ****************************************************
                                ;               RECEIVE 1 BYTE FROM IC-CARD
                                ;       ****************************************************
                                ;
                                ;       NOTE:
                                ;               Receive data from IC-card.
                                ;               This routine must be called by all interrupt disable.
                                ;       <> entry parameter      <>
                                ;               NONE
                                ;       <> return parameter     <>
                                ;               CY : Return information
                                ;                       =0 : Normally received
                                ;                               A reg. is received data
                                ;                       =1 : Time out error
                                ;       <> preserved registers  <>
                                ;               BC,DE,HL
                                ;
6286                            IRDCOM:
6286    CD 62B8                         CALL    IRESWRT         ; Reset write mode
6289    C3 6228                         JP      SRVBYT          ; Receive 1 byte from serial (IC-card)
                                ;
                                ;       This routine receives the data and sets it to work.
                                ;
628C                            ICRDBYT:
628C    CD 6286                         CALL    IRDCOM          ; Receive data
628F    D8                              RET     C               ; Time out error
6290    77                              LD      (HL),A          ; Save received data
6291    C9                              RET                     ;
                                ;
                                ;       ****************************************************
                                ;               SEND 1 BYTE TO IC-CARD
                                ;       ****************************************************
                                ;
                                ;       NOTE:
                                ;               Send data to IC-card
                                ;
                                ;       <> entry parameter      <>
                                ;               C  : Sending data
                                ;       <> return parameter     <>
                                ;               NONE
                                ;       <> preserved registers  <>
                                ;               BC,DE,HL
                                ;
6292                            IWRCOM:
6292    3A F0DA                         LD      A,(RZARTCR)     ;
6295    F5                              PUSH    AF              ;
6296    E6 F3                           AND     11110011B       ; Disable Rx-enable status
6298    32 F0DA                         LD      (RZARTCR),A     ;
629B    D3 16                           OUT     (ZARTCR),A      ;
                                ;
629D    CD 62B4                         CALL    ISETWRT         ; Set write mode
62A0    79                              LD      A,C             ;
62A1    CD 62B3                         CALL    SSDBYT          ; Send 1 byte to serial (IC-card)
                                ;
62A4                            IWRC10:
62A4    DB 15                           IN      A,(ZARTSR)      ; Check Tx-empty
62A6    E6 04                           AND     00000100B       ; (Wait until send the data to serial)
62A8    28 FA                           JR      Z,IWRC10        ;
                                ;
62AA    CD 62B8                         CALL    IRESWRT         ; Reset write mode
62AD    F1                              POP     AF              ; Restore Rx-enable status
62AE    32 F0DA                         LD      (RZARTCR),A     ;
62B1    D3 16                           OUT     (ZARTCR),A      ;
```

```
62B3    C9                          RET                             ;
                           ;
                           ;        *****************************************************
                           ;                SET OR RESET WRITE MODE
                           ;        *****************************************************
                           ;
                           ;        NOTE:
                           ;                There are two routines.
                           ;                One is setting write mode and another is resetting
                           ;              write mode.
                           ;
                           ;        <> entry parameter          <>
                           ;                NONE
                           ;        <> return parameter         <>
                           ;                NONE
                           ;        <> preserved registers      <>
                           ;                BC,DE,HL
                           ;
62B4                       ISETWMT:
62B4    3A F0D8                     LD      A,(RZICCTLB)        ; Set write mode
62B7    E6 FB                       AND     11111011B           ;
62B9    18 05                       JR      ISETSRES            ;
                           ;
62BB                       IRESWMT:
62BB    3A F0D8                     LD      A,(RZICCTLB)        ; Reset write mode
62BE    F6 04                       OR      00000100B           ;
                           ;
62C0                       ISETSRES:
62C0    32 F0D8                     LD      (RZICCTLB),A        ;
62C3    D3 17                       OUT     (ZICCTLB),A         ;
62C5    C9                          RET                         ;
                           ;
                           ;        *****************************************************
                           ;                WAIT ANY TIME
                           ;        *****************************************************
                           ;
                           ;        NOTE:
                           ;                Software timer.
                           ;                If interrupt is happened, then the waiting time
                           ;              is longer.
                           ;
                           ;        <> entry parameter          <>
                           ;                BC : Time (Per millisecond)
                           ;        <> return parameter         <>
                           ;                NONE
                           ;        <> preserved registers      <>
                           ;                AF,DE,HL
                           ;
62C6                       WAIT:
62C6    F5                          PUSH    AF                  ; Save register
62C7                       WAIT10:
62C7    3E E6                       LD      A,230               ; Loop counter
62C9                       WAIT20:
62C9    3D                          DEC     A                   ; Wait 1 millisecond
62CA    20 FD                       JR      NZ,WAIT20           ;
                           ;
62CC    0B                          DEC     BC                  ; Wait X millisecond
62CD    78                          LD      A,B                 ;
62CE    B1                          OR      C                   ;
62CF    20 F6                       JR      NZ,WAIT10           ;
62D1    F1                          POP     AF                  ;
62D2    C9                          RET                         ;
                           ;
                           ;        *****************************************************
                           ;                SEND DATA TO SERIAL
                           ;        *****************************************************
                           ;
                           ;        NOTE:
                           ;                Send data to serial.
                           ;
                           ;        <> entry parameter          <>
                           ;                A  : Sending data
                           ;        <> return parameter         <>
                           ;                NONE
                           ;        <> preserved registers      <>
                           ;                All registers
                           ;
62D3                       SSDBYT:
62D3    F5                          PUSH    AF                  ; Save sending data.
62D4    3A F02B                     LD      A,(ICCPWTM)         ; Set IC-card power off time
62D7    32 F2CE                     LD      (ICCPWCNT),A        ; (Default is 30 seconds)
62DA    3E 01                       LD      A,01H               ; Set IC-card power off check flag
62DC    32 F203                     LD      (ICCPWFLG),A        ;
                           ;
62DF                       SSDB10:
62DF    DB 15                       IN      A,(ZARTSR)          ; Wait until Tx-ready
62E1    E6 01                       AND     00000001B           ;
62E3    28 FA                       JR      Z,SSDB10            ;
62E5    F1                          POP     AF                  ;
62E6    D3 14                       OUT     (ZARTDOR),A         ; Send the data to serial
62E8    C9                          RET                         ;
```

```
                          ;
                          ;       **************************************************
                          ;               RECEIVE DATA FROM SERIAL
                          ;       **************************************************
                          ;
                          ;       NOTE:
                          ;               Receive data from serial.
                          ;
                          ;       <> entry parameter        <>
                          ;               NONE
                          ;       <> return parameter       <>
                          ;               CY : Return information
                          ;                       =0 : Normally received
                          ;                               A reg. is received data
                          ;                       =1 : Time out error
                          ;       <> preserved registers    <>
                          ;               BC,DE,HL
                          ;
62E9                      RSVBYT:
62E9    C5                        PUSH    BC
62EA    ED 4B F1A8                LD      BC,(RPISBTO)      ; Set time-out counter
62EE    CB 21                     SLA     C                 ; (RPIRBTO) * 4 is counter value
62F0    CB 10                     RL      B                 ;
62F2    CB 21                     SLA     C                 ;
62F4    CB 10                     RL      B                 ;
62F6                      RSVB10:
62F6    0B                        DEC     BC                ; Check time-out
62F7    78                        LD      A,B               ;
62F8    B1                        OR      C                 ;
62F9    3E 06                     LD      A,006H            ; Time-out error code
62FB    28 14                     JR      Z,RSVB80          ; Time-out error
                          ;
62FD    DB 15                     IN      A,(ZARTSR)        ; Wait until Rx-ready
62FF    E6 02                     AND     00000010B         ;
6301    28 F3                     JR      Z,RSVB10          ;
                          ;
6303    3A F026                   LD      A,(ICCPWTM)       ; Set IC-card power off time
6306    32 F202                   LD      (ICCPWCNT),A      ; (Default is 30 seconds)
6309    3E 01                     LD      A,01H             ; Set IC-card power off check flag
630B    32 F203                   LD      (ICCPWFLG),A      ;
630E    DB 14                     IN      A,(ZARTDIR)       ; Read received data
6310    37                        SCF                       ;
                          ;
6311                      RSVB80:
6311    3F                        CCF                       ; Set return information
6312    C1                        POP     BC                ;
6313    C9                        RET                       ;
                          ;
                          ;       **************************************************
                          ;               CLOSE IC-CARD
                          ;       **************************************************
                          ;
                          ;       NOTE:
                          ;               Close IC-card.
                          ;               This routine must be run in interrupt disable.
                          ;
                          ;       <> entry parameter        <>
                          ;               NONE
                          ;       <> return parameter       <>
                          ;               NONE
                          ;       <> preserved registers    <>
                          ;               NONE
                          ;
6314                      ICCLS00:
6314    3A F00A                   LD      A,(RZARTCR)       ; Save current ART-command register
6317    F5                        PUSH    AF                ;
6318    E6 FB                     AND     11111011B         ; Reset Rx-enable
631A    D3 16                     OUT     (ZARTCR),A        ;
                          ;
631C    3A F00D                   LD      A,(RZIOCTLR)      ; Set SST to high level
631F    F6 04                     OR      00000100B         ;
6321    32 F00D                   LD      (RZIOCTLR),A      ;
6324    D3 19                     OUT     (ZIOCTLR),A       ;
                          ;
6326    3A F00D                   LD      A,(RZIOCTLR)      ; Disable cover interrupt
6329    E6 FC                     AND     11111100B         ; Stop power-supply
632B    32 F00D                   LD      (RZIOCTLR),A      ; Stop clock-supply
632E    D3 19                     OUT     (ZIOCTLR),A       ;
                          ;
6330    AF                        XOR     A                 ; Reset IC-card flags
6331    32 F202                   LD      (ICCPWFLG),A      ; Power control flag
6334    32 F662                   LD      (ICCUSEDV),A      ; Using device flag
6337    32 F660                   LD      (ICONFLG),A       ; Power on flag
633A    32 F661                   LD      (ICCVFLG),A       ; Cover status flag
                          ;
633D    0E 01                     LD      C,01H             ; Set serial port to RS-232C
633F    CD 6346                   CALL    SELSER            ;
6342    F1                        POP     AF                ;
6343    D3 16                     OUT     (ZARTCR),A        ;
6345    C9                        RET                       ;
```

```
                                ;
                                ;       ******************************************************
                                ;               SELECT SERIAL LINE
                                ;       ******************************************************
                                ;
                                ;       NOTE:
                                ;               Select serial line & change it.
                                ;
                                ;       () entry parameter       ()
                                ;               C   : line number
                                ;       () return parameter      ()
                                ;               NONE
                                ;       () preserved registers   ()
                                ;               NONE
                                ;
6346                            SELS20:
6346    3A F241                         LD      A,(SBMODE)      ; Check current serial status.
6349    91                              SUB     C               ;
634A    C8                              RET     Z               ; Same as new one
                                ;
634B                            SELS10:
634B    DB 15                           IN      A,(ZARTSR)      ; Wait until Tx-empty
634D    E6 04                           AND     00000100B       ;
634F    28 FA                           JR      Z,SELS10        ;
6351    79                              LD      A,C             ; Set new serial line
6352    32 F241                         LD      (SBMODE),A      ;
6355    07                              RLCA                    ; Get new serial parameter table address
6356    07                              RLCA                    ;
6357    81                              ADD     A,C             ;
6358    4F                              LD      C,A             ;
6359    06 00                           LD      B,00H           ;
635B    21 F196                         LD      HL,SBTABL       ;
635E    09                              ADD     HL,BC           ;
                                ;
635F    3A F0DA                         LD      A,(BZARTCR)     ; Disable Rx-enable & Tx-enable
6362    E6 FA                           AND     0FAH            ;
6364    CD 63A4                         CALL    SELS50          ;
                                ;
6367    7E                              LD      A,(HL)          ; Set CTLR1 register
6368    E6 F0                           AND     0F0H            ; (Bit rate)
636A    77                              LD      (HL),A          ;
636B    3A F0D6                         LD      A,(BZCTLR1)     ;
636E    E6 0F                           AND     00FH            ;
6370    B6                              OR      (HL)            ;
6371    32 F0D6                         LD      (BZCTLR1),A     ;
6374    D3 00                           OUT     (ZCTLR1),A      ;
                                ;
6376    23                              INC     HL              ;
6377    46                              LD      B,(HL)          ;
6378    23                              INC     HL              ;
6379    7E                              LD      A,(HL)          ; Set SWR register
637A    E6 0C                           AND     00CH            ; (Serial line)
637C    77                              LD      (HL),A          ;
637D    3A F0DC                         LD      A,(BZSWR)       ;
6380    E6 F3                           AND     0F3H            ;
6382    B6                              OR      (HL)            ;
6383    32 F0DC                         LD      (BZSWR),A       ;
6386    D3 19                           OUT     (ZSWR),A        ;
                                ;
6388    23                              INC     HL              ; Set ARTCR register
6389    CD 6393                         CALL    SELS40          ;
638C    78                              LD      A,B             ; Set ARTMR register
638D    32 F0D9                         LD      (BZARTMR),A     ; (Parity, Bit length, Stop bit)
6390    D3 14                           OUT     (ZARTMR),A      ;
6392    C9                              RET                     ;
                                ;
                                ;
6393                            SELS40:
6393    3A F0DC                         LD      A,(BZSWR)       ; Check serial line
6396    E6 08                           AND     08H             ;
6398    7E                              LD      A,(HL)          ;
6399    20 09                           JR      NZ,SELS50       ; Not RS-232C
                                ;
639B    E6 3D                           AND     03DH            ; Set RTS & DTR control line
639D    4F                              LD      C,A             ;
639E    3A F0DA                         LD      A,(BZARTCR)     ;
63A1    E6 C2                           AND     0C2H            ;
63A3    B1                              OR      C               ;
                                ;
63A4                            SELS50:
63A4    32 F0DA                         LD      (BZARTCR),A     ; Set ARTCR register
63A7    D3 16                           OUT     (ZARTCR),A      ;
63A9    C9                              RET                     ;
```

```
                                  ;
                                  ;       ****************************************************
                                  ;               CALCULATE RECORD
                                  ;       ****************************************************
                                  ;
                                  ;       NOTE:
                                  ;               Calculate physical record number & loop counter
                                  ;              by seek track/sector.
                                  ;
                                  ;       <> entry parameter       <>
                                  ;               (ICRECSZ)  -- 1 physical record size
                                  ;               (TPICCARD) -- Directory record number
                                  ;               (SEKTRK)   -- Seek track number
                                  ;               (SEKSEC)   -- Seek sector number
                                  ;       <> return parameter      <>
                                  ;               CY : 1 -- Virtual record
                                  ;                    0 -- Physical record
                                  ;                         DE : Record number
                                  ;                         B  : Loop counter
                                  ;       <> preserved registers  <>
                                  ;               NONE
63AA                              ICALRCD:
63AA    3A F85F                           LD      A,(TPICCARD)      ; Check virtual record
63AD    47                                LD      B,A               ; Directory record number
63AE    3A F4C6                           LD      A,(SEKSEC)        ;
63B1    B8                                CP      B                 ; TPICCARD sector to 1 sector of 0 track
63B2    38 0B                             JR      C,IBLK10          ;  then virtual record
63B4    FE 05                             CP      05H               ;
63B6    30 07                             JR      NC,IBLK10         ;
63B8    2A F4C2                           LD      HL,(SEKTRK)       ;
63BB    7C                                LD      A,H               ;
63BC    B5                                OR      L                 ;
63BD    37                                SCF                       ;
63BE    C9                                RET     Z                 ; Virtual record
                                  ;
63BF                              IBLK10:
63BF    3A F17E                           LD      A,(DPB2+7)        ; Directory number
63C2    3C                                INC     A                 ;
63C3    CD 63F9                           CALL    CALKADR           ; Calculate logical address
63C6    2A F08A                           LD      HL,(LOCADR)       ;
63C9    CD 63CE                           CALL    ICRECCHK          ; Calculate physical address
63CC    B7                                OR      A                 ;
63CD    C9                                RET                       ;
                                  ;
                                  ;       ****************************************************
                                  ;               CALACULATE PHYSICAL RECORD NUMBER
                                  ;       ****************************************************
                                  ;
                                  ;       NOTE:
                                  ;               Calculate physical record number
                                  ;
                                  ;       <> entry parameter       <>
                                  ;               HL : Logical address
                                  ;       <> return parameter      <>
                                  ;               DE : Physical record number
                                  ;               B  : Loop counter
                                  ;       <> preserved registers  <>
                                  ;               NONE
                                  ;
63CE                              ICRECCHK:
63CE    AF                                XOR     A                 ; Calculate record number (Per 128 bytes)
63CF    CB 15                             RL      L                 ; HL / 128 --> DE
63D1    CB 14                             RL      H                 ;
63D3    17                                RLA                       ;
63D4    57                                LD      D,A               ;
63D5    5C                                LD      E,H               ;
                                  ;
63D6                              ICREC10:
63D6    62                                LD      H,D               ; Copy data
63D7    6B                                LD      L,E               ;
63D8    3A F125                           LD      A,(ICRECSZ)       ; 1 physical record size
63DB    4F                                LD      C,A               ; Check physical record number
63DC    3E 80                             LD      A,128             ;
63DE    06 01                             LD      B,01H             ;
                                  ;
63E0                              ICREC20:
63E0    91                                SUB     C                 ; B reg. is loop counter
63E1    C8                                RET     Z                 ; DE reg. is physical record number
63E2    D8                                RET     C                 ;
63E3    04                                INC     B                 ;
63E4    EB                                EX      DE,HL             ;
63E5    19                                ADD     HL,DE             ; Record number (Per ICRECSZ)
63E6    EB                                EX      DE,HL             ;
63E7    D8                                RET     C                 ; Record number overflow
63E8    18 F6                             JR      ICREC20           ;
```

```
                                  ;
                                  ;       *****************************************************
                                  ;               CHECK TRACK/SECTOR OVERFLOW
                                  ;       *****************************************************
                                  ;
                                  ;       NOTE:
                                  ;               This routine checks overflow of track/sector.
                                  ;               But it neglects first byte of track, so first byte
                                  ;               must be always zero.
                                  ;
                                  ;       <> entry parameter        <>
                                  ;               DE : Maximum track/record number
                                  ;               (SEKTRK) -- Current track number
                                  ;               (SEKSEC) -- Current sector number
                                  ;       <> return parameter       <>
                                  ;               CY : 1 -- Track/sector overflow
                                  ;                    0 -- Address O.K.
                                  ;       <> preserved registers    <>
                                  ;               BC,DE,HL
                                  ;
63EA                              CCBKMAX:
63EA       3A F4C2                        LD      A,(SEKTRK)      ; Check track number
63ED       BA                             CP      D               ; Track number unmatch?
63EE       3F                             CCF                     ;
63EF       D0                             RET     NC              ; Yes
63F0       C0                             RET     NZ              ; Track number overflow
63F1       3A F4C4                        LD      A,(SEKSEC)      ; Check sector number
63F4       BB                             CP      E               ;
63F5       C8                             RET     Z               ; Match
63F6       3F                             CCF                     ;
63F7       C9                             RET                     ;
                                  ;
                                  ;       *****************************************************
                                  ;               CALCULATE LOGICAL ADDRESS
                                  ;       *****************************************************
                                  ;
                                  ;       NOTE:
                                  ;               Calculate logical address by seeking track/sector.
                                  ;
                                  ;       <> entry parameter        <>
                                  ;               A  : Directory No  (If correct)
                                  ;               (SEKTRK)  -- Seek track number
                                  ;               (SEKSEC)  -- Seek sector number
                                  ;       <> return parameter       <>
                                  ;               (LOCADR)  -- Logcal address
                                  ;       <> preserved registers    <>
                                  ;               ALL
                                  ;
63F8                              CALKADR:
63F8       F5                             PUSH    AF              ; Save all registers
63F9       C5                             PUSH    BC              ;
63FA       D5                             PUSH    DE              ;
63FB       E5                             PUSH    HL              ;
                                  ;
63FC       21 0000                        LD      HL,0000H        ;
63FF       B7                             OR      A               ; Check offset record
6400       28 17                          JR      Z,CAL20         ; Non need of correct
6402       47                             LD      B,A             ;
6403       3A F4C2                        LD      A,(SEKTRK)      ; Current record is in directory area?
6406       B7                             OR      A               ;
6407       20 07                          JR      NZ,CAL10        ; No
6409       3A F4C4                        LD      A,(SEKSEC)      ;
640C       FE 05                          CP      05H             ;
640E       38 09                          JR      C,CAL20         ; Yes
                                  ;
6410                              CAL10:
6410       3E 20                          LD      A,32            ; Calculate correct size
6412       90                             SUB     B               ; (32 - Directry No.) * 32 --> HL
6413       6F                             LD      L,A             ;
6414       1E 05                          LD      E,05H           ;
6416       CD 6453                        CALL    MULTI           ;
                                  ;
6419                              CAL20:
6419       E5                             PUSH    HL              ; Save correct size
641A       2A F4C2                        LD      HL,(SEKTRK)     ; Change track number to sector number
641D       26 00                          LD      H,00H           ;  Track number * 54
641F       06 36                          LD      B,36H           ;
6421       CD 6453                        CALL    MULTI           ;
6424       EB                             EX      DE,HL           ; Calculate total sector number
6425       2A F4C4                        LD      HL,(SEKSEC)     ;
6428       26 00                          LD      H,00H           ;
642A       19                             ADD     HL,DE           ;
                                  ;
642B       EB                             EX      DE,HL           ; Calculate logical address
642C       21 0000                        LD      HL,0000H        ;  Total sector number * 128
642F       06 07                          LD      B,07H           ;
                                  ;
6431                              CAL30:
6431       CB 23                          SLA     E               ; (HLDE) * 2 -- (HLDE)
6433       CB 12                          RL      D               ;
6435       CB 15                          RL      L               ;
6437       CB 14                          RL      H               ;
6439       10 F6                          DJNZ    CAL30           ;
                                  ;
```

```
843B    C1                  POP     BC              ; Correct address
843C    EB                  EX      DE,HL           ; BC reg. is offset value
843D    B7                  OR      A               ; (HLDE) - (BC) --> (HLDE)
843E    ED 42               SBC     HL,BC           ;
8440    EB                  EX      DE,HL           ;
8441    01 0000             LD      BC,0000H        ;
8444    ED 42               SBC     HL,BC           ;
                        ;
8446    ED 53 F06A          LD      (LOCADR),DE     ; Set logical address
844A    7D                  LD      A,L             ;
844B    32 F06C             LD      (LOCADR+2),A    ;
844E    E1                  POP     HL              ; Restore all registers
844F    D1                  POP     DE              ;
8450    C1                  POP     BC              ;
8451    F1                  POP     AF              ;
8452    C9                  RET                     ;
                        ;
                        ;       Shift HL register by B register.
                        ;
8453                    MULTI:
8453    CB 25               SLA     L               ; HL * 2 ** B --> HL
8455    CB 14               RL      H               ;
8457    10 FA               DJNZ    MULTI           ;
8459    C9                  RET                     ;
                        ;
                        ;       ****************************************************
                        ;               UPDATE IC-CARD RECORD NUMBER
                        ;       ****************************************************
                        ;
                        ;       NOTE:
                        ;               Select serial line & change it.
                        ;
                        ;       <> entry parameter      <>
                        ;               DE : Record number
                        ;               HL : Logical address
                        ;       <> return parameter     <>
                        ;               DE : Next record number
                        ;               HL : Next logical address
                        ;       <> preserved registers  <>
                        ;               BC
                        ;
845A                    ICRECNX:
845A    D5                  PUSH    DE              ;
845B    3A F1E5             LD      A,(ICRECSZ)     ; 1 physical record size
845E    5F                  LD      E,A             ; Calculate next logical address
845F    16 00               LD      D,00H           ;
8461    19                  ADD     HL,DE           ;
8462    D1                  POP     DE              ;
8463    13                  INC     DE              ; Next physical record number
8464    C9                  RET                     ;
                        ;
                        ;       ****************************************************
                        ;               GET IC-CARD MAX TRACK/SECTOR
                        ;       ****************************************************
                        ;
                        ;       NOTE:
                        ;               Get maximum track/sector of IC-card.
                        ;
                        ;       <> entry parameter      <>
                        ;               (ICMAXREC) -- Maximum record number
                        ;       <> return parameter     <>
                        ;               CY : =1 -- No IC-card.
                        ;                    =0 -- IC-card in.
                        ;                               D : Sector number
                        ;                               E : Track number
                        ;       <> preserved registers  <>
                        ;               NONE
                        ;
8465                    ICGETMAX:
8465    3A F560             LD      A,(ICONFLG)     ; Check IC-card power on status
8468    FE 01               CP      01H             ;
846A    C4 81D5             CALL    NZ,ICMOUNT      ; Power off? Then mount check
846D    D8                  RET     C               ; Mount error
                        ;
846E    2A F65C             LD      HL,(ICMAXREC)   ; Check max record number
8471    7C                  LD      A,H             ;
8472    B5                  OR      L               ;
8473    37                  SCF                     ;
8474    C8                  RET     Z               ; No record
                        ;
8475    2B                  DEC     HL              ;
8476    3A F65F             LD      A,(TPICCARD)    ; Add the size of virtual record
8479    47                  LD      B,A             ;
847A    3E 08               LD      A,8             ;
847C    90                  SUB     B               ;
847D    4F                  LD      C,A             ;
847E    06 00               LD      B,00H           ;
8480    09                  ADD     HL,BC           ;
                        ;
8481    0E 40               LD      C,40H           ; Calculate max track/sector
8483    16 00               LD      D,00H           ;
                        ;
8485                    ICGET10:
8485    ED 42               SBC     HL,BC           ; Subtract records per track
```

```
6487    38 03                   JR      C,ICCET20       ;
6489    14                      INC     D               ; Track counter 1 up
648A    18 F9                   JR      ICCET10         ;
                        ;
648C                    ICCET20:
648C    09                      ADD     HL,BC           ;
648D    5D                      LD      E,L             ; Set record number
648E    B7                      OR      A               ;
648F    C9                      RET                     ;
                        ;
                        ;       ************************************************************
                        ;               CHECK IC-CARD INTERRUPT
                        ;       ************************************************************
                        ;
                        ;       NOTE:
                        ;               Check IC-card status
                        ;
                        ;       <> entry parameter      <>
                        ;               (ICMAXREC) -- Maximum record number
                        ;       <> return parameter     <>
                        ;               CY : =1 -- Error
                        ;                          A reg. is error code
                        ;                    =0 -- Status is O.K
                        ;       <> preserved registers  <>
                        ;               BC,DE,HL
                        ;
6490                    ICINTCHK:
6490    3A F661                 LD      A,(ICCVFLG)     ; Check IC-card cover status
6493    B7                      OR      A               ;
6494    3E 0B                   LD      A,ERR11         ; Error code
6496    37                      SCF                     ;
6497    C0                      RET     NZ              ; Cover is opened
                        ;
6498    3A F680                 LD      A,(ICONFLG)     ; Check IC-card power status
649B    3C                      INC     A               ;
649C    3E 09                   LD      A,ERR9          ; Error code
649E    C8                      RET     Z               ; Power is off
649F    AF                      XOR     A               ;
64A0    C9                      RET                     ; Normal return
                        ;
                        ;       ************************************************************
                        ;               SEND COMMAND & RECEIVE ANSWER
                        ;       ************************************************************
                        ;
                        ;       NOTE:
                        ;               Send command and receive answer.
                        ;
                        ;       <> entry parameter      <>
                        ;               C  : Command code
                        ;                       =01 -- Read record
                        ;                       =02 -- Write record
                        ;               DE : Record number
                        ;               HL : Buffer for data sending/receiving
                        ;       <> return parameter     <>
                        ;               CY : =1 -- Error
                        ;                    =0 -- Normal end
                        ;       <> preserved registers  <>
                        ;               BC,DE,HL
                        ;
64A1                    ICMDFNC:
64A1    CD 6490                 CALL    ICINTCHK        ; Check IC-card interrupt
64A4    D8                      RET     C               ;
64A5    79                      LD      A,C             ; Check logical command code.
64A6    FE 01                   CP      01H             ;
64A8    28 06                   JR      Z,ICMD100       ; Read function
64AA    FE 02                   CP      02H             ;
64AC    28 1A                   JR      Z,ICMD200       ; Write function
64AE    37                      SCF                     ;
64AF    C9                      RET                     ;
                        ;
                        ;       Following programs are reading from or writing into
                        ;       IC-card. But they are concerned to IC-card's protocol.
                        ;       So they are made to be secret.
                        ;       You make these routines as your IC-card working.
                        ;
                        .LIST
                                END

Macros:

Symbols:
000A    ASSCNT          0002    APLBOM          6410    CAL10
6419    CAL20           6431    CAL30           63F5    CALKADB
F252    CNS2L           EB09    CONIN           EB0C    CONOUT
0001    CRC16           F405    DMAADR          F177    DPB2
F17E    DBM2L           F17C    DSM2L           F1AB    EP1BBTO
000B    ERR11           0006    ERR6            0007    ERR7
0009    ERR9            627A    FILLNULL        6291    FILNUL10
FF6F    HOOKADDR        6080    HOOKDATA        617F    HOOKRET
63BF    IBLK10          645A    IBLKCHK         63AA    ICALRCD
63EA    ICCHKMAX        6314    ICCLS00         F202    ICCPWCNT
F203    ICCPWFLG        F026    ICCPWTM         F661    ICCVFLG
6485    ICCET10         648C    ICCET20         6465    ICCETMAX
F1E6    ICHOKDTI        6490    ICINTCHK        F65C    ICMAXREC
64B0    ICMD100         64C8    ICMD200         64A1    ICMDFNC
```

?001	ICOMRED	0010	ICOMTRN	0002	ICOMWRT
?660	ICONFLG	6595	ICRC10	657E	ICRCCHK
?1B8	ICBDSWT	626C	ICRDDT	63D6	ICREC10
?3E0	ICBEC20	63CE	ICRECCHK	F1E5	ICRECS2
?534	ICBED10	653E	ICRED20	6523	ICREDDAT
619D	ICBR00	61AD	ICBR10	61BA	ICRB20
?1AE	ICBSTPNT	61BC	ICBW00	61C8	ICRW10
61D6	ICBW20	F663	ICTDDT	F1E9	ICTSDT
F662	ICUSEDV	654E	ICWBT10	6542	ICWBTDAT
6578	IBDBCRC	6256	IBDCOM	6566	IBDCBC
651F	IBEDS20	64FB	IBEDSTS	6288	IBESWBT
62C0	ISETSBES	62B4	ISETWBT	64DE	ISNDCOM
6575	IWBSCRC	62A9	IWAC10	6282	IWBCOM
655D	IWBCRC	F06A	LOCADR	1000	MAINSP
0064	MODE	6453	MULTI	F65E	QTICCARD
0001	RDCMD	0020	RECSIZE	F0DA	RZARTCR
F0D8	RZARTMR	F006	RZCTLR1	F0DE	RZCTLR3
F0DB	RZICCTLR	F42E	RZIER	F0D0	RZIOCTLR
F0DC	RZSWR	F4C4	SEKSEC	F4C2	SEKTBE
6318	SELS10	6393	SELS50	[illegible]	SELS90
6346	SELSEB	F241	SEMODE	F196	SETABL
62F6	SEVE10	6311	SEVE90	62E9	SEVBYT
620F	SSDB10	62D3	SSDBYT	6055	START
0002	STX	F197	STSARTMR	F65E	TPICCARD
0038	TSBTYP	6005	UBDATA	65A8	UBDTBDT
?005	UBDTTOP	6157	UICD110	615E	UICD120
6171	UICD130	617B	UICD150	617C	UICD190
61E0	UICD700	61F6	UICD720	6215	UICD900
623D	UICD910	6264	UICD620	6268	UICD820
620F	UICD840	60ED	UICDHE1	617F	UICDHE2
617F	UICDHE3	617F	UICDHE4	6198	UICDHE5
6191	UICDHE6	61D9	UICDHE7	6209	UICDHES
62C6	WAIT	62C7	WAIT10	62C9	WAIT20
0050	WAITTIME	F803	WBOOT	0002	WTCMD
F0B5	YPOPST	0016	ZARTCR	0014	ZARTDIR
0018	ZARTDDR	0015	ZARTMR	0015	ZARTSR
0000	ZCTLR1	0023	ZCTLR3	0017	ZICCTLR
0004	ZIER	0019	ZIOCTLR	C016	ZIOSTR
0019	ZSWR				

No Fatal error(s)

(34) SAMPLE 34. FDD Format/Copy utility for the EHT-10/EHT-10/2

< Overview >

This program enables FDD format/copy by using PF-10 or TF-15 via RS-232C I/F of EHT-10/EHT-10/2. This program does not work by itself. It should be included in an application.

< Function >

This program has the following functions.

(1) Format "D" drive.

(2) Copy from "D" drive to "E" drive. (In this case, you should use Dual type TF-15)

< How to use >

(1) Include this program in your application.

(2) Added the process of extended subroutine. (CALSUB1)

You can do the special process by using CALSUB1 (Ex. Display the track No. on the LCD.)

(3) Set the address of CALSUB1 to CALLAD1, and call FDDUTY.

```
                        ;       <> Assemble condition   <>
                        ;
                                .Z80
                        ;
                        ;       <> Start address        <>
                        ;
                                .PHASE          100H
                                .COMMENT        *
                        ;
                        ;
                        ;               Copyright (c) EPSON 1986
                        ;
                        ;               Created   by    T.Tanaka
                        ;                               1986. 7.25      ver 1.0
                        ;
                        ;
                        ;********************************
                        ;*      OPERATION GUIDE         *
                        ;********************************
                        ;
                        ;[COMMAND NAME]
                        ;               FDDUTY
                        ;
                        ;[HANDLE DRIVE]
                        ;               (D,E)
                        ;
                        ;[FUNCTION]             a) DISK VOLUME COPY
                        ;                               Copy from D drive to E drive
                        ;                       b) FORMAT
                        ;                               Format D drive
                        ;
                                SUBTTL  SYSTEM CONSTANT
                        ;
                        ;   *******     PROGRAM CONSTANT     ********
                        ;
001F                    EPSPSND         EQU     001FH   ; Data send to TF & receive from TF
0022                    EPSPRCV         EQU     0022H   ; Only receive data ... receive ACK

F0B4                    YPOFDS          EQU     0F0B4H  ; Power off interrupt disable flag
F0B5                    YPOFST          EQU     0F0B5H  ; Power off status flag
F0B6                    YALMDS          EQU     0F0B6H  ; Alarm/Wake interrupt disable flag
F0B7                    YALMST          EQU     0F0B7H  ; Alarm/Wake status flag
F220                    CSTOPFLG        EQU     0F220H  ; CTRL/STOP flag
F418                    DISBNK          EQU     0F418H  ; Destination bank

EB69                    CALLX           EQU     0EB69H
FF96                    JPSTBIOS        EQU     0FF96H

0028                    TRKMAX          EQU     40      ; Track max num

                        ;
                        ;       Work area arrange
                        ;
FBAA                    WKSTART         EQU     0FBAAH  ; System work area

FBAA                    CALLAD1         EQU     WKSTART +00
                                                        ; 1 track access
                        ;
                        ;
```

```
                                ;        Following are temporary work area
                                ;
FBAC                            DINTFLG          EQU     CALLADR +02
                                                                 ; Interrupt status (Always use)
FBAD                            TRACK            EQU     DINTFLG +01
                                                                 ; Current track number (always use)
                                ;
                                ;        Use EPSP data send/receive
                                ;
FBAE                            PACKET           EQU     TRACK   +01
                                                                 ;
FBAE                            FMT      EQU     PACKET          ; Format  0=from master to slave
FBAF                            DID      EQU     FMT+1           ; Destination device ID  31h=D,E
FBB0                            SID      EQU     DID+1           ; Source device ID  23h=master
FBB1                            FNC      EQU     SID+1           ; Function
FBB2                            SIZ      EQU     FNC+1           ; Data size  0= data 1, 1= data 2,.....
FBB3                            DAT      EQU     SIZ+1           ; TRK, DRV, SEC, TYPE, 1 sector data
                                                                 ; etc...
FC3C                            BOT      EQU     DAT+137
                                         SUBTTL  SAMPLE PROGRAM

                                ;        ********************************
                                ;                Sample main program
                                ;        ********************************

1000                            STACK    EQU     1000H

EB03                            WBOOT    EQU     0EB03H
EB09                            CONIN    EQU     0EB09H
EB0C                            CONOUT   EQU     0EB0CH
EB6F                            PUTPFK   EQU     0EB6FH

000A                            LF       EQU     0AH
000D                            CR       EQU     0DH
0020                            SPACE    EQU     20H

0100                            START:
0100       31 1000                       LD      SP,STACK        ; Set stack pointer

0103       0E 01                         LD      C,01            ; Define key table for RC-10
0105       06 01                         LD      B,01            ;
0107       3E 01                         LD      A,01            ;
0109       CD EB6F                       CALL    PUTPFK          ;

010C       21 016C                       LD      HL,MSG00        ; Display opening message
010F       CD 0159                       CALL    DSPMSG          ;
                                ;
                                ;        Make packet data
                                ;
0112       21 0137                       LD      HL,SUB1         ; Subroutine call
0115       22 FBAA                       LD      (CALLADR),HL    ;
0118       CD 0193                       CALL    FDOUTT          ; Go !!
011B       38 09                         JR      C,ERROR         ; Error

011D       21 0179                       LD      HL,MSG01        ; Display ending message
0120       CD 0165                       CALL    DSPMSCRLN       ;
0123       C3 EB03                       JP      WBOOT           ;
                                ;
                                ;        Display error code
                                ;
0126                            ERROR:
0126       F5                            PUSH    AF              ; Save error code
0127       21 0188                       LD      HL,MSG02        ; Display error message
012A       CD 0159                       CALL    DSPMSG          ;
012D       F1                            POP     AF              ;
012E       CD 0143                       CALL    DSPHEX          ; Display error code by hexa code
0131       CD EB09                       CALL    CONIN           ;
0134       C3 EB03                       JP      WBOOT           ;
                                ;
                                ;        Subroutine 1 (Called when 1 track access)
                                ;
0137                            SUB1:
0137       3A FBAD                       LD      A,(TRACK)       ; Current track number
013A       CD 0143                       CALL    DSPHEX          ; Display it by hexa code
013D       0E 20                         LD      C,SPACE         ; Display space code
013F       CD EB0C                       CALL    CONOUT          ;
0142       C9                            RET                     ;
                                ;
                                ;        Display binary data by hexa code
                                ;
0143                            DSPHEX:
0143       F5                            PUSH    AF              ;
0144       0F                            RRCA                    ;
0145       0F                            RRCA                    ;
0146       0F                            RRCA                    ;
0147       0F                            RRCA                    ;
0148       CD 014C                       CALL    DSP20           ; Display upper 4 bits
014B       F1                            POP     AF              ;
014C                            DSP20:
014C       E6 0F                         AND     0FH             ;
014E       C6 90                         ADD     A,90H           ;
0150       27                            DAA                     ;
0151       CE 40                         ADC     A,40H           ;
0153       27                            DAA                     ;
```

```
0154    4F                      LD      C,A             ;
0155    CD 280C                 CALL    CONOUT          ;
0158    C9                      RET                     ;
                        ;
                        ;       Display strings until  find 00 code
                        ;
0159                    DSPMSG:
0159    7E                      LD      A,(HL)          ; Get data
015A    B7                      OR      A               ;
015B    C8                      RET     Z               ; Delimeter
015C    4F                      LD      C,A             ;
015D    E5                      PUSH    HL              ;
015E    CD 280C                 CALL    CONOUT          ;
0161    E1                      POP     HL              ;
0162    23                      INC     HL              ;
0163    18 F4                   JR      DSPMSG          ;
                        ;
                        ;       Display strings & wait until any key in
                        ;
0165                    DSPMSGWIN:
0165    CD 0159                 CALL    DSPMSG          ;
0165    CD 8809                 CALL    CONIN           ;
0168    C9                      RET                     ;
                        ;
                        ;       Message data
                        ;
016C                    MSG00:
016C    20 46 6F 72             DB      ' Formating',CR,LF,00
0170    6D 61 74 69
0174    6E 67 0D 0A
0178    00
0179                    MSG01:
0179    0D 0A                   DB      CR,LF
017B    20 43 6F 6D             DB      ' Completed',CR,LF,00
017F    70 6C 65 74
0183    65 64 0D 0A
0187    00
0188                    MSG02:
0188    0D 0A                   DB      CR,LF
018A    45 72 72 6F             DB      'Error --'
018E    72 20 2D 2D
                                DB      00
0192    00                      SUBTTL  MAIN ROUTINE

                        ;       ********************************
                        ;               Floppy disk utility
                        ;       ********************************
                        ;
                        ;       <> Entry parameter      <>
                        ;               None
                        ;       <> Return parameter     <>
                        ;               CY-flag : Return information
                        ;                         =0  : Normally end
                        ;                         =1  : Error happened
                        ;                               A reg. is error code
                        ;                                 =01 : Force stop
                        ;                                  61 : Device non-connect
                        ;                                  62 : Communication error
                        ;                                  63 : Time over
                        ;                                  64 : Force stop
                        ;                                  FA : Read error
                        ;                                  FB : Write error
                        ;                                  FC : Drive select error
                        ;                                  FD : Write protect error
                        ;                                  FE :
                        ;                                  FF :
                        ;       <> Preserved registers  <>
                        ;               BC,DE,HL
0193                    FDDUTY:
0193    E5                      PUSH    HL              ; Save registers
0194    D5                      PUSH    DE              ;
0195    C5                      PUSH    BC              ;
                        ;
                        ;       Reset disk system
                        ;
0196    CD 01FD                 CALL    RESET           ; Reset drive
0199    38 03                   JR      C,FDDUTY90      ; Error
                        ;
                        ;       Start of disk operations
                        ;
                        ;               If you want to copy diskette, you replace
                        ;               the statement 'CALL FORMAT' to 'CALL DISKCOPY'.
                        ;
019B    CD 01A2                 CALL    FORMAT          ; Do formating!!

019E                    FDDUTY90:
019E    C1                      POP     BC              ; Restore registers
019F    D1                      POP     DE              ;
01A0    E1                      POP     HL              ;
01A1    C9                      RET                     ;
```

```
                                        SUBTTL  FORMAT
                                ;       ***********************************
                                ;               Format D drive
                                ;       ***********************************
                                ;
                                ;       <> Entry parameter        <>
                                ;               None
                                ;       <> Return parameter       <>
                                ;               CY-flag : =0 -- Normal end
                                ;                         =1 -- Error happened
                                ;                                (A reg. is error code)
                                ;       <> Preserved registers    <>
                                ;               All without A reg.
                                ;       <> Note                   <>
                                ;               If you want to format other drive,
                                ;               you must change 'DAT+0' data.
01A2                            FORMAT:
01A2    E5                              PUSH    HL              ; Save registers
01A3    D5                              PUSH    DE              ;
01A4    C5                              PUSH    BC              ;
                                ;
                                ;       Make formating packet
                                ;
01A5    21 02C3                         LD      HL,PACKFT       ; Format packet address
01A8    CD 0242                         CALL    COPYA           ; Copy command data
                                ;
                                ;       Format command (Format first track)
                                ;
01AB                            XXXXXX:
01AB    AF                              XOR     A               ; Set initial track counter
01AC    32 F8AD                         LD      (TRACK),A       ;  (In formating, it isn't used)
01AF    CD 0219                         CALL    DINTPWALM       ; Disable power off & alarm interrupt
01B2    CD 02AD                         CALL    CALSUB1         ;
01B5    B7                              OR      A               ;
01B6    CD 026E                         CALL    SENDSLAVE       ; Send command to FDD & Get 0 track ACK

01B9    38 33                           JR      C,FORMAT90      ;  EPSP error
01BB    3A F8B5                         LD      A,(DAT+2)       ;
01BE    B7                              OR      A               ; Check return code
01BF    37                              SCF                     ;
01C0    20 2C                           JR      NZ,FORMAT90     ;  FDD error
                                ;
01C2    CD 0227                         CALL    EINTPW          ; Enable power off interrupt
01C5    38 27                           JR      C,FORMAT90      ;  Force stop
                                ;
                                ;       Track number 01..39 formating
                                ;
01C7    06 27                           LD      B,TRKMAX-1      ; Loop counter

01C9                            FORMAT10:
01C9    3A F8AD                         LD      A,(TRACK)       ; Set next access track number
01CC    3C                              INC     A               ;
01CD    32 F8AD                         LD      (TRACK),A       ;
01D0    CD 0219                         CALL    DINTPWALM       ; Disable interrupt (power off & alarm)
01D3    CD 02AD                         CALL    CALSUB1         ;
01D6    37                              SCF                     ;
01D7    CD 026E                         CALL    SENDSLAVE       ; Receive ACK code

01DA    38 12                           JR      C,FORMAT90      ;  EPSP error
01DC    3A F8B5                         LD      A,(DAT+2)       ;
01DF    B7                              OR      A               ; Check return code
01E0    37                              SCF                     ;
01E1    20 0B                           JR      NZ,FORMAT90     ;  FDD error

01E3    CD 0227                         CALL    EINTPW          ; Enable power off interrupt
01E6    38 06                           JR      C,FORMAT90      ;  Force stop
01E8    10 DF                           DJNZ    FORMAT10        ; Loop 39 times

01EA    CD 0250                         CALL    EINTALM         ; Enable alarm interrupt
01ED    B7                              OR      A               ; Normal return

01EE                            FORMAT90:
01EE    C1                              POP     BC              ; Restore registers
01EF    D1                              POP     DE              ;
01F0    E1                              POP     HL              ;
01F1    C9                              RET                     ;

                                        SUBTTL  DISKCOPY
```

```
                ;       ********************************
                ;               Copy from D drive to E drive
                ;       ********************************
                ;
                ;       <> Entry parameter      <>
                ;               None
                ;       <> Return parameter     <>
                ;               CY-flag : =0 -- Normal end
                ;                         =1 -- Error happened
                ;                              (A reg. is error code)
                ;       <> Preserved registers  <>
                ;               All without A reg.
                ;       <> Note                 <>
                ;               If you want to format other drive,
                ;               you must change 'DAT+0' data.
                ;
0172            DISKCOPY:
0172    E5              PUSH    HL              ; Save registers
0173    D5              PUSH    DE              ;
0174    C5              PUSH    BC              ;
                ;
                ;       Make volume copy command packet
                ;
0175    21 02C9         LD      HL,PACKCP       ; Volume copy command
0178    CD 02A2         CALL    COPYA           ;
                ;
                ;       Send volume copy command & Get answer
                ;
017B    18 AE           JR      XXXXXX          ; Common program with formating

                        SUBTTL  Reset disk drive
                ;       ********************************
                ;               Reset disk drive
                ;       ********************************
                ;
                ;       <> Entry parameter      <>
                ;               None
                ;       <> Return parameter     <>
                ;               CY-flag : =0 -- Normal end
                ;                         =1 -- Error happened
                ;                              (A reg. is error code)
                ;       <> Preserved registers  <>
                ;               All without A reg.
                ;
017D            RESET:
017D    E5              PUSH    HL              ; Save registers
017E    D5              PUSH    DE              ;
017F    C5              PUSH    BC              ;

0200    21 028D         LD      HL,PACKRS       ; Reset drive packet
0203    CD 02A2         CALL    COPYA           ;
                ;
                ;
                ;       Reset disk drive system subroutine
                ;
0206    B7              OR      A               ;
0207    CD 026E         CALL    SENDSLAVE       ; Send command to slave

020A    C1              POP     BC              ; Restore registers
020B    D1              POP     DE              ;
020C    E1              POP     HL              ;

020D    D8              RET     C               ;
020E    3A F883         LD      A,(DAT)         ; Check return code
0211    B7              OR      A               ;
0212    C8              RET     Z               ; Normal return
0213    37              SCF                     ;
0214    C9              RET                     ; Error return

                        SUBTTL  Other subroutines

                ;       ******************************************
                ;               Interrupt managements
                ;       ******************************************
                ;
                ;       Disable power-off & alarm interrupt
                ;
                ;       <> Entry parameter      <>
                ;               None
                ;       <> Return parameter     <>
                ;               None
                ;       <> Preserved registers  <>
                ;               All without A reg.
0215            DINTPWALM:
0215    E5              PUSH    HL              ;
0216    21 F084         LD      HL,YPOFDS       ; Disable power off interrupt
0219    CB F6           SET     6,(HL)          ;
021B    21 F086         LD      HL,YALMDS       ; Disable alarm interrupt
021E    CB F6           SET     6,(HL)          ;
0220    3E 01           LD      A,1             ; Disable flag on
0222    32 F8AC         LD      (DINTFLG),A     ;
0225    E1              POP     HL              ;
0226    C9              RET                     ;

                ;
```

```
                        ;       Enable power-off interrupt
                        ;
                        ;       <> Entry parameter        <>
                        ;               None
                        ;       <> Return parameter       <>
                        ;               CY-flag : Return information
                        ;                         =0 : Normal end
                        ;                         =1 : Force stop ( Areg. is 1)
                        ;       <> Preserved registers    <>
                        ;               All without A reg.
0227                    EINTPW:
0227    E5                      PUSH    HL                      ; Save registers
0228    D5                      PUSH    DE                      ;
0229    C5                      PUSH    BC                      ;

022A    21 F0B4                 LD      HL,YPOFDS               ; Enable power off
022D    CB B6                   RES     6,(HL)                  ;
                        ;
                        ;       Check power off interrupt
                        ;
022F    21 F0B5                 LD      HL,YPOFST               ;
0232    CB 76                   BIT     6,(HL)                  ; Check flag
0234    CB B6                   RES     6,(HL)                  ; Reset flag
0236    28 08                   JR      Z,EINT20                ;  Power-off not happened
                        ;
                        ;       Power off bu using PSTBIOS routine
                        ;
0238    CB FE                   SET     7,(HL)                  ; Follows  do power off
023A    CD 0250                 CALL    EINTALM                 ; Enable alarm interrupt
023D    CD FF96                 CALL    JPSTBIOS                ; Do power off !!!
                        ;
                        ;       Check force-stop
                        ;
0240                    EINT20:
0240    3A F220                 LD      A,(CSTOPFLG)            ; Check force-stop flag
0243    B7                      OR      A                       ;
0244    28 06                   JR      Z,EINT90                ; Not force-stop.
                        ;
0246    CD 0250                 CALL    EINTALM                 ; Enable alarm interrupt
0249    3E 01                   LD      A,01                    ; Force-stop error code
024B    37                      SCF                             ;

024C                    EINT90:
024C    C1                      POP     BC                      ; Restore registers
024D    D1                      POP     DE                      ;
024E    E1                      POP     HL                      ;
024F    C9                      RET                             ;
                        ;
                        ;       Enable alarm interrupt
                        ;
                        ;       <> Entry parameter        <>
                        ;               None
                        ;       <> Return parameter       <>
                        ;               None
                        ;       <> Preserved registers    <>
                        ;               All without A reg.
0250                    EINTALM:
0250    E5                      PUSH    HL                      ; Save registers
0251    D5                      PUSH    DE                      ;
0252    C5                      PUSH    BC                      ;

0253    21 F0B8                 LD      HL,YALMDS               ; Enable alarm interrupt
0256    CB B6                   RES     6,(HL)                  ;
                        ;
                        ;       Check alarm interrupt
                        ;
0258    21 F0B7                 LD      HL,YALMST               ;
025B    CB 76                   BIT     6,(HL)                  ; Check alarm
025D    CB B6                   RES     6,(HL)                  ; Reset flag
025F    28 05                   JR      Z,EINTA90               ; Not happened alarm interrupt
                        ;
                        ;       Alarm screen display
                        ;
0261    CB FE                   SET     7,(HL)                  ;
0263    CD FF96                 CALL    JPSTBIOS                ; Do alarm

0266                    EINTA90:
0266    AF                      XOR     A                       ; Disable flag off
0267    32 F8AC                 LD      (DINTFLG),A             ;

026A    C1                      POP     BC                      ; Restore registers
026B    D1                      POP     DE                      ;
026C    E1                      POP     HL                      ;
026D    C9                      RET                             ;
```

```
                        ;       ****************************************
                        ;                  Send command to TF
                        ;       ****************************************
                        ;
                        ;       <> Entry parameter         <>
                        ;                  CY-flag : Command type
                        ;                            =0  : Send command & receive data
                        ;                            =1  : Only receive data
                        ;       <> Return parameter        <>
                        ;                  CY-flag : Return parameter
                        ;                            =0 : Normal return
                        ;                            =1 : Error
                        ;                                 (A reg. is error code)
                        ;       <> Preserved registers     <>
                        ;                  All without A reg.
                        ;
026E                    SENDSLAVE:
026E    DD E5                   PUSH    IX                  ; Save registers
0270    E5                      PUSH    HL                  ;
0271    D5                      PUSH    DE                  ;
0272    C5                      PUSH    BC                  ;

0273    DD 21 0022              LD      IX,EPSPRCV          ;
0277    38 06                   JR      C,SENDS50           ; Use <receive ACK only routine> !!

0279    3E 01                   LD      A,1                 ; Send and receive data
027B    DD 21 001F              LD      IX,EPSPSND          ;

027F                    SENDS50:
027F    21 F41B                 LD      HL,DISBNK           ; Set bank to OS ROM
0282    36 FF                   LD      (HL),-1             ;
0284    21 FBAE                 LD      HL,PACKET           ; Communication buffer address
0287    CD EB69                 CALL    CALLX               ;
028A    B7                      OR      A                   ;
028B    28 0F                   JR      Z,SENDS90           ; Normal end
                        ;
                        ;       EPSP error
                        ;
028D    F5                      PUSH    AF                  ; Save return code
028E    3A FBAC                 LD      A,(DINTFLG)         ; Check interrupt disable ?
0291    B7                      OR      A                   ;
0292    28 06                   JR      Z,SENDS80           ;

0294    CD 0227                 CALL    EINTPW              ; Enable power off interrupt
0297    CD 0250                 CALL    EINTALM             ; Enable alarm interrupt
029A                    SENDS80:
029A    F1                      POP     AF                  ;
029B    37                      SCF                         ; Error return

029C                    SENDS90:
029C    C1                      POP     BC                  ; Restore registers
029D    D1                      POP     DE                  ;
029E    E1                      POP     HL                  ;
029F    DD E1                   POP     IX                  ;
02A1    C9                      RET                         ;
                        ;       ****************************************
                        ;             Copy data into command buffer
                        ;       ****************************************
                        ;
                        ;       <> Entry parameter         <>
                        ;                  HL reg. : Source address
                        ;       <> Return parameter        <>
                        ;                  None
                        ;       <> Preserved registers     <>
                        ;                  BC
                        ;
02A2                    COPYA:
02A2    C5                      PUSH    BC                  ;
02A3    11 FBAE                 LD      DE,FMT              ; Command buffer address
02A6    01 0006                 LD      BC,0006             ; Copy length
02A9    ED B0                   LDIR                        ; Copy
02AB    C1                      POP     BC                  ;
02AC    C9                      RET                         ;

                        ;       ****************************************
                        ;                  Subroutine call
                        ;       ****************************************
                        ;
                        ;       <> Entry parameter         <>
                        ;                  None
                        ;       <> Return parameter        <>
                        ;                  None
                        ;       <> Preserved registers     <>
                        ;                  BC,DE,HL
                        ;
02AD                    CALSUB1:
02AD    E5                      PUSH    HL                  ; Save registers
02AE    D5                      PUSH    DE                  ;
02AF    C5                      PUSH    BC                  ;

02B0    2A FBAA                 LD      HL,(CALLAD1)        ; Execute address
02B3                    CALSUB10:
02B3    E5                      PUSH    HL                  ;
02B4    21 02B9                 LD      HL,CALSUB20         ; Return address
02B7    E3                      EX      (SP),HL             ; Return addr --> (SP)
```

```
02B8    E9                      JP      (HL)            ; Go subroutine

02B9                    CALSUB20:
02B9    C1                      POP     BC              ; Restore registers
02BA    D1                      POP     DE              ;
02BB    E1                      POP     HL              ;
02BC    C9                      RET                     ;

                                SUBTTL  COMMENT DATA AREA
                        ;
                        ;
                        ; COMMUNICATION  DATA & WORK
                        ;
                        ;       <> Slave send packet data <>
                        ;
02BD                    PACKRS:                         ; Reset
02BD    00 31 23 0D             DB      00h,31h,23h,0Dh,00h,00h
02C1    00 00

02C3                    PACKFT:                         ; Format (01h means D drive)
02C3    00 31 23 7C             DB      00h,31h,23h,7Ch,00h,01h
02C7    00 01

02C9                    PACKCP:                         ; Copy (01h means from D drive to E drive)

02C9    00 31 23 7A             DB      00h,31h,23h,7Ah,00h,01h
02CD    00 01

                                END

Macros:

Symbols:
FC3C    BOT             FBAA    CALLAD1         EB69    CALLX
02AD    CALSUB1         02B3    CALSUB10        02B9    CALSUB20
EB09    CONIN           EB0C    CONOUT          02A2    COPYA
000D    CR              F220    CSTOPFLG        FBB3    DAT
FBAF    DID             FBAC    DINTFLG         0215    D.NTPWALM
F41[illegible]    DISBNK          01F[illegible]    DISKCOPY        014C    DSP20
0143    DSPHEX          0159    DSPMSG          0165    DSPMSGSIN
024[illegible]    EINT20          024C    EINT90          0266    EINTA90
0250    EINTALM         0227    EINTPW          0022    EPSPRCV
0[illegible]1F    EPSPSND         0126    ERROR           0193    FDDUTY
019E    FDDUTY90        FBAE    FMT             FBB1    FNC
01A2    FORMAT          01C9    FORMAT10        01EE    FORMAT90
FF96    JPSTBIOS        000A    LF              016C    MSG00
0179    MSG01           0188    MSG02           02C9    PACKCP
FBAE    PACKET          02C3    PACKFT          02BD    PACKRS
EB6F    PUTPFK          01FD    RESET           027F    SENDS50
0294    SENDS80         029C    SENDS90         0268    SENDSLAVE
FBB0    SID             FBB2    SIZ             0020    SPACE
1000    STACK           0100    START           0137    SUB1
FBAD    TRACK           0028    TRKMAX          EB03    WBOOT
FBAA    WKSTART         01AB    XXXXXX          F0B6    YALMDS
F0B7    YALMST          F0B4    YPOFDS          F0B5    YPOFST


No Fatal error(s)
```