

TITLE W2X15Q.SYS for the alphaTroni  
c P30 + R0204 (REST)

PAGE 60,132

Hard Disk Drive for Version 2.x of MSDOS.

= 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

= FFEA

= FFEA

= FFE9

-----  
PIOOUT EQU OFFFEAH ;Port to output one Byte to the 8085  
PIOIN EQU OFFFEAH ;Port to input one Byte from the 8085  
PIOSTS EQU OFFFE9H ;Port for Bufferflags (IBF & OBF)  
;OBF is connected to TEST Input of the CPU,  
;IBF is connected to INT2 of PIC 8259A

= 0010

= 0011

= 0012

= 0014

hd\_read equ 10H  
hd\_writ equ 11H  
HD\_STS EQU 12H ;get Status, if Winni on-line  
HD\_INIT EQU 14H ;init. XEBEC  
-----

0000

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



W2X19Q.SYS for the alphaTronic P30 + R0204 (REST)  
Device driver tables.

PAGE

SUBTTL Dispatch tables for each device.

0013	0183	R	DSK_TBL:DW	DSK_INI	!0	- Initialize Driver.
0015	00BC	R	DW	MEDIAC	!1	- Return current media code.
0017	00C7	R	DW	GET_BP	!2	- Get Bios Parameter Block.
0019	007F	R	DW	CMDERR	!3	- Reserved. (currently returns error)
001B	00D7	R	DW	DSK_RED	!4	- Block read.
001D	007B	R	DW	BUS_EXIT	!5	- (Not used; return busy flag)
001F	0086	R	DW	EXIT	!6	- Return status. (Not used)
0021	0086	R	DW	EXIT	!7	- Flush input buffer. (Not used.)
0023	00DD	R	DW	DSK_WRT	!8	- Block write.
0025	00DD	R	DW	DSK_WRV	!9	- Block write with verify.
0027	0086	R	DW	EXIT	!10	- Return output status.
0029	0086	R	DW	EXIT	!11	- Flush output buffer. (Not used.)
002B	0086	R	DW	EXIT	!12	- IO Control.

Dispatch tables for each device.

```

                                PAGE
                                SUBTTL Strategy and Software Interrupt routines.

; 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 E   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

002D  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.

0031  STRATP  PROC  FAR

0031  STRATEGY:
0031  2E: 89 1E 002D R  MOV  WORD PTR CS:[PTRSAV],BX
0036  2E: 8C 06 002F R  MOV  WORD PTR CS:[PTRSAV+2],ES
003B  CB  RET

003C  STRATP  ENDP

003C  DSK_INT:
003C  56  PUSH  SI
003D  BE 0013 R  MOV  SI,OFFSET DSK_TBL

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

0040  50  ENTRY:  PUSH  AX           ;Save all nessacary registers.
0041  51  PUSH  CX
0042  52  PUSH  DX
0043  57  PUSH  DI
0044  55  PUSH  BP
0045  1E  PUSH  DS
0046  06  PUSH  ES
0047  53  PUSH  BX

0048  2E: C5 1E 002D R  LDS  BX,CS:[PTRSAV] ;Retrieve pointer to I/O Packet.

004D  8A 47 01  MOV  AL,[BX.UNIT]   ;AL = Unit code.
0050  8A 67 0D  MOV  AH,[BX.MEDIA]  ;AH = Media descriptor.

```

## Strategy and Software Interrupt routines.

```
0053 8B 4F 12      MOV     CX,[BX.COUNT]    ;CX = Contains byte/sector count.
0056 8B 57 14      MOV     DX,[BX.START]   ;DX = Starting Logical sector.
0059 97             XCHG   DI,AX            ;Save Unit and Media Temporarily.
005A 8A 47 02      MOV     AL,[BX.CMD]     ;Retrieve Command type. (1 => 11)
005D 32 E4        XOR     AH,AH           ;Clear upper half of AX for calculation.
005F 03 F0        ADD     SI,AX           ;Compute entry pointer in dispatch table.
0061 03 F0        ADD     SI,AX
0063 3C 0B        CMP     AL,11           ;Verify that not more than 11 commands.
0065 77 18        JA     CMDERR           ;Ah, well, error out.

0067 97             XCHG   AX,DI
0068 C4 7F 0E      LES     DI,[BX.TRANS]   ;DI contains address of Transfer address.
                                ;ES contains segment.

006B 0E          PUSH   CS
006C 1F          POP    DS              ;Data segment same as Code segment.

006D 2E: 80 3E 0012 R 00  CMP     [WINSTS],0
0073 75 04      JNZ    DOIT             ;Winni is present

0075 B0 02      MOV     AL,2            ; not ready error
0077 EB 08      JMP     SHORT ERR_EXIT

0079 FF 24      DOIT:  JMP     [SI]        ;Perform I/O packet command.
```

PAGE

SUBTTL Common error and exit points.

```
007B          BUS_EXIT:          ;Device busy exit.
007B B4 03          MOV         AH,00000011B ;Set busy and done bits.
007D EB 09          JMP         SHORT EXIT1
007F B0 03          CMDERR: MOV     AL,3      ;Set unknown command error #.
0081          ERR_EXIT:
0081 B4 81          MOV         AH,10000001B ;Set error and done bits.
0083 F9            STC         ;Set carry bit also.
0084 EB 02          JMP         SHORT EXIT1 ;Quick way out.
0086          EXITP  PROC  FAR          ;Normal exit for device drivers.
0086 B4 01          EXIT:  MOV     AH,00000001B ;Set done bit for MSDOS.
0088 2E: C5 1E 002D R EXIT1: LDS     BX,CS:[PTRSAV]
008D 89 47 03          MOV     [BX.STATUS],AX ;Save operation complete and status.
0090 5B            POP         BX          ;Restore registers.
0091 07            POP         ES
0092 1F            POP         DS
0093 5D            POP         BP
0094 5F            POP         DI
0095 5A            POP         DX
0096 59            POP         CX
0097 58            POP         AX
0098 5E            POP         SI
0099 CB            RET
009A          EXITP  ENDP          ;RESTORE REGS AND RETURN
```

Common error and exit points.

PAGE  
SUBTTL Common Drive parameter block

DBP STRUC

```
0000 03 [    JMPNEAR DB    3 DUP (?)    ;Jump 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    ?                ;Number of heads per cylinder.
001C ????    HIDDEN  DW    ?                ;Number of hidden sectors.
```

001E DBP ENDS

```
009A 03 [    HDDRIVE DBP    (,,512,4,0,2,1024,10234,0FFH,8,17,2,0)
      ??      ]
```

```
009D 08 [
      ??      ]
```

```
00A5 0200
00A7 04
00A8 0000
00AA 02
00AB 0400
00AD 27FA
00AF FF
00B0 0008
00B2 0011
00B4 0002
00B6 0000
```

```
00B8 00A5 R    INI_TAB DW    OFFSET HDDRIVE.SEC SIZE
00BA 00A5 R    DW    OFFSET HDDRIVE.SEC SIZE
```

Common Drive parameter block

PAGE  
SUBTTL Media check routine

; Media check routine.  
; On entry:  
; AL = memory driver 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.

00BC 2E: C5 1E 002D R      MEDIAC: LDS      BX,CS:[PTRSAV]  
00C1 C6 47 0E 00            MOV      BYTE PTR [BX.TRANS],0    ;removable media  
00C5 EB BF                  JMP      EXIT

; 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.

00C7                      GET\_BPFB:  
00C7 BE 00A5 R              MOV      SI,OFFSET HDDRIVE+11  
00CA 2E: C5 1E 002D R      LDS      BX,CS:[PTRSAV]  
00CF 89 77 12              MOV      WORD PTR [BX.COUNT],SI  
00D2 8C 4F 14              MOV      WORD PTR [BX.COUNT+2],CS  
00D5 EB AF                  JMP      EXIT

PAGE  
SUBTTL    Hard Disk drive control.

```

; Disk READ/WRITE functions.

; On entry:
;     AL = Disk I/O driver 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.

```

```

00D7                   DSK_RED:
00D7   8A E0            mov     ah,al
00D9   B0 10            MOV     AL,HD_READ
00DB   EB 04            JMP     SHORT DSK_COM
00DD                   DSK_WRV:
00DD                   DSK_WRT:
00DD   8A E0            mov     ah,al
00DF   B0 11            MOV     AL,HD_WRIT
00E1                   DSK_COM:
00E1   50              push    ax

00E2   E8 0146 R        CALL    OUTDAT            ;send Function # to IOCS-85

00E5   8A C4            mov     al,ah            ;send drive # to IOCS-85
00E7   E8 0146 R        call    outdat

00EA   8A C2            MOV     AL,DL
00EC   E8 0146 R        CALL    OUTDAT

00EF   8A C6            MOV     AL,DH            ;send starting block to IOCS-85
00F1   E8 0146 R        CALL    OUTDAT

00F4   8A C1            MOV     AL,CL
00F6   E8 0146 R        CALL    OUTDAT            ;send # of sectors

00F9   FC              CLD
00FA   8B C1            MOV     AX,CX
00FC   B1 08            MOV     CL,8            ;*512/2
00FE   D3 E0            SHL     AX,CL
0100   8B C8            MOV     CX,AX           ; <CX> = # of bytes to transfer
0102   58              POP     AX

0103   3C 10            CMP     AL,HD_READ
0105   75 2D            JNZ     WRTWIN

0107   FA              CLI
0108   BA FFE9         MOV     DX,PIOSTS

```

Hard Disk drive control.

```

010B          GET_IT:
010B EC      INDAT2: IN      AL,DX
010C A8 01   TEST      AL,1
010E 75 FB   JNZ      INDAT2

0110 42      INC      DX          ;Dataregister
0111 EC      IN      AL,DX
0112 8A E0   MOV      AH,AL

0114 4A      DEC      DX          ;Statusregister
0115 EC      INDAT3: IN      AL,DX
0116 A8 01   TEST      AL,1
0118 75 FB   JNZ      INDAT3

011A 42      INC      DX          ;Dataregister
011B EC      IN      AL,DX
011C 86 E0   XCHG     AH,AL
011E AB      STOS     WORD PTR ES:[DI]
011F 4A      DEC      DX          ;Statusregister
0120 E2 E9   LOOP     GET_IT

0122 FB      WINEXI: STI
;-----
0123 BA FFE9 MOV      DX,PIDSTS
0126 EC      INDAT1: IN      AL,DX
0127 A8 01   TEST      AL,1
0129 75 FB   JNZ      INDAT1

012B 42      INC      DX
012C EC      IN      AL,DX
;-----
012D 0A C0   OR      AL,AL
012F 75 1D   JNZ      DERROR

0131 E9 0086 R JMP      EXIT          ;All done.
;-----

0134 8B F7   WRTWIN: MOV      SI,DI
0136 FA      CLI
0137 BA FFEA MOV      DX,PIDOUT
013A 26: AD  WRTWI1: LODS     WORD PTR ES:[SI]
013C 9B      WAIT      ;for OBF = 0 (1)
013D EE      OUT      DX,AL
013E 8A C4   MOV      AL,AH
0140 9B      WAIT      ;for OBF = 0 (1)
0141 EE      OUT      DX,AL
0142 E2 F6   LOOP     WRTWI1

0144 EB DC   JMP      SHORT WINEXI

0146 52      OUTDAT: PUSH     DX
0147 BA FFEA MOV      DX,PIDOUT
014A 9B      WAIT      ;for OBF = 0 (1)
014B EE      OUT      DX,AL
014C 5A      POP      DX

```





PAGE  
SUBTTL Hard Disk Drive initialization routine.

```

0183          DSK_INI:
0183 B0 14          MOV     AL,HD_INIT
0185 E8 0146 R      CALL    OUTDAT

0188 B0 80          MOV     AL,80H
018A E8 0146 R      CALL    OUTDAT

018D B9 0008        MOV     CX,8
0190 BE 0242 R      MOV     SI,OFFSET WINTAB
0193          DSK_IN_1:
0193 2E: AC          LODS   BYTE PTR CS:[SI]
0195 E8 0146 R      CALL    OUTDAT
0198 E2 F9          LOOP   DSK_IN_1

019A B0 12          MOV     AL,HD_STS
019C E8 0146 R      CALL    OUTDAT          ;request the Winni Status

019F BA FFE9        MOV     DX,PIOSTS
01A2 EC          IN      AL,DX
01A3 A8 01          TEST   AL,1
01A5 75 FB          JNZ    INDAT4

01A7 42          INC     DX
01A8 EC          IN      AL,DX
01A9 2E: A2 0012 R  MOV     WINSTS,AL          ;00, if not online (present), () 0, if present

01AD 2E: C5 1E 002D R  LDS     BX,CS:[PTRSAV]
01B2 C6 47 0D 02      MOV     BYTE PTR [BX.MEDIA],2
01B4 C7 47 0E 0183 R  MOV     WORD PTR [BX.TRANS],OFFSET DSK_INI
01BB 8C 4F 10        MOV     WORD PTR [BX.TRANS+2],CS

01BE C7 47 12 00B8 R  MOV     WORD PTR [BX.COUNT],OFFSET INI_TAB
01C3 8C 4F 14        MOV     WORD PTR [BX.COUNT+2],CS

01C6 0E          PUSH   CS
01C7 1F          POP    DS
01C8 E8 021E R      CALL   PRINT
01CB 0D 0A        DB     13,10
01CD 61 6C 70 68 61 54  DB     'alphaTronic P40 Hard Disk Driver  V1.51'
      72 6F 6E 69 63 20
      50 34 30 20 48 61
      72 64 20 44 69 73
      6B 20 44 72 69 76
      65 72 20 20 20 56
      31 2E 35 31
01F5 20 20 20 32 58 31  DB     ' 2X15Q          BS  10-Nov-84'
      53 51 20 20 20 20
      20 20 20 20 20 20
      20 20 20 42 53 20
      20 20 31 30 2D 4E
      6F 76 2D 38 34
0218 0D 0A FF        DB     13,10,-1

```

```
021B E9 0086 R          JMP     EXIT

021E 8B EC          PRINT: MOV     BP,SP
0220 87 5E 00          XCHG    BX,[BP]
0223 8A 07          PRINT1: MOV    AL,[BX]
0225 E8 0235 R          CALL   OUTCHR
0228 43              INC     BX
0229 80 3F FF          CMP     BYTE PTR [BX],-1
022C 75 F5          JNE    PRINT1

022E 43              INC     BX
022F 8B EC          MOV     BP,SP
0231 87 5E 00          XCHG    BX,[BP]
0234 C3              RET

0235 8A C8          OUTCHR: MOV    CL,AL
0237 B0 03          MOV     AL,3
0239 E8 0146 R          CALL   OUTDAT
023C 8A C1          MOV    AL,CL
023E E8 0146 R          CALL   OUTDAT
0241 C3              RET

0242 01 32          WINTAB: DB     01H,32H
0244 02              DB     02H
0245 00 80          DB     00H,80H
0247 00 40          DB     00H,40H
0249 0B              DB     0BH

024A              CODE  ENDS

                          END
```

Structures and records:

Name	Width Shift	# fields Width Mask	Initial
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		

Segments and groups:

Name	Size	align	combine class
CODE . . . . .	024A	PARA	NONE

Symbols:

Name	Type	Value	Attr
BADSIZ . . . . .	L NEAR	016F	CODE
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	

Structures and records:

Name	Width Shift	# fields Width Mask	Initial
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		

Segments and groups:

Name	Size	align	combine class
CODE . . . . .	024A	PARA	NONE

Symbols:

Name	Type	Value	Attr
BADSIZ . . . . .	L NEAR	016F	CODE
BIT0 . . . . .	Number	0001	
BIT1 . . . . .	Number	0002	
BIT10 . . . . .	Number	0400	
BIT11 . . . . .	Number	0800	
BIT12 . . . . .	Number	1000	
BIT13 . . . . .	Number	2000	
BIT14 . . . . .	Number	4000	
BIT15 . . . . .	Number	8000	
BIT2 . . . . .	Number	0004	
BIT3 . . . . .	Number	0008	
BIT4 . . . . .	Number	0010	
BIT5 . . . . .	Number	0020	
BIT6 . . . . .	Number	0040	
BIT7 . . . . .	Number	0080	
BIT8 . . . . .	Number	0100	
BIT9 . . . . .	Number	0200	

BUS_EXIT . . . . .	L NEAR 007B	CODE
CMDERR . . . . .	L NEAR 007F	CODE
DERROR . . . . .	L NEAR 014E	CODE
DERROR2 . . . . .	L NEAR 0163	CODE
DERROR3 . . . . .	L NEAR 0171	CODE
DERRTAB . . . . .	L BYTE 0176	CODE
DOIT . . . . .	L NEAR 0079	CODE
DSKDEV . . . . .	L NEAR 0000	CODE
DSK_COM . . . . .	L NEAR 00E1	CODE
DSK_INI . . . . .	L NEAR 0183	CODE
DSK_INT . . . . .	L NEAR 003C	CODE
DSK_IN_1 . . . . .	L NEAR 0193	CODE
DSK_RED . . . . .	L NEAR 00D7	CODE
DSK_TBL . . . . .	L NEAR 0013	CODE
DSK_WRT . . . . .	L NEAR 00DD	CODE
DSK_WRV . . . . .	L NEAR 00DD	CODE
ENTRY . . . . .	L NEAR 0040	CODE
ERR_EXIT . . . . .	L NEAR 0081	CODE
EXIT . . . . .	L NEAR 0086	CODE
EXIT1 . . . . .	L NEAR 0088	CODE
EXITP . . . . .	F PROC 0086	CODE
GET_BPB . . . . .	L NEAR 00C7	CODE
GET_IT . . . . .	L NEAR 010B	CODE
HDDRIVE . . . . .	L 001E 009A	CODE
HD_INIT . . . . .	Number 0014	
HD_READ . . . . .	Number 0010	
HD_STS . . . . .	Number 0012	
HD_WRIT . . . . .	Number 0011	
INDAT1 . . . . .	L NEAR 0126	CODE
INDAT2 . . . . .	L NEAR 010B	CODE
INDAT3 . . . . .	L NEAR 0115	CODE
INDAT4 . . . . .	L NEAR 01A2	CODE
INI_TAB . . . . .	L WORD 00B8	CODE
MEDIAC . . . . .	L NEAR 00BC	CODE
MEMMAX . . . . .	L BYTE 000A	CODE
OUTCHR . . . . .	L NEAR 0235	CODE
OUTDAT . . . . .	L NEAR 0146	CODE
PIOIN . . . . .	Number FFEA	
PIOOUT . . . . .	Number FFEA	
PIOSTS . . . . .	Number FFE9	
PRINT . . . . .	L NEAR 021E	CODE
PRINT1 . . . . .	L NEAR 0223	CODE
PTRSAV . . . . .	L DWORD 002D	CODE
STRATEGY . . . . .	L NEAR 0031	CODE
STRATP . . . . .	F PROC 0031	CODE
WINEXI . . . . .	L NEAR 0122	CODE
WINSTS . . . . .	L BYTE 0012	CODE
WINTAB . . . . .	L NEAR 0242	CODE
WRTWI1 . . . . .	L NEAR 013A	CODE
WRTWIN . . . . .	L NEAR 0134	CODE
Y . . . . .	Number 0000	

Length =0014

Length =000B

Warning Severe  
Errors Errors  
0 0

