```
                                TITLE   IO.SYS for the alphaTronic P30

                                PAGE    70,115

                                ;       Date 02-Nov-83

                                ;       I/O system for Version 2.x of MSDOS

                                ;       This BIOS designed to be linked with the SYSINIT
                                ;       and SYSIMES module provided by Microsoft

 = 1000                         BIOSIZ  EQU     4096            ;Size of BIOS in bytes (incl. SYSINIT+SYSIMES)
 = 0100                         BIOSIZS EQU     BIOSIZ / 16     ;Size of BIOS in Paragraphs
 = 0000                         ANSI    EQU     0               ;Ansi switch
 = 0001                         DPBRD   EQU     1               ;set to '1', if not read DPB from Disk

                                ;       Things needed to communicate with SYSINIT

                                EXTRN   SYSINIT:FAR                     ;The entry point of SYSINIT
                                EXTRN   CURRENT_DOS_LOCATION:WORD       ;Where the DOS is when SYSINIT called
                                EXTRN   FINAL_DOS_LOCATION:WORD         ;Where I want SYSINIT to put the DOS
                                EXTRN   DEVICE_LIST:DWORD               ;Pointer to the DEVICE list
                                EXTRN   MEMORY_SIZE:WORD                ;Size in paragraphs of Physical memory
                                EXTRN   DEFAULT_DRIVE:BYTE              ;Def. Drive to use when system booted
                                EXTRN   BUFFERS:BYTE                    ;Number of default buffers
                                                                        ;Leave as is and SYSINIT uses only 2
                                PUBLIC  RE_INIT

                                ;               Link the Object File with SYSINIT.OBJ & SYSIMES.OBJ !!!


 = 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
 = FFEA                         PIOIN   EQU     0FFEAH          ;Port to input  one Byte from the 8085
```

```
 = FFE9                         PIOSTS  EQU     0FFE9H          ;Port for Bufferflags (IBF & OBF)
                                                                ;OBF is connected to TEST Input of the CPU,
                                                                ;IBF is connected to INT2 of PIC 8259A
                                ;------------------------------------------------------------------------
 = FFE0                         PICR0   EQU     0FFE0H          ;ICW1, OCW2, OCW3
 = FFE1                         PICR1   EQU     0FFE1H          ;ICW2, ICW3, ICW4, OCW1
                                                                ;Int. Controller Port Address

                                ;               Bits in ICW1
 =                              LTIM    EQU     BIT3            ;Level trigg. = 1, Edge = 0
 =                              SNGL    EQU     BIT1            ;Single = 1, Cascade Mode = 0
 =                              ICW4    EQU     BIT0            ;ICW4 needed = 1, no ICW4 needed = 0

                                ;               Bits in ICW2
                                ;               set 5 MSB's of INT Vector as ICW2

                                ;               Bits in ICW3
                                ;               **** only in Slave Mode ****

                                ;               Bits in ICW 4
 =                              SNFM    EQU     BIT4            ;special fully nested mode = 1
                                                                ;not special fully nested mode = 0
 =                              BUF     EQU     BIT3            ;0 X non buffered Mode
 =                              MS      EQU     BIT2            ;1 0 buffered Mode Slave
                                                                ;1 1 buffered Mode Master
 =                              AEOI    EQU     BIT1            ;Auto EOI = 1, normal EOI = 0
 =                              UPM     EQU     BIT0            ;8086/8088 = 1, 8085 = 0

 =                              EOI     EQU     BIT5            ;non specified End-of-Interrupt
                                ;------------------------------------------------------------------------
 = FFE4                         TIMER0  EQU     0FFE4H
 = FFE5                         TIMER1  EQU     0FFE5H          ;CLK=100KHz, OUT1=IR1
 = FFE6                         TIMER2  EQU     0FFE6H          ;CLK=100KHz, OUT2=IR2
 = FFE7                         TIMCMD  EQU     0FFE7H          ;Timer controll Register

                                ;       Steuerwort Format
 = 0000                         SEL0    EQU     0 SHL 6
 = 0040                         SEL1    EQU     1 SHL 6
 = 0080                         SEL2    EQU     2 SHL 6

 = 0000                         LATCH   EQU     0
 =                              RLMSB   EQU     BIT5
 =                              RLLSB   EQU     BIT4
 = 0030                         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
                                ;------------------------------------------------------------------------
 = 0100                         BNKSIZ  EQU     100H            ;# of 64k Banks (left here by BOOT EPROM)
                                ;------------------------------------------------------------------------

                                SUBTTL  Device driver tables
```

```
                              Device driver tables

                              PAGE
 0000                         CODE    SEGMENT BYTE PUBLIC

                              ASSUME  CS:CODE,DS:CODE,ES:CODE,SS:CODE

                              PUBLIC  INIT

 0000                                 ORG     0               ;Starts at an offset of zero

 0000  E9 04A0 R              INIT:   JMP     HWINIT

                              ;-----------------------------------------------+
                              ;       DWORD pointer to next device            | 1 word offset
                              ;               (-1,-1 if last device)          | 1 word segement
                              ;-----------------------------------------------+
                              ;       Device attribute WORD                   ; 1 word
                              ;       Bit 15 = 1 for chacter devices.         ;
                              ;               0 for Block devices.            ;
                              ;                                               ;
                              ;       Charcter 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.                 ;
                              ;-----------------------------------------------+

 0003                         DEVSTART LABEL WORD

 0003                         CONDEV:                         ;Header for device CON
 0003  0015 R 0000                    DW      AUXDEV,0        ;Link to next device
 0007  8003                           DW      BIT15+BIT1+BIT0 ;Attributes - console input, output device
 0009  00E1 R                         DW      STRATEGY        ;Srategy entry point
 000B  00EC R                         DW      CON_INT         ;Interrupt entry point
 000D  43 4F 4E 20 20 20              DB      "CON     "      ;Device name
       20 20

 0015                         AUXDEV:                         ;Header for device AUX
 0015  0027 R 0000                    DW      PRNDEV,0
 0019  8000                           DW      BIT15
 001B  00E1 R                         DW      STRATEGY
 001D  00F2 R                         DW      AUX_INT
 001F  41 55 58 20 20 20              DB      "AUX     "
       20 20
 0027                         PRNDEV:                          ;Header for device PRN
 0027  0039 R 0000                    DW      TIMDEV,0
 002B  8000                           DW      BIT15
 002D  00E1 R                         DW      STRATEGY
```

```
       20 20
```

```
 0039                          TIMDEV:                            ;Header for device CLOCK
 0039   0049 R 0000                    DW      DSKDEV,0
 003D   8008                           DW      BIT15+BIT3         ;8008H
 003F   00E1 R                         DW      STRATEGY
 0041   00FE R                         DW      TIM_INT
 0043   43 4C 4F 43 4B 09              DB      "CLOCK  "

 0049                          DSKDEV:                             ;Header for disk devices
 0049   FFFF FFFF                      DW      -1,-1
 004D   2000                           DW      BIT13              ;2000H          ;Is a block device
 004F   00E1 R                         DW      STRATEGY
 0051   0104 R                         DW      DSK_INT

 0053   02                             DB      2                  ;Number of Units
 0054      07 [                        DB      7 DUP (?)
                 ??
                     ]


                               SUBTTL  Dispatch tables for each device
```

```
                                        Dispatch tables for each device

                                        PAGE

005B  0299 R                            DSKTBL: DW      DSK_INIT        ;0  - Initialize Driver
005D  02A6 R                                    DW      MEDIAC          ;1  - Return current media code
005F  02B4 R                                    DW      GET_BPB         ;2  - Get Bios Parameter Block
0061  013B R                                    DW      CMDERR          ;3  - Reserved. (currently returns error)
0063  02D7 R                                    DW      DSK_RED         ;4  - Block read
0065  0137 R                                    DW      BUS_EXIT        ;5  - (Not used, return busy flag)
0067  0142 R                                    DW      EXIT            ;6  - Return status. (Not used)
0069  0142 R                                    DW      EXIT            ;7  - Flush input buffer. (Not used.)
006B  02DB R                                    DW      DSK_WRT         ;8  - Block write
006D  02DB R                                    DW      DSK_WRV         ;9  - Block write with verify
006F  0142 R                                    DW      EXIT            ;10 - Return output status
0071  0142 R                                    DW      EXIT            ;11 - Flush output buffer. (Not used.)
0073  0142 R                                    DW      EXIT            ;12 - IO Control

0075  0142 R                            CONTBL: DW      EXIT            ;0  - Init. (Not used)
0077  0142 R                                    DW      EXIT            ;1  - Media check (Not used)
0079  0142 R                                    DW      EXIT            ;2  - Get Bios Parameter Block (Not used)
007B  013B R                                    DW      CMDERR          ;3  - Reserved. (Currently returns error)
007D  0192 R                                    DW      CON_READ        ;4  - Character read. (Destructive)
007F  0181 R                                    DW      CON_RDND        ;5  - Character read. (Non-destructive)
0081  0142 R                                    DW      EXIT            ;6  - Return status. (Not used)
0083  019A R                                    DW      CON_FLSH        ;7  - Flush Input buffer
0085  01A3 R                                    DW      CON_WRIT        ;8  - Character write
0087  01A3 R                                    DW      CON_WRIT        ;9  - Character write with Verify
0089  01A1 R                                    DW      CON_WRST        ;10 - Character write status
008B  0142 R                                    DW      EXIT            ;11 - Flush output buffer. (Not used.)
008D  0142 R                                    DW      EXIT            ;12 - IO Control

008F  0142 R                            AUXTBL: DW      EXIT            ;0  - Init. (Not used)
0091  0142 R                                    DW      EXIT            ;1  - Media check (Not used)
0093  0142 R                                    DW      EXIT            ;2  - Get Bios Parameter Block (Not used)
0095  013B R                                    DW      CMDERR          ;3  - Reserved. (Returns an error)
0097  0208 R                                    DW      AUX_READ        ;4  - Character read. (Destructive)
0099  01FE R                                    DW      AUX_RDND        ;5  - Character read. (Non-destructive)
009B  0142 R                                    DW      EXIT            ;6  - Return status. (Not used)
009D  0211 R                                    DW      AUX_CLR         ;7  - Flush Input buffer
009F  021C R                                    DW      AUX_WRIT        ;8  - Character write
00A1  021C R                                    DW      AUX_WRIT        ;9  - Character write with verify
00A3  0219 R                                    DW      AUX_WRST        ;10 - Character write status
00A5  0142 R                                    DW      EXIT            ;11 - Flush output buffer. (Not used.)
00A7  0142 R                                    DW      EXIT            ;12 - IO Control
```

```
                                        Dispatch tables for each device

                                        PAGE

 00A9  0142 R                           TIMTBL: DW      EXIT            ;0  - Init. (Not used)
 00AB  0142 R                                   DW      EXIT            ;1  - Media check (Not used)
 00AD  0142 R                                   DW      EXIT            ;2  - Get Bios Parameter Block (Not used)
 00AF  013B R                                   DW      CMDERR          ;3  - Reserved. (Currently ret. an error)
 00B1  0245 R                                   DW      TIM_RED         ;4  - Character read. (Destructive)
 00B3  0137 R                                   DW      BUS_EXIT        ;5  - (Not used, returns busy flag.)
 00B5  0142 R                                   DW      EXIT            ;6  - Return status. (Not used)
 00B7  0142 R                                   DW      EXIT            ;7  - Flush Input buffer. (Not used)
 00B9  0230 R                                   DW      TIM_WRT         ;8  - Character write
 00BB  0230 R                                   DW      TIM_WRT         ;9  - Character write with verify
 00BD  0142 R                                   DW      EXIT            ;10 - Character write status. (Not used)
 00BF  0142 R                                   DW      EXIT            ;11 - Flush output buffer. (Not used)
 00C1  0142 R                                   DW      EXIT            ;12 - IO Control

 00C3  0142 R                           PRNTBL: DW      EXIT            ;0  - (Not used)
 00C5  0142 R                                   DW      EXIT            ;1  - (Not used)
 00C7  0142 R                                   DW      EXIT            ;2  - Block (Not used)
 00C9  013B R                                   DW      CMDERR          ;3  - Reserved. (currently returns error)
 00CB  0142 R                                   DW      EXIT            ;4  - (Not used)
 00CD  0137 R                                   DW      BUS_EXIT        ;5  - (Not used, returns busy flag.)
 00CF  0142 R                                   DW      EXIT            ;6  - (Not used)
 00D1  0142 R                                   DW      EXIT            ;7  - (Not used)
 00D3  01BF R                                   DW      PRN_WRT         ;8  - Character write
 00D5  01BF R                                   DW      PRN_WRT         ;9  - Character write with verify
 00D7  01B8 R                                   DW      PRN_STA         ;10 - Character write status
 00D9  0142 R                                   DW      EXIT            ;11 - (Not used.)
 00DB  0142 R                                   DW      EXIT            ;12 - IO Control

                                        SUBTTL  Strategy and Software Interrupt routines
```

```
                                Strategy and Software Interrupt routines

                                PAGE

                                ;       Define offsets for io data packet

                                IODAT   STRUC
 0000  ??                       CMDLEN  DB      ?               ;LENGTH OF THIS COMMAND
 0001  ??                       UNIT    DB      ?               ;SUB UNIT SPECIFIER
 0002  ??                       CMD     DB      ?               ;COMMAND CODE
 0003  ????                     STATUS  DW      ?               ;STATUS
 0005     08 [                          DB      8 DUP (?)
                 ??
                     ]

 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                           IODAT   ENDS

 00DD  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
                                ;

 00E1                           STRATP  PROC    FAR

 00E1                           STRATEGY:
 00E1  2E: 89 1E 00DD R                 MOV     WORD PTR CS:[PTRSAV],BX
 00E6  2E: 8C 06 00DF R                 MOV     WORD PTR CS:[PTRSAV+2],ES
 00EB  CB                               RET

 00EC                           STRATP  ENDP
```

```
                              Strategy and Software Interrupt routines

                              PAGE

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

 00EC                         CON_INT:
 00EC  56                             PUSH    SI
 00ED  BE 0075 R                      MOV     SI,OFFSET CONTBL
 00F0  EB 16                          JMP     SHORT ENTRY

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

 00F2                         AUX_INT:
 00F2  56                             PUSH    SI
 00F3  BE 008F R                      MOV     SI,OFFSET AUXTBL
 00F6  EB 10                          JMP     SHORT ENTRY

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

 00F8                         PRN_INT:
 00F8  56                             PUSH    SI
 00F9  BE 00C3 R                      MOV     SI,OFFSET PRNTBL
 00FC  EB 0A                          JMP     SHORT ENTRY

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

 00FE                         TIM_INT:
 00FE  56                             PUSH    SI
 00FF  BE 00A9 R                      MOV     SI,OFFSET TIMTBL
 0102  EB 04                          JMP     SHORT ENTRY
```

```
                                Strategy and Software Interrupt routines

                                PAGE

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


 0104                           DSK_INT:
 0104  56                               PUSH    SI
 0105  BE 005B R                        MOV     SI,OFFSET DSKTBL

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

 0108  50                       ENTRY:  PUSH    AX              ;Save all nessacary registers
 0109  51                               PUSH    CX
 010A  52                               PUSH    DX
 010B  57                               PUSH    DI
 010C  55                               PUSH    BP
 010D  1E                               PUSH    DS
 010E  06                               PUSH    ES
 010F  53                               PUSH    BX

 0110  2E: C5 1E 00DD R                 LDS     BX,CS:[PTRSAV]  ;Retrieve pointer to I/O Packet
 0115  8A 47 01                         MOV     AL,[BX.UNIT]    ;AL = Unit code
 0118  8A 67 0D                         MOV     AH,[BX.MEDIA]   ;AH = Media descriptor
 011B  8B 4F 12                         MOV     CX,[BX.COUNT]   ;CX = Contains byte/sector count
 011E  8B 57 14                         MOV     DX,[BX.START]   ;DX = Starting Logical sector

 0121  97                               XCHG    DI,AX           ;Move Unit & Media into DI temporarily
 0122  8A 47 02                         MOV     AL,[BX.CMD]     ;Retrieve Command type. (1 => 11)
 0125  32 E4                            XOR     AH,AH           ;Clear upper half of AX for calculation
 0127  03 F0                            ADD     SI,AX           ;Compute entry pointer in dispatch table
 0129  03 F0                            ADD     SI,AX
 012B  3C 0B                            CMP     AL,11           ;Verify that not more than 11 commands
 012D  77 0C                            JA      CMDERR          ;Ah, well, error out

 012F  97                               XCHG    AX,DI           ;Move Unit & Media back where they belong
 0130  C4 7F 0E                         LES     DI,[BX.TRANS]   ;DI contains addess of Transfer address
                                                                        ;ES contains segment
 0133  0E                               PUSH    CS
 0134  1F                               POP     DS              ;Data segment same as Code segment
 0135  FF 24                            JMP     [SI]            ;Perform I/O packet command

                                SUBTTL  Common error and exit points
```

```
                                        Common error and exit points

                                        PAGE

 0137                                   BUS_EXIT:                       ;Device busy exit
 0137  B4 03                                    MOV     AH,00000011B    ;Set busy and done bits
 0139  EB 09                                    JMP     SHORT EXIT1

 013B  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 = Unkown 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
                                        ;

 013D                                   ERR_EXIT:
 013D  B4 81                                    MOV     AH,10000001B    ;Set error and done bits
 013F  F9                                       STC                     ;Set carry bit also
 0140  EB 02                                    JMP     SHORT EXIT1     ;Quick way out

 0142                                   EXITP   PROC    FAR             ;Normal exit for device drivers

 0142  B4 01                            EXIT:   MOV     AH,00000001B    ;Set done bit for MSDOS
 0144  2E: C5 1E 00DD R                 EXIT1:  LDS     BX,CS:[PTRSAV]
 0149  89 47 03                                 MOV     [BX.STATUS],AX  ;Save operation compete and status

 014C  5B                                       POP     BX              ;Restore registers
 014D  07                                       POP     ES
 014E  1F                                       POP     DS
 014F  5D                                       POP     BP
 0150  5F                                       POP     DI
 0151  5A                                       POP     DX
 0152  59                                       POP     CX
 0153  58                                       POP     AX
 0154  5E                                       POP     SI
 0155  CB                                       RET

 0156                                   EXITP   ENDP

                                        SUBTTL  Main console I/O section
```

```
                                  Main console I/O section

                                  PAGE

 0156  ??                         CHAR    DB      ?               ;Small typeahead buffer for now

                                  ;---------------------------------------------------------------------------
                                  ; Console keyboard handler
                                  ;

 0157  51                         CISTAT: PUSH    CX              ;Save CX pair
 0158  A0 0156 R                          MOV     AL,[CHAR]
 015B  0A C0                              OR      AL,AL
 015D  75 15                              JNZ     CISTA9          ;Character still in buffer

 015F                             CISTA1:
                                  ;---------------------------------------------------------------------------
 015F  B0 01                              MOV     AL,CSTAT        ;**********************
 0161  E8 043A R                          CALL    OUTIN           ;send Command to IOCS85 and get result
                                  ;---------------------------------------------------------------------------
 0164  84 C0                              TEST    AL,AL
 0166  74 0C                              JZ      CISTA9

                                  ;---------------------------------------------------------------------------
 0168  B0 02                              MOV     AL,CONIN        ;**********************
 016A  E8 043A R                          CALL    OUTIN           ;send Command to IOCS85 and get result
                                  ;---------------------------------------------------------------------------

 016D  0A C0                              OR      AL,AL
 016F  74 EE                              JZ      CISTA1          ;Got a null character

 0171  A2 0156 R                          MOV     [CHAR],AL
 0174  59                         CISTA9: POP     CX              ;Can't lose CX pair
 0175  C3                                 RET

                                  ;---------------------------------------------------------------------------
                                  ; Get a character from the buffer queue
                                  ;

 0176  E8 0157 R                  CINP:   CALL    CISTAT          ;Check for character ready in queue
 0179  74 FB                              JZ      CINP            ;Cycle until one ready

 017B  C6 06 0156 R 00                    MOV     [CHAR],0        ;We have character in AL, clear type a head
 0180  C3                                 RET

                                  ;---------------------------------------------------------------------------
                                  ; Console read non-destructive
                                  ;

 0181                             CON_RDND:
 0181  E8 0157 R                          CALL    CISTAT          ;See if character ready
 0184  74 0A                              JZ      CON_RDN2        ;No, return busy signal

 0186                             CON_RDN1:
 0186  2E: C5 1E 00DD R                   LDS     BX,CS:[PTRSAV]
 018B  88 47 0D                           MOV     [BX.MEDIA],AL
 018E  EB B2                              JMP     EXIT

 0190                             CON_RDN2:
 0190  EB A5                              JMP     BUS_EXIT

                                  ;---------------------------------------------------------------------------
                                  ; Console destructive read
                                  ;
 0192  E8 0176 R                       CALL    CINP             ;Get character
```

```
 0195  AA                              STOSB                    ;Save it in users buffer
 0196  E2 FA                           LOOP    CON_READ         ;Loop until CX is exhausted
 0198  EB A8                           JMP     EXIT

                                ;--------------------------------------------------------------------------
                                ; Console flush routine. (ctrl-c, ctrl-f, or ctrl-s inspired)
                                ;

 019A                           CON_FLSH:
 019A  C6 06 0156 R 00                 MOV     [CHAR],0         ;Clear small type a head buffer
 019F  EB A1                           JMP     EXIT
```

```
                                        Main console I/O section

                                        PAGE
                                        ;-------------------------------------------------------------------------
                                        ; Console output status routine
                                        ;

 01A1                                   CON_WRST:
 01A1  EB 9F                                    JMP     EXIT            ;Yes, normal exit

                                        ;-------------------------------------------------------------------------
                                        ; Console output routine
                                        ;

 01A3                                   CON_WRIT:
 01A3  8B F7                                    MOV     SI,DI           ;Get destination to source
 01A5                                   CON_WRI1:
 01A5  26: AC                                   LODS    BYTE PTR ES:[SI]
 01A7  51                                       PUSH    CX

                                                ENDIF

                                                IFE     ANSI
 01A8  E8 01B0 R                                CALL    OUTCHR
                                                ENDIF

 01AB  59                                       POP     CX
 01AC  E2 F7                                    LOOP    CON_WRI1        ;Keep going until user buffer through
 01AE  EB 92                                    JMP     EXIT

                                        ;-------------------------------------------------------------------------
                                        ; Console character output routine
                                        ;

 01B0                                   OUTCHR:                         ;Character to output in AL
                                        ;-------------------------------------------------------------------------
 01B0  8A C8                                    MOV     CL,AL           ;save Output character
 01B2  B0 03                                    MOV     AL,COOUT        ;first send command to IOCS-85, then character
 01B4  E8 042D R                                CALL    SENCHR
                                        ;-------------------------------------------------------------------------
 01B7  C3                                       RET

                                                ENDIF

                                        SUBTTL  Printer buffer handler
```

```
                               PAGE
                               ;------------------------------------------------------------------------------
                               ;       Printer status routine
                               ;

 01B8                          PRN_STA:
 01B8  B0 04                           MOV     AL,LPSTS
 01BA  E8 043A R                       CALL    OUTIN           ;first send command, then byte in CL
 01BD  EB 83                           JMP     EXIT

                               ;------------------------------------------------------------------------------
                               ; Printer write routine
                               ;

 01BF                          PRN_WRT:
 01BF  8B F7                           MOV     SI,DI           ;Set source = destination index

 01C1  26: AC                  PRN_WR1:LODS    BYTE PTR ES:[SI];Get a data byte
 01C3  51                              PUSH    CX              ;Print character in AL

                               ;------------------------------------------------------------------------------
 01C4  8A C8                           MOV     CL,AL           ;save character for later
 01C6  B0 05                           MOV     AL,LPOUT        ;send command to IOCS-85
 01C8  E8 042D R                       CALL    SENCHR          ;then send the byte
                               ;------------------------------------------------------------------------------

 01CB  59                              POP     CX
 01CC  E2 F3                           LOOP    PRN_WR1
 01CE  E9 0142 R                       JMP     EXIT

                               SUBTTL  Auxilary I/O routines
```

```
                                Auxilary I/O routines

                                PAGE

 01D1   00                      AUXCHAR DB      0               ;Temporary AUX ahead storage

                                ;------------------------------------------------------------
                                ; Status routine for Auxilary port
                                ;

 01D2   A0 01D1 R               AISTAT: MOV     AL,[AUXCHAR]
 01D5   84 C0                           TEST    AL,AL
 01D7   75 0E                           JNZ     AISTA9          ;Character already waiting

                                ;------------------------------------------------------------
 01D9   B0 06                           MOV     AL,AUXSTS
 01DB   E8 043A R                       CALL    OUTIN           ;send command, then get result
                                ;------------------------------------------------------------

 01DE   84 C0                           TEST    AL,AL
 01E0   74 05                           JZ      AISTA9          ;Still none waiting

                                ;------------------------------------------------------------
 01E2   B0 07                           MOV     AL,AUXIN
 01E4   E8 043A R                       CALL    OUTIN           ;send command, then get byte
                                ;------------------------------------------------------------

 01E7   A2 01D1 R               AISTA9: MOV     [AUXCHAR],AL
 01EA   C3                              RET


                                ;------------------------------------------------------------
                                ; Auxilary port read
                                ;

 01EB   E8 01D2 R               AIN:    CALL    AISTAT          ;Get status and/or char
 01EE   74 FB                           JZ      AIN             ;Cycle until one is ready

 01F0   C6 06 01D1 R 00                 MOV     [AUXCHAR],0
 01F5   C3                              RET

                                ;------------------------------------------------------------
                                ; Write routine for Auxilary port
                                ;

 01F6                           AOUT:
                                ;------------------------------------------------------------
 01F6   8A C8                           MOV     CL,AL
 01F8   B0 08                           MOV     AL,AUXOUT
 01FA   E8 042D R                       CALL    SENCHR          ;first send command, then send byte
                                ;------------------------------------------------------------
 01FD   C3                              RET

                                ;------------------------------------------------------------
                                ; Non-Destructive Auxilary read routine
                                ;

 01FE                           AUX_RDND:
 01FE   E8 01D2 R                       CALL    AISTAT          ;Get status and copy of char. waiting if any
 0201   74 02                           JZ      AUX_RDN2        ;No character waiting, exit

 0203   EB 81                           JMP     CON_RDN1

 0205                           AUX_RDN2:
```

```
                                  Auxilary I/O routines

                                  ; Destructive Auxilary read routine
                                  ;

 0208                             AUX_READ:
 0208  E8 01EB R                          CALL    AIN               ;Get data character
 020B  AA                                 STOSB                     ;Save it through DI
 020C  E2 FA                              LOOP    AUX_READ          ;Cycle until user buffer full
 020E  E9 0142 R                          JMP     EXIT

                                  ;--------------------------------------------------------------------------
                                  ; Auxilary clear type a head
                                  ;

 0211                             AUX_CLR:
 0211  C6 06 01D1 R 00                    MOV     [AUXCHAR],0
 0216  E9 0142 R                          JMP     EXIT

                                  ;--------------------------------------------------------------------------
                                  ; Auxilary write port status
                                  ;

 0219                             AUX_WRST:
 0219  E9 0142 R                          JMP     EXIT

                                  ;--------------------------------------------------------------------------
                                  ; Auxilary write
                                  ;

 021C                             AUX_WRIT:
 021C  8B F7                              MOV     SI,DI
 021E                             AUX_WRI1:
 021E  26: AC                             LODS    BYTE PTR ES:[SI]          ;Get char. from users buffer
 0220  51                                 PUSH    CX                ;is destroyed by AOUT Routine
 0221  E8 01F6 R                          CALL    AOUT              ;Send it to device
 0224  59                                 POP     CX
 0225  E2 F7                              LOOP    AUX_WRI1          ;Cycle until all done
 0227  E9 0142 R                          JMP     EXIT

                                  SUBTTL  Date/Time Routines
```

```
                                        Date/Time Routines

                                        PAGE

 022A  0579                             TIM_DAYS: DW     1401            ;Number of days since 1-1-80 (02-Nov-83)
 022C  00                               TIM_MINS: DB     0               ;Minutes
 022D  08                               TIM_HRS:  DB     8               ;Hours
 022E  00                               TIM_HSEC: DB     0               ;Hundreths of a second
 022F  00                               TIM_SECS: DB     0               ;Seconds

                                        ;---------------------------------------------------------------------------
                                        ; Time write routine
                                        ;

 0230                                   TIM_WRT:
 0230  BE 022A R                                MOV     SI,OFFSET TIM_DAYS
 0233  87 F7                                    XCHG    SI,DI
 0235  06                                       PUSH    ES
 0236  8C D8                                    MOV     AX,DS
 0238  1F                                       POP     DS
 0239  8E C0                                    MOV     ES,AX
 023B  B9 0006                                  MOV     CX,6
 023E  F3/ A4                                   REP     MOVSB
 0240  B0 00                                    MOV     AL,0
 0242  E9 0142 R                                JMP     EXIT

                                        ;---------------------------------------------------------------------------
                                        ; Time read routine
                                        ;

 0245                                   TIM_RED:
 0245  BE 022A R                                MOV     SI,OFFSET TIM_DAYS
 0248  B9 0006                                  MOV     CX,6
 024B  F3/ A4                                   REP     MOVSB
 024D  B0 00                                    MOV     AL,0
 024F  E9 0142 R                                JMP     EXIT

                                        SUBTTL  Drive Tables
```

```
                                   Drive Tables

                                   PAGE

                                   SIOPB   STRUC

0000  ??                           OPCODE  DB       ?               ;I/O operation code
0001  ??                           DRIVE   DB       ?               ;Logical drive spec
0002  ????                         TRACK   DW       ?               ;Logical track number
0004  ??                           SIDE    DB       ?               ;Logical head number
0005  ??                           SECTOR  DB       ?               ;Logical sector to start with
0006  ??                           SCOUNT  DB       ?               ;Number of logical sectors in buffer
0007  ????                         DMAOFF  DW       ?               ;Buffer offset address
0009  ????                         DMASEG  DW       ?               ;Buffer segment

000B                               SIOPB   ENDS

0252  00                           IOPB    SIOPB   <0,0,0,0,0,0,0,0>
0253  00
0254  0000
0256  00
0257  00
0258  00
0259  0000
025B  0000



                                   ;-------------------------------------------------------------------------
                                   ; MSDOS drive initialization tables and other what not
                                   ;
                                   ;       Drive 0 is:

                                   ;       Drive 1 is: the same as Drive 0

                                   DBP     STRUC

0000     03 [                      JMPNEAR DB       3 DUP (?)       ;Jmp Near xxxx  for boot
                ??
                    ]

0003     08 [                      NAMEVER DB       8 DUP (?)       ;Name / Version of OS
                ??
                    ]


                                   ;-- Start of Drive Parameter Block

000B  ????                         SECSIZE DW       ?               ;Sector size in bytes.                 (dpb)
000D  ??                           ALLOC   DB       ?               ;Number of sectors per alloc. block.   (dpb)
000E  ????                         RESSEC  DW       ?               ;Reserved sectors.                     (dpb)
0010  ??                           FATS    DB       ?               ;Number of FAT's.                      (dpb)
0011  ????                         MAXDIR  DW       ?               ;Number of root directory entries.     (dpb)
0013  ????                         SECTORS DW       ?               ;Number of sectors per diskette.       (dpb)
0015  ??                           MEDIAID DB       ?               ;Media byte ID.                        (dpb)
0016  ????                         FATSEC  DW       ?               ;Number of FAT Sectors.                (dpb)

                                   ;-- End of Drive Parameter Block

0018  ????                         SECTRK  DW       ?               ;Number of Sectors per track
001A  ????                         HEADS   DW       ?               ;# of heads
001C  ????                         HIDDEN  DW       ?               ;# of hidden sectors

001E                               DBP     ENDS
```

```
                    ??
```

```
                          ]
 0260      08 [
                   ??
                          ]
 0268  0400
 026A  02
 026B  0005
 026D  02
 026E  00A0
 0270  0320
 0272  FF
 0273  0001
 0275  0005
 0277  0002
 0279  0000

 027B      03 [                         LDDRIV2 DBP      (,, 1024,2,5,2,160,800,0FFH,1, 5,2,0)
                   ??
                          ]
 027E      08 [
                   ??
                          ]
 0286  0400
 0288  02
 0289  0005
 028B  02
 028C  00A0
 028E  0320
 0290  FF
 0291  0001
 0293  0005
 0295  0002
 0297  0000


 0299                                   DSK_INIT:
 0299  B8 0002                                  MOV     AX,2
 029C  BE 02A2 R                                MOV     SI,OFFSET INITTAB
 029F  EB 1E 90                                 JMP     GET_BPS

 02A2                                   INITTAB:
 02A2  0268 R                                   DW      LDDRIV1.SECSIZE
 02A4  0286 R                                   DW      LDDRIV2.SECSIZE

                                        SUBTTL  Media check routine
                                        Media check routine

                                        PAGE

                                        ;----------------------------------------------------------------------------
                                        ; Media check routine
                                        ; On entry:
                                        ;       AL = disk unit number
                                        ;       AH = media byte
                                        ; On exit:
                                        ;
                                        ;       [MEDIA FLAG] = -1 (FF hex) if disk is changed
                                        ;       [MEDIA FLAG] = 0 if don't know
                                        ;       [MEDIA FLAG] = 1 if not changed
                                        ;
                                        ;       [MEDIA] = OFFH for alphaTronic disks

                                        MEDIAS  STRUC
0000     0D [                                   DB      13 DUP(?)               ;Static request header
               ??
                   ]

000D  ??                                MEDIAS1 DB      ?                       ;Media byte
000E  ??                                MEDIAS2 DB      ?                       ;Media status byte flag
000F                                    MEDIAS  ENDS

02A6  B4 00                             MEDIAC: MOV     AH,0            ;don't know if media changed
02A8  C5 1E 00DD R                      MEDIA1: LDS     BX,[PTRSAV]     ;Udate media section of data block
02AC  88 67 0E                                  MOV     [BX.MEDIAS2],AH
02AF  B0 00                                     MOV     AL,0
02B1  E9 0142 R                                 JMP     EXIT

                                        SUBTTL  Build and return Bios Parameter Block for a diskette
```

```
                                Build and return Bios Parameter Block for a diskette

                                PAGE

                                ;---------------------------------------------------------------------------
                                ;       Build Bios Parameter Blocks
                                ;
                                ;       On entry:  ES:DI contains the address of a scratch sector buffer
                                ;                        AL = Unit number
                                ;                        AH = Current media byte
                                ;
                                ;       On exit:         Return a DWORD pointer to the associated BPB
                                ;                        in the Request packet
                                ;

                                BPBS    STRUC
0000     0D [                           DB      13 DUP(?)               ;Static request header
              ??
                 ]

000D  ??                        BPB1    DB      ?                       ;Media byte
000E  ????                      BPB2    DW      ?                       ;DWORD transfer address
0010  ????                              DW      ?
0012  ????                      BPB3    DW      ?                       ;DWORD pointer to BPB
0014  ????                              DW      ?
0016                            BPBS    ENDS

                                ; Build Bios Parameter Blocks.
                                ;
                                ;       On entry:  ES:BX contains the address of a scratch sector buffer.
                                ;                  AL = Unit number.
                                ;                  AH = Current media byte.
                                ;
                                ;       On exit:   Return a DWORD pointer to the associated BPB
                                ;                  in the Request packet.
                                ;

                                        IF      DPBRD
02B4                            GET_BPB:
02B4  BE 025D R                         MOV     SI,OFFSET LDDRIV1
02B7  83 C6 0B                          ADD     SI,11
02BA  B0 FF                             MOV     AL,0FFH         ;Media Byte
02BC  EB 01 90                          JMP     GET_BP5

                                        ENDIF

02BF  C5 1E 00DD R              GET_BP5:LDS     BX,[PTRSAV]             ;Update I/O data packet
02C3  88 47 0D                          MOV     [BX.BPB1],AL            ;Media byte
02C6  89 77 12                          MOV     [BX.BPB3],SI            ;DPB pointer
02C9  8C 4F 14                          MOV     [BX.BPB3+2],CS          ;Code segment

02CC  B0 00                             MOV     AL,0                    ;assume that all is ok
02CE  E9 0142 R                         JMP     EXIT

02D1  B8 0007                   GET_BP6:MOV     AX,7                    ;unknown Media discovered
02D4  E9 013D R                         JMP     ERR_EXIT

                                SUBTTL  MSDOS 2.x Disk I/O drivers
                                MSDOS 2.x Disk I/O drivers

                                PAGE

                                ;--------------------------------------------------------------
                                ; Disk READ / WRITE functions
                                ;
                                ; On entry:
                                ;       AL = Disk Drive number
                                ;       AH = Media byte

                                ;       ES = Disk transfer segment
                                ;       DI = Disk transfer offset in ES

                                ;       CX = Number of sectors to transfer
                                ;       DX = Logical starting sector

                                ; On exit:
                                ;         Normal exit through common exit  routine
                                ;       Abnormal exit through common error routine
                                ;

 02D7                           DSK_RED:
 02D7  B3 09                            MOV     BL,READ                 ;Set read mode and Error mask
 02D9  EB 02                            JMP     SHORT DSK_COM

 02DB                           DSK_WRV:
 02DB  B3 0A                    DSK_WRT:MOV     BL,WRITE                ;Set write mode and Error mask
 02DD  BE 025D R                DSK_COM:MOV     SI,OFFSET LDDRIV1
 02E0  3A 64 15                         CMP     AH,[SI.MEDIAID]
 02E3  74 0D                            JE      DSK_C02

 02E5  BE 027B R                        MOV     SI,OFFSET LDDRIV2
 02E8  3A 64 15                         CMP     AH,[SI.MEDIAID]
 02EB  74 05                            JE      DSK_C02

 02ED  B0 07                            MOV     AL,7                    ;Unknown media discovered
 02EF  E9 013D R                        JMP     ERR_EXIT
                                ;--------------------------------------------------------------
 02F2                           DSK_C02:
 02F2  8C 06 025B R                     MOV     [IOPB.DMASEG],ES        ;Setup Buffer segment
 02F6  89 3E 0259 R                     MOV     [IOPB.DMAOFF],DI        ;Setup buffer offset

 02FA  88 1E 0252 R                     MOV     [IOPB.OPCODE],BL        ;R/W opcode
 02FE  A2 0253 R                        MOV     [IOPB.DRIVE],AL         ;Drive with density select
 0301  8B E9                            MOV     BP,CX                   ;Save number of sectors to R/W

 0303  52                       DSK_C04:PUSH    DX                      ;Save starting sector
 0304  8B C2                            MOV     AX,DX
 0306  BA 0000                          MOV     DX,0                    ;32 bit divide coming up
 0309  8B 4C 18                         MOV     CX,[SI.SECTRK]
 030C  F7 F1                            DIV     CX                      ;Get track+head and start sector

                                        ;       DX:AX / CX = AX mod DX
                                        ;       log. Sector # / .SECTRK = AX(.TRACK) mod DX(.SECTOR)

 030E  FE C2                            INC     DL                      ;Sector # starts with 1
 0310  88 16 0257 R                     MOV     [IOPB.SECTOR],DL        ;Starting sector
 0314  8A DA                            MOV     BL,DL                   ;Save starting sector for later

 0316  50                               PUSH    AX
 0317  D1 F8                            SAR     AX,1                    ;generate Track & Side
 0319  A3 0254 R                        MOV     [IOPB.TRACK],AX         ;Track to read/write
 031C  58                               POP     AX
```

sks
STEINMETZ, KRISCHKE
SYSTEMTECHNIK – GMBH
Maybachstraße 10
7500 Karlsruhe 41

Helmut Wiertalla
Abteilungsleiter Software

Telefon 07 21 / 40 30 76/77/78
Telex : 7 826 934 sksg–d

```
0323  B0 00                       MOV     AL,0
```

```
0325  8B 44 18                    MOV     AX,[SI.SECTRK]        ;Now see how many sectors
0328  FE C0                       INC     AL                    ;  we can burst read
032A  2A C3                       SUB     AL,BL                 ;BL is the starting sector
032C  B4 00                       MOV     AH,0
032E  5A                          POP     DX                    ;Retrieve logical sector start
032F  3B C5                       CMP     AX,BP                 ;See if on last partial track+head
0331  7F 06                       JG      DSK_C05               ;Yes, on last track+head

0333  2B E8                       SUB     BP,AX                 ;No, update number of sectors left
0335  03 D0                       ADD     DX,AX                 ;Update next starting sector
0337  EB 05                       JMP     SHORT DSK_C06

0339  8B C5               DSK_C05:MOV     AX,BP                 ;Only read enough of sector
033B  BD 0000                     MOV     BP,0                  ;to finish buffer and clear # left
033E  A2 0258 R           DSK_C06:MOV     [IOPB.SCOUNT],AL
0341  8B F8                       MOV     DI,AX                 ;Save number sectors for later

                          ;------------------------------------------------------------
                          ;       Common Disk Read / Write routine
                          ;       Reads or writes [IOPB.SECCOUNT] Sectors
                          ;------------------------------------------------------------

0343  FC                          CLD                           ;set direction to forward
0344  E8 03D1 R                   CALL    SNDPAR
0347  74 06                       JZ      POSOK

0349  E8 03FA R                   CALL    DERROR
034C  E9 013D R                   JMP     ERR_EXIT
                          ;--------------------------------------------------------------------
034F  80 3E 0252 R 09     POSOK:  CMP     [IOPB.OPCODE],READ
                          ;       CMP     AH,READ               ;now check, what's to do
0354  74 2E                       JZ      DSKREAD

0356  56                          PUSH    SI
0357  A0 0258 R                   MOV     AL,[IOPB.SCOUNT]
035A  B4 00                       MOV     AH,0
035C  B1 09                       MOV     CL,9                  ;* 512
035E  D3 E0                       SHL     AX,CL
0360  8B C8                       MOV     CX,AX                 ;# of bytes / 2 in CX

0362  8E 06 025B R                MOV     ES,[IOPB.DMASEG]
0366  8B 36 0259 R                MOV     SI,[IOPB.DMAOFF]
036A  26: AD              DSKOU1: LODS    WORD PTR ES:[SI]      ;get 2 bytes from DMASEG:DMAOFF
036C  E8 0432 R                   CALL    OUTDAT
036F  8A C4                       MOV     AL,AH
0371  E8 0432 R                   CALL    OUTDAT
0374  E2 F4                       LOOP    DSKOU1

0376  5E                          POP     SI

0377  E8 043D R                   CALL    INDAT                 ;all Bytes ok ?
037A  0A C0                       OR      AL,AL
037C  74 2A                       JZ      GEMEXI

037E  E8 03FA R                   CALL    DERROR
0381  E9 013D R                   JMP     ERR_EXIT
                          ;------------------------------------------------------------
0384                      DSKREAD:
0384  57                          PUSH    DI
0385  A0 0258 R                   MOV     AL,[IOPB.SCOUNT]
0388  B4 00                       MOV     AH,0
```

```
 038E  8B C8                          MOV     CX,AX                    ;# of bytes in CX
```

```
 0390  8B 3E 0259 R                   MOV     DI,[IOPB.DMAOFF]
 0394  8E 06 025B R                   MOV     ES,[IOPB.DMASEG]
 0398  E8 043D R              DSKIN1: CALL    INDAT                    ;returns byte in AL
 039B  8A D8                          MOV     BL,AL                    ;save byte for later
 039D  E8 043D R                      CALL    INDAT
 03A0  8A E0                          MOV     AH,AL
 03A2  8A C3                          MOV     AL,BL                    ;16 bit value constructed
 03A4  AB                             STOS    WORD PTR ES:[DI]
 03A5  E2 F1                          LOOP    DSKIN1

 03A7  5F                             POP     DI
                              ;----------------------------------------------------------

 03A8  8B C7                  GEMEXI: MOV     AX,DI                    ;Retrieve number of sectors read
 03AA  8B 4C 0B                       MOV     CX,[SI.SECSIZE]          ;Number of bytes per sector
 03AD  52                             PUSH    DX
 03AE  F7 E1                          MUL     CX                       ;DX:AX=AX*CX
 03B0  5A                             POP     DX
 03B1  A8 0F                          TEST    AL,0FH                   ;Make sure no strange sizes
 03B3  75 17                          JNZ     DSK_C07                  ;Illegal sector size found

 03B5  B1 04                          MOV     CL,4                     ; / 16 !! (1 Segment = 16 bytes)
 03B7  D3 E8                          SHR     AX,CL                    ;Convert number of bytes to para
 03B9  03 06 025B R                   ADD     AX,[IOPB.DMASEG]
 03BD  A3 025B R                      MOV     [IOPB.DMASEG],AX
 03C0  0B ED                          OR      BP,BP
 03C2  74 03                          JZ      DSK_OK

 03C4  E9 0303 R                      JMP     DSK_C04                  ;Still more to do

 03C7  B0 00                  DSK_OK: MOV     AL,0
 03C9  E9 0142 R                      JMP     EXIT                     ;All done

 03CC  B0 0C                  DSK_C07:MOV     AL,12                    ;general failure
 03CE  E9 013D R                      JMP     ERR_EXIT
                              ;----------------------------------------------------------
 03D1  A0 0252 R              SNDPAR: MOV     AL,[IOPB.OPCODE]         ;send Floppy Parameters to IOCS
 03D4  8A E0                          MOV     AH,AL                    ;save Op-Code for later
 03D6  E8 0432 R                      CALL    OUTDAT                   ;send Op-Code to IOCS
 03D9  A0 0253 R                      MOV     AL,[IOPB.DRIVE]
 03DC  E8 0432 R                      CALL    OUTDAT
 03DF  A0 0256 R                      MOV     AL,[IOPB.SIDE]
 03E2  E8 0432 R                      CALL    OUTDAT
 03E5  A1 0254 R                      MOV     AX,[IOPB.TRACK]          ;Track # is 16 bit !!
 03E8  E8 0432 R                      CALL    OUTDAT
 03EB  A0 0257 R                      MOV     AL,[IOPB.SECTOR]
 03EE  E8 0432 R                      CALL    OUTDAT
 03F1  A0 0258 R                      MOV     AL,[IOPB.SCOUNT]         ;send # of Sectors (1 to 5)
 03F4  E8 043A R                      CALL    OUTIN                    ;

                              ;               **** IOCS works (position to specified Parameters ****


 03F7  0A C0                          OR      AL,AL                    ;check for error condition,
                                                                       ;Then positioning is ok
 03F9  C3                             RET

                              SUBTTL  Disk Error processing
```

```
                                   Disk Error processing

                                   PAGE

                                   ;-----------------------------------------------------------------------
                                   ;       Disk error routine
                                   ;-----------------------------------------------------------------------

 03FA  2E: C5 1E 00DD R            DERROR: LDS     BX,CS:[PTRSAV]
 03FF  C7 47 12 0000                       MOV     [BX.COUNT],0
 0404  0E                                  PUSH    CS
 0405  1F                                  POP     DS                      ;DS = CS

 0406  B3 FF                               MOV     BL,-1
 0408  8A E0                               MOV     AH,AL
 040A  B7 0E                               MOV     BH,14           ;Lenght of table
 040C  BE 0420 R                           MOV     SI,OFFSET DERRTAB
 040F  FE C3                       DERROR2:INC     BL              ;Increment to next error code
 0411  2E: AC                              LODS    BYTE PTR CS:[SI]
 0413  3A E0                               CMP     AH,AL           ;See if error code matches disk status
 0415  74 06                               JZ      DERROR3         ;Got the right error, exit

 0417  FE CF                               DEC     BH
 0419  75 F4                               JNZ     DERROR2         ;Keep checking table

 041B  B3 0C                               MOV     BL,12           ;Set general type of error
 041D  8A C3                       DERROR3:MOV     AL,BL           ;Now we've got the code
 041F  C3                                  RET


                                   ;-----------------------------------------------------------------------
                                   ;       The codes in the table are the codes that the IOCS
                                   ;       returns after Disk I/O.
                                   ;-----------------------------------------------------------------------

 0420  10                          DERRTAB DB      10H             ; 0. Write protect error
 0421  00                                  DB      00H             ; 1. Unknown unit
 0422  40                                  DB      40H             ; 2. Not ready error
 0423  08                                  DB      08H             ; 3. Unknown command
 0424  80                                  DB      80H             ; 4. CRC error
 0425  00                                  DB      00H             ; 5. Bad drive request
 0426  00                                  DB      00H             ; 6. Seek error
 0427  00                                  DB      00H             ; 7. Unknown media
 0428  20                                  DB      20H             ; 8. Sector not found
 0429  00                                  DB      00H             ; 9. (Not used.)
 042A  01                                  DB      01H             ;10. Write fault
 042B  00                                  DB      00H             ;11. Read fault
 042C  00                                  DB      00H             ;12. General type of failure

                                   SUBTTL  Interrupt Routines and IOCS-Call
```

```
                                  Interrupt Routines and IOCS-Call

                                  PAGE

042D  E8 0432 R                   SENCHR: CALL    OUTDAT          ;send command
0430  8A C1                               MOV     AL,CL
0432  52                          OUTDAT: PUSH    DX
0433  BA FFEA                             MOV     DX,PIOOUT
0436  9B                                  WAIT                    ;for OBF = 0 (1)
0437  EE                                  OUT     DX,AL
0438  5A                                  POP     DX
0439  C3                                  RET

                                  ;------------------------------------------------------------------
043A  E8 0432 R                   OUTIN:  CALL    OUTDAT
043D  52                          INDAT:  PUSH    DX
043E  BA FFE9                             MOV     DX,PIOSTS
0441  EC                          INDAT1: IN      AL,DX
0442  A8 01                               TEST    AL,1
0444  75 FB                               JNZ     INDAT1


0446  42                                  INC     DX
0447  EC                                  IN      AL,DX
0448  5A                                  POP     DX
0449  C3                                  RET
                                  ;------------------------------------------------------------------
044A                              CLK_INTER:
044A  1E                                  PUSH    DS
044B  52                                  PUSH    DX
044C  50                                  PUSH    AX

044D  0E                                  PUSH    CS
044E  1F                                  POP     DS

044F  80 06 022E R 02                     ADD     BYTE PTR TIM_HSEC,2     ;this INT is generated every 20 ms
0454  80 3E 022E R 64                     CMP     BYTE PTR TIM_HSEC,100   ;one second passed ?
0459  75 39                               JNE     INT_END

045B  C6 06 022E R 00                     MOV     BYTE PTR TIM_HSEC,0     ;clear counter for hundreds of seconds
0460  FE 06 022F R                        INC     BYTE PTR TIM_SECS       ;increment the second counter
0464  80 3E 022F R 3C                     CMP     BYTE PTR TIM_SECS,60    ;one minute passed ?
0469  75 29                               JNE     INT_END

046B  C6 06 022F R 00                     MOV     BYTE PTR TIM_SECS,0
0470  FE 06 022C R                        INC     BYTE PTR TIM_MINS       ;same stuff as above
0474  80 3E 022C R 3C                     CMP     BYTE PTR TIM_MINS,60
0479  75 19                               JNE     INT_END

047B  C6 06 022C R 00                     MOV     BYTE PTR TIM_MINS,0
0480  FE 06 022D R                        INC     BYTE PTR TIM_HRS
0484  80 3E 022D R 18                     CMP     BYTE PTR TIM_HRS,24
0489  75 09                               JNE     INT_END

048B  C6 06 022D R 00                     MOV     BYTE PTR TIM_HRS,0
0490  FF 06 022A R                        INC     WORD PTR TIM_DAYS
0494                              INT_END:
0494  BA FFE0                             MOV     DX,PICR0                ;send OCW1
0497  B0 20                               MOV     AL,EOI                  ;End-of-Interrupt
0499  EE                                  OUT     DX,AL

049A  58                                  POP     AX
049B  5A                                  POP     DX
049C  1F                                  POP     DS
```

```
;DUMMY: PUSH    DX
;       PUSH    AX
;       MOV     DX,PICR0
;       MOV     AL,EOI
;       OUT     DX,AL
;       POP     AX
;       POP     DX
;       STI
;       IRET

SUBTTL  Initalization code and temporary work areas
```

```
                                        Initalization code and temporary work areas

                                        PAGE

 049F                                   REINI   PROC    FAR

 049F                                   RE_INIT:
 049F   CB                                      RET

 04A0                                   REINI   ENDP

                                        ;---------------------------------------------------------------------------

 04A0   EB 18 90                        HWINIT: JMP     HWINI1

 04A3   8B EC                           PRINT:  MOV     BP,SP
 04A5   87 5E 00                                XCHG    BX,[BP]
 04A8   8A 07                           PRINT1: MOV     AL,[BX]
 04AA   E8 01B0 R                               CALL    OUTCHR
 04AD   43                                      INC     BX
 04AE   80 3F FF                                CMP     BYTE PTR [BX],-1
 04B1   75 F5                                   JNE     PRINT1

 04B3   43                                      INC     BX
 04B4   8B EC                                   MOV     BP,SP
 04B6   87 5E 00                                XCHG    BX,[BP]
 04B9   C3                                      RET

 04BA   FA                              HWINI1: CLI
 04BB   8C C8                                   MOV     AX,CS
 04BD   8E D8                                   MOV     DS,AX
 04BF   8E C0                                   MOV     ES,AX

                                        ;---------------------------------------------------------------------------
                                        ;TIMER0 EQU     0FFE4H
                                        ;TIMER1 EQU     0FFE5H          ;CLK=100KHz, OUT1=IR1
                                        ;TIMER2 EQU     0FFE6H          ;CLK=100KHz, OUT2=IR0
                                        ;TIMCMD EQU     0FFE7H          ;Timer controll Register

                                        ;               Steuerwort Format

                                        ;SEL0   EQU     0 SHL 6         LATCH   EQU     0
                                        ;SEL1   EQU     1 SHL 6         RLMSB   EQU     BIT5
                                        ;SEL2   EQU     2 SHL 6         RLLSB   EQU     BIT4
                                        ;                               RLLMSB  EQU     BIT4+BIT5

                                        ;MODE0  EQU     0               ;Int. on zero-count
                                        ;MODE1  EQU     BIT1            ;prog. Monoflop
                                        ;MODE2  EQU     BIT2            ;synch. divider by n
                                        ;MODE3  EQU     BIT1+BIT2       ;square wave generator
                                        ;MODE4  EQU     BIT3            ;software controlled strobe
                                        ;MODE5  EQU     BIT3+BIT1       ;hardware controlled strobe

                                        ;BCD    EQU     BIT0            ;4 decade BCD-counter
                                        ;BIN    EQU     0               ;16 bit binary counter
                                        ;---------------------------------------------------------------------------

 04C1   E8 04A3 R                               CALL    PRINT
 04C4   0D 0A                                   DB      13,10
 04C6   61 6C 70 68 61 54                       DB      'alphaTronic BIOS P30                   V1.3'
        72 6F 6E 69 63 20
        42 49 4F 53 20 50
        33 30 20 20 20 20
        20 20 20 20 20 20
```

```
04ED  20 20 20 20 20 20           DB    '                      28-Oct-83 ',13,10,-1
```

```
      20 20 20 20 20 20
      20 20 20 20 20 20
      20 20 20 20 20 20
      20 20 20 32 38 2D
      4F 63 74 2D 38 33
      0D 0A FF

0514  BA FFE7                          MOV     DX,TIMCMD
0517  B0 36                            MOV     AL,SEL0+RLLMSB+MODE3+BIN
0519  EE                               OUT     DX,AL           ;select Timer 0

051A  BA FFE4                          MOV     DX,TIMER0
051D  B8 07D0                          MOV     AX,2000
0520  EE                               OUT     DX,AL
0521  8A C4                            MOV     AL,AH
0523  EE                               OUT     DX,AL

0524  BA FFE7                          MOV     DX,TIMCMD
0527  B0 76                            MOV     AL,SEL1+RLLMSB+MODE3+BIN
0529  EE                               OUT     DX,AL           ;select Timer 1

052A  BA FFE5                          MOV     DX,TIMER1
052D  B8 C350                          MOV     AX,50000
0530  EE                               OUT     DX,AL
0531  8A C4                            MOV     AL,AH
0533  EE                               OUT     DX,AL           ;Timer 1 generates an INT every .5 s

0534  BA FFE7                          MOV     DX,TIMCMD
0537  B0 B6                            MOV     AL,SEL2+RLLMSB+MODE3+BIN
0539  EE                               OUT     DX,AL           ;select Timer 2

053A  BA FFE6                          MOV     DX,TIMER2
053D  B8 07D0                          MOV     AX,2000         ;100.000 / 50
0540  EE                               OUT     DX,AL
0541  8A C4                            MOV     AL,AH
0543  EE                               OUT     DX,AL           ;Timer 2 generates an INT every 20 ms

                                ;--------------------------------------------------------------------------
                                ;PICR0  EQU     0FFE0H          ;ICW1, OCW2, OCW3
                                ;PICR1  EQU     0FFE1H          ;ICW2, ICW3, ICW4, OCW1
                                                                ;Int. Controller Port Address


                                ;               Bits in ICW1
                                ;LTIM   EQU     BIT3            ;Level trigg. = 1, Edge = 0
                                ;SNGL   EQU     BIT1            ;Single = 1, Cascade Mode = 0
                                ;ICW4   EQU     BIT0            ;ICW4 needed = 1, no ICW4 needed = 0

                                ;               Bits in ICW2
                                ;               set 5 MSB's of INT Vector as ICW2

                                ;               Bits in ICW3
                                ;               **** only in Slave Mode ****

                                ;               Bits in ICW 4
                                ;SNFM   EQU     BIT4            ;special fully nested mode = 1
                                                                ;not special fully nested mode = 0
                                ;BUF    EQU     BIT3            ;0 X non buffered Mode
                                ;MS     EQU     BIT2            ;1 0 buffered Mode Slave
                                                                ;1 1 buffered Mode Master
                                ;AEOI   EQU     BIT1            ;Auto EOI = 1, normal EOI = 0
                                ;UPM    EQU     BIT0            ;8086/8088 = 1, 8085 = 0
```

```
                                        ;EOI    EQU     BIT5            ;non specified End-of-Interrupt
```

```
                                        Initalization code and temporary work areas

                                        ;---------------------------------------------------------------------------

 0544  BA FFE0                                  MOV     DX,PICRO        ;Init the 8259
 0547  B0 13                                    MOV     AL,10H+ICW4+SNGL        ;AND (NOT LTIM)
 0549  EE                                       OUT     DX,AL           ;= ICW1

 054A  42                                       INC     DX
 054B  B0 08                                    MOV     AL,8            ;= ICW2 (INT Vector for IR 0)
 054D  EE                                       OUT     DX,AL

 054E  B0 0D                                    MOV     AL,BUF+MS+UPM
 0550  EE                                       OUT     DX,AL           ;= ICW4

 0551  B0 FF                                    MOV     AL,BIT0+BIT1+BIT2+BIT3+BIT4+BIT5+BIT6+BIT7
 0553  EE                                       OUT     DX,AL           ;= OCW1 (mask all IR's)

                                        ;---------------------------------------------------------------------------
 0000                                   INTSEG  SEGMENT AT 0

 0020                                           ORG     8*4

 0020                                   INTVECTOR LABEL WORD

 0020                                   INTSEG  ENDS

                                        ;---------------------------------------------------------------------------

                                        ASSUME  DS:INTSEG

 0554  33 C0                                    XOR     AX,AX           ;set up some Interrupt Vectors
 0556  8E D8                                    MOV     DS,AX
                                                                        ;Set the clock interrupt
 0558  C7 06 0020 R 044A R                      MOV     INTVECTOR,OFFSET CLK_INTER
 055E  8C 0E 0022 R                             MOV     INTVECTOR+2,CS

 0562  8A 26 0100                               MOV     AH,DS:[BNKSIZ]          ;left by BOOT EPROM
 0566  32 C0                                    XOR     AL,AL
 0568  B1 04                                    MOV     CL,4
 056A  D2 C4                                    ROL     AH,CL
 056C  8B C8                                    MOV     CX,AX                   ;top of memory in CX
                                        ASSUME  DS:CODE
                                        ;---------------------------------------------------------------------------
 056E  8C C8                                    MOV     AX,CS
 0570  8E D8                                    MOV     DS,AX
 0572  A3 0005 R                                MOV     WORD PTR CONDEV+2,AX
 0575  A3 0017 R                                MOV     WORD PTR AUXDEV+2,AX
 0578  A3 0029 R                                MOV     WORD PTR PRNDEV+2,AX
 057B  A3 003B R                                MOV     WORD PTR TIMDEV+2,AX
                                        ;---------------------------------------------------------------------------

                                        ;---------------------------------------------------------------------------
 057E  B8 ---- E                                MOV     AX,SEG SYSINIT
 0581  8E D8                                    MOV     DS,AX

                                        ASSUME  DS:SEG SYSINIT

 0583  8C C8                                    MOV     AX,CS
 0585  05 0100                                  ADD     AX,BIOSIZS              ;current DOS Location = CS:+BIOSIZS
 0588  A3 0000 E                                MOV     DS:[CURRENT_DOS_LOCATION],AX

 058B  8B C1                                    MOV     AX,CX
 058D  A3 0000 E                                MOV     DS:[MEMORY_SIZE],AX
 0592  A3 0002 E                           MOV     WORD PTR DS:[DEVICE_LIST+2],AX
```

```
 0595  C7 06 0000 E 0003 R                 MOV     WORD PTR DS:[DEVICE_LIST],OFFSET DEVSTART

 059B  8C C8                               MOV     AX,CS
 059D  05 004D                             ADD     AX,((OFFSET HWINIT - OFFSET INIT)+50) /16
 05A0  A3 0000 E                           MOV     DS:[FINAL_DOS_LOCATION],AX
                                                                          ;there I want the DOS to be
 05A3  BA FFE1                             MOV     DX,PICR1
 05A6  B0 FE                               MOV     AL,NOT BIT0            ;enable Timer 2 INT as IR0
 05A8  EE                                  OUT     DX,AL

 05A9  EA 0000 ---- E                      JMP     SYSINIT

 05AE                              CODE    ENDS

                                           END     INIT
 The Microsoft MACRO Assembler              10-08-85       PAGE     Symbols-1
IO.SYS for the alphaTronic P30


Structures and records:

                N a m e                 Width   # fields
                                        Shift   Width   Mask    Initial

BPBS . . . . . . . . . . . . . .        0016    0006
  BPB1 . . . . . . . . . . . . . .      000D
  BPB2 . . . . . . . . . . . . . .      000E
  BPB3 . . . . . . . . . . . . . .      0012
DBP. . . . . . . . . . . . . . .        001E    000D
  JMPNEAR. . . . . . . . . . . . .      0000
  NAMEVER. . . . . . . . . . . . .      0003
  SECSIZE. . . . . . . . . . . . .      000B
  ALLOC. . . . . . . . . . . . . .      000D
  RESSEC . . . . . . . . . . . . .      000E
  FATS . . . . . . . . . . . . . .      0010
  MAXDIR . . . . . . . . . . . . .      0011
  SECTORS. . . . . . . . . . . . .      0013
  MEDIAID. . . . . . . . . . . . .      0015
  FATSEC . . . . . . . . . . . . .      0016
  SECTRK . . . . . . . . . . . . .      0018
  HEADS. . . . . . . . . . . . . .      001A
  HIDDEN . . . . . . . . . . . . .      001C
IODAT. . . . . . . . . . . . . .        0016    0009
  CMDLEN . . . . . . . . . . . . .      0000
  UNIT . . . . . . . . . . . . . .      0001
  CMD. . . . . . . . . . . . . . .      0002
  STATUS . . . . . . . . . . . . .      0003
  MEDIA. . . . . . . . . . . . . .      000D
  TRANS. . . . . . . . . . . . . .      000E
  COUNT. . . . . . . . . . . . . .      0012
  START. . . . . . . . . . . . . .      0014
MEDIAS . . . . . . . . . . . . .        000F    0003
  MEDIAS1. . . . . . . . . . . . .      000D
  MEDIAS2. . . . . . . . . . . . .      000E
SIOPB. . . . . . . . . . . . . .        000B    0008
  OPCODE . . . . . . . . . . . . .      0000
  DRIVE. . . . . . . . . . . . . .      0001
  TRACK. . . . . . . . . . . . . .      0002
  SIDE . . . . . . . . . . . . . .      0004
  SECTOR . . . . . . . . . . . . .      0005
  SCOUNT . . . . . . . . . . . . .      0006
  DMAOFF . . . . . . . . . . . . .      0007
  DMASEG . . . . . . . . . . . . .      0009

Segments and groups:

                N a m e                 Size    align   combine class

CODE . . . . . . . . . . . . . .        05AE    BYTE    PUBLIC
INTSEG . . . . . . . . . . . . .        0020    AT      0000

Symbols:

                N a m e                 Type    Value   Attr

AEOI . . . . . . . . . . . . . .        Alias   BIT1
AIN. . . . . . . . . . . . . . .        L NEAR  01EB    CODE
AISTA9 . . . . . . . . . . . . .        L NEAR  01E7    CODE
AISTAT . . . . . . . . . . . . .        L NEAR  01D2    CODE
ANSI . . . . . . . . . . . . . .        Number  0000
AOUT . . . . . . . . . . . . . .        L NEAR  01F6    CODE
```

```
AUXIN. . . . . . . . . . . . . . . .      Number  0007
AUXOUT . . . . . . . . . . . . . . .      Number  0008
```

```
AUXSTS . . . . . . . . . . . . . . .      Number  0006
AUXTBL . . . . . . . . . . . . . . .      L NEAR  008F    CODE
AUX_CLR. . . . . . . . . . . . . . .      L NEAR  0211    CODE
AUX_INT. . . . . . . . . . . . . . .      L NEAR  00F2    CODE
AUX_RDN2 . . . . . . . . . . . . . .      L NEAR  0205    CODE
AUX_RDND . . . . . . . . . . . . . .      L NEAR  01FE    CODE
AUX_READ . . . . . . . . . . . . . .      L NEAR  0208    CODE
AUX_WRI1 . . . . . . . . . . . . . .      L NEAR  021E    CODE
AUX_WRIT . . . . . . . . . . . . . .      L NEAR  021C    CODE
AUX_WRST . . . . . . . . . . . . . .      L NEAR  0219    CODE
BCD. . . . . . . . . . . . . . . . .      Alias   BIT0
BIN. . . . . . . . . . . . . . . . .      Number  0000
BIOSIZ . . . . . . . . . . . . . . .      Number  1000
BIOSIZS. . . . . . . . . . . . . . .      Number  0100
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
BNKSIZ . . . . . . . . . . . . . . .      Number  0100
BUF. . . . . . . . . . . . . . . . .      Alias   BIT3
BUFFERS. . . . . . . . . . . . . . .      V BYTE  0000            External
BUS_EXIT . . . . . . . . . . . . . .      L NEAR  0137    CODE
CHAR . . . . . . . . . . . . . . . .      L BYTE  0156    CODE
CINP . . . . . . . . . . . . . . . .      L NEAR  0176    CODE
CISTA1 . . . . . . . . . . . . . . .      L NEAR  015F    CODE
CISTA9 . . . . . . . . . . . . . . .      L NEAR  0174    CODE
CISTAT . . . . . . . . . . . . . . .      L NEAR  0157    CODE
CLK_INTER. . . . . . . . . . . . . .      L NEAR  044A    CODE
CMDERR . . . . . . . . . . . . . . .      L NEAR  013B    CODE
CONDEV . . . . . . . . . . . . . . .      L NEAR  0003    CODE
CONIN. . . . . . . . . . . . . . . .      Number  0002
CONTBL . . . . . . . . . . . . . . .      L NEAR  0075    CODE
CON_FLSH . . . . . . . . . . . . . .      L NEAR  019A    CODE
CON_INT. . . . . . . . . . . . . . .      L NEAR  00EC    CODE
CON_RDN1 . . . . . . . . . . . . . .      L NEAR  0186    CODE
CON_RDN2 . . . . . . . . . . . . . .      L NEAR  0190    CODE
CON_RDND . . . . . . . . . . . . . .      L NEAR  0181    CODE
CON_READ . . . . . . . . . . . . . .      L NEAR  0192    CODE
CON_WRI1 . . . . . . . . . . . . . .      L NEAR  01A5    CODE
CON_WRIT . . . . . . . . . . . . . .      L NEAR  01A3    CODE
CON_WRST . . . . . . . . . . . . . .      L NEAR  01A1    CODE
COOUT. . . . . . . . . . . . . . . .      Number  0003
CSTAT. . . . . . . . . . . . . . . .      Number  0001
CURRENT_DOS_LOCATION . . . . . . . .      V WORD  0000            External
DEFAULT_DRIVE. . . . . . . . . . . .      V BYTE  0000            External
DERROR . . . . . . . . . . . . . . .      L NEAR  03FA    CODE
DERROR2. . . . . . . . . . . . . . .      L NEAR  040F    CODE
DERROR3. . . . . . . . . . . . . . .      L NEAR  041D    CODE
DERRTAB. . . . . . . . . . . . . . .      L BYTE  0420    CODE
DEVICE_LIST. . . . . . . . . . . . .      V DWORD 0000            External
```

```
DSKDEV . . . . . . . . . . . . . . . .          L NEAR  0049    CODE
```

```
DSKIN1 . . . . . . . . . . . . . . . .          L NEAR  0398    CODE
DSKOU1 . . . . . . . . . . . . . . . .          L NEAR  036A    CODE
DSKREAD. . . . . . . . . . . . . . . .          L NEAR  0384    CODE
DSKTBL . . . . . . . . . . . . . . . .          L NEAR  005B    CODE
DSK_CO2. . . . . . . . . . . . . . . .          L NEAR  02F2    CODE
DSK_CO4. . . . . . . . . . . . . . . .          L NEAR  0303    CODE
DSK_CO5. . . . . . . . . . . . . . . .          L NEAR  0339    CODE
DSK_CO6. . . . . . . . . . . . . . . .          L NEAR  033E    CODE
DSK_CO7. . . . . . . . . . . . . . . .          L NEAR  03CC    CODE
DSK_COM. . . . . . . . . . . . . . . .          L NEAR  02DD    CODE
DSK_INIT . . . . . . . . . . . . . . .          L NEAR  0299    CODE
DSK_INT. . . . . . . . . . . . . . . .          L NEAR  0104    CODE
DSK_OK . . . . . . . . . . . . . . . .          L NEAR  03C7    CODE
DSK_RED. . . . . . . . . . . . . . . .          L NEAR  02D7    CODE
DSK_WRT. . . . . . . . . . . . . . . .          L NEAR  02DB    CODE
DSK_WRV. . . . . . . . . . . . . . . .          L NEAR  02DB    CODE
ENTRY. . . . . . . . . . . . . . . . .          L NEAR  0108    CODE
EOI. . . . . . . . . . . . . . . . . .          Alias   BIT5
ERR_EXIT . . . . . . . . . . . . . . .          L NEAR  013D    CODE
EXIT . . . . . . . . . . . . . . . . .          L NEAR  0142    CODE
EXIT1. . . . . . . . . . . . . . . . .          L NEAR  0144    CODE
EXITP. . . . . . . . . . . . . . . . .          F PROC  0142    CODE    Length =0014
FINAL_DOS_LOCATION . . . . . . . . . .          V WORD  0000            External
FNERR. . . . . . . . . . . . . . . . .          Number  0000
FORMAT . . . . . . . . . . . . . . . .          Number  000B
GEMEXI . . . . . . . . . . . . . . . .          L NEAR  03A8    CODE
GET_BP5. . . . . . . . . . . . . . . .          L NEAR  02BF    CODE
GET_BP6. . . . . . . . . . . . . . . .          L NEAR  02D1    CODE
GET_BPB. . . . . . . . . . . . . . . .          L NEAR  02B4    CODE
HWINI1 . . . . . . . . . . . . . . . .          L NEAR  04BA    CODE
HWINIT . . . . . . . . . . . . . . . .          L NEAR  04A0    CODE
ICW4 . . . . . . . . . . . . . . . . .          Alias   BIT0
INDAT. . . . . . . . . . . . . . . . .          L NEAR  043D    CODE
INDAT1 . . . . . . . . . . . . . . . .          L NEAR  0441    CODE
INIT . . . . . . . . . . . . . . . . .          L NEAR  0000    CODE    Global
INITTAB. . . . . . . . . . . . . . . .          L NEAR  02A2    CODE
INTVECTOR. . . . . . . . . . . . . . .          L WORD  0020    INTSEG
INT_END. . . . . . . . . . . . . . . .          L NEAR  0494    CODE
IOPB . . . . . . . . . . . . . . . . .          L 000B  0252    CODE
LATCH. . . . . . . . . . . . . . . . .          Number  0000
LDDRIV1. . . . . . . . . . . . . . . .          L 001E  025D    CODE
LDDRIV2. . . . . . . . . . . . . . . .          L 001E  027B    CODE
LPOUT. . . . . . . . . . . . . . . . .          Number  0005
LPSTS. . . . . . . . . . . . . . . . .          Number  0004
LTIM . . . . . . . . . . . . . . . . .          Alias   BIT3
MEDIA1 . . . . . . . . . . . . . . . .          L NEAR  02A8    CODE
MEDIAC . . . . . . . . . . . . . . . .          L NEAR  02A6    CODE
MEMORY_SIZE. . . . . . . . . . . . . .          V WORD  0000            External
MODE0. . . . . . . . . . . . . . . . .          Number  0000
MODE1. . . . . . . . . . . . . . . . .          Alias   BIT1
MODE2. . . . . . . . . . . . . . . . .          Alias   BIT2
MODE3. . . . . . . . . . . . . . . . .          Number  0006
MODE4. . . . . . . . . . . . . . . . .          Alias   BIT3
MODE5. . . . . . . . . . . . . . . . .          Number  000A
MS . . . . . . . . . . . . . . . . . .          Alias   BIT2
OUTCHR . . . . . . . . . . . . . . . .          L NEAR  01B0    CODE
OUTDAT . . . . . . . . . . . . . . . .          L NEAR  0432    CODE
OUTIN. . . . . . . . . . . . . . . . .          L NEAR  043A    CODE
PICR0. . . . . . . . . . . . . . . . .          Number  FFE0
PICR1. . . . . . . . . . . . . . . . .          Number  FFE1
PIOIN. . . . . . . . . . . . . . . . .          Number  FFEA
PIOOUT . . . . . . . . . . . . . . . .          Number  FFEA
```

```
PRINT. . . . . . . . . . . . . . . .          L NEAR  04A3     CODE


 The Microsoft MACRO Assembler               10-08-85      PAGE    Symbols-4
IO.SYS for the alphaTronic P30


PRINT1 . . . . . . . . . . . . . . .          L NEAR  04A8     CODE
PRNDEV . . . . . . . . . . . . . . .          L NEAR  0027     CODE
PRNTBL . . . . . . . . . . . . . . .          L NEAR  00C3     CODE
PRN_INT. . . . . . . . . . . . . . .          L NEAR  00F8     CODE
PRN_STA. . . . . . . . . . . . . . .          L NEAR  01B8     CODE
PRN_WR1. . . . . . . . . . . . . . .          L NEAR  01C1     CODE
PRN_WRT. . . . . . . . . . . . . . .          L NEAR  01BF     CODE
PTRSAV . . . . . . . . . . . . . . .          L DWORD 00DD     CODE
READ . . . . . . . . . . . . . . . .          Number  0009
REINI. . . . . . . . . . . . . . . .          F PROC  049F     CODE    Length =0001
RE_INIT. . . . . . . . . . . . . . .          L NEAR  049F     CODE    Global
RLLMSB . . . . . . . . . . . . . . .          Number  0030
RLLSB. . . . . . . . . . . . . . . .          Alias   BIT4
RLMSB. . . . . . . . . . . . . . . .          Alias   BIT5
SEL0 . . . . . . . . . . . . . . . .          Number  0000
SEL1 . . . . . . . . . . . . . . . .          Number  0040
SEL2 . . . . . . . . . . . . . . . .          Number  0080
SENCHR . . . . . . . . . . . . . . .          L NEAR  042D     CODE
SETKEY . . . . . . . . . . . . . . .          Number  000C
SETPRN . . . . . . . . . . . . . . .          Number  000E
SETSIO . . . . . . . . . . . . . . .          Number  000D
SNDPAR . . . . . . . . . . . . . . .          L NEAR  03D1     CODE
SNFM . . . . . . . . . . . . . . . .          Alias   BIT4
SNGL . . . . . . . . . . . . . . . .          Alias   BIT1
STRATEGY . . . . . . . . . . . . . .          L NEAR  00E1     CODE
STRATP . . . . . . . . . . . . . . .          F PROC  00E1     CODE    Length =000B
SYSINIT. . . . . . . . . . . . . . .          L FAR   0000             External
TIMCMD . . . . . . . . . . . . . . .          Number  FFE7
TIMDEV . . . . . . . . . . . . . . .          L NEAR  0039     CODE
TIMER0 . . . . . . . . . . . . . . .          Number  FFE4
TIMER1 . . . . . . . . . . . . . . .          Number  FFE5
TIMER2 . . . . . . . . . . . . . . .          Number  FFE6
TIMTBL . . . . . . . . . . . . . . .          L NEAR  00A9     CODE
TIM_DAYS . . . . . . . . . . . . . .          L NEAR  022A     CODE
TIM_HRS. . . . . . . . . . . . . . .          L NEAR  022D     CODE
TIM_HSEC . . . . . . . . . . . . . .          L NEAR  022E     CODE
TIM_INT. . . . . . . . . . . . . . .          L NEAR  00FE     CODE
TIM_MINS . . . . . . . . . . . . . .          L NEAR  022C     CODE
TIM_RED. . . . . . . . . . . . . . .          L NEAR  0245     CODE
TIM_SECS . . . . . . . . . . . . . .          L NEAR  022F     CODE
TIM_WRT. . . . . . . . . . . . . . .          L NEAR  0230     CODE
UPM. . . . . . . . . . . . . . . . .          Alias   BIT0
WRITE. . . . . . . . . . . . . . . .          Number  000A
Y. . . . . . . . . . . . . . . . . .          Number  0000

Warning Severe
Errors  Errors
0       0
```