



```

BS      TITLE  UHR.SYS - REAL-TIME CLOCKDRIVER

        PAGE   70,115

;       Date 24-Jul-84
  
```

```

= 0001  Y      =      1
        IRP    X,(0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15)
        BIT&X =      Y
        Y      =      Y SHL 1
        ENDM
  
```

```

;-----
; Additional EQUates for alphaTronic CO-Processor
;-----
  
```

```

= 0000  FNERR  EQU    0          ;Function # to stop IOCS-85

= 0001  CSTAT  EQU    1          ;Function # for Console Status request
= 0002  CONIN  EQU    2          ;Function # for Console Input one character
= 0003  COOUT  EQU    3          ;Function # for Console Output one Character

= 0004  LPSTS  EQU    4          ;Function # for Printer Status request
= 0005  LPOUT  EQU    5          ;Function # for Printer Output one character

= 0006  AUXSTS EQU    6          ;Function # for Auxillary Port Input Status
= 0007  AUXIN  EQU    7          ;Function # for Auxillary Port Input one Char
= 0008  AUXOUT EQU    8          ;Function # for Auxillary Port Output one Char

= 0009  READ   EQU    9          ;Function # for read one sector from Disk
= 000A  WRITE  EQU   10          ;Function # for write one sector to Disk

= 000B  FORMAT EQU   11          ;Function # for format one Track on Disk

= 000C  SETKEY EQU   12          ;Function # for setting codes to FN. Keys
= 000D  SETSIO EQU   13          ;Function # for setting AUX Parameters
= 000E  SETPRN EQU   14          ;Function # for setting printer Baudrate

= 0015  IUREAD EQU   21          ;Function # for reading Clock
= 0016  IUWRIT EQU   22          ;Function # for writing Clock

;-----
= FFEA  PIOOUT EQU   OFFEAH      ;Port to output one Byte to the 8085
= FFEA  PIOIN  EQU   OFFEAH      ;Port to input one Byte from the 8085

= FFE9  PIOSTS EQU   OFFE9H      ;Port for Bufferflags (IBF & OBF)
;OBF is connected to TEST Input of the CPU,
;IBF is connected to INT2 of PIC 8259A

;-----
= FFE4  TIMERO EQU   OFFE4H
= FFE5  TIMER1 EQU   OFFE5H      ;CLK=100KHz, OUT1=IR1
= FFE6  TIMER2 EQU   OFFE6H      ;CLK=100KHz, OUT2=IR2
  
```

```
=  
= 0030 RLLSB EQU BIT4  
RLLMSB EQU BIT4+BIT5  
  
= 0000 MODE0 EQU 0 ;Int. on zero-count  
= MODE1 EQU BIT1 ;prog. Monoflop  
= MODE2 EQU BIT2 ;synch. divider by n  
= 0006 MODE3 EQU BIT1+BIT2 ;square wave generator  
= MODE4 EQU BIT3 ;software controlled strobe  
= 000A MODE5 EQU BIT3+BIT1 ;hardware controlled strobe  
  
= BCD EQU BIT0 ;4 decade BCD-counter  
= 0000 BIN EQU 0 ;16 bit binary counter
```

-----  
;

SUBTTL Device driver tables

Device driver tables

```

0000 PAGE
      CODE SEGMENT

      ASSUME CS:CODE,DS:NOTHING,ES:NOTHING,SS:NOTHING

0000 ORG 0 ;Starts at an offset of zero

;-----+
;   DWORD pointer to next device      | 1 word offset
;   (-1,-1 if last device)           | 1 word segment
;-----+
;   Device attribute WORD              | 1 word
;   Bit 15 = 1 for character devices.
;   0 for Block devices.
;
;   Character devices. (Bit 15=1)
;   Bit 0 = 1 current sti device.
;   Bit 1 = 1 current sto device.
;   Bit 2 = 1 current NUL device.
;   Bit 3 = 1 current Clock device.
;
;   Bit 13 = 1 for non IBM machines.
;   0 for IBM machines only.
;   Bit 14 = 1 IOCTL control bit.
;-----+
;   Device strategy pointer.           | 1 word offset
;-----+
;   Device interrupt pointer.         | 1 word offset
;-----+
;   Device name field.                | 8 bytes
;   Character devices are any valid name
;   left justified, in a space filled
;   field.
;   Block devices contain # of units in
;   the first byte.
;-----+

```

```

0000 TIMDEV: ;Header for device CLOCK
0000 FFFF FFFF DW -1,-1
0004 8008 DW BIT15+BIT3 ;8008H
0006 0030 R DW STRATEGY
0008 003B R DW TIM_INT
000A 42 53 2D 43 4C 4F DB "BS-CLOCK"
      43 4B

```

SUBTTL Dispatch tables for each device

Dispatch tables for each device

PAGE

0012	0223	R	TIMTBL:	DW	CL_INI	!0	- Init. (Not used)
0014	007B	R		DW	EXIT	!1	- Media check (Not used)
0016	007B	R		DW	EXIT	!2	- Get Bios Parameter Block (Not used)
0018	0074	R		DW	CMDERR	!3	- Reserved. (Currently ret. an error)
001A	00D8	R		DW	TIM_RED	!4	- Character read. (Destructive)
001C	0070	R		DW	BUS_EXIT	!5	- (Not used, returns busy flag.)
001E	007B	R		DW	EXIT	!6	- Return status. (Not used)
0020	007B	R		DW	EXIT	!7	- Flush Input buffer. (Not used)
0022	00BA	R		DW	TIM_WRT	!8	- Character write
0024	00BA	R		DW	TIM_WRT	!9	- Character write with verify
0026	007B	R		DW	EXIT	!10	- Character write status. (Not used)
0028	007B	R		DW	EXIT	!11	- Flush output buffer. (Not used)
002A	007B	R		DW	EXIT	!12	- IO Control

SUBTTL Strategy and Software Interrupt routines

Strategy and Software Interrupt routines

PAGE

; Define offsets for io data packet

```
0000 ??          IODAT  STRUC
0001 ??          CMDLEN DB      ?          ;LENGTH OF THIS COMMAND
0002 ??          UNIT   DB      ?          ;SUB UNIT SPECIFIER
0003 ????       CMD     DB      ?          ;COMMAND CODE
0005      08 [   STATUS  DW      ?          ;STATUS
                ??
                ]
000D ??          MEDIA  DB      ?          ;MEDIA DESCRIPTOR
000E ?????????? TRANS  DD      ?          ;TRANSFER ADDRESS
0012 ????       COUNT  DW      ?          ;COUNT OF BLOCKS OR CHARACTERS
0014 ????       START  DW      ?          ;FIRST BLOCK TO TRANSFER
0016
002C 00 00 00 00 PTRSAV DD      0          ;Strategy pointer save
```

```
-----
; Simplistic Strategy routine for non-multi-Tasking system
;
; Currently just saves I/O packet pointers in PTRSAV for
; later processing by the individual interrupt routines
;
```

```
0030          STRATP  PROC      FAR
0030          STRATEGY:
0030 2E: 89 1E 002C R      MOV     WORD PTR CS:[PTRSAV],BX
0035 2E: 8C 06 002E R      MOV     WORD PTR CS:[PTRSAV+2],ES
003A CB                  RET
003B          STRATP  ENDP
```

Strategy and Software Interrupt routines

PAGE

-----  
 ; Clock interrupt routine for processing I/O packets  
 ;

```

003B      TIM_INT:
003B      54          PUSH     SI
003C      BE 0012 R   MOV      SI,OFFSET TIMTBL
003F      EB 00      JMP      SHORT ENTRY
  
```

-----  
 ; Common program for handling the simplistic I/O packet  
 ; processing scheme in MSDOS 2.0  
 ;

```

0041      50          ENTRY:  PUSH    AX           ;Save all nessacary registers
0042      51          PUSH    CX
0043      52          PUSH    DX
0044      57          PUSH    DI
0045      55          PUSH    BP
0046      1E          PUSH    DS
0047      06          PUSH    ES
0048      53          PUSH    BX

0049      2E: C5 1E 002C R  LDS     BX,CS:[PTRSAV] ;Retrieve pointer to I/O Packet
004E      8A 47 01      MOV     AL,[BX.UNIT]   ;AL = Unit code
0051      8A 67 0D      MOV     AH,[BX.MEDIA] ;AH = Media descriptor
0054      8B 4F 12      MOV     CX,[BX.COUNT] ;CX = Contains byte/sector count
0057      8B 57 14      MOV     DX,[BX.START] ;DX = Starting Logical sector

005A      97          XCHG   DI,AX           ;Move Unit & Media into DI temporarily
005B      8A 47 02      MOV     AL,[BX.CMD]   ;Retrieve Command type. (1 => 11)
005E      32 E4          XOR     AH,AH         ;Clear upper half of AX for calculation
0060      03 F0          ADD     SI,AX         ;Compute entry pointer in dispatch table
0062      03 F0          ADD     SI,AX
0064      3C 0B          CMP     AL,11        ;Verify that not more than 11 commands
0066      77 0C          JA     CMDERR        ;Ah; well; error out

0068      97          XCHG   AX,DI           ;Move Unit & Media back where they belong
0069      C4 7F 0E      LES     DI,[BX.TRANS] ;DI contains address of Transfer address
                                ;ES contains segment

006C      0E          PUSH    CS
006D      1F          POP     DS           ;Data segment same as Code segment
006E      FF 24          JMP     [SI]         ;Perform I/O packet command
  
```

SUBTTL Common error and exit points

Common error and exit points

PAGE

```

0070          BUS_EXIT:          ;Device busy exit
0070 B4 03      MOV      AH,00000011B ;Set busy and done bits
0072 EB 09      JMP      SHORT EXIT1
;
0074 B0 03      CMDERR: MOV      AL,3      ;Set unknown command error #
;
; Common error processing routine
; AL contains actual error code
;
; Error # 0 = Write Protect violation
;          1 = Unknown unit
;          2 = Drive not ready
;          3 = Unknown command in I/O packet
;          4 = CRC error
;          5 = Bad drive request structure length
;          6 = Seek error
;          7 = Unknown media discovered
;          8 = Sector not found
;          9 = Printer out of paper
;         10 = Write fault
;         11 = Read fault
;         12 = General failure
;
0076          ERR_EXIT:
0076 B4 81      MOV      AH,10000001B ;Set error and done bits
0078 F9        STC      ;Set carry bit also
0079 EB 02      JMP      SHORT EXIT1 ;Quick way out
;
007B          EXITP  PROC  FAR      ;Normal exit for device drivers
;
007B B4 01      EXIT:  MOV      AH,00000001B ;Set done bit for MSDOS
007D 2E: C5 1E 002C R EXIT1: LDS  BX,CS:[PTRSAV]
0082 89 47 03      MOV      [BX.STATUS],AX ;Save operation complete and status
;
0085 5B        POP      BX          ;Restore registers
0086 07        POP      ES
0087 1F        POP      DS
0088 5D        POP      BP
0089 5F        POP      DI
008A 5A        POP      DX
008B 59        POP      CX
008C 58        POP      AX
008D 5E        POP      SI
008E CB        RET
;
008F          EXITP  ENDP

```

SUBTTL Date/Time Routines

Date/Time Routines

PAGE

008F	0658	TIM_DAYS:	DW	1624	!Number of days since 1-1-80
0091	38	TIM_MINS:	DB	56	!Minutes
0092	17	TIM_HRS:	DB	23	!Hours
0093	00	TIM_HSEC:	DB	0	!Hundreths of a second
0094	00	TIM_SECS:	DB	0	!Seconds

0095	OD [	UBUF:	DB	13 DUP (?)	
	??				
	]				

00A2		MO_D_TAB:			
00A2	0000 001F 003B 005A		DW	0,31,59,90,120,151	
	0078 0097				
00AE	00B5 00D4 00F3 0111		DW	181,212,243,273,304,334	
	0130 014E				

-----  
 ! Time write routine  
 !

00BA		TIM_WRT:			
00BA	1E		PUSH	DS	
00BB	06		PUSH	ES	
00BC	BE 008F R		MOV	SI,OFFSET TIM_DAYS	
00BF	87 F7		XCHG	SI,DI	
00C1	06		PUSH	ES	
00C2	8C D8		MOV	AX,DS	
00C4	1F		POP	DS	
00C5	8E C0		MOV	ES,AX	
00C7	B9 0006		MOV	CX,6	
00CA	F3/ A4		REP	MOVSB	
00CC	07		POP	ES	
00CD	1F		POP	DS	
00CE	E8 01C9 R		CALL	UTNAW	
00D1	E8 01B1 R		CALL	UPUT	
00D4	B0 00		MOV	AL,0	
00D6	EB A3		JMP	EXIT	

-----  
 ! Time read routine  
 !

00D8		TIM_RED:			
00D8	E8 00ED R		CALL	UGET	
00DB	E8 0107 R		CALL	UTWAN	
00DE	E8 013E R		CALL	UDWAN	
00E1	BE 008F R		MOV	SI,OFFSET TIM_DAYS	



Date/Time Routines

```

00F2 53          PUSH    BX
00F3 51          PUSH    CX
00F4 BB 0095 R    MOV     BX,OFFSET UBUF
00F7 B9 000D     MOV     CX,13
00FA          UGETL:
00FA E8 0216 R    CALL   INDAT
00FD 24 0F        AND     AL,0FH
00FF 88 07        MOV     [BX],AL
0101 43          INC     BX
0102 E2 F6        LOOP   UGETL

0104 59          POP     CX
0105 5B          POP     BX
0106 C3          RET

0107          UTWAN:
0107 53          PUSH    BX
0108 BB 0091 R    MOV     BX,OFFSET TIM_MINS
010B 51          PUSH    CX
010C B9 000A     MOV     CX,10
010F 2E: A0 0098 R  MOV     AL,BYTE PTR UBUF+3
0113 F6 E1        MUL     CL
0115 2E: 02 06 0097 R  ADD     AL,BYTE PTR UBUF+2
011A 88 07        MOV     [BX],AL
011C 43          INC     BX
011D 2E: A0 009A R  MOV     AL,BYTE PTR UBUF+5
0121 24 03        AND     AL,3
0123 F6 E1        MUL     CL
0125 2E: 02 06 0099 R  ADD     AL,BYTE PTR UBUF+4
012A 88 07        MOV     [BX],AL
012C 43          INC     BX
012D 43          INC     BX
012E 2E: A0 0096 R  MOV     AL,BYTE PTR UBUF+1
0132 F6 E1        MUL     CL
0134 2E: 02 06 0095 R  ADD     AL,BYTE PTR UBUF
0139 88 07        MOV     [BX],AL
013B 59          POP     CX
013C 5B          POP     BX
013D C3          RET

013E          UDWAN:
013E 53          PUSH    BX
013F 51          PUSH    CX
0140 52          PUSH    DX

0141 B1 0A        MOV     CL,10
0143 2E: A0 00A1 R  MOV     AL,BYTE PTR UBUF+12
0147 F6 E1        MUL     CL
0149 2E: 02 06 00A0 R  ADD     AL,BYTE PTR UBUF+11
014E 2C 50        SUB     AL,80
0150 32 E4        XOR     AH,AH
    
```

Date/Time Routines

```

0164 8A D8          MOV     BL,AL
0166 32 FF          XOR     BH,BH
0168 58             POP     AX
0169 03 C3          ADD     AX,BX
016B 50             PUSH    AX
016C B1 0A          MOV     CL,10
016E 2E: A0 009F R  MOV     AL,BYTE PTR UBUF+10
0172 F6 E1          MUL     CL
0174 2E: 02 06 009E R ADD     AL,BYTE PTR UBUF+9
0179 FE C8          DEC     AL
017B 8A C8          MOV     CL,AL
017D 32 E4          XOR     AH,AH
017F 03 C0          ADD     AX,AX
0181 05 00A2 R      ADD     AX,OFFSET MO_D_TAB
0184 8B D8          MOV     BX,AX
0186 8B 07          MOV     AX,[BX]
0188 5B             POP     BX
0189 03 C3          ADD     AX,BX
018B 80 F9 03       CMP     CL,3
018E 7C 06          JL     NO_SCHALT
0190 80 FD 00       CMP     CH,0
0193 75 01          JNE    NO_SCHALT
0195 40             INC     AX
0196              NO_SCHALT:
0196 50             PUSH    AX
0197 B1 0A          MOV     CL,10
0199 2E: A0 009D R  MOV     AL,BYTE PTR UBUF+8
019D F6 E1          MUL     CL
019F 2E: 02 06 009C R ADD     AL,BYTE PTR UBUF+7
01A4 48             DEC     AX
01A5 5B             POP     BX
01A6 03 C3          ADD     AX,BX

01A8 BB 008F R      MOV     BX,OFFSET TIM_DAYS
01AB 89 07          MOV     [BX],AX

01AD 5A             POP     DX
01AE 59             POP     CX
01AF 5B             POP     BX
01B0 C3             RET

01B1              UPUT:
01B1 B0 16          MOV     AL,IUWRIT
01B3 E8 020B R      CALL    OUTDAT

01B6 53             PUSH    BX
01B7 51             PUSH    CX
01B8 BB 0095 R      MOV     BX,OFFSET UBUF
01BB B9 000D       MOV     CX,13
01BE              UPUTL:
01BE 8A 07          MOV     AL,[BX]
;              OR     AL,30H
01C0 E8 020B R      CALL    OUTDAT
    
```

Date/Time Routines

```
01C9          UTNAW:
01C9 53          PUSH    BX
01CA BB 0091 R   MOV     BX,OFFSET TIM_MINS
01CD 51          PUSH    CX
01CE B9 000A     MOV     CX,10
01D1 8A 07     MOV     AL,[BX]
01D3 32 E4     XOR     AH,AH
01D5 43          INC     BX
01D6 F6 F1     DIV     CL
01D8 2E: A2 0098 R MOV     BYTE PTR UBUF+3,AL
01DC 2E: 88 26 0097 R MOV     BYTE PTR UBUF+2,AH
01E1 8A 07     MOV     AL,[BX]
01E3 43          INC     BX
01E4 43          INC     BX
01E5 32 E4     XOR     AH,AH
01E7 F6 F1     DIV     CL
01E9 0C 08     OR      AL,8
01EB 2E: A2 009A R MOV     BYTE PTR UBUF+5,AL
01EF 2E: 88 26 0099 R MOV     BYTE PTR UBUF+4,AH
01F4 8A 07     MOV     AL,[BX]
01F6 32 E4     XOR     AH,AH
01F8 F6 F1     DIV     CL
01FA 2E: A2 0096 R MOV     BYTE PTR UBUF+1,AL
01FE 2E: 88 26 0095 R MOV     BYTE PTR UBUF,AH
0203 59          POP     CX
0204 5B          POP     BX
0205 C3          RET
```

SUBTTL Interrupt Routines and IOCS-Call

Interrupt Routines and IOCS-Call

PAGE

```

0206 E8 020B R      SENCHR: CALL   OUTDAT      ;send command
0209 8A C1          MOV     AL,CL
020B 52             OUTDAT: PUSH   DX
020C BA FFEA       MOV     DX,PIOUT
020F 9B            WAIT          ;for OBF = 0 (1)
0210 EE            OUT     DX,AL
0211 5A            POP     DX
0212 C3            RET
  
```

-----

```

0213 E8 020B R      OUTIN:  CALL   OUTDAT
0216 52             INDAT:  PUSH   DX
0217 BA FFE9       MOV     DX,PIOSTS
021A EC            INDAT1: IN     AL,DX
021B A8 01         TEST    AL,1
021D 75 FB         JNZ    INDAT1
021F 42             INC     DX
0220 EC            IN     AL,DX
0221 5A            POP     DX
0222 C3            RET
  
```

-----

```

0223              CL_INI:
0223 2E: C5 1E 002C R  LDS     BX,CS:[PTRSAV]
0228 C6 47 0D 01     MOV     BYTE PTR [BX.MEDIA],1
022C C7 47 0E 0223 R MOV     WORD PTR [BX.TRANS],OFFSET CL_INI
0231 8C 4F 10       MOV     WORD PTR [BX.TRANS+2],CS
  
```

```

0234 0E             PUSH   CS
0235 1F             POP    DS
0236 E8 028C R      CALL   PRINT
0239 0D 0A         DB     13,10
023B 43 6C 6F 63 6B 2D DB     'Clock-Driver EZU8607 V 1.0 - BS '
023B 44 72 69 76 65 72
023B 20 45 5A 55 38 36
023B 30 37 20 20 20 20
023B 56 20 31 2E 30 20
023B 20 20 20 2D 20 42
023B 53 20 20 20 20
  
```

```

0264 20 20 20 20 20 20 DB     '                24-Jul-84'
0264 20 20 20 20 20 20
0264 20 20 20 20 20 20
0264 20 20 20 20 20 20
0264 20 32 34 2D 4A 75
0264 6C 2D 38 34
  
```

```

0286 0D 0A FF       DB     13,10,-1
  
```

```

0289 E9 007B R      JMP    EXIT
  
```

```

028C 8B EC          PRINT: MOV    BP,SP
  
```

Interrupt Routines and IOCS-Call

029F 87 5E 00  
02A2 C3

XCHG BX,[BP]  
RET

02A3  
02A3 8A C8  
02A5 B0 03  
02A7 E8 020B R  
02AA 8A C1  
02AC E8 020B R  
02AF C3

OUTCHR:

MOV CL,AL  
MOV AL,COOUT  
CALL OUTDAT  
MOV AL,CL  
CALL OUTDAT  
RET

02B0

CODE ENDS  
END

Structures and records:

Name	Width Shift	# fields Width Mask	Initial
IODAT. . . . .	0016	0009	
CMDLEN . . . . .	0000		
UNIT . . . . .	0001		
CMD. . . . .	0002		
STATUS . . . . .	0003		
MEDIA. . . . .	000D		
TRANS. . . . .	000E		
COUNT. . . . .	0012		
START. . . . .	0014		

Segments and groups:

Name	Size	align	combine class
CODE . . . . .	02B0	PARA	NONE

Symbols:

Name	Type	Value	Attr
AUXIN. . . . .	Number	0007	
AUXOUT . . . . .	Number	0008	
AUXSTS . . . . .	Number	0006	
BCD. . . . .	Alias	BIT0	
BIN. . . . .	Number	0000	
BIT0 . . . . .	Number	0001	
BIT1 . . . . .	Number	0002	
BIT10. . . . .	Number	0400	
BIT11. . . . .	Number	0800	
BIT12. . . . .	Number	1000	
BIT13. . . . .	Number	2000	
BIT14. . . . .	Number	4000	
BIT15. . . . .	Number	8000	
BIT2 . . . . .	Number	0004	
BIT3 . . . . .	Number	0008	
BIT4 . . . . .	Number	0010	
BIT5 . . . . .	Number	0020	
BIT6 . . . . .	Number	0040	
BIT7 . . . . .	Number	0080	
BIT8 . . . . .	Number	0100	
BIT9 . . . . .	Number	0200	
BUS_EXIT . . . . .	L NEAR	0070	CODE
CL_INI . . . . .	L NEAR	0223	CODE
CMDERR . . . . .	L NEAR	0074	CODE
CONIN. . . . .	Number	0002	
COOUT. . . . .	Number	0003	
CSTAT. . . . .	Number	0001	
ENTRY. . . . .	L NEAR	0041	CODE
ERR_EXIT . . . . .	L NEAR	0076	CODE
EXIT . . . . .	L NEAR	007B	CODE
EXIT1. . . . .	L NEAR	007D	CODE