

APPENDIX A — IMPLEMENTATION ON IMS EQUIPMENT

Drivers furnished for IMS Equipment

BRT4420 Baud-rate tables for IMS 442/480 with baud-rate oscillator.
BRT442S Baud-rate tables for IMS 442 without baud-rate oscillator.
BRT740 Baud-rate tables for IMS 740 slave board.

CON192 Console driver for ASCII CRT at 19,200 baud.
CON96 Console driver for ASCII CRT at 9,600 baud.

DSK401 Disk driver for IMS 401 8-inch floppy disk controller.
DSK431 Disk driver for IMS 431 5-inch floppy disk controller.
DSK490 Disk driver for IMS 490 hard disk controller.
DSKIMS58 Disk driver for both IMS 401 and 431 floppy disk controllers.
DSKM10 Disk driver for Morrow 10 Mb Winchester.
DSKM20 Disk driver for Morrow 20 Mb Winchester.
DSKM26 Disk driver for Morrow 26 Mb Winchester.
DSKFMT5 Disk specification tables for 5-inch floppy disks.
DSKFMT8 Disk specification tables for 8-inch floppy disks.
DSKFMTA Disk specification tables for both 5-inch and 8-inch floppy disks.

HDWNIT Hardware initialization.

LST300 Printer driver for no handshaking, 300 baud (e.g., Teletype 43).
LSTCTS Printer driver for CTS handshaking, 9600 baud (e.g., TI-810).
LSTETX Printer driver for ETX/ACK handshaking, 1200 baud (e.g., NEC 5510).
LSTXON Printer driver for XON/XOFF handshaking, 1200 baud (e.g., Diablo 630).
LSTIMS Printer driver for Centronics parallel.
LSTNEC Printer driver for parallel NEC 5500D.

MPENIT Memory parity initialization for IMS 461 64K RAM board.

N740M Network driver for master, using IMS 740 slave boards.
N740S Network driver for slaves, using IMS 740 slave boards.
NET80M Network driver for master, using MuSYS NET/80 slave boards.
NET80S Network driver for slaves, using MuSYS NET/80 slave boards.

RTC442 Real-time clock driver for IMS 442 ROM-I/O board.

Configuration Guide to TurboDOS

Copyright (C) 1981 by Software 2000 Inc.

Implementation on IMS Equipment

- SER480 Serial subroutines for IMS 480 4-port serial board.
- SP442 Serial and parallel subroutines for IMS 442 ROM-I/O board.
- SP740 Serial and parallel subroutines for IMS 740 slave board.
- SPN80 Serial and parallel subroutines for MuSYS NET/80.

Symbolic Patch Points in IMS Driver Modules

Parameters in the IMS hardware-dependent driver modules which may be useful to patch include the following, shown with their standard values:

In CON96 Modules:

CONBR = 8E Baud rate code (9,600 baud w/attention detect)
 FFCHR = 0C Clear-the-screen character at cold-start

In CON192 Modules:

CONBR = 8F Baud rate code (19,200 baud w/attention detect)
 FFCHR = 0C Clear-the-screen character at cold-start

In LST300 Modules:

LST3BR = 25 Baud rate code (300 baud, output-only)
 LST3FF = 0C Top-of-form character at end-of-print

In LSTCTS Modules:

CTSBR = 6E Baud rate code (9,600 baud, CTS handshaking)
 CTSFF = 0C Top-of-form character at end-of-print

In LSTETX Modules:

ETXBR = 07 Baud rate code (1,200 baud, input/output)
 ETXFF = 0C Top-of-form character at end-of-print
 ETXLEN = 8C Length of output between ETXs
 ETXSEQ = 03 Length of maximal escape sequence

In LSTIMS Modules:

IMSFF = 0C Top-of-form character at end-of-print

In LSTXON Modules:

XONBR = 07 Baud rate code (1,200 baud, input/output)
 XONFF = 0C Top-of-form character at end-of-print

In N740M Module:

SLVPAT = 40,44,48,4C,50,54,58,5C,60,64,68,6C,70,74,78,7C
 Slave board port assignment table

In NET80M Module:

N80PAT = 20,22,24,26,28,2A,2C,2E,30,32,34,36,38,3A,3C,3E
 Slave board port assignment table

Configuration Guide to TurboDOS

Copyright (C) 1981 by Software 2000 Inc.

Implementation on IMS Equipment

In SP442 Modules

SERBSZ = 40 Type-ahead buffer size

In SP740 Modules

SERBSZ = 40 Type-ahead buffer size

IMS Logic Board Set-Up for TurboDOS

401 8" Floppy Disk Controller

- Shunt JA Address to port 80-8F hex (standard etch)
JB Install horizontal shunt at "5" (interrupts)
JC Install two horizontal shunts at top and bottom (precomp)
JD Install horizontal shunt (delay complete)

431 5" Floppy Disk Controller

- Shunt JA Address to port C0-CF hex (cut trace, install jumper)
JB Install horizontal shunt at "5" (interrupts)
JC Install vertical shunt (two-sided) if MPI 52 drives installed
Remove IC at "5-B" (74LS74), lift pin 13 clear of socket, connect to pin 12 on the IC itself, and re-install in socket (ready logic modification).

442 I/O Board

- Shunt JA Install horizontal shunt on topmost pair
JB Install vertical shunt
JC For parallel NEC printer, install horizontal shunts on all 16 pairs
JD Standard etch
JE Standard etch
JF Standard etch
JG Install only 1 horizontal shunt on bottom pair (ROM enable at FC00 hex)
JH Cut trace between top pair, jumper bottom pair (timer enable)
JJ Install horizontal shunt at "V11"
JK Install horizontal shunt at "V13"
JL Install horizontal shunt at "V13"
JM No shunt
JN No shunt
JP Install 3 horizontal shunts at "A7", "A6", and "A5" (not "A4")

Install oscillator at "13-D" and install 74LS161 IC at "13-C". Install TurboDOS boot PROM at "6.5-B". For parallel NEC printer, install 220/330 resistor pack at IC "12-A".

450 Z80 CPU Board

- Shunt JA No shunt (wait states)
JB Install 2 horizontal shunts on bottom (Jump to FC00 hex)
JC Standard etch (4 MHz)
JD Standard etch

For accurate real-time clock performance, add 10 pf capacitor (mica or NPO) in series with one leg of 16 Mhz crystal.

Configuration Guide to TurboDOS

Copyright (C) 1981 by Software 2000 Inc.

Implementation on IMS Equipment

461 64K Dynamic RAM Board

- Shunt JA Install vertical shunt on top pair (high-speed RAM)
 JB Install horizontal shunt (I/O port enabled)
 JC Install 8 horizontal shunts (port 00)
 JD Install shunt (phantom)
 JE No shunt (unmapped)
 JF No shunt (Z80 timing)
 JG No shunt (vectored interrupt)
 JH Install horizontal shunt on left pair (no front panel)

480 Four-Port Serial Board

- Shunt JA No shunts (port address E0-FF hex)
 JB Install horizontal shunt on bottom pair (use oscillator)
 JC Install horizontal shunt at "VB" (vectored interrupt)
 JD Install horizontal shunt at "VB" (vectored interrupt)
 JE Install horizontal shunt at "VB" (vectored interrupt)
 JF Install horizontal shunt at "VB" (vectored interrupt)

490 Cartridge Module Drive Controller

- Shunt JA Standard etch (DMA channel 2)
 JC Standard etch (address port 90-97 hex)
 JD Install vertical shunt at "VI4" (vectored interrupt)

740 I/O Processor Board (Slave uP)

- Shunt JA Address boards to hex 40, 44, 48, 4C, etc.
 JB No shunt (disable vectored interrupt)

APPENDIX B — IMPLEMENTATION ON TRS-80 MODEL II

Drivers furnished for TRS-80 Model II

- CONTRS Console driver for TRS-80 keyboard/display.
- DSKTRS Disk driver for TRS-80 8-inch floppy disk controller.
- DSKFMT8 Disk specification tables for 8-inch floppy disks.
- HDWNIT Hardware initialization.
- LSTTRS Printer driver for TRS-80 Centronics parallel interface.
- LST300 Printer driver for no handshaking, 300 baud (e.g., Teletype 43).
- LSTCTS Printer driver for CTS handshaking, 9600 baud (e.g., TI-810).
- LSTETX Printer driver for ETX/ACK handshaking, 1200 baud (e.g., NEC 5510).
- LSTXON Printer driver for XON/XOFF handshaking, 1200 baud (e.g., Diablo 630).
- RTCTRS Real-time clock driver for TRS-80 CTC.
- SPTRS Serial and parallel subroutines for TRS-80.

Configuration Guide to TurboDOS

Copyright (C) 1981 by Software 2000 Inc.

Implementation on TRS-80 Model II

Symbolic Patch Points in TRS-80 Model II Driver Modules

Parameters in the TRS-80 Model II hardware-dependent driver modules which may be useful to patch include the following, shown with their standard values:

In LSTTRS Modules:

TRSFF = 0C Top-of-form character at end-of-print

In LST300 Modules:

LST3BR = 25 Baud rate code (300 baud, output-only)

LST3FF = 0C Top-of-form character at end-of-print

In LSTCTS Modules:

CTSBR = 6E Baud rate code (9,600 baud, CTS handshaking)

CTSFF = 0C Top-of-form character at end-of-print

In LSTETX Modules:

ETXBR = 07 Baud rate code (1,200 baud, input/output)

ETXFF = 0C Top-of-form character at end-of-print

ETXLEN = 8C Length of output between ETXs

ETXSEQ = 03 Length of maximal escape sequence

In LSTXON Modules:

XONBR = 07 Baud rate code (1,200 baud, input/output)

XONFF = 0C Top-of-form character at end-of-print

In SPTRS Modules:

SERBSZ = 40 Type-ahead buffer size

Configuration Guide to TurboDOS
Copyright (C) 1981 by Software 2000 Inc.
Sample Driver Listings

APPENDIX C — SAMPLE DRIVER LISTINGS

California State Bar
Copyright © 1981 by California State Bar
San Francisco, California

APPENDIX II - SAMPLE DRIVER LICENSE

QUATE - TURBODOS OPERATING SYSTEM SYMBOLIC EQUIVALENCES
 Copyright (C) 1981 by Software 2000, Inc.

```

;
; IDENT EQUATE
;
; ASCII EQUIVALENCES
;
0000 ANUL      == 00H  ;NULL
0001 ASCH     == 01H  ;SOH
0002 ASTX     == 02H  ;STX
0003 AETX     == 03H  ;ETX
0004 AEOT     == 04H  ;EOT
0005 AENQ     == 05H  ;ENQ
0006 AACK     == 06H  ;ACK
0007 ABEL     == 07H  ;BELL
0008 ABS      == 08H  ;BS
0009 AHT      == 09H  ;HT
000A ALF      == 0AH  ;LF
000B AVT      == 0BH  ;VT
000C AFF      == 0CH  ;FF
000D ACR      == 0DH  ;CR
000E ASO      == 0EH  ;SO
000F ASI      == 0FH  ;SI
0010 ADLE     == 10H  ;DLE
0011 ADC1     == 11H  ;DC1
0012 ADC2     == 12H  ;DC2
0013 ADC3     == 13H  ;DC3
0014 ADC4     == 14H  ;DC4
0015 ANAK     == 15H  ;NAK
0016 ASYN     == 16H  ;SYN
0017 AETB     == 17H  ;ETB
0018 ACAN     == 18H  ;CAN
0019 AEM      == 19H  ;EM
001A ASUB     == 1AH  ;SUB
001B AESC     == 1BH  ;ESC
001C AFS      == 1CH  ;FS
001D AGS      == 1DH  ;GS
001E ARS      == 1EH  ;RS
001F AUS      == 1FH  ;US
0020 ASP      == 20H  ;SPACE
007F ARUB     == 7FH  ;RUBOUT (DEL)

```

EQUATE - TURBODOS OPERATING SYSTEM SYMBOLIC EQUIVALENCES
Copyright (C) 1981 by Software 2000, Inc.

```

;
0000      WBOOT      == 0000H      ;WARM START ENTRYPOINT
0003      IOBYTE     == 0003H      ;I/O CONFIGURATION BYTE
0004      CURDRV     == 0004H      ;CURRENT DEFAULT DRIVE
0005      OPSYS      == 0005H      ;OPERATING SYSTEM ENTRYPOINT
005C      TFCB       == 005CH      ;DEFAULT FILE CONTROL BLOCK
0080      TBUF       == 0080H      ;DEFAULT DISK BUFFER ADDRESS
0100      TPA        == 0100H      ;TRANSIENT PROGRAM AREA BASE
;
0000      ;          .LOC      0      ;WORKING STORAGE RELATIVE TO 0
;
0000      PDRDP:     ;          ;PD REQUEST DESCRIPTOR PACKET
0000      PDRFCN:   .BLKB      1      ;PD REQUEST FUNCTION NUMBER
0001      PDRDRV:   .BLKB      1      ;PD REQUEST DRIVE NUMBER
0002      PDRTRK:   .BLKW      1      ;PD REQUEST TRACK NUMBER
0004      PDRSEC:   .BLKW      1      ;PD REQUEST SECTOR NUMBER
0006      PDRSC:    .BLKW      1      ;PD REQUEST SECTOR COUNT
0008      PDRTC:    .BLKW      1      ;PD REQUEST TRANSFER COUNT
000A      PDRDMA:   .BLKW      1      ;PD REQUEST DMA ADDRESS
000C      PDRDST:   .BLKW      1      ;PD REQUEST DRIVE SPEC TABLE A
DDR
000E      PDRLEN    == .-PDRDP      ;PD REQUEST DESCRIPTOR PACKET
LENGTH
000E      DSKNFO:   ;          ;DISK TYPE INFORMATION
000E      BLKSIZ:   .BLKB      1      ;BLOCK SIZE
000F      NMBLKS:   .BLKW      1      ;NUMBER OF BLOCKS
0011      NMBDIR:   .BLKB      1      ;NUMBER OF DIRECTORY BLOCKS
0012      SECSIZ:   .BLKB      1      ;PHYSICAL SECTOR SIZE (2^N*128
)
0013      SECTRK:   .BLKW      1      ;PHYSICAL SECTORS PER TRACK
0015      TRKDSK:   .BLKW      1      ;PHYSICAL TRACKS PER DISK
0017      RESTRK:   .BLKW      1      ;NUMBER OF RESERVED TRACKS
000B      DNFOL     == .-DSKNFO      ;DISK INFO LENGTH
;
.END

```

HDWNIT - TURBODOS OPERATING SYSTEM HARDWARE INITIALIZATION
 COPYRIGHT (C) 1981, SOFTWARE 2000, INC.

```

;
; COPYRIGHT (C) 1981, SOFTWARE 2000, INC.
;
; AUTHORS: RONALD E. RAIKES
;          MICHAEL D. BUSCH
;
; VERSION: 09/08/81
;
; IDENT HDWNIT ;MODULE ID
;
; INSERT DREQUATE ;DRIVER SYMBOLIC EQUIVALENCES
;
0000' .LOC .PROG.# ;LOCATE IN PROGRAM AREA
;
0000' 0020 .WORD NITLEN+2 ;INITIALIZATION CODE LENGTH
;
0002' CD 0000:04 HDWNIT::CALL MPENIT# ;INITIALIZE MEMORY PARITY
0005' CD 0000:05 CALL SPINIT# ;INITIALIZE SERIAL/PARALLEL I/O
0008' CD 0000:06 CALL RTCNIT# ;INITIALIZE REAL TIME CLOCK
000B' CD 0000:07 CALL DSKINA# ;INITIALIZE DISK DEVICE A
000E' CD 0000:08 CALL DSKINB# ;INITIALIZE DISK DEVICE B
0011' CD 0000:09 CALL DSKINC# ;INITIALIZE DISK DEVICE C
0014' CD 0000:0A CALL DSKIND# ;INITIALIZE DISK DEVICE D
0017' CD 0000:0B CALL NETNIT# ;INITIALIZE NETWORK DRIVER
001A' 21 0002' LXI H,HDWNIT ;GET INITIALIZATION CODE ADDRESS
001D' C3 0000:0C JMP DEALOC# ;DE-ALLOCATE INITIALIZATION CODE
;
001E NITLEN = .-HDWNIT ;INITIALIZATION CODE LENGTH
;
; .END

```

CON96 - TURBODOS OPERATING SYSTEM NULL CONSOLE DRIVER
 COPYRIGHT (C) 1981, SOFTWARE 2000, INC.

```

;
; COPYRIGHT (C) 1981, SOFTWARE 2000, INC.
;
; AUTHORS: RONALD E. RAIKES
;          MICHAEL D. BUSCH
;
; VERSION: 09/08/81
;
; IDENT CON96 ;MODULE ID
;
; INSERT DREQUATE ;DRIVER SYMBOLIC EQUIVALENCES
;
0000" ;
; .LOC .DATA.# ;LOCATE IN DATA AREA
;
0000" 8E CONBR:: .BYTE 0EH:80H ;CONSOLE BAUD RATE CODE (9600 BAUD)
0001" 0C FFCHR:: .BYTE AFF ;FORM FEED CHARACTER
0002" 00 INITC: .BYTE 0 ;INITIALIZATION COMPLETE FLAG
;
0000' ;
; .LOC .PROG.# ;LOCATE IN PROGRAM AREA
;
0000' 21 0002" CONDRZ::LXI H,INITC ;GET INITIALIZATION COMPLETE FLAG
0003' 7E MOV A,M
0004' B7 ORA A ;INITIALIZATION COMPLETE FLAG SET
0005' CC 0013' CZ ..INIT ;IF NOT, INITIALIZE CONSOLE BAUD RA
E
0008' 7B ..CDRV: MOV A,E ;GET FUNCTION NUMBER
0009' D608 SUI 8 ;FUNCTION NUMBER=8?
000B' 2821 JRZ CONSO ;IF SO, ERROR SHIFT OUT
000D' 3D DCR A ;FUNCTION NUMBER=9?
000E' 281E JRZ CONSI ;IF SO, ERROR SHIFT IN
0010' C3 0000:04 JMP SERIAL# ;ELSE, CONTINUE
0013' 35 ..INIT: DCR M ;SET INITIALIZATION COMPLETE FLAG
0014' C5 PUSH B ;SAVE CHANNEL NUMBER/CHARACTER
0015' D5 PUSH D ;SAVE FUNCTION NUMBER
0016' 3A 0000" LDA CONBR ;GET CONSOLE BAUD RATE CODE
0019' 4F MOV C,A ;TELEVIDEO BAUD RATE CODE TO C-REG
001A' 1E03 MVI E,3 ;SET FUNCTION NUMBER=3
001C' CD 0000:04 CALL SERIAL# ;SET CHANNEL BUAD RATE
001F' 3A 0001" LDA FFCHR ;GET FORM FEED CHARACTER
0022' B7 ORA A ;FORM FEED CHARACTER=0?
0023' 2806 JRZ ..NITX ;IF SO, CONTINUE
0025' 4F MOV C,A ;ELSE, FORM FEED CHARACTER TO C-REG
0026' 1E02 MVI E,2 ;SET FUNCTION NUMBER=2
0028' CD 0008' CALL ..CDRV ;OUTPUT FORM FEED
002B' D1 ..NITX: POP D ;RESTORE FUNCTION NUMBER
002C' C1 POP B ;RESTORE CHANNEL NUMBER/CHARACTER
002D' C9 RET ;DONE
;
002E' CONSO:
002E' CD 0000:05 CONSI: CALL DMS# ;POSITION TO NEXT LINE
0031' OD8A .ASCIS [ACR] [ALF]
0033' C9 RET ;DONE
;
.END

```

STXON - TURBODOS OPERATING SYSTEM XON/XOFF PRINTER DRIVER
 COPYRIGHT (C) 1981 BY SOFTWARE 2000, INC.

```

;
; COPYRIGHT (C) 1981 BY SOFTWARE 2000, INC.
;
; AUTHORS: RONALD E. RAIKES
;          MICHAEL D. BUSCH
;
; VERSION: 09/08/81
;
; IDENT LSTXON ;MODULE ID
;
; INSERT DREQUATE ;DRIVER SYMBOLIC EQUIVALENCES
;
0000" ;
; .LOC .DATA.# ;LOCATE IN DATA AREA
;
0000" 07 XONBR: .BYTE 7 ;BAUD RATE CODE (1200 BAUD)
0001" 0C XONFF: .BYTE AFF ;FORM FEED CHARACTER
0002" 000000000000 INITC: .BYTE [16]0 ;INITIALIZATION COMPLETE FLAGS
;
0000' ;
; .LOC .PROG.# ;LOCATE IN PROGRAM AREA
;
0000' 21 0002" LSTDR%: LXI H,INITC ;GET INITIALIZATION COMPLETE FLAGS
0003' D5 PUSH D ;SAVE FUNCTION NUMBER
0004' 58 MOV E,B ;CHANNEL NUMBER TO DE-REG
0005' 1600 MVI D,0 ;DOUBLE LENGTH
0007' 19 DAD D ;INDEX INTO FLAGS TABLE
0008' D1 POP D ;RESTORE FUNCTION NUMBER
0009' 7E MOV A,M ;GET INITIALIZATION COMPLETE FLAG
000A' B7 ORA A ;INITIALIZATION COMPLETE FLAG SET?
000B' CC 0018' CZ ..INIT ;IF NOT, INITIALIZE LIST CHANNEL
000E' 7B MOV A,E ;GET FUNCTION NUMBER
000F' FE02 CPI 2 ;FUNCTION NUMBER=2?
0011' 281A JRZ LSTOUT ;IF SO, CONTINUE
0013' FE07 CPI 7 ;FUNCTION NUMBER=7?
0015' 2810 JRZ LSTWSR ;IF SO, CONTINUE
0017' C9 RET ;ELSE, DONE
0018' 35 ;
; ..INIT: DCR M ;SET INITIALIZATION COMPLETE FLAG
0019' D5 PUSH D ;SAVE FUNCTION NUMBER
001A' C5 PUSH B ;SAVE CHANNEL NUMBER/CHARACTER
001B' 3A 0000" LDA XONBR ;GET BAUD RATE CODE
001E' 4F MOV C,A ;BAUD RATE CODE TO C-REG
001F' 1E03 MVI E,3 ;SET FUNCTION NUMBER=3
0021' CD 0000:04 CALL SERIAL# ;SET CHANNEL BUAD RATE
0024' C1 POP B ;RESTORE CHANNEL NUMBER/CHARACTER
0025' D1 POP D ;RESTORE FUNCTION NUMBER
0026' C9 RET ;DONE
;
0027' 3A 0001" LSTWSR: LDA XONFF ;GET FORM FEED CHARACTER
002A' 4F MOV C,A ;FORM FEED CHARACTER TO C-REG
002B' 1E02 MVI E,2 ;SET FUNCTION NUMBER=2
;
002D' CD 0048' LSTOUT: CALL ..SST ;GET SERIAL STATUS
0030' B7 ORA A ;CHARACTER AVAILABLE?
0031' 2812 JRZ ..OUT ;IF NOT, CONTINUE
0033' CD 0051' CALL ..SIN ;ELSE, GET SERIAL INPUT

```

SP442 - TURBODOS OPERATING SYSTEM SERIAL/PARALLEL I/O DRIVER (IMS 442)
 COPYRIGHT (C) 1981, SOFTWARE 2000, INC.

```

;
; COPYRIGHT (C) 1981, SOFTWARE 2000, INC.
;
; AUTHORS: RONALD E. RAIKES
;          MICHAEL D. BUSCH
;
; VERSION: 09/08/81
;
; IDENT SP442                ;MODULE ID
;
; INSERT DREQUATE           ;DRIVER SYMBOLIC EQUIVALENCES
;
0010 IOBASE = 10H            ;SERIAL/PARALLEL I/O PORT BASE
;
0010 S0CTRL = IOBASE+00H    ;SERIAL 0 CONTROL/STATUS REGISTER
0011 S0DATA = IOBASE+01H    ;SERIAL 0 DATA REGISTER
0012 S1CTRL = IOBASE+02H    ;SERIAL 1 CONTROL/STATUS REGISTER
0013 S1DATA = IOBASE+03H    ;SERIAL 1 DATA REGISTER
0014 TIM0 = IOBASE+04H      ;TIMER 0 DATA REGISTER
0015 TIM1 = IOBASE+05H      ;TIMER 1 DATA REGISTER
0016 TIM2 = IOBASE+06H      ;TIMER 2 DATA REGISTER
0017 TIMCTL = IOBASE+07H    ;TIMER CONTROL REGISTER
0018 SINTE = IOBASE+08H     ;SERIAL INTERRUPT ENABLE REGISTER
0019 T2RES = IOBASE+09H     ;TIMER 2 INTERRUPT RESET
001C PODATA = IOBASE+0CH    ;PARALLEL 0 DATA REGISTER
001D P1DATA = IOBASE+0DH    ;PARALLEL 1 DATA REGISTER
001E P2DATA = IOBASE+0EH    ;PARALLEL 2 DATA REGISTER
001F PPCTL = IOBASE+0FH     ;PARALLEL PORT CONTROL REGISTER
;
0000 RDA = 0                ;RECEIVED DATA AVAILABLE BIT
0001 TBE = 1                ;TRANSMIT BUFFER EMPTY BIT
0007 CTSN = 7               ;CLEAR TO SEND (NOT) BIT
;
0000 ROMDIS = 0             ;ROM DISABLE BIT
0001 RTCENA = 1             ;REAL TIME CLOCK ENABLE BIT
0002 S1TXIE = 2             ;SERIAL 1 TX INTERRUPT ENABLE BIT
0003 S1RXIE = 3             ;SERIAL 1 RX INTERRUPT ENABLE BIT
0004 S1RTSN = 4             ;SERIAL 1 REQ TO SEND (NOT) BIT
0005 S0TXIE = 5             ;SERIAL 0 TX INTERRUPT ENABLE BIT
0006 S0RXIE = 6             ;SERIAL 0 RX INTERRUPT ENABLE BIT
0007 S0RTSN = 7             ;SERIAL 0 REQ TO SEND (NOT) BIT
;
0036 TOCMD = 36H            ;TIMER 0 COMMAND
0076 T1CMD = 76H            ;TIMER 1 COMMAND
00B6 T2CMD = 0B6H          ;TIMER 2 COMMAND
;
0089 PPMODE = 89H           ;PARALLEL PORT MODE WORD
0019 SPMODE = 19H           ;SERIAL PORT MODE WORD
;PARITY INHIBIT/1 STOP BIT/8 BITS
;
FC00 BOOTPR = 0FC00H        ;BOOTSTRAP LOADER EPROM BASE
;
0000" .LOC .DATA.# ;LOCATE IN DATA AREA
;

```


STXON - TURBODOS OPERATING SYSTEM XON/XOFF PRINTER DRIVER
 COPYRIGHT (C) 1981 BY SOFTWARE 2000, INC.

```

0036: E67F ANI 7FH ;STRIP SIGN BIT
0038: FE13 CPI ADC3 ;CHARACTER=DC3 (XOFF)?
003A: 20F1 JRNZ LSTOUT ;IF NOT, WAIT
003C: CD 0051: ..WAIT: CALL ..SIN ;GET SERIAL INPUT
003F: E67F ANI 7FH ;STRIP SIGN BIT
0041: FE11 CPI ADC1 ;CHARACTER=DC1 (XON)?
0043: 20F7 JRNZ ..WAIT ;IF NOT, WAIT
0045: C3 0000:04 ..OUT: JMP SERIAL# ;OUTPUT CHARACTER
0048: C5 ..SST: PUSH B ;SAVE CHANNEL NUMBER/CHARACTER
0049: D5 PUSH D ;SAVE FUNCTION NUMBER
004A: 1E00 MVI E,0 ;SET FUNCTION NUMBER=0
004C: CD 0000:04 CALL SERIAL# ;GET SERIAL STATUS
004F: 1807 JMPR ..SSIC ;CONTINUE
0051: C5 ..SIN: PUSH B ;SAVE CHANNEL NUMBER/CHARACTER
0052: D5 PUSH D ;SAVE FUNCTION NUMBER
0053: 1E01 MVI E,1 ;SET FUNCTION NUMBER=1
0055: CD 0000:04 CALL SERIAL# ;GET SERIAL STATUS
0058: D1 ..SSIC: POP D ;RESTORE FUNCTION NUMBER
0059: C1 POP B ;RESTORE CHANNEL NUMBER/CHARACTER
005A: C9 RET ;DONE
;
;END

```

SP442 - TURBODOS OPERATING SYSTEM SERIAL/PARALLEL I/O DRIVER (IMS 442)
 COPYRIGHT (C) 1981, SOFTWARE 2000, INC.

```

0000n 0040      SERBSZ: .WORD 64      ;SERIAL BUFFER SIZE
0002n 49        INTMSK: .BYTE 1<ROMDIS!1<SORXIE!1<S1RXIE ;INTERRUPT MASK
0003n 00        SOBR:  .BYTE 0      ;SERIAL 0 BAUD RATE
0004n 00        S1BR:  .BYTE 0      ;SERIAL 1 BAUD RATE
0005n 0000      SERBUF: .WORD 0      ;SERIAL BUFFER ADDRESS
0007n 0000      SERPTR: .WORD 0      ;SERIAL BUFFER POINTER
0009n 00        SOOCHR: .BYTE 0     ;SERIAL 0 OUTPUT CHARACTER
000An 00        S1OCHR: .BYTE 0     ;SERIAL 1 OUTPUT CHARACTER
;
000Bn          ;SOISPH:          ;SERIAL 0 INPUT SEMAPHORE
000Bn 0000      .WORD 0          ;SEMAPHORE COUNT
000Dn 000Dn    ..SOIH: .WORD ..SOIH ;SEMAPHORE P/D HEAD
000Fn 000Dn    .WORD ..SOIH
;
0011n 0000      S1ISPH: .WORD 0          ;SERIAL 1 INPUT SEMAPHORE
0013n 0013n    ..S1IH: .WORD ..S1IH ;SEMAPHORE COUNT
0015n 0013n    .WORD ..S1IH ;SEMAPHORE P/D HEAD
;
0017n 0000      SOOSPH: .WORD 0          ;SERIAL 0 OUTPUT SEMAPHORE
0019n 0019n    ..SOOH: .WORD ..SOOH ;SEMAPHORE COUNT
001Bn 0019n    .WORD ..SOOH ;SEMAPHORE P/D HEAD
;
001Dn 0000      S1OSPH: .WORD 0          ;SERIAL 1 OUTPUT SEMAPHORE
001Fn 001Fn    ..S1OH: .WORD ..S1OH ;SEMAPHORE COUNT
0021n 001Fn    .WORD ..S1OH ;SEMAPHORE P/D HEAD
;
0000'          ; .LOC .PROG.# ;LOCATE IN PROGRAM AREA
;
0000' 0038      .WORD NITLEN+2 ;INITIALIZATION CODE LENGTH
;
0002' 3E89      SPINIT: .MVI A,PPMODE ;INITIALIZE 8255
0004' D31F      OUT PPCTL
0006' 3EFF      MVI A,OFFH ;CLEAR PARALLEL PORTS
0008' D31C      OUT PODATA
000A' D31D      OUT P1DATA
000C' 3E19      MVI A,SPMODE ;INITIALIZE UARTS
000E' D310      OUT SOCTRL
0010' D312      OUT S1CTRL
0012' 3EC3      MVI A,JMP ;SET UP SERIAL 0 INTERRUPT VECTOR
0014' 32 0018   STA 3*8
0017' 21 0107'  LXI H,SERISR
001A' 22 0019   SHLD (3*8)+1
001D' 3A 0002n LDA INTMSK ;GET INTERRUPT MASK
0020' D318      OUT SINTE ;ENABLE INTERRUPT MASKS
0022' 2A 0000n LHLD SERBSZ ;GET SERIAL BUFFER SIZE
0025' 29        DAD H ;X2
0026' CD 0000:04 CALL ALLOC# ;ALLOCATE PACKET FOR SERIAL BUFFER
0029' 22 0005n SHLD SERBUF ;SAVE SERIAL BUFFER ADDRESS
002C' 22 0007n SHLD SERPTR ;SET SERIAL BUFFER POINTER
002F' CD 0000:05 CALL NIT480# ;INITIALIZE IMS 480 SERIAL PORTS
0032' 21 0002'  LXI H,SPINIT ;GET INITIALIZATION CODE ADDRESS

```

P442 - TURBODOS OPERATING SYSTEM SERIAL/PARALLEL I/O DRIVER (IMS 442)
 COPYRIGHT (C) 1981, SOFTWARE 2000, INC.

```

0035'   C3 0000:06           JMP      DEALOC# ;DE-ALLOCATE INITIALIZATION CODE
0036                                     ;
                                ;NITLEN = .-SPINIT           ;INITIALIZATION CODE LENGTH
                                ;
0038'   7B                   ;SERIAL::MOV      A,E           ;GET FUNCTION NUMBER
0039'   B7                   ORA      A           ;FUNCTION NUMBER=0?
003A'   2817                 JRZ     SERST        ;IF SO, CONTINUE
003C'   3D                   DCR     A           ;FUNCTION NUMBER=1?
003D'   282A                 JRZ     SERIN        ;IF SO, CONTINUE
003F'   3D                   DCR     A           ;FUNCTION NUMBER=2?
0040'   284A                 JRZ     SEROUT       ;IF SO, CONTINUE
0042'   3D                   DCR     A           ;FUNCTION NUMBER=3?
0043'   CA 01A2'            JZ     SERSBR       ;IF SO, CONTINUE
0046'   3D                   DCR     A           ;FUNCTION NUMBER=4?
0047'   CA 01D5'            JZ     SERRBR       ;IF SO, CONTINUE
004A'   3D                   DCR     A           ;FUNCTION NUMBER=5?
004B'   CA 01E4'            JZ     SERSMC       ;IF SO, CONTINUE
004E'   3D                   DCR     A           ;FUNCTION NUMBER=6?
004F'   CA 0208'            JZ     SERRMC       ;IF SO, CONTINUE
0052'   C9                   RET                    ;ELSE, DONE

0053'   2A 0005"           ;SERST: LHL      SERBUF      ;GET SERIAL BUFFER ADDRESS
0056'   ED5B 0007"        LDED     SERPTR      ;GET SERIAL BUFFER POINTER
005A'   E5                   ;..STL: PUSH     H           ;SAVE SERIAL BUFFER ADDRESS
005B'   AF                   XRA     A           ;CLEAR CARRY/PRESET RETURN CODE=0
005C'   ED52                 DSB     D           ;END OF SERIAL BUFFER?
005E'   E1                   POP     H           ;RESTORE SERIAL BUFFER ADDRESS
005F'   C8                   RZ                    ;IF END OF SERIAL BUFFER, DONE
0060'   78                   MOV     A,B         ;ELSE, GET CHANNEL NUMBER
0061'   96                   SUB     M           ;NEXT CHARACTER=REQUESTED CHANNEL
0062'   23                   INX     H           ;ADVANCE TO CHARACTER
0063'   4E                   MOV     C,M         ;GET CHARACTER FROM BUFFER
0064'   23                   INX     H           ;ADVANCE TO NEXT CHANNEL NUMBER
0065'   2F                   CMA                    ;PRESET RETURN CODE=OFFH
0066'   C8                   RZ                    ;IF REQUESTED CHANNEL, DONE
0067'   18F1                 JMPR    ..STL       ;CONTINUE

0069'   78                   ;SERIN: MOV     A,B         ;GET CHANNEL NUMBER
006A'   FE02                 CPI     2           ;CHANNEL NUMBER=0/1?
006C'   3805                 JRC     ..S01I      ;IF SO, CONTINUE
006E'   CD 0000:07         CALL    IN480#      ;ELSE, GET IMS 480 IN SEMAPHORE
0071'   1809                 JMPR    ..ICOM      ;CONTINUE
0073'   21 000B"           ;..S01I: LXI     H,SOISPH ;GET SERIAL 0 IN SEMAPHORE
0076'   B7                   ORA     A           ;CHANNEL NUMBER=0?
0077'   2803                 JRZ     ..ICOM      ;IF SO, CONTINUE
0079'   21 0011"           LXI     H,S1ISPH   ;ELSE, GET SERIAL 1 IN SEMAPHORE
007C'   CD 0000:08         ;..ICOM: CALL    WAIT#      ;WAIT FOR CONSOLE INPUT
007F'   CD 0053'           CALL    SERST       ;GET SERIAL CHANNEL STATUS
0082'   B7                   ORA     A           ;CHARACTER AVAILABLE?
0083'   28E4                 JRZ     SERIN       ;IF NOT, CONTINUE
0085'   79                   MOV     A,C         ;ELSE, GET INPUT CHARACTER
0086'   F3                   DI                    ;DISABLE INTERRUPTS
0087'   CD 018C'           CALL    MOVBUF      ;MOVE BUFFER TAIL DOWN
008A'   FB                   EI                    ;ENABLE INTERRUPTS

```

SP442 - TURBODOS OPERATING SYSTEM SERIAL/PARALLEL I/O DRIVER (IMS 442)
 COPYRIGHT (C) 1981, SOFTWARE 2000, INC.

```

008B'   C9                RET                ;DONE
;
008C'   78                SEROUT: MOV        A,B          ;GET CHANNEL NUMBER
008D'   FE02              CPI            2            ;CHANNEL NUMBER=0/1?
008F'   3805              JRC            ..S010 ;IF SO, CONTINUE
0091'   CD 0000:09        CALL           OUT480# ;ELSE, GET IMS 480 OUT SEMAPHORE
0094'   1817              JMPR           ..OCOM  ;CONTINUE
0096'   21 0009"         ..S010: LXI        H,SOOCHR ;GET SERIAL 0 OUTPUT CHARACTER
0099'   B7                ORA            A            ;CHANNEL NUMBER=0?
009A'   2801              JRZ            ..S01C ;IF SO, CONTINUE
009C'   23                INX            H            ;GET SERIAL 1 OUTPUT CHARACTER
009D'   71                ..S01C: MOV       M,C          ;SAVE OUTPUT CHARACTER
009E'   21 0017"         LXI        H,SOOSPH ;GET SERIAL 0 OUT SEMAPHORE
00A1'   11 00BF'         LXI        D,SOOPOL ;GET SERIAL 0 OUT POLL ROUTINE
00A4'   B7                ORA            A            ;CHANNEL NUMBER=0?
00A5'   2806              JRZ            ..OCOM ;IF SO, CONTINUE
00A7'   21 001D"         LXI        H,S10SPH ;GET SERIAL 1 OUT SEMAPHORE
00AA'   11 00E3'         LXI        D,S10POL ;GET SERIAL 1 OUT POLL ROUTINE
00AD'   E5                ..OCOM: PUSH     H            ;SAVE SEMAPHORE ADDRESS
00AE'   D5                PUSH     D            ;SAVE POLL ROUTINE ADDRESS
00AF'   CD 0000:0A        CALL        LNKPOL# ;CREATE POLL ROUTINE
00B2'   21 00BB'         LXI        H,..RET ;GET RETURN ADDRESS
00B5'   E3                XTHL           ;SIMULATE CALL/GET POLL ROUTINE
00B6'   23                INX            H            ;ADVANCE PAST LINK POINTERS
00B7'   23                INX            H
00B8'   23                INX            H
00B9'   23                INX            H
00BA'   E9                PCHL           ;EXECUTE POLL ROUTINE
00BB'   E1                ..RET: POP     H            ;RESTORE SEMAPHORE ADDRESS
00BC'   C3 0000:08        JMP         WAIT# ;DISPATCH IF NECESSARY
;
00BF'   ;                SOOPOL: ;SERIAL 0 OUTPUT POLL ROUTINE
00BF'   0000              .WORD        0            ;SUCCESSOR LINK POINTER
00C1'   0000              .WORD        0            ;PREDECESSOR LINK POINTER
;
00C3'   DB10              IN            SOCTRL ;GET SERIAL 0 STATUS
00C5'   CB4F              BIT            TBE,A     ;TRANSMIT BUFFER EMPTY?
00C7'   C8                RZ            ;IF NOT, DONE
00C8'   21 0003"         LXI        H,SOBR ;ELSE, GET SERIAL 0 BAUD RATE CODE
00CB'   CB76              BIT            6,M       ;CTS HANDSHAKING REQUESTED?
00CD'   2803              JRZ            ..NCTS ;IF NOT, CONTINUE
00CF'   CB7F              BIT            CTSN,A   ;CHECK CLEAR TO SEND (NOT) STATUS
00D1'   C0                RNZ           ;IF CLEAR TO SEND FALSE, DONE
00D2'   3A 0009"         ..NCTS: LDA     SOOCHR ;GET SERIAL 0 OUTPUT CHARACTER
00D5'   D311              OUT          SODATA ;OUTPUT CHARACTER
00D7'   21 00BF'         LXI        H,SOOPOL ;GET SERIAL 0 OUT POLL ROUTINE
00DA'   CD 0000:0B        CALL        UNLINK# ;UNLINK POLL ROUTINE
00DD'   21 0017"         LXI        H,SOOSPH ;GET SERIAL 0 OUT SEMAPHORE
00E0'   C3 0000:0C        JMP         SIGNAL# ;SIGNAL PROCESS AS READY
;
00E3'   ;                S10POL: ;SERIAL 1 OUTPUT POLL ROUTINE
00E3'   0000              .WORD        0            ;SUCCESSOR LINK POINTER
00E5'   0000              .WORD        0            ;PREDECESSOR LINK POINTER
;

```

442 - TURBODOS OPERATING SYSTEM SERIAL/PARALLEL I/O DRIVER (IMS 442)
 COPYRIGHT (C) 1981, SOFTWARE 2000, INC.

```

00E7' DB12          IN          S1CTRL ;GET SERIAL 1 STATUS
00E9' CB4F          BIT          TBE,A   ;TRANSMIT BUFFER EMPTY?
00EB' C8            RZ           ;IF NOT, DONE
00EC' 21 0004#     LXI          H,S1BR  ;ELSE, GET SERIAL 1 BAUD RATE CODE
00EF' CB76          BIT          6,M    ;CTS HANDSHAKING REQUESTED?
00F1' 2803         JRZ          ..NCTS ;IF NOT, CONTINUE
00F3' CB7F          BIT          CTSN,A  ;CHECK CLEAR TO SEND (NOT) STATUS
00F5' C0            RNZ          ;IF CLEAR TO SEND FALSE, DONE
00F6' 3A 000A#     ..NCTS: LDA     S1OCHR ;GET SERIAL 1 OUTPUT CHARACTER
00F9' D313         OUT          S1DATA ;OUTPUT CHARACTER
00FB' 21 00E3'     LXI          H,S1OPOL ;GET SERIAL 1 OUT POLL ROUTINE
00FE' CD 0000:0B  CALL      UNLINK# ;UNLINK POLL ROUTINE
0101' 21 001D#     LXI          H,S1OSPH ;GET SERIAL 1 OUT SEMAPHORE
0104' C3 0000:0C  JMP          SIGNAL# ;SIGNAL PROCESS AS READY

0107' ED73 0000:0D ;SERISR: SSPD   INTSP# ;SAVE STACK POINTER
010B' 31 0000:0E  LXI          SP,INTSTK# ;SET UP AUX STACK POINTER
010E' F5           PUSH         PSW      ;SAVE REGISTERS
010F' C5           PUSH         B
0110' D5           PUSH         D
0111' E5           PUSH         H
0112' CD 0126'    CALL      ..SOI   ;CHECK FOR SERIAL 0 INPUT
0115' CD 013D'    CALL      ..S1I   ;CHECK FOR SERIAL 1 INPUT
0118' CD 0000:0F  CALL      ISR480# ;CHECK FOR IMS 480 INPUT
011B' ET          POP          H      ;RESTORE REGISTERS
011C' D1           POP          D
011D' C1           POP          B
011E' F1           POP          PSW
011F' ED7B 0000:0D LSPD     INTSP# ;RESTORE STACK POINTER
0123' C3 0000:10  JMP      ISRXIT# ;CONTINUE
0126' DB10         ..SOI: IN          SOCTRL ;GET SERIAL 0 STATUS
0128' CB47         BIT          RDA,A  ;CHARACTER AVAILABLE
012A' C8            RZ           ;IF NOT, DONE
012B' 21 000B#     LXI          H,SOISPH ;GET SERIAL 0 INPUT SEMAPHORE
012E' E5           PUSH         H      ;SAVE SERIAL 0 INPUT SEMAPHORE
012F' CD 0000:0C  CALL      SIGNAL# ;SIGNAL PROCESS AS READY
0132' D1           POP          D      ;RESTORE SERIAL 0 INPUT SEMAPHORE
0133' DB11         IN          S0DATA  ;GET SERIAL 0 DATA CHARACTER
0135' 4F           MOV          C,A    ;SERIAL 0 DATA CHARACTER TO C-REG
0136' 0600         MVI          B,0    ;SET CHANNEL NUMBER=0
0138' 21 0003#     LXI          H,SOBR  ;GET SERIAL 0 BAUD RATE
013B' 1815         JMPR       SERISC  ;CONTINUE
013D' DB12         ..S1I: IN          S1CTRL ;GET SERIAL 1 STATUS
013F' CB47         BIT          RDA,A  ;CHARACTER AVAILABLE
0141' C8            RZ           ;IF NOT, DONE
0142' 21 0011#     LXI          H,S1ISPH ;GET SERIAL 1 INPUT SEMAPHORE
0145' E5           PUSH         H      ;SAVE SERIAL 1 INPUT SEMAPHORE
0146' CD 0000:0C  CALL      SIGNAL# ;SIGNAL PROCESS AS READY
0149' D1           POP          D      ;RESTORE SERIAL 1 INPUT SEMAPHORE
014A' DB13         IN          S1DATA  ;GET SERIAL 1 DATA CHARACTER
014C' 4F           MOV          C,A    ;SERIAL 1 DATA CHARACTER TO C-REG
014D' 0601         MVI          B,1    ;SET CHANNEL NUMBER=1
014F' 21 0004#     LXI          H,S1BR  ;GET SERIAL 1 BAUD RATE

```

SP442 - TURBODOS OPERATING SYSTEM SERIAL/PARALLEL I/O DRIVER (IMS 442)
 COPYRIGHT (C) 1981, SOFTWARE 2000, INC.

```

0152'  CB7E          SERISC::BIT      7,M      ;SIGN BIT ON BAUD RATE CODE?
0154'  281C          JRZ          ..NCC      ;IF NOT, CONTINUE
0156'  CBB9          RES          7,C      ;ELSE, STRIP SIGN BIT ON CHARACTER
0158'  CD 0000:11    CALL         SLVRES#   ;CHECK FOR SLAVE RESET
015B'  3A 0000:12    LDA          ATNCHR#   ;GET ATTENTION CHARACTER
015E'  91            SUB          C          ;CHARACTER=ATTENTION CHARACTER?
015F'  2011          JRNZ         ..NCC      ;IF NOT, CONTINUE
0161'  3C            INR          A          ;ELSE, GET SEMAPHORE COUNT=1
0162'  12            STAX         D          ;SET SEMAPHORE COUNT=1
0163'  C5            ;..FBL: PUSH     B          ;SAVE CHARACTER/CHANNEL NUMBER
0164'  CD 0053'     CALL         SERST     ;GET SERIAL CHANNEL STATUS
0167'  C1            POP          B          ;RESTORE CHARACTER/CHANNEL NUMBER
0168'  B7            ORA          A          ;CHARACTER AVAILABLE?
0169'  2807          JRZ          ..NCC      ;IF NOT, CONTINUE
016B'  C5            PUSH     B          ;SAVE CHARACTER/CHANNEL NUMBER
016C'  CD 018C'     CALL         MOVBUF    ;MOVE BUFFER TAIL DOWN
016F'  C1            POP          B          ;RESTORE CHARACTER/CHANNEL NUMBER
0170'  18F1          JMPR         ..FBL     ;CONTINUE
0172'  2A 0000"     ;..NCC: LHL D      SERBSZ    ;GET SERIAL BUFFER SIZE
0175'  29            DAD          H          ;X2
0176'  ED5B 0005"   LDED        SERBUF    ;GET SERIAL BUFFER ADDRESS
017A'  19            DAD          D          ;CALC END OF SERIAL BUFFER ADDRESS
017B'  ED5B 0007"   LDED        SERPTR    ;GET SERIAL BUFFER POINTER
017F'  B7            ORA          A          ;CLEAR CARRY FLAG
0180'  ED52          DSB C       D          ;SERIAL BUFFER FULL?
0182'  C8            RZ          ;IF SO, DONE
0183'  EB            XCHG         ;SERIAL BUFFER POINTER TO HL-REG
0184'  70            MOV          M,B       ;STORE CHANNEL NUMBER IN BUFFER
0185'  23            INX          H
0186'  71            MOV          M,C       ;STORE INPUT CHARACTER IN BUFFER
0187'  23            INX          H
0188'  22 0007"     SHLD        SERPTR    ;UPDATE SERIAL BUFFER POINTER
018B'  C9            RET          ;DONE

;
018C'  EB            ;MOVBUF: XCHG         ;SOURCE ADDRESS TO DE-REG
018D'  2A 0007"     LHL D      SERPTR    ;GET SERIAL BUFFER POINTER
0190'  ED52          DSB C       D          ;CALC LENGTH OF TAIL TO MOVE DOWN
0192'  4D            MOV          C,L       ;LENGTH OF TAIL TO BC-REG
0193'  44            MOV          B,H
0194'  EB            XCHG         ;SOURCE ADDRESS TO HL-REG
0195'  5D            MOV          E,L       ;COPY SOURCE ADDRESS INTO DE-REG
0196'  54            MOV          D,H
0197'  1B            DCX         D          ;CALC DESTINATION ADDRESS
0198'  1B            DCX         D
0199'  2802          JRZ          ..X       ;IF LENGTH OF TAIL=0, CONTINUE
019B'  ED B0          LDIR        ;ELSE, MOVE TAIL DOWN
019D'  ED53 0007"   ;..X: SDED        SERPTR    ;UPDATE SERIAL BUFFER POINTER
01A1'  C9            RET          ;DONE

;
01A2'  78            ;SERSBR: MOV        A,B       ;GET CHANNEL NUMBER
01A3'  FE02          CPI          2         ;CHANNEL NUMBER=0/1?
01A5'  D2 0000:13    JNC         SBR480#    ;IF NOT, CONTINUE
01A8'  21 0003"     LXI         H,SOBR     ;ELSE, GET SERIAL 0 BAUD RATE
01AB'  B7            ORA          A          ;CHANNEL NUMBER=0?

```

9442 - TURBODOS OPERATING SYSTEM SERIAL/PARALLEL I/O DRIVER (IMS 442)
 COPYRIGHT (C) 1981, SOFTWARE 2000, INC.

```

01AC: 2801          JRZ          ..COM1 ;IF SO, CONTINUE
01AE: 23           INX          H          ;ELSE, GET SERIAL 1 BAUD RATE
01AF: 71           ..COM1: MOV     M,C       ;SAVE BAUD RATE CODE
01B0: CD 01C6:    CALL       GETBTV    ;GET BAUD RATE TIMER VALUE
01B3: 78           MOV     A,B         ;GET CHANNEL NUMBER
01B4: B7           ORA          A          ;CHANNEL NUMBER=0?
01B5: 3E36        MVI     A,TOCMD     ;GET TIMER 0 COMMAND
01B7: 0E14        MVI     C,TIMO      ;GET TIMER 0 DATA REGISTER
01B9: 2804          JRZ          ..COM2 ;IF CHANNEL NUMBER=0, CONTINUE
01BB: 3E76        MVI     A,T1CMD     ;ELSE, GET TIMER 1 COMMAND
01BD: 0E15        MVI     C,TIM1     ;GET TIMER 1 DATA REGISTER
01BF: D317        ..COM2: OUT    TIMCTL  ;SELECT TIMER
01C1: ED59        OUTP     E          ;OUTPUT LSB OF TIMER VALUE
01C3: ED51        OUTP     D          ;OUTPUT MSB OF TIMER VALUE
01C5: C9           RET              ;DONE

;
01C6: 79           GETBTV::MOV    A,C      ;GET REQUESTED BAUD RATE CODE
01C7: E60F        ANI     OFH         ;EXTRACT RELEVANT BITS
01C9: 87           ADD     A           ;X2
01CA: 5F           MOV     E,A         ;TO E-REG
01CB: 1600        MVI     D,0         ;MAKE IT DOUBLE LENGTH
01CD: 21 0000:14 LXI     H,BRTBL#    ;GET BAUD RATE TABLE
01D0: 19           DAD     D           ;INDEX INTO TABLE
01D1: 5E           MOV     E,M         ;GET TIMER VALUE
01D2: 23           INX     H
01D3: 56           MOV     D,M
01D4: C9           RET              ;DONE

;
01D5: 78           ;SERRBR: MOV    A,B      ;GET CHANNEL NUMBER
01D6: FE02        CPI     2           ;CHANNEL NUMBER=0/1?
01D8: D2 0000:15 JNC     RBR480#    ;IF NOT, CONTINUE
01DB: 21 0003" LXI     H,SOBR     ;ELSE, GET SERIAL 0 BAUD RATE
01DE: B7           ORA     A           ;CHANNEL NUMBER=0?
01DF: 2801        JRZ     ..COM      ;IF SO, CONTINUE
01E1: 23           INX     H           ;ELSE, GET SERIAL 1 BAUD RATE
01E2: 7E           ..COM: MOV    A,M    ;GET CURRENT BAUD RATE CODE
01E3: C9           RET              ;DONE

;
01E4: 78           ;SERSMC: MOV    A,B      ;GET CHANNEL NUMBER
01E5: FE02        CPI     2           ;CHANNEL NUMBER=0/1?
01E7: D2 0000:16 JNC     SMC480#    ;IF NOT, CONTINUE
01EA: B7           ORA     A           ;CHANNEL NUMBER=0?
01EB: 3A 0002" LDA     INTMSK     ;GET INT MASK
01EE: 200A        JRNZ    ..CH1     ;IF CHANNEL NUMBER NOT=0, CONTINUE
01F0: CBBF        RES     SORTSN,A   ;CLEAR SERIAL 0 REQ TO SEND (NOT)
01F2: CB79        BIT     7,C        ;SERIAL 0 REQ TO SEND TO BE ON?
01F4: 200C        JRNZ    ..COM     ;IF SO, CONTINUE
01F6: CBFF        SET     SORTSN,A   ;ELSE, SET SERIAL 0 RTS (NOT)
01F8: 1808        ..COM: JMPR    ..COM ;CONTINUE
01FA: CBA7        ..CH1: RES     S1RTSN,A ;CLEAR SERIAL 1 REQ TO SEND (NOT)
01FC: CB79        BIT     7,C        ;SERIAL 1 REQ TO SEND TO BE ON?
01FE: 2002        JRNZ    ..COM     ;IF SO, CONTINUE
0200: CBE7        SET     S1RTSN,A   ;ELSE, SET SERIAL 1 RTS (NOT)
0202: 32 0002" ..COM: STA     INTMSK ;UPDATE INT MASK

```

SP442 - TURBODOS OPERATING SYSTEM SERIAL/PARALLEL I/O DRIVER (IMS 442)
 COPYRIGHT (C) 1981, SOFTWARE 2000, INC.

```

0205'   D318           OUT      SINTE      ;SET SERIAL 1 REQUEST TO SEND
0207'   C9            RET          ;DONE

;
0208'   78            SERRMC: MOV     A,B      ;GET CHANNEL NUMBER
0209'   FE02          CPI          2         ;CHANNEL NUMBER=0/1?
020B'   D2 0000:17   JNC          RMC480# ;IF NOT, CONTINUE
020E'   B7            ORA          A         ;CHANNEL NUMBER=0?
020F'   DB12          IN           S1CTRL   ;GET SERIAL 0 STATUS
0211'   2802          JRZ          ..COM    ;IF CHANNEL NUMBER=0, CONTINUE
0213'   DB12          IN           S1CTRL   ;ELSE, GET SERIAL 1 STATUS
0215'   E680          ..COM: ANI     1<CTSN ;EXTRACT CLEAR TO SEND (NOT)
0217'   EE80          XRI          1<CTSN   ;COMPLIMENT IT
0219'   C9            RET          ;DONE

;
021A'   D31C          POOUT:: OUT     PODATA  ;OUTPUT BYTE TO PARALLEL 0
021C'   C9            RET          ;DONE

;
021D'   D31D          P1OUT:: OUT     P1DATA  ;OUTPUT BYTE TO PARALLEL 1
021F'   C9            RET          ;DONE

;
0220'   DB1E          P2IN::  IN      P2DATA  ;INPUT BYTE FROM PARALLEL 2
0222'   C9            RET          ;DONE

;
0223'   21 0002"     EBROM::LXI     H,INTMSK ;GET INTERRUPT MASK
0226'   CB86          RES          ROMDIS,M ;RESET ROM DISABLE BIT
0228'   7E            MOV          A,M      ;GET INTERRUPT MASK
0229'   D318          OUT          SINTE    ;TURN ON BOOTSTRAP ROM
022B'   21 FC00      LXI          H,BOOTPR ;RETURN BOOT PROM ADDR
022E'   C9            RET          ;DONE

;
022F'   21 0002"     DBROM::LXI     H,INTMSK ;GET INTERRUPT MASK
0232'   CBC6          SET          ROMDIS,M ;SET ROM DISABLE BIT
0234'   7E            MOV          A,M      ;GET INTERRUPT MASK
0235'   D318          OUT          SINTE    ;TURN OFF BOOTSTRAP ROM
0237'   C9            RET          ;DONE

;
.END

```


SER480 - TURBODOS OPERATING SYSTEM SERIAL DRIVER (IMS 480)
 COPYRIGHT (C) 1981, SOFTWARE 2000, INC.

```

;
; COPYRIGHT (C) 1981, SOFTWARE 2000, INC.
;
; AUTHORS: RONALD E. RAIKES
;          MICHAEL D. BUSCH
;
; VERSION: 09/08/81
;
; IDENT SER480 ;MODULE ID
;
; INSERT DREQUATE ;DRIVER SYMBOLIC EQUIVALENCES
;
00E0 IOBASE = 0E0H ;I/O PORT BASE
;
00E0 S2DATA = IOBASE+00H ;SERIAL 2 DATA REGISTER
00E1 S2IER = IOBASE+01H ;SERIAL 2 INTERRUPT ENABLE REGISTER
00E2 S2IIDR = IOBASE+02H ;SERIAL 2 INTERRUPT ID REGISTER
00E3 S2LCR = IOBASE+03H ;SERIAL 2 LINE CONTROL REGISTER
00E4 S2MCR = IOBASE+04H ;SERIAL 2 MODEM CONTROL REGISTER
00E5 S2LSR = IOBASE+05H ;SERIAL 2 LINE STATUS REGISTER
00E6 S2MSR = IOBASE+06H ;SERIAL 2 MODEM STATUS REGISTER
;
00E8 S3DATA = IOBASE+08H ;SERIAL 3 DATA REGISTER
00E9 S3IER = IOBASE+09H ;SERIAL 3 INTERRUPT ENABLE REGISTER
00EA S3IIDR = IOBASE+0AH ;SERIAL 3 INTERRUPT ID REGISTER
00EB S3LCR = IOBASE+0BH ;SERIAL 3 LINE CONTROL REGISTER
00EC S3MCR = IOBASE+0CH ;SERIAL 3 MODEM CONTROL REGISTER
00ED S3LSR = IOBASE+0DH ;SERIAL 3 LINE STATUS REGISTER
00EE S3MSR = IOBASE+0EH ;SERIAL 3 MODEM STATUS REGISTER
;
00F0 S4DATA = IOBASE+10H ;SERIAL 4 DATA REGISTER
00F1 S4IER = IOBASE+11H ;SERIAL 4 INTERRUPT ENABLE REGISTER
00F2 S4IIDR = IOBASE+12H ;SERIAL 4 INTERRUPT ID REGISTER
00F3 S4LCR = IOBASE+13H ;SERIAL 4 LINE CONTROL REGISTER
00F4 S4MCR = IOBASE+14H ;SERIAL 4 MODEM CONTROL REGISTER
00F5 S4LSR = IOBASE+15H ;SERIAL 4 LINE STATUS REGISTER
00F6 S4MSR = IOBASE+16H ;SERIAL 4 MODEM STATUS REGISTER
;
00F8 S5DATA = IOBASE+18H ;SERIAL 5 DATA REGISTER
00F9 S5IER = IOBASE+19H ;SERIAL 5 INTERRUPT ENABLE REGISTER
00FA S5IIDR = IOBASE+1AH ;SERIAL 5 INTERRUPT ID REGISTER
00FB S5LCR = IOBASE+1BH ;SERIAL 5 LINE CONTROL REGISTER
00FC S5MCR = IOBASE+1CH ;SERIAL 5 MODEM CONTROL REGISTER
00FD S5LSR = IOBASE+1DH ;SERIAL 5 LINE STATUS REGISTER
00FE S5MSR = IOBASE+1EH ;SERIAL 5 MODEM STATUS REGISTER
;
0001 IERCW = 01H ;INT ENABLE REGISTER CONTROL WORD
0003 LCRCW = 03H ;LINE CONTROL REGISTER CONTROL WORD
0003 MCRCW = 03H ;MODEM CONTROL REGISTER CONTROL WORD
;
0000 RDA = 0 ;RECEIVED DATA AVAILABLE BIT
0005 TBE = 5 ;TRANSMIT BUFFER EMPTY BIT
0004 CTS = 4 ;CLEAR TO SEND BIT

```

SER480 - TURBODOS OPERATING SYSTEM SERIAL DRIVER (IMS 480)
 COPYRIGHT (C) 1981, SOFTWARE 2000, INC.

```

;
0000"          ;          .LOC      .DATA.# ;LOCATE IN DATA AREA
;
0000"  00      S2BR:  .BYTE  0      ;SERIAL 2 BAUD RATE
0001"  00      S3BR:  .BYTE  0      ;SERIAL 3 BAUD RATE
0002"  00      S4BR:  .BYTE  0      ;SERIAL 4 BAUD RATE
0003"  00      S5BR:  .BYTE  0      ;SERIAL 5 BAUD RATE
0004"  00      S2OCHR: .BYTE  0      ;SERIAL 2 OUTPUT CHARACTER
0005"  00      S3OCHR: .BYTE  0      ;SERIAL 3 OUTPUT CHARACTER
0006"  00      S4OCHR: .BYTE  0      ;SERIAL 4 OUTPUT CHARACTER
0007"  00      S5OCHR: .BYTE  0      ;SERIAL 5 OUTPUT CHARACTER
;
0008"          ;S2ISPH:          ;SERIAL 2 INPUT SEMAPHORE
0008"  0000      .WORD  0          ;SEMAPHORE COUNT
000A"  000A"    ..S2IH: .WORD  ..S2IH ;SEMAPHORE P/D HEAD
000C"  000A"    .WORD  ..S2IH
;
000E"  0000      S3ISPH: .WORD  0          ;SERIAL 3 INPUT SEMAPHORE
0010"  0010"    ..S3IH: .WORD  ..S3IH ;SEMAPHORE COUNT
0012"  0010"    .WORD  ..S3IH ;SEMAPHORE P/D HEAD
;
0014"          ;S4ISPH:          ;SERIAL 4 INPUT SEMAPHORE
0014"  0000      .WORD  0          ;SEMAPHORE COUNT
0016"  0016"    ..S4IH: .WORD  ..S4IH ;SEMAPHORE P/D HEAD
0018"  0016"    .WORD  ..S4IH
;
001A"  0000      S5ISPH: .WORD  0          ;SERIAL 5 INPUT SEMAPHORE
001C"  001C"    ..S5IH: .WORD  ..S5IH ;SEMAPHORE COUNT
001E"  001C"    .WORD  ..S5IH ;SEMAPHORE P/D HEAD
;
0020"          ;S2OSPH:          ;SERIAL 2 OUTPUT SEMAPHORE
0020"  0000      .WORD  0          ;SEMAPHORE COUNT
0022"  0022"    ..S2OH: .WORD  ..S2OH ;SEMAPHORE P/D HEAD
0024"  0022"    .WORD  ..S2OH
;
0026"          ;S3OSPH:          ;SERIAL 3 OUTPUT SEMAPHORE
0026"  0000      .WORD  0          ;SEMAPHORE COUNT
0028"  0028"    ..S3OH: .WORD  ..S3OH ;SEMAPHORE P/D HEAD
002A"  0028"    .WORD  ..S3OH
;
002C"          ;S4OSPH:          ;SERIAL 4 OUTPUT SEMAPHORE
002C"  0000      .WORD  0          ;SEMAPHORE COUNT
002E"  002E"    ..S4OH: .WORD  ..S4OH ;SEMAPHORE P/D HEAD
0030"  002E"    .WORD  ..S4OH
;
0032"          ;S5OSPH:          ;SERIAL 5 OUTPUT SEMAPHORE
0032"  0000      .WORD  0          ;SEMAPHORE COUNT
0034"  0034"    ..S5OH: .WORD  ..S5OH ;SEMAPHORE P/D HEAD
0036"  0034"    .WORD  ..S5OH
;
0000"          ;          .LOC      .PROG.# ;LOCATE IN PROGRAM AREA
;

```

ER480 - TURBODOS OPERATING SYSTEM SERIAL DRIVER (IMS 480)
 COPYRIGHT (C) 1981, SOFTWARE 2000, INC.

```

0000' 0026          .WORD  NITLEN+2 ;INITIALIZATION CODE LENGTH
;
0002' 3E03          ;NIT480::MVI  A,LCRCW ;GET LINE CONTROL REGISTER VALUE
0004' D3E3          OUT    S2LCR  ;SET LINE CONTROL REGISTER 0
0006' D3EB          OUT    S3LCR  ;SET LINE CONTROL REGISTER 1
0008' D3F3          OUT    S4LCR  ;SET LINE CONTROL REGISTER 2
000A' D3FB          OUT    S5LCR  ;SET LINE CONTROL REGISTER 3
000C' 3E03          MVI  A,MCRCW ;GET MODEM CONTROL REGISTER VALUE
000E' D3E4          OUT    S2MCR  ;SET MODEM CONTROL REGISTER 0
0010' D3EC          OUT    S3MCR  ;SET MODEM CONTROL REGISTER 1
0012' D3F4          OUT    S4MCR  ;SET MODEM CONTROL REGISTER 2
0014' D3FC          OUT    S5MCR  ;SET MODEM CONTROL REGISTER 3
0016' 3E01          MVI  A,IERCW ;GET INT ENABLE REGISTER VALUE
0018' D3E1          OUT    S2IER  ;SET INT ENABLE REGISTER 0
001A' D3E9          OUT    S3IER  ;SET INT ENABLE REGISTER 1
001C' D3F1          OUT    S4IER  ;SET INT ENABLE REGISTER 2
001E' D3F9          OUT    S5IER  ;SET INT ENABLE REGISTER 3
0020' 21 0002'     LXI  H,NIT480 ;GET INITIALIZATION CODE ADDRESS
0023' C3 0000:04   JMP  DEALOC# ;DE-ALLOCATE INITIALIZATION CODE
;
0024          ;NITLEN = .-NIT480 ;INITIALIZATION CODE LENGTH
;
0026' 21 0008"     ;IN480::LXI  H,S2ISPH ;GET SERIAL 2 IN SEMPAHORE
0029' D602          SUI  2 ;REMOVE CHANNEL NUMBER BIAS
002B' C8            RZ ;IF CHANNEL NUMBER=2, DONE
002C' 21 000E"     LXI  H,S3ISPH ;ELSE, GET SERIAL 3 IN SEMPAHORE
002F' 3D            DCR  A ;CHANNEL NUMBER=3?
0030' C8            RZ ;IF SO, DONE
0031' 21 0014"     LXI  H,S4ISPH ;ELSE, GET SERIAL 4 IN SEMPAHORE
0034' 3D            DCR  A ;CHANNEL NUMBER=4?
0035' C8            RZ ;IF SO, DONE
0036' 21 001A"     LXI  H,S5ISPH ;ELSE, GET SERIAL 5 IN SEMPAHORE
0039' C9            RET ;DONE
;
003A' 21 0004"     ;OUT480::LXI  H,S2OCHR ;GET SERIAL 2 OUTPUT CHARACTER
003D' CD 01B7'     CALL CHNMB ;DO COMMON SETUP
0040' 71            MOV  M,C ;SAVE OUTPUT CHARACTER
0041' 21 0020"     LXI  H,S2OSPH ;GET SERIAL 2 OUT SEMAPHORE
0044' 11 005F'     LXI  D,S2OPOL ;GET SERIAL 2 OUT POLL ROUTINE
0047' C8            RZ ;IF CHANNEL NUMBER=2, DONE
0048' 21 0026"     LXI  H,S3OSPH ;ELSE, GET SERIAL 3 OUT SEMAPHORE
004B' 11 0085'     LXI  D,S3OPOL ;GET SERIAL 3 OUT POLL ROUTINE
004E' 3D            DCR  A ;CHANNEL NUMBER=3?
004F' C8            RZ ;IF SO, DONE
0050' 21 002C"     LXI  H,S4OSPH ;ELSE, GET SERIAL 4 OUT SEMAPHORE
0053' 11 00AB'     LXI  D,S4OPOL ;GET SERIAL 4 OUT POLL ROUTINE
0056' 3D            DCR  A ;CHANNEL NUMBER=4?
0057' C8            RZ ;IF SO, DONE
0058' 21 0032"     LXI  H,S5OSPH ;ELSE, GET SERIAL 5 OUT SEMAPHORE
005B' 11 00D1'     LXI  D,S5OPOL ;GET SERIAL 5 OUT POLL ROUTINE
005E' C9            RET ;DONE
;
005F'          ;S2OPOL:
005F' 0000          .WORD  0 ;SERIAL 2 OUTPUT POLL ROUTINE
;SUCCESSOR LINK POINTER

```

SER480 - TURBODOS OPERATING SYSTEM SERIAL DRIVER (IMS 480)
 COPYRIGHT (C) 1981, SOFTWARE 2000, INC.

```

0061' 0000          .WORD 0          ;PREDECESSOR LINK POINTER
;
0063' DBE5          IN      S2LSR      ;GET SERIAL 2 LINE STATUS REGISTER
0065' CB6F          BIT      TBE,A      ;TRANSMIT BUFFER EMPTY?
0067' C8            RZ              ;IF NOT, DONE
0068' 21 0000"     LXI      H,S2BR     ;ELSE, GET SERIAL 2 BAUD RATE
006B' CB76          BIT      6,M      ;CLEAR TO SEND HANDSHAKING REQUESTEI
?
006D' 2805          JRZ      ..NCTS    ;IF NOT, CONTINUE
006F' DBE6          IN      S2MSR     ;GET SERIAL 2 MODEM STATUS REGISTER
0071' CB67          BIT      CTS,A     ;CLEAR TO SEND STATUS TRUE?
0073' C8            RZ              ;IF NOT, DONE
0074' 3A 0004"     ..NCTS: LDA      S2OCHR   ;GET SERIAL 2 OUTPUT CHARACTER
0077' D3E0          OUT      S2DATA   ;OUTPUT CHARACTER
0079' 21 005F"     LXI      H,S2OPOL  ;GET SERIAL 2 OUT POLL ROUTINE
007C' CD 0000:05   CALL     UNLINK#  ;UNLINK POLL ROUTINE
007F' 21 0020"     LXI      H,S2OSPH  ;GET SERIAL 2 OUT SEMAPHORE
0082' C3 0000:06   JMP      SIGNAL#  ;SIGNAL PROCESS AS READY
;
0085'              ;S3OPOL:          ;SERIAL 3 OUTPUT POLL ROUTINE
0085' 0000          .WORD 0          ;SUCCESSOR LINK POINTER
0087' 0000          .WORD 0          ;PREDECESSOR LINK POINTER
;
0089' DBED          IN      S3LSR     ;GET SERIAL 3 LINE STATUS REGISTER
008B' CB6F          BIT      TBE,A     ;TRANSMIT BUFFER EMPTY?
008D' C8            RZ              ;IF NOT, DONE
008E' 21 0001"     LXI      H,S3BR     ;ELSE, GET SERIAL 3 BAUD RATE
0091' CB76          BIT      6,M      ;CLEAR TO SEND HANDSHAKING REQUESTEI
?
0093' 2805          JRZ      ..NCTS    ;IF NOT, CONTINUE
0095' DBEE          IN      S3MSR     ;GET SERIAL 3 MODEM STATUS REGISTER
0097' CB67          BIT      CTS,A     ;CLEAR TO SEND STATUS TRUE?
0099' C8            RZ              ;IF NOT, DONE
009A' 3A 0005"     ..NCTS: LDA      S3OCHR   ;GET SERIAL 3 OUTPUT CHARACTER
009D' D3E8          OUT      S3DATA   ;OUTPUT CHARACTER
009F' 21 0085"     LXI      H,S3OPOL  ;GET SERIAL 3 OUT POLL ROUTINE
00A2' CD 0000:05   CALL     UNLINK#  ;UNLINK POLL ROUTINE
00A5' 21 0026"     LXI      H,S3OSPH  ;GET SERIAL 3 OUT SEMAPHORE
00A8' C3 0000:06   JMP      SIGNAL#  ;SIGNAL PROCESS AS READY
;
00AB'              ;S4OPOL:          ;SERIAL 4 OUTPUT POLL ROUTINE
00AB' 0000          .WORD 0          ;SUCCESSOR LINK POINTER
00AD' 0000          .WORD 0          ;PREDECESSOR LINK POINTER
;
00AF' DBF5          IN      S4LSR     ;GET SERIAL 4 LINE STATUS REGISTER
00B1' CB6F          BIT      TBE,A     ;TRANSMIT BUFFER EMPTY?
00B3' C8            RZ              ;IF NOT, DONE
00B4' 21 0002"     LXI      H,S4BR     ;ELSE, GET SERIAL 4 BAUD RATE
00B7' CB76          BIT      6,M      ;CLEAR TO SEND HANDSHAKING REQUESTEI
?
00B9' 2805          JRZ      ..NCTS    ;IF NOT, CONTINUE
00BB' DBF6          IN      S4MSR     ;GET SERIAL 4 MODEM STATUS REGISTER
00BD' CB67          BIT      CTS,A     ;CLEAR TO SEND STATUS TRUE?
00BF' C8            RZ              ;IF NOT, DONE

```

SR480 - TURBODOS OPERATING SYSTEM SERIAL DRIVER (IMS 480)
 COPYRIGHT (C) 1981, SOFTWARE 2000, INC.

```

00C0' 3A 0006"    ..NCTS: LDA    S4OCHR ;GET SERIAL 4 OUTPUT CHARACTER
00C3' D3F0        OUT    S4DATA ;OUTPUT CHARACTER
00C5' 21 00AB'   LXI    H,S4OPOL ;GET SERIAL 4 OUT POLL ROUTINE
00C8' CD 0000:05 CALL   UNLINK# ;UNLINK POLL ROUTINE
00CB' 21 002C"   LXI    H,S4OSPH ;GET SERIAL 4 OUT SEMAPHORE
00CE' C3 0000:06 JMP    SIGNAL# ;SIGNAL PROCESS AS READY

;
00D1'           ;S5OPOL:
00D1' 0000        .WORD  0 ;SERIAL 5 OUTPUT POLL ROUTINE
00D3' 0000        .WORD  0 ;SUCCESSOR LINK POINTER
;
00D5' DBFD        IN    S5LSR ;GET SERIAL 5 LINE STATUS REGISTER
00D7' CB6F        BIT   TBE,A ;TRANSMIT BUFFER EMPTY?
00D9' C8          RZ     ;IF NOT, DONE
00DA' 21 0003"   LXI    H,S5BR ;ELSE, GET SERIAL 5 BAUD RATE
00DD' CB76        BIT   6,M ;CLEAR TO SEND HANDSHAKING REQUESTER

?
00DF' 2805        JRZ   ..NCTS ;IF NOT, CONTINUE
00E1' DBFE        IN    S5MSR ;GET SERIAL 5 MODEM STATUS REGISTER
00E3' CB67        BIT   CTS,A ;CLEAR TO SEND STATUS TRUE?
00E5' C8          RZ     ;IF NOT, DONE
00E6' 3A 0007"   ..NCTS: LDA    S5OCHR ;GET SERIAL 5 OUTPUT CHARACTER
00E9' D3F8        OUT    S5DATA ;OUTPUT CHARACTER
00EB' 21 00D1'   LXI    H,S5OPOL ;GET SERIAL 5 OUT POLL ROUTINE
00EE' CD 0000:05 CALL   UNLINK# ;UNLINK POLL ROUTINE
00F1' 21 0032"   LXI    H,S5OSPH ;GET SERIAL 5 OUT SEMAPHORE
00F4' C3 0000:06 JMP    SIGNAL# ;SIGNAL PROCESS AS READY

;
00F7' CD 0104'   ;ISR480::CALL ..S2I ;CHECK FOR SERIAL 2 INPUT
00FA' CD 0116'   CALL ..S3I ;CHECK FOR SERIAL 3 INPUT
00FD' CD 0128'   CALL ..S4I ;CHECK FOR SERIAL 4 INPUT
0100' CD 013A'   CALL ..S5I ;CHECK FOR SERIAL 5 INPUT
0103' C9        RET     ;DONE

;
0104' DBE5        ..S2I: IN    S2LSR ;GET SERIAL 2 STATUS
0106' CB47        BIT   RDA,A ;CHARACTER AVAILABLE
0108' C8          RZ     ;IF NOT, DONE
0109' 21 0008"   LXI    H,S2ISPH ;GET SERIAL 2 INPUT SEMAPHORE
010C' E5          PUSH   H ;SAVE SERIAL 2 INPUT SEMPAPHORE
010D' CD 0000:06 CALL   SIGNAL# ;SIGNAL PROCESS AS READY
0110' DBE0        IN    S2DATA ;GET SERIAL 2 DATA CHARACTER
0112' 0602        MVI   B,2 ;SET CHANNEL NUMBER=2
0114' 1834        JMPR  ..SIC ;CONTINUE
0116' DBED        ..S3I: IN    S3LSR ;GET SERIAL 3 STATUS
0118' CB47        BIT   RDA,A ;CHARACTER AVAILABLE
011A' C8          RZ     ;IF NOT, DONE
011B' 21 000E"   LXI    H,S3ISPH ;GET SERIAL 3 INPUT SEMAPHORE
011E' E5          PUSH   H ;SAVE SERIAL 3 INPUT SEMPAPHORE
011F' CD 0000:06 CALL   SIGNAL# ;SIGNAL PROCESS AS READY
0122' DBE8        IN    S3DATA ;GET SERIAL 3 DATA CHARACTER
0124' 0603        MVI   B,3 ;SET CHANNEL NUMBER=3
0126' 1822        JMPR  ..SIC ;CONTINUE
0128' DBF5        ..S4I: IN    S4LSR ;GET SERIAL 4 STATUS
012A' CB47        BIT   RDA,A ;CHARACTER AVAILABLE

```

SER480 - TURBODOS OPERATING SYSTEM SERIAL DRIVER (IMS 480)
 COPYRIGHT (C) 1981, SOFTWARE 2000, INC.

```

012C'   C8           RZ           ;IF NOT, DONE
012D'   21 0014"    LXI           H,S4ISPH ;GET SERIAL 4 INPUT SEMAPHORE
0130'   E5           PUSH        H       ;SAVE SERIAL 4 INPUT SEMPAHORE
0131'   CD 0000:06  CALL        SIGNAL# ;SIGNAL PROCESS AS READY
0134'   DBF0         IN          S4DATA  ;GET SERIAL 4 DATA CHARACTER
0136'   0604         MVI         B,4     ;SET CHANNEL NUMBER=4
0138'   1810         JMPR        ..SIC  ;CONTINUE
013A'   DBFD         ..S5I: IN      S5LSR  ;GET SERIAL 5 STATUS
013C'   CB47         BIT         RDA,A   ;CHARACTER AVAILABLE
013E'   C8           RZ           ;IF NOT, DONE
013F'   21 001A"    LXI           H,S5ISPH ;GET SERIAL 5 INPUT SEMAPHORE
0142'   E5           PUSH        H       ;SAVE SERIAL 5 INPUT SEMPAHORE
0143'   CD 0000:06  CALL        SIGNAL# ;SIGNAL PROCESS AS READY
0146'   DBF8         IN          S5DATA  ;GET SERIAL 5 DATA CHARACTER
0148'   0605         MVI         B,5     ;SET CHANNEL NUMBER=5
014A'   4F           ..SIC: MOV      C,A   ;SERIAL DATA CHARACTER TO C-REG
014B'   78           MOV      A,B     ;GET CHANNEL NUMBER
014C'   21 0000"    LXI           H,S2BR  ;GET SERIAL 2 BAUD RATE
014F'   CD 01B7'    CALL        CHNMBC  ;DO COMMON SETUP
0152'   D1           POP         D       ;RESTORE SERIAL INPUT SEMAPHORE
0153'   C3 0000:07 JMP         SERISC# ;CONTINUE

0156'   21 0000"    ;SBR480::LXI   H,S2BR  ;GET SERIAL 2 BAUD RATE
0159'   CD 01B7'    CALL        CHNMBC  ;DO COMMON SETUP
015C'   F5           PUSH        PSW     ;SAVE CHANNEL NUMBER
015D'   71           MOV         M,C     ;SAVE BAUD RATE CODE
015E'   CD 0000:08  CALL        GETBT#  ;GET BAUD RATE TIMER VALUE
0161'   F1           POP         PSW     ;RESTORE CHANNEL NUMBER
0162'   87           ADD         A       ;X2
0163'   87           ADD         A       ;X2=X4
0164'   87           ADD         A       ;X2=X8
0165'   F6E3        ORI         IOBASE+3 ;CALC LINE CONTROL REGISTER
0167'   4F           MOV         C,A     ;LINE CONTROL REGISTER TO C-REG
0168'   3E83        MVI        A,LCRCW180H ;GET DIVISOR LATCH ACCESS BIT
016A'   ED79        OUTP       A       ;SELECT DIVISOR LATCH
016C'   0D           DCR         C       ;CALC DATA REGISTER
016D'   0D           DCR         C
016E'   0D           DCR         C
016F'   ED59        OUTP       E       ;OUTPUT LSB OF BAUD RATE TIMER VALUE

0171'   0C           INR         C       ;CALC DATA REGISTER+1
0172'   ED51        OUTP       D       ;OUTPUT MSB OF BAUD RATE TIMER VALUE

0174'   0C           INR         C       ;CALC LINE CONTROL REGISTER
0175'   0C           INR         C
0176'   3E03        MVI        A,LCRCW  ;GET LINE CONTROL REGISTER VALUE
0178'   ED79        OUTP       A       ;DE-SELECT DIVISOR LATCH
017A'   C9           RET         ;DONE

017B'   21 0000"    ;RBR480::LXI   H,S2BR  ;GET SERIAL 2 BAUD RATE
017E'   CD 01B7'    CALL        CHNMBC  ;DO COMMON SETUP
0181'   7E           MOV         A,M     ;GET CURRENT BAUD RATE
0182'   C9           RET         ;DONE

```

ER480 - TURBODOS OPERATING SYSTEM SERIAL DRIVER (IMS 480)
 COPYRIGHT (C) 1981, SOFTWARE 2000, INC.

```

0183: 79          SMC480::MOV    A,C      ;GET REQUESTED MODEM CONTROLS
0184: 0F          RRC           ;SHIFT MODEM CONTROLS INTO BITS 6/7
0185: 0F          RRC
0186: 57          MOV          D,A      ;MODEM CONTROL REGISTER VALUE TO D-E
          EG
0187: 78          MOV          A,B      ;GET CHANNEL NUMBER
0188: D602       SUI          2        ;REMOVE CHANNEL NUMBER BIAS
018A: 87          ADD          A        ;X2
018B: 87          ADD          A        ;X2=X4
018C: 87          ADD          A        ;X2=X8
018D: F6E4       ORI          IOBASE+4 ;CALC MODEM CONTROL REGISTER
018F: 4F          MOV          C,A      ;MODEM CONTROL REGISTER TO C-REG
0190: ED51       OUTP        D        ;OUTPUT MODEM CONTROLS
0192: C9          RET           ;DONE

          ;
0193: 78          RMC480::MOV    A,B      ;GET CHANNEL NUMBER
0194: D602       SUI          2        ;REMOVE CHANNEL NUMBER BIAS
0196: 87          ADD          A        ;X2
0197: 87          ADD          A        ;X2=X4
0198: 87          ADD          A        ;X2=X8
0199: F6E6       ORI          IOBASE+6 ;CALC MODEM STATUS REGISTER
019B: 4F          MOV          C,A      ;MODEM STATUS REGISTER TO C-REG
019C: ED50       INP          D        ;GET MODEM STATUS REGISTER
019E: AF          XRA          A        ;SET RETURN CODE=0
019F: CB62       BIT          4,D      ;CLEAR TO SEND BIT SET?
01A1: 2802       JRZ         ..NCTS   ;IF NOT, CONTINUE
01A3: CBFF       SET          7,A      ;ELSE, SET CLEAR TO SEND BIT
01A5: CB6A       ..NCTS: BIT    5,D      ;DATA SET READY BIT SET?
01A7: 2802       JRZ         ..NDSR   ;IF NOT, CONTINUE
01A9: CBF7       SET          6,A      ;ELSE, SET DATA SET READY BIT
01AB: CB7A       ..NDSR: BIT    7,D      ;DATA CARRIER DETECT BIT SET?
01AD: 2802       JRZ         ..NDCD   ;IF NOT, CONTINUE
01AF: CBEF       SET          5,A      ;ELSE, SET DATA CARRIER DETECT BIT
01B1: CB72       ..NDCD: BIT    6,D      ;RING INDICATOR BIT SET?
01B3: C8          RZ           ;IF NOT, DONE
01B4: CBET       SET          4,A      ;ELSE, SET RING INDICATOR BIT
01B6: C9          RET           ;DONE

          ;
01B7: D602       CHNMBC: SUI    2        ;REMOVE CHANNEL NUMBER BIAS
01B9: 5F          MOV          E,A      ;CHANNEL NUMBER TO DE-REG
01BA: 1600       MVI         D,0      ;DOUBLE LENGTH
01BC: 19          DAD         D        ;INDEX INTO CHARACTER SAVE AREA
01BD: C9          RET           ;DONE

          ;
          .END

```

BRT442 - TURBODOS OPERATING SYSTEM IMS SERIAL PORT BAUD RATE TABLE (OPTIONAL)
 COPYRIGHT (C) 1980 BY SOFTWARE 2000, INC.

```

;
; COPYRIGHT (C) 1980 BY SOFTWARE 2000, INC.
;
; AUTHORS: RONALD E. RAIKES
;          MICHAEL D. BUSCH
;
; VERSION: 09/08/81
;
; .IDENT BRT442 ;MODULE ID
;
; .INSERT DREQUATE ;DRIVER SYMBOLIC EQUIVALENCES
;
0600 BR50 = 1536 ;50 BAUD TIMER VALUE
0400 BR75 = 1024 ;75 BAUD TIMER VALUE
02BA BR110 = 698 ;110 BAUD TIMER VALUE
023B BR1345 = 571 ;134.5 BAUD TIMER VALUE
0200 BR150 = 512 ;150 BAUD TIMER VALUE
0100 BR300 = 256 ;300 BAUD TIMER VALUE
0080 BR600 = 128 ;600 BAUD TIMER VALUE
0040 BR1200 = 64 ;1200 BAUD TIMER VALUE
002B BR1800 = 43 ;1800 BAUD TIMER VALUE
0026 BR2000 = 38 ;2000 BAUD TIMER VALUE
0020 BR2400 = 32 ;2400 BAUD TIMER VALUE
0015 BR3600 = 21 ;3600 BAUD TIMER VALUE
0010 BR4800 = 16 ;4800 BAUD TIMER VALUE
000B BR7200 = 11 ;7200 BAUD TIMER VALUE
0008 BR9600 = 8 ;9600 BAUD TIMER VALUE
0004 BR192K = 4 ;19200 BAUD TIMER VALUE

;
5000 RTCCNT =: 20480 ;RTC COUNT (1/60 SECOND TICK)
003C TICSEC =: 60 ;RTC TICKS PER SECOND
;
0000' ; .LOC .PROG.# ;LOCATE IN PROGRAM AREA
;
0000' 0600 BRTBL:: .WORD BR50 ;50 BAUD TIMER VALUE
0002' 0400 .WORD BR75 ;75 BAUD TIMER VALUE
0004' 02BA .WORD BR110 ;110 BAUD TIMER VALUE
0006' 023B .WORD BR1345 ;134.5 BAUD TIMER VALUE
0008' 0200 .WORD BR150 ;150 BAUD TIMER VALUE
000A' 0100 .WORD BR300 ;300 BAUD TIMER VALUE
000C' 0080 .WORD BR600 ;600 BAUD TIMER VALUE
000E' 0040 .WORD BR1200 ;1200 BAUD TIMER VALUE
0010' 002B .WORD BR1800 ;1800 BAUD TIMER VALUE
0012' 0026 .WORD BR2000 ;2000 BAUD TIMER VALUE
0014' 0020 .WORD BR2400 ;2400 BAUD TIMER VALUE
0016' 0015 .WORD BR3600 ;3600 BAUD TIMER VALUE
0018' 0010 .WORD BR4800 ;4800 BAUD TIMER VALUE
001A' 000B .WORD BR7200 ;7200 BAUD TIMER VALUE
001C' 0008 .WORD BR9600 ;9600 BAUD TIMER VALUE
001E' 0004 .WORD BR192K ;19200 BAUD TIMER VALUE
;
; .END

```


3K401 - TURBODOS OPERATING SYSTEM IMS FLOPPY DISK DRIVER
 COPYRIGHT (C) 1981, SOFTWARE 2000, INC.

```

;
; COPYRIGHT (C) 1981, SOFTWARE 2000, INC.
;
; AUTHORS: RONALD E. RAIKES
;          MICHAEL D. BUSCH
;
; VERSION: 09/10/81
;
; IDENT   DSK401                ;MODULE ID
;
; INSERT DREQUATE              ;DRIVER SYMBOLIC EQUIVALENCES
;
0082     CH1DMA = 82H            ;CHANNEL 1 DMA REGISTER (FDC)
0083     CH1TC  = 83H            ;CHANNEL 1 TERMINAL COUNT (FDC)
0088     DMACTL = 88H            ;DMA COMMAND AND STATUS REGISTERS
008A     DSKSEL = 8AH            ;DISK SELECT PORT
008C     DSKCTL = 8CH            ;STATUS AND INT MASK (BOARD)
008E     FDCST  = 8EH            ;DISK CONTROLLER STATUS (uPD-765)
008F     FDCDAT = 8FH            ;DISK CONTROLLER DATA (uPD-765)
;
0042     CH1ENA = 42H            ;DMA CHANNEL 1 ENABLE COMMAND
0000     DMAVFY = 00H            ;DMA VERIFY COMMAND
0040     DMARD  = 40H            ;DMA READ COMMAND
0080     DMAWR  = 80H            ;DMA WRITE COMMAND
;
0003     FDCSFY = 03H            ;FDC SPECIFY COMMAND
0004     FDCSDS = 04H            ;FDC SENSE DRIVE STATUS COMMAND
0007     FDCRCL = 07H            ;FDC RECALIBRATE COMMAND
0008     FDCSIS = 08H            ;FDC SENSE INTERRUPT STATUS COMMAND
000A     FDCRID = 0AH            ;FDC READ ID COMMAND
000D     FDCFMT = 0DH            ;FDC FORMAT TRACK COMMAND
000F     FDCSK  = 0FH            ;FDC SEEK COMMAND
0005     FDCWR  = 05H            ;FDC WRITE COMMAND
0006     FDCRD  = 06H            ;FDC READ COMMAND
;
0000     DSKENI = 0              ;DISK CONTROLLER ENABLE INTERRUPTS
0007     DSKDLC = 7              ;DISK CONTROLLER DELAY COMPLETE
;
0006     FDCMFM = 6              ;FDC DOUBLE-DENSITY BIT
0007     FDCMT  = 7              ;FDC MULTI-TRACK BIT
;
0004     FDCBSY = 4              ;FDC BUSY STATUS
0005     FDCSE  = 5              ;FDC SEEK END
0006     FDCOUT = 6              ;FDC OUTPUT MODE
0007     FDCRDY = 7              ;FDC READY FOR DATA
;
00D0     SRT8R  = (16-3)<4       ;8 INCH FDD STEP RATE (3 MS-REMEX)
00A0     SRT8S  = (16-6)<4       ;8 INCH FDD STEP RATE (6 MS-SHUG)
;
0024     HDLT   = 18*2           ;FDD HEAD LOAD TIME (36 MS)
0001     HDUT   = 1              ;FDD HEAD UNLOAD TIME (16 MS)
;
0003     STONR  = 3              ;STATUS REGISTER 0 NOT READY
0004     STOEC  = 4              ;STATUS REGISTER 0 EQUIP CHECK

```

DSK401 - TURBODOS OPERATING SYSTEM IMS FLOPPY DISK DRIVER
 COPYRIGHT (C) 1981, SOFTWARE 2000, INC.

```

0005          STOSE      = 5          ;STATUS REGISTER 0 SEEK END
;
0000          ST1MA     = 0          ;STATUS REGISTER 1 MISSING ADDR MK
0001          ST1NW     = 1          ;STATUS REGISTER 1 NOT WRITABLE
0002          ST1ND     = 2          ;STATUS REGISTER 1 NO DATA
0004          ST1OR     = 4          ;STATUS REGISTER 1 OVER RUN
0005          ST1DE     = 5          ;STATUS REGISTER 1 DATA ERROR
;
0003          ST3TS     = 3          ;STATUS REGISTER 3 TWO-SIDED
0004          ST3TO     = 4          ;STATUS REGISTER 3 TRACK 0
0005          ST3RDY    = 5          ;STATUS REGISTER 3 READY
0006          ST3WP     = 6          ;STATUS REGISTER 3 WRITE PROTECTED
;
000A          MAXTRY    = 10         ;MAX DISK TRY COUNT
;
0000          SLOWSR    = 0          ;SLOW STEP RATE (FLAGS)
;
0002          TSD       = 2          ;TWO-SIDED DISK BIT (TYPE CODE)
0003          DDD       = 3          ;DOUBLE DENSITY DISK BIT (TYPE CODE)
;
0004          MINI      = 4          ;MINI-FLOPPY DISK BIT (TYPE CODE)
;
0000'         ;          .LOC      .PROG.# ;LOCATE IN PROGRAM AREA
;
0000'         21 0000"    DSKDR%::LXI   H,DMXSPH ;GET MUTUAL EXCLUSION SEMAPHORE
0003'         CD 0000:04 CALL    WAIT# ;DISPATCH IF NECESSARY
0006'         CD 0012'   CALL    ..DD  ;CALL DISK DRIVER
0009'         F5        PUSH   PSW     ;SAVE RETURN CODE
000A'         21 0000"    LXI   H,DMXSPH ;GET MUTUAL EXCLUSION SEMAPHORE
000D'         CD 0000:05 CALL    SIGNAL# ;SIGNAL PROCESS AS READY
0010'         F1        POP    PSW     ;RESTORE RETURN CODE
0011'         C9        RET          ;DONE
;
0012'         ED73 0012" ..DD:  SSPD   RETSP  ;SAVE ERROR RETURN STACK POINTER
0016'         DD7 E00   MOV    A,PDRFCN(X) ;GET PD REQ FUNCTION NUMBER
0019'         B7        ORA    A        ;PD REQ FUNCTION NUMBER=0?
001A'         283D     JRZ    RDDSK    ;IF SO, CONTINUE
001C'         3D        DCR    A        ;PD REQ FUNCTION NUMBER=1?
001D'         284C     JRZ    WRDSK    ;IF SO, CONTINUE
001F'         3D        DCR    A        ;PD REQ FUNCTION NUMBER=2?
0020'         CA 028A'  JZ     RETDST  ;IF SO, CONTINUE
0023'         3D        DCR    A        ;PD REQ FUNCTION NUMBER=3?
0024'         CA 0303'  JZ     RETRDY  ;IF SO, CONTINUE
0027'         3D        DCR    A        ;PD REQ FUNCTION NUMBER=4?
0028'         285C     JRZ    FMTDSK  ;IF SO, CONTINUE
002A'         C9        RET          ;ELSE, DONE
;
002B'         002E     ;          .WORD  NITLEN+2 ;INITIALIZATION CODE LENGTH
;
002D'         DB8E     DSKIN%::IN   FDCST  ;GET FDC STATUS
002F'         3C        INR    A        ;CONTROLLER PRESENT?
0030'         2821     JRZ    ..X     ;IF NOT, CONTINUE
0032'         3EC3     MVI    A,JMP   ;ELSE, INITIALIZE INTERRUPT VECTOR
0034'         32 0028. STA    5*8    ;(VECTORED INTERRUPT-5)

```

3K401 - TURBODOS OPERATING SYSTEM IMS FLOPPY DISK DRIVER
 COPYRIGHT (C) 1981, SOFTWARE 2000, INC.

```

0037' 21 0387' LXI H,DSKISR
003A' 22 0029 SHLD (5*8)+1
003D' AF XRA A
003E' D388 OUT DMACTL ;DISABLE DMA CONTROLLER
0040' 3E03 MVI A,FDCSFY ;GET FDC SPECIFY COMMAND
0042' CD 0412' CALL CMDRDY ;OUTPUT COMMAND TO FDC
0045' 3ED1 MVI A,SRT8R!HDUT ;GET REMEX STEP RT/HEAD UNLD
0047' CD 0418' CALL DATOUT ;OUTPUT IT TO FDC
004A' 3E24 MVI A,HDLT ;GET HEAD LOAD TIME/NON-DMA BIT
004C' CD 0418' CALL DATOUT ;OUTPUT IT TO FDC
004F' 3E01 MVI A,1<DSKENI
0051' D38C OUT DSKCTL ;ENABLE CONTROLLER INTERRUPTS
0053' 21 002D' ..X: LXI H,DSKIN% ;GET INITIALIZATION CODE ADDRESS
0056' C3 0000:06 JMP DEALOC# ;DE-ALLOCATE INITIALIZATION CODE

;
002C ;NITLEN = .-DSKIN% ;INITIALIZATION CODE LENGTH
;
0059' 3E0A RDDSK: MVI A,MAXTRY ;GET MAX TRY COUNT
005B' 32 000C' STA TRYCNT ;SET TRY COUNT
005E' 3E06 ..RD: MVI A,FDCRD ;GET FDC READ COMMAND
0060' 0E40 MVI C,DMARD ;GET DMA READ COMMAND
0062' CD 00F2' CALL DSKCOM ;CALL COMMON CODE
0065' C8 RZ ;NO ERRORS, RET A=0
0066' CD 0149' CALL RETRY ;ERRORS, RECALIBRATE
0069' 18F3 JMPR ..RD ;TRY AGAIN

;
006B' 3E0A WRDSK: MVI A,MAXTRY ;GET MAX TRY COUNT
006D' 32 000C' STA TRYCNT ;SET TRY COUNT
0070' 3E05 ..WR: MVI A,FDCWR ;GET FDC WRITE COMMAND
0072' 0E80 MVI C,DMAWR ;GET DMA WRITE COMMAND
0074' CD 00F2' CALL DSKCOM ;CALL COMMON CODE
0077' 2008 JRNZ ..RT ;IF ERRORS, RETRY
0079' 3E06 MVI A,FDCRD ;ELSE, GET FDC READ COMMAND
007B' 0E00 MVI C,DMAVFY ;GET DMA VERIFY COMMAND
007D' CD 00F2' CALL DSKCOM ;CALL COMMON CODE
0080' C8 RZ ;NO ERRORS, RET A=0
0081' CD 0149' ..RT: CALL RETRY ;ERRORS, RECALIBRATE
0084' 18EA JMPR ..WR ;TRY AGAIN

;
0086' DD7E02 FMTDSK: MOV A,PDRTK(X) ;GET PD REQ TRACK NUMBER
0089' B7 ORA A ;PD REQUEST TRACK NUMBER=0?
008A' 2006 JRNZ ..NTRO ;IF NOT, CONTINUE
008C' CD 0350' CALL SELCUR ;ELSE, SELECT I/O DISK
008F' CD 024F' CALL RECAL ;RECALIBRATE DRIVE
0092' 3E0A ..NTRO: MVI A,MAXTRY ;GET MAX TRY COUNT
0094' 32 000C' STA TRYCNT ;SET TRY COUNT
0097' CD 01D4' ..FMT: CALL SEEK ;SELECT DISK AND SEEK
009A' 3E80 MVI A,DMAWR ;GET DMA WRITE COMMAND
009C' 32 0011' STA IODMAC ;SET DMA COMMAND
009F' DD6E08 MOV L,PDRTC(X) ;GET PD REQ TRANSFER COUNT
00A2' DD6609 MOV H,PDRTC+1(X)
00A5' DD5E0A MOV E,PDRCMA(X) ;GET PD REQUEST DMA ADDRESS
00A8' DD560B MOV D,PDRCMA+1(X)
00AB' CD 016B' CALL DMANIT ;INITIALIZE DMA CONTROLLER

```

DSK401 - TURBODOS OPERATING SYSTEM IMS FLOPPY DISK DRIVER
 COPYRIGHT (C) 1981, SOFTWARE 2000, INC.

```

00AE' 3E0D          MVI    A,FDCEMT ;GET FORMAT TRACK COMMAND
00B0' DDCB047E    BIT    7,PDRSEC(X) ;DOUBLE DENSITY FLAG SET?
00B4' 2802        JRZ    ..SD ;IF NOT, CONTINUE
00B6' CBF7        SET    FDCMF,M,A ;ELSE, SET DOUBLE DENSITY BIT
00B8' CD 0412'    ..SD:  CALL  CMDRDY ;SEND FORMAT COMMAND TO FDC
00BB' DD7E01        MOV    A,PDRDRV(X) ;GET PD REQUEST DRIVE NUMBER
00BE' DDCB057E    BIT    7,PDRSEC+1(X) ;HEAD NUMBER ONE FLAG SET?
00C2' 2802        JRZ    ..HDO ;IF NOT, CONTINUE
00C4' CBD7        SET    2,A ;ELSE, SET HEAD ONE BIT
00C6' CD 0418'    ..HDO:  CALL  DATOUT ;OUTPUT UNIT NUMBER TO FDC
00C9' DD7E04        MOV    A,PDRSEC(X) ;GET PD REQUEST SECTOR (LSB)
00CC' E603        ANI    3 ;EXTRACT FORMAT SECTOR SIZE
00CE' CD 0418'    CALL  DATOUT ;OUTPUT FORMAT SECTOR SIZE TO FDC
00D1' DD7E06        MOV    A,PDRSC(X) ;GET PD REQUEST SECTOR COUNT
00D4' CD 0418'    CALL  DATOUT ;OUTPUT SECTORS/TRACK TO FDC
00D7' DD7E05        MOV    A,PDRSEC+1(X) ;GET PD REQUEST SECTOR (MSB)
00DA' E67F        ANI    7FH ;EXTRACT FORMAT GAP LENGTH
00DC' CD 0418'    CALL  DATOUT ;OUTPUT FORMAT GAP LENGTH TO FDC
00DF' 3EE5        MVI    A,OE5H ;GET FORMAT FILLER BYTE
00E1' CD 0418'    CALL  DATOUT ;OUTPUT FORMAT FILLER BYTE TO FDC
00E4' CD 0380'    CALL  WTINT ;WAIT FOR INTERRUPT
00E7' 3A 0021"    LDA    STO ;GET STATUS REGISTER 0
00EA' E6C0        ANI    OCOH ;ANY ERRORS?
00EC' C8          RZ    ;NO ERRORS, RET A=0
00ED' CD 0149'    CALL  RETRY ;ERRORS, RECALIBRATE
00FO' 18A5        JMPR  ..FMT ;TRY AGAIN

;
00F2' 32 0010"    DSKCOM: STA  IORWC ;SET FDC READ/WRITE COMMAND
00F5' 79          MOV    A,C ;GET DMA COMMAND
00F6' 32 0011"    STA  IODMAC ;SET DMA COMMAND
00F9' DD7E04        MOV    A,PDRSEC(X) ;GET PD REQ SECTOR NUMBER
00FC' 32 0015"    STA  CURSEC ;SET CURRENT SECTOR
00FF' DD6E0A        MOV    L,PDRDMA(X) ;GET PD REQUEST DMA ADDRESS
0102' DD660B        MOV    H,PDRDMA+1(X)
0105' 22 0016"    SHLD  CURADR ;SET CURRENT DMA ADDRESS
0108' DD7E06        MOV    A,PDRSC(X) ;GET PD REQ SECTOR COUNT
010B' 32 0018"    STA  CURSC ;SET CURRENT SECTOR COUNT
010E' CD 01D4'    CALL  SEEK ;SELECT DISK AND SEEK
0111' AF          XRA    A
0112' 32 0019"    STA  IOERR ;CLEAR I/O ERROR STATUS
0115' CD 0183'    ..RWL:  CALL  SETID ;SET UP SECTOR ID INFO
0118' CD 0159'    CALL  SETUP ;SETUP READ/WRITE DMA
011B' CD 03E1'    CALL  CMDOUT ;SEND SECTOR ID INFO TO FDC
011E' CD 0380'    CALL  WTINT ;WAIT FOR INTERRUPT
0121' 21 0019"    LXI  H,IOERR ;GET I/O ERROR STATUS
0124' 3A 0021"    LDA  STO ;GET STATUS REGISTER 0
0127' B6          ORA  M ;ADD NEW STATUS
0128' 77          MOV  M,A ;UPDATE I/O ERROR STATUS
0129' CD 0453'    CALL  GETXLT ;GET TRANSLATION TABLE ADDRESS
012C' 2815        JRZ  ..NI ;IF TRANSLATION NOT REQUIRED, CONTINUE
012E' 21 0015"    LXI  H,CURSEC ;ELSE, GET CURRENT SECTOR NUMBER
0131' 34          INR  M ;INCREMENT CURRENT SECTOR
0132' CD 0448'    CALL  CALCSS ;CALC SECTOR SIZE
0135' EB          XCHG ;SECTOR SIZE TO DE-REG

```

DISK401 - TURBODOS OPERATING SYSTEM IMS FLOPPY DISK DRIVER
 COPYRIGHT (C) 1981, SOFTWARE 2000, INC.

```

0136' 2A 0016"      LHLD  CURADR ;GET CURRENT DMA ADDRESS
0139' 19            DAD    D ;CALC NEXT DMA ADDRESS
013A' 22 0016"      SHLD  CURADR ;UPDATE CURRENT DMA ADDRESS
013D' 21 0018"      LXI   H,CURSC ;GET CURRENT SECTOR COUNT
0140' 35            DCR   M ;DECREMENT CURRENT SECTOR COUNT
0141' 20D2          JRNZ  ..RWL ;IF TRANSFER NOT COMPLETE, CONTINUE
0143' 3A 0019"      LDA    IOERR ;GET I/O ERROR STATUS
0146' E6C0          ANI   OCOH ;EXTRACT COMPLETION STATUS
0148' C9            RET    ;DONE

;
0149' 0E07          ;RETRY: MVI   C,ABEL ;GET BELL CHARACTER
014B' CD 0000:07   CALL  CONOUT# ;OUTPUT TO CONSOLE
014E' CD 024F'     CALL  RECAL  ;RECALIBRATE DRIVE
0151' 21 000C"     LXI   H,TRYCNT ;GET RETRY COUNT
0154' 35            DCR   M ;DECREMENT RETRY COUNT
0155' C0            RNZ   ;IF COUNT NOT EXHAUSTED, TRY AGAIN
0156' C3 0470'     JMP   FATAL ;CONTINUE

;
0159' CD 0453'     ;SETUP: CALL  GETXLT ;GET TRANSLATION TABLE ADDRESS
015C' DD6E08        MOV   L,PDRTC(X) ;GET PD REQ TRANSFER COUNT
015F' DD6609        MOV   H,PDRTC+1(X)
0162' 2803          JRZ   ..NI ;IF NO TRANSLATION RQRD, CONTINUE
0164' CD 0448'     CALL  CALCSS ;ELSE, CALC SECTOR SIZE
0167' ED5B 0016"   ..NI: LDED  CURADR ;GET CURRENT DMA ADDRESS

;
016B' AF            ;DMANIT: XRA   A
016C' D388          OUT  DMACTL ;RESET DMA CONTROLLER
016E' 2B            DCX   H ;TERMINAL COUNT-1 FOR 8257
016F' 7D            MOV   A,L ;GET LSB OF TERMINAL COUNT
0170' D383          OUT  CH1TC ;SEND LSB OF TERMINAL COUNT
0172' 3A 0011"     LDA    IODMAC ;GET I/O DMA COMMAND
0175' B4            ORA   H ;ADD TO MSB OF TERMINAL COUNT
0176' D383          OUT  CH1TC ;SEND MSB OF TERMINAL COUNT
0178' 7B            MOV   A,E ;GET LSB
0179' D382          OUT  CH1DMA ;OUTPUT IT TO DMA CONTROLLER
017B' 7A            MOV   A,D ;GET MSB
017C' D382          OUT  CH1DMA ;OUTPUT IT TO DMA CONTROLLER
017E' 3E42          MVI   A,CH1ENA ;GET CHANNEL 1 ENABLE COMMAND
0180' D388          OUT  DMACTL ;ENABLE DMA CONTROLLER
0182' C9            RET    ;DONE

;
0183' DD7E02        ;SETID: MOV   A,PDRTK(X) ;GET PD REQ TRACK NUMBER
0186' 32 001A"     STA   CYL ;SET CYLINDER
0189' 3A 0015"     LDA   CURSEC ;GET CURRENT SECTOR
018C' 4F            MOV   C,A ;SECTOR NUMBER TO C-REG
018D' CD 0453'     CALL  GETXLT ;GET TRANSLATION TABLE ADDRESS
0190' 2804          JRZ   ..NI ;IF TRANSLATION NOT REQUIRED, CONT
0192' 0600          MVI   B,0 ;ELSE, MAKE SECTOR DOUBLE LENGTH
0194' 09            DAD   B ;INDEX INTO TRANSLATION TABLE
0195' 4E            MOV   C,M ;GET TRANSLATED SECTOR NUMBER
0196' 0C            ..NI: INR   C ;CONVERT SECTOR TO BASE 1
0197' DD4613        MOV   B,SECTRK(X) ;GET NUMBER OF SECTORS/TRACK
019A' CD 0461'     CALL  GETTCA ;GET DISK TYPE CODE ADDRESS
019D' CB56          BIT   TSD,M ;TWO SIDED DISK?

```

DSK401 - TURBODOS OPERATING SYSTEM IMS FLOPPY DISK DRIVER
 COPYRIGHT (C) 1981, SOFTWARE 2000, INC.

```

019F' 2802          JRZ      ..SSD    ;IF NOT, CONTINUE
01A1' CB38          SRLR     B          ;ELSE, CALC NUMBER OF SECTORS/SIDE
01A3' 78           ..SSD:  MOV     A,B      ;GET NUMBER OF SECTORS/SIDE
01A4' 32 001E'    STA     EOT      ;SET END OF TRACK SECTOR NUMBER
01A7' B9           CMP     C          ;FRONT SIDE OF DISK?
01A8' 3E00        MVI     A,0      ;PRESET FOR FRONT SIDE
01AA' 3005        JRNC    ..FS      ;IF FRONT SIDE, CONTINUE
01AC' 79           MOV     A,C      ;GET SECTOR NUMBER
01AD' 90           SUB     B          ;SUBTRACT ONE SIDES WORTH
01AE' 4F           MOV     C,A      ;TO C-REG
01AF' 3E01        MVI     A,1      ;GET HEAD #1
01B1' 32 001B'    ..FS:  STA     HEAD     ;SET HEAD NUMBER
01B4' 79           MOV     A,C      ;GET SECTOR NUMBER
01B5' 32 001C'    STA     REC      ;SET RECORD NUMBER
01B8' DD7E12      MOV     A,SECSIZ(X) ;GET SECTOR SIZE
01BB' 32 001D'    STA     SIZE     ;SET RECORD SIZE
01BE' B7           ORA     A          ;N=0?
01BF' 3E80        MVI     A,128    ;PRESET DTL=128
01C1' 2802        JRZ      ..NO      ;IF N=0, CONTINUE
01C3' 3EFF        MVI     A,OFFH   ;ELSE, DTL=OFFH
01C5' 32 0020'    ..NO:  STA     DTL      ;SET DATA LENGTH
01C8' CD 0469'    CALL   GETDST   ;GET DST ADDRESS
01CB' 11 0000:08 LXI     D,GAPLEN# ;GET OFFSET TO GAP LENGTH
01CE' 19           DAD     D          ;CALC GAP LENGTH ADDRESS
01CF' 7E           MOV     A,M      ;GET GAP LENGTH
01D0' 32 001F'    STA     GPL      ;SET GAP LENGTH
01D3' C9           RET          ;DONE

;
01D4' CD 0350'    ;SEEK: CALL   SELCUR   ;SELECT I/O DISK
01D7' DD7E01      MOV     A,PDRDRV(X) ;GET PD REQ DISK NUMBER
01DA' 3C           INR     A          ;INCREMENT IT
01DB' 47           MOV     B,A      ;TO B-REG
01DC' 37           STC          ;SET CARRY FLAG
01DD' 21 0000     LXI     H,0      ;INITIALIZE MASK
01E0' ED6A        ..SL:  DADC     H          ;GET DRIVE MASK
01E2' 10FC        DJNZ   ..SL
01E4' ED5B 000D' LDED   CALTBL   ;GET DRIVE CALIBRATED TABLE
01E8' 2C           INR     L
01E9' 2D           DCR     L          ;DRIVE 0-7?
01EA' 2006        JRNZ   ..D07    ;IF SO, CONTINUE
01EC' 7A           MOV     A,D      ;GET CALIBRATED MAP
01ED' B4           ORA     H          ;SET CALIBRATED BIT
01EE' BA           CMP     D          ;WAS IT CALIBRATED?
01EF' 57           MOV     D,A      ;UPDATE MAP
01F0' 1804        JMPR   ..UM
01F2' 7B           ..D07: MOV     A,E      ;GET CALIBRATED MAP
01F3' B5           ORA     L          ;SET CALIBRATED BIT
01F4' BB           CMP     E          ;WAS IT CALIBRATED?
01F5' 5F           MOV     E,A      ;UPDATE MAP
01F6' ED53 000D' ..UM:  SDED   CALTBL   ;UPDATE TABLE
01FA' 2844        JRZ      ..NRR   ;IF DRIVE CALIBRATED, CONTINUE
01FC' 3A 000F'    LDA     FLAGS    ;ELSE, GET FLAGS
01FF' CB47        BIT     SLOWSR,A ;SLOW STEP RATE SET?
0201' 203A        JRNZ   ..RD      ;IF SO, CONTINUE

```

DSK401 - TURBODOS OPERATING SYSTEM IMS FLOPPY DISK DRIVER
 COPYRIGHT (C) 1981, SOFTWARE 2000, INC.

```

0203' DD7E01          MOV      A,PDRDRV(X) ;GET PD REQ DISK NUMBER
0206' CD 0259'      CALL     RECCMD ;SEND RECALIBRATE COMMAND
0209' 201D          JRNZ    ..SSSR ;IF ERRORS, CONTINUE
020B' DD7E01          MOV      A,PDRDRV(X) ;GET PD REQ DISK NUMBER
020E' 47            MOV      B,A ;CONTROLLER DISK TO B-REG
020F' 0E4C          MVI      C,76 ;CYLINDER 76 TO C-REG
0211' CD 026E'      CALL     SEKCMD ;SEND SEEK COMMAND
0214' 2012          JRNZ    ..SSSR ;IF ERRORS, CONTINUE
0216' CD 028A'      CALL     RETDST ;ELSE, READ DISK ID
0219' 3A 0024*     LDA      RCYL ;GET PRESENT CYLINDER NUMBER
021C' FE4C          CPI      76 ;DRIVE STEP TO CYLINDER 76?
021E' 2008          JRNZ    ..SSSR ;IF NOT, CONTINUE
0220' DD7E01          MOV      A,PDRDRV(X) ;GET PD REQ DISK NUMBER
0223' CD 0259'      CALL     RECCMD ;SEND RECALIBRATE COMMAND
0226' 2818          JRZ     ..NRR ;IF NO ERRORS, CONTINUE
0228' 3E03          ;..SSSR: MVI     A,FDCSFY ;GET FDC SPECIFY COMMAND
022A' CD 0412'      CALL     CMDRDY ;OUTPUT COMMAND TO FDC
022D' 3EA1          MVI     A,SRT8S!HDUT ;GET SHUGHART STEP RATE/HEAD (
;
022F' CD 0418'      NLOAD   CALL     DATOUT ;OUTPUT IT TO FDC
0232' 3E24          MVI     A,HDLT ;GET HEAD LOAD TIME/NON-DMA BIT
0234' CD 0418'      CALL     DATOUT ;OUTPUT IT TO FDC
0237' FB            EI      ;ENABLE INTERRUPTS
0238' 21 000F*     LXI     H,FLAGS ;GET FLAGS
023B' CBC6          SET     SLOWSR,M ;SET SLOW STEP RATE BIT
023D' CD 024F'      ;..RD: CALL    RECAL ;RE-CALIBRATE DRIVE
0240' DD7E01          ;..NRR: MOV     A,PDRDRV(X) ;GET PD REQ DISK NUMBER
0243' 47            MOV     B,A ;CONTROLLER DISK TO B-REG
0244' DD7E02          MOV     A,PDRTRK(X) ;GET PD REQ TRACK NUMBER
0247' 4F            MOV     C,A ;CYLINDER TO C-REG
0248' CD 026E'      CALL     SEKCMD ;SEND SEEK COMMAND
024B' C8            RZ     ;IF NO ERRORS, DONE
024C' C3 0470'     JMP     FATAL ;CONTINUE
;
024F' DD7E01          ;RECAL: MOV    A,PDRDRV(X) ;GET PD REQ DISK NUMBER
0252' CD 0259'      CALL     RECCMD ;SEND RECALIBRATE COMMAND
0255' C8            RZ     ;IF NO ERRORS, DONE
0256' C3 0470'     JMP     FATAL ;CONTINUE
;
0259' F5            ;RECCMD: PUSH   PSW ;SAVE CONTROLLER DISK
025A' 3E07          MVI     A,FDCRCL ;GET FDC RECALIBRATE COMMAND
025C' CD 0412'      CALL     CMDRDY ;OUTPUT COMMAND TO FDC
025F' F1            POP     PSW ;RESTORE CONTROLLER DISK
0260' CD 0418'      CALL     DATOUT ;OUTPUT IT TO FDC
0263' CD 0380'      CALL     WTINT ;WAIT FOR INTERRUPT
0266' 3A 0021*     LDA     STO ;GET STATUS REGISTER 0
0269' E6E0          ANI    OCOH!1<FDCSE ;EXTRACT COMPLETION STATUS
026B' FE20          CPI    1<FDCSE ;ANY ERRORS?
026D' C9            RET     ;DONE
;
026E' C5            ;SEKCMD: PUSH   B ;SAVE DISK/TRACK
026F' 3E0F          MVI     A,FDCSK ;GET FDC SEEK COMMAND
0271' CD 0412'      CALL     CMDRDY ;OUTPUT COMMAND TO FDC
0274' C1            POP     B ;RESTORE DISK/TRACK
    
```

DSK401 - TURBODOS OPERATING SYSTEM IMS FLOPPY DISK DRIVER
COPYRIGHT (C) 1981, SOFTWARE 2000, INC.

```

0275†   C5           PUSH   B           ;SAVE DISK/TRACK
0276†   78           MOV    A,B         ;GET CONTROLLER DISK
0277†   CD 0418†    CALL  DATOUT      ;OUTPUT IT TO FDC
027A†   C1           POP    B           ;RESTORE DISK/TRACK
027B†   79           MOV    A,C         ;GET CYLINDER NUMBER
027C†   CD 0418†    CALL  DATOUT      ;OUTPUT IT TO FDC
027F†   CD 0380†    CALL  WTINT       ;WAIT FOR INTERRUPT
0282†   3A 0021†    LDA   STO         ;GET STATUS REGISTER 0
0285†   E6E0        ANI   0COH11<FDCSE ;EXTRACT COMPLETION STATUS
0287†   FE20        CPI   1<FDCSE     ;ANY ERRORS?
0289†   C9           RET                    ;DONE

;
028A†   CD 0303†    RETDST: CALL  RETRDY    ;RETURN READY STATUS
028D†   E7           ORA   A           ;DRIVE READY?
028E†   C8           RZ                    ;IF NOT, DONE
028F†   0E00        MVI   C,0         ;ELSE, GET INITIAL TYPE VALUE
0291†   3A 0029†    LDA   ST3         ;GET STATUS REGISTER 3
0294†   CB5F        BIT   ST3TS,A     ;ONE-SIDED DISK?
0296†   2802        JRZ   ..OS        ;YES
0298†   CBD1        SET   TSD,C       ;SET TWO-SIDED DISK BIT
029A†   DD7E01      ..OS: MOV   A,PDRDRV(X) ;GET PD REQ DISK NUMBER
029D†   32 0014†    STA  RIDDSK      ;SET READ ID DISK
02A0†   CD 02EE†    CALL ..FD        ;FIND DISK DENSITY
02A3†   280F        JRZ   ..DF        ;IF DENSITY FOUND, CONTINUE
02A5†   C5           PUSH  B           ;ELSE, SAVE DISK TYPE CODE
02A6†   DD7E01      MOV   A,PDRDRV(X) ;GET PD REQ DISK NUMBER
02A9†   CD 0259†    CALL  RECCMD     ;RECALIBRATE DRIVE
02AC†   C1           POP    B           ;RESTORE DISK TYPE CODE
02AD†   2032        JRNZ  ..NR        ;IF UNABLE TO RECALIBRATE, CONTINUE
02AF†   CD 02EE†    CALL  ..FD        ;ELSE, ATTEMPT TO FIND DISK DENSITY
02B2†   202D        JRNZ  ..NR        ;IF DENSITY NOT FOUND, CONTINUE
02B4†   B1           ..DF: ORA   C           ;ADD SECTOR SIZE TO TYPE CODE
02B5†   4F           MOV   C,A         ;
02B6†   CB51        BIT   TSD,C       ;TWO SIDED BIT SET?
02B8†   2814        JRZ   ..FDI       ;IF NOT, CONTINUE
02BA†   C1 0014†    LXI  H,RIDDSK    ;GET READ ID DISK
02BD†   CBD6        SET   2,M         ;SET HEAD BIT
02BF†   3E4A        MVI  A,FDCRID!1<FDCMFM ;GET READ ID CMD (DD)
02C1†   CB59        BIT   DDD,C       ;DOUBLE DENSITY BIT SET?
02C3†   2002        JRNZ  ..DD        ;IF SO, CONTINUE
02C5†   CBB7        RES  FDCMFM,A     ;ELSE, RESET MFM BIT
02C7†   CD 02FD†    ..DD: CALL ..RID      ;ATTEMPT TO READ ID ON BACK SIDE
02CA†   2802        JRZ   ..FDI       ;IF READABLE, CONTINUE
02CC†   CB91        RES  TSD,C       ;ELSE, RESET TWO SIDED BIT
02CE†   11 0000:09 ..FDI: LXI  D,DSTBLS# ;GET DISK SPEC TABLES
02D1†   79           ..SL2: MOV   A,C         ;GET DISK TYPE CODE
02D2†   21 0000:0A LXI  H,DTCO#     ;GET OFFSET TO DISK TYPE CODE
02D5†   19           DAD  D           ;CALC DISK TYPE CODE ADDRESS
02D6†   BE           CMP  M           ;DISK SPEC TABLE FOUND?
02D7†   280A        JRZ   ..DSTF     ;IF SO, CONTINUE
02D9†   EB           XCHG              ;DISK SPEC TABLE ADDRESS TO HL-REG
02DA†   5E           MOV  E,M         ;GET DISK SPEC TABLE LINK POINTER
02DB†   23           INX  H           ;
02DC†   56           MOV  D,M         ;

```


SK401 - TURBODOS OPERATING SYSTEM IMS FLOPPY DISK DRIVER
COPYRIGHT (C) 1981, SOFTWARE 2000, INC.

```

02DD' 7A          MOV      A,D
02DE' B3          ORA      E          ;END OF LIST?
02DF' 20F0       JRNZ   ..SL2      ;IF NOT, CONTINUE
02E1' AF         ..NR:   XRA      A          ;ELSE, SET RETURN CODE=0
02E2' C9         RET
02E3' 13         ..DSTF: INX     D          ;DONE
02E4' 13         INX     D          ;ADVANCE PAST LINK POINTER
02E5' DD730C     MOV     PDRDST(X),E ;SET DISK SPEC TABLE ADDRESS
02E8' DD720D     MOV     PDRDST+1(X),D
02EB' 3EFF       MVI     A,OFFH     ;SET RETURN CODE=OFFH
02ED' C9         RET
02EE' 3E0A       ..FD:   MVI     A,FDCRID ;GET FDC READ ID COMMAND (SD)
02F0' CD 02FD'   CALL   ..RID      ;ATTEMPT TO READ SINGLE-DENSITY
02F3' C8         RZ          ;IF SINGLE-DENSITY, DONE
02F4' 3E4A       MVI     A,FDCRID+1<FDCMF ;GET READ ID CMD (DD)
02F6' CD 02FD'   CALL   ..RID      ;ATTEMPT TO READ DOUBLE-DENSITY
02F9' C0         RNZ       ;IF UNABLE, DONE
02FA' CBD9       SET     DDD,C     ;SET DOUBLE-DENSITY DISK BIT
02FC' C9         RET
02FD' C5         ..RID:  PUSH   B          ;DONE
02FE' CD 0326'   CALL   READID    ;SAVE BC
0301' C1         POP     B         ;READ DISK ID
0302' C9         RET          ;RESTORE BC
;
0303' DD7E01     ;RETRDY: MOV    A,PDRDRV(X) ;GET PD REQ DISK NUMBER
0306' FE04       CPI     4          ;TEST FOR VALID DRIVE NUMBER
0308' 3E00       MVI     A,0       ;PRESET RETURN CODE=0
030A' D0         RNC      ;IF INVALID DRIVE, RETURN NOT READY
030B' DB8E       IN      FDCST    ;GET FDC STATUS
030D' 3C         INR     A         ;CONTROLLER PRESENT?
030E' C8         RZ          ;IF NOT, DONE
030F' CD 0350'   CALL   SELCUR    ;ELSE, SELECT REQUESTED DRIVE
0312' CD 031C'   CALL   ..RDY     ;CHECK IF DRIVE READY
0315' C0         RNZ       ;IF SO, DONE
0316' 21 0001   LXI     H,1       ;ELSE, DELAY ONE TICK...
0319' CD 0000:0B CALL   DELAY#    ;...SO 765 CAN SCAN
031C' CD 036D'   ..RDY:  CALL   SENSDDS ;SENSE DRIVE STATUS
031F' CB6F       BIT     ST3RDY,A ;DRIVE READY?
0321' 3E00       MVI     A,0       ;PRESET RETURN CODE=0
0323' C8         RZ          ;IF DRIVE NOT READY, DONE
0324' 2F         CMA      ;ELSE, SET RETURN CODE=OFFH
0325' C9         RET          ;DONE
;
0326' CD 0412'   ;READID: CALL   CMDRDY   ;OUTPUT COMMAND TO FDC
0329' 3A 0014"   LDA     RIDDSK   ;GET READ ID DISK
032C' CD 0418'   CALL   DATOUT   ;OUTPUT IT TO FDC
032F' CD 0380'   CALL   WTINT    ;WAIT FOR INTERRUPT
0332' 3A 0021"   LDA     STO      ;GET STATUS REGISTER 0
0335' E6C0       ANI     OCOH    ;EXTRACT COMPLETION STATUS
0337' 3A 0027"   LDA     RSIZE   ;RETURN SECTOR SIZE
033A' C9         RET          ;DINE
;
033B'           ;DLCPOL:
033B' 0000       .WORD 0          ;DELAY COMPLETE POLL ROUTINE

```

DSK401 - TURBODOS OPERATING SYSTEM IMS FLOPPY DISK DRIVER
COPYRIGHT (C) 1981, SOFTWARE 2000, INC.

```

033D' 0000          .WORD 0

;
033F' DB8C          ;DLCPR: IN      DSKCTL  ;GET DISK CONTROLLER STATUS
0341' CB7F          BIT      DSKDLC,A ;DELAY COMPLETE (MOTORS RUNNING)?
0343' C8            RZ          ;IF NOT, DONE
0344' 21 033B'     LXI      H,DLCPOL ;ELSE, GET POLL ROUTINE
0347' CD 0000:0C   CALL     UNLINK# ;UNLINK POLL ROUTINE FROM POLL LIST
034A' 21 0006"     LXI      H,DWTSPH ;GET DISK WAIT SEMAPHORE
034D' C3 0000:05   JMP      SIGNAL# ;CONTINUE

;
0350' DB8C          ;SELCUR: IN      DSKCTL  ;GET DISK CONTROLLER STATUS
0352' 0F           RRC          ;EXTRACT SELECTED DRIVE
0353' E603         ANI          3
0355' 4F           MOV      C,A      ;TO C-REG
0356' DD7E01.     MOV      A,PDRDRV(X) ;GET PD REQ DISK NUMBER
0359' B9           CMP      C      ;DRIVE ALREADY SELECTED?
035A' 2802         JRZ          ..DAS ;IF SO, CONTINUE
035C' D38A         OUT          DSKSEL ;ELSE, SELECT CONTROLLER DISK
035E' 11 033B'     ..DAS: LXI      D,DLCPOL ;GET POLL ROUTINE
0361' CD 0000:0D   CALL     LNKPOL# ;CREATE POLL ROUTINE
0364' CD 033F'     CALL     DLCPR   ;EXECUTE POLL ROUTINE
0367' 21 0006"     LXI      H,DWTSPH ;GET DISK WAIT SEMAPHORE
036A' C3 0000:04   JMP      WAIT#   ;DISPATCH IF NECESSARY

;
036D' 3E04          ;SENSDS: MVI     A,FDCSDS ;GET FDC SENSE DRIVE STATUS CMD
036F' CD 0412'     CALL     CMDRDY ;OUTPUT COMMAND TO FDC
0372' DD7E01       MOV      A,PDRDRV(X) ;GET PD REQ DISK NUMBER
0375' CD 0418'     CALL     DATOUT  ;OUTPUT IT TO FDC
0378' CD 041F'     CALL     DATAIN ;GET STATUS REGISTER 3
037B' 32 0029"     STA      ST3    ;SAVE STATUS REGISTER 3
037E' FB           EI          ;ENABLE INTERRUPTS
037F' C9           RET          ;DONE

;
0380' FB           ;WTINT: EI          ;ENABLE INTERRUPTS
0381' 21 0006"     LXI      H,DWTSPH ;GET DISK WAIT SEMAPHORE
0384' C3 0000:04   JMP      WAIT#   ;DISPATCH IF NECESSARY

;
0387' ED73 0000:0E DSKISR: SSPD   INTSP# ;SAVE INTERRUPT STACK POINTER
038B' 31 0000:0F   LXI      SP,INTSTK# ;SET UP AUX STACK
038E' F5           PUSH     PSW    ;SAVE REGISTERS
038F' C5           PUSH     B
0390' D5           PUSH     D
0391' E5           PUSH     H
0392' DB8E         ..RQML: IN      FDCST  ;GET FDC STATUS
0394' CB7F          BIT      FDCRDY,A ;FDC READY FOR CONVERSATION?
0396' 28FA         JRZ          ..RQML ;IF NOT, WAIT
0398' 32 0028"     STA      MAINST ;SAVE MAIN STATUS REGISTER
039B' CB77          BIT      FDCOUT,A ;FDC IN OUTPUT MODE?
039D' 2020         JRNZ     ..RW   ;IF SO, PROCESS
039F' 3E08         MVI      A,FDCSIS ;GET SENSE INTERRUPT STATUS CMD
03A1' D38F         OUT      FDCDAT ;OUTPUT IT TO FDC DATA REGISTER
03A3' CD 041F'     CALL     DATAIN ;GET STATUS REGISTER 0
03A6' 4F           MOV      C,A      ;SAVE IT IN C-REG
03A7' E6C0         ANI      OCOH   ;EXTRACT COMPLETION STATUS

```

BK401 - TURBODOS OPERATING SYSTEM IMS FLOPPY DISK DRIVER
 COPYRIGHT (C) 1981, SOFTWARE 2000, INC.

```

03A9'   FE80           CPI      80H      ;INTERRUPT STACK EMPTY?
03AB'   2829           JRZ      ..X      ;IF SO, DONE
03AD'   CD 041F'      CALL     DATAIN   ;GET PRESENT CYLINDER NUMBER
03B0'   CB69           BIT      STOSE,C  ;READY LINE CHANGE STATE?
03B2'   28DE           JRZ      ..RQML   ;IF SO, IGNORE
03B4'   32 0024'      STA      RCYL    ;ELSE, SAVE PCN
03B7'   79            MOV      A,C      ;GET STATUS REGISTER 0
03B8'   32 0021'      STA      STO     ;SAVE IT
03BB'   3E01           MVI     A,1      ;SET INTERRUPT COMPLETION STATUS
03BD'   180F           JMPR    ..SIGC   ;CONTINUE
03BF'   21 0021'      ..RW:   LXI     H,RESULT ;GET RESULT TABLE
03C2'   0607           MVI     B,7      ;GET LENGTH OF RESULT PHASE
03C4'   CD 041F'      ..RL:   CALL    DATAIN   ;GET RESULT BYTE FROM FDC
03C7'   77            MOV      M,A      ;STORE IN RESULT AREA
03C8'   23            INX     H        ;INCREMENT POINTER
03C9'   10F9           DJNZ    ..RL     ;READ ALL SEVEN BYTES
03CB'   AF            XRA     A
03CC'   D388           OUT    DMACTL    ;DISABLE DMA CONTROLLER
03CE'   21 0006'      ..SIGC: LXI     H,DWTSPH ;GET DISK WAIT SEMAPHORE
03D1'   CD 0000:05   CALL    SIGNAL#   ;SIGNAL PROCESS AS READY
03D4'   18BC           JMPR    ..RQML   ;FLUSH ANY REMAINING INTERRUPTS
03D6'   E1            ..X:   POP     H    ;REGISTERS
03D7'   D1            POP     D
03D8'   C1            POP     B
03D9'   F1            POP     PSW
03DA'   ED7B 0000:0E LSPD    INTSP#   ;RESTORE STACK POINTER
03DE'   C3 0000:10   JMP     ISRXIT#  ;CONTINUE

;
03E1'   CD 0461'      ;CMDOUT: CALL   GETTCA  ;GET DISK TYPE CODE ADDRESS
03E4'   3A 0010'      LDA     IORWC    ;GET READ/WRITE COMMAND
03E7'   CB5E           BIT     DDD,M    ;DOUBLE DENSITY DISK?
03E9'   2802           JRZ     ..SD     ;IF NOT, SINGLE DENSITY
03EB'   CBF7           SET     FDCMFM,A ;ELSE, SET DOUBLE DENSITY BIT
03ED'   CB56           ..SD:   BIT     TSD,M ;TWO-SIDED DISK?
03EF'   2802           JRZ     ..SS     ;IF NOT, SINGLE SIDED
03F1'   CBF7           SET     FDCMT,A ;ELSE, SET MULTI-TRACK BIT
03F3'   CD 0412'      ..SS:   CALL    CMDRDY  ;SEND COMMAND TO FDC
03F6'   DD7E01        MOV     A,PDRDRV(X) ;GET PD REQ DISK NUMBER
03F9'   21 001B'      LXI     H,HEAD   ;GET HEAD NUMBER
03FC'   CB46           BIT     0,M      ;HEAD #0?
03FE'   2802           JRZ     ..FS     ;IF SO, CONTINUE
0400'   CBD7           SET     2,A      ;ELSE, SET HEAD #1 BIT IN I/O DISK
0402'   CD 0418'      ..FS:   CALL    DATOUT  ;OUTPUT IT TO FDC
0405'   21 001A'      LXI     H,IDINFO ;GET SECTOR ID INFO
0408'   0607           MVI     B,7      ;B=LENGTH OF ID INFO
040A'   7E            ..IDL:  MOV     A,M      ;GET BYTE FROM LIST
040B'   23            INX     H        ;INCREMENT POINTER
040C'   CD 0418'      CALL    DATOUT  ;OUTPUT BYTE TO FDC
040F'   10F9           DJNZ    ..IDL   ;SEND ENTIRE LIST
0411'   C9            RET            ;DONE

;
0412'   CD 042A'      ;CMDRDY: CALL   OUTRDY  ;WAIT FOR FDC READY
0415'   F3            DI            ;DISABLE INTERRUPTS
0416'   1803           JMPR    OUTCOM  ;JOIN COMMON CODE

```

DSK401 - TURBODOS OPERATING SYSTEM IMS FLOPPY DISK DRIVER
 COPYRIGHT (C) 1981, SOFTWARE 2000, INC.

```

0418'   CD 042A'   ;
                ;DATOUT: CALL   OUTRDY   ;WAIT FOR FDC READY
041B'   79        ;
041C'   D38F     ;OUTCOM: MOV   A,C       ;RESTORE OUTPUT BUTE
041E'   C9        ;           OUT   FDCDAT   ;OUTPUT BYTE TO FDC DATA REGISTER
                ;           RET           ;DONE
041F'   DB8E     ;
0421'   07        ;DATAIN: IN    FDCST   ;GET FDC STATUS
0422'   30FB     ;           RLC       ;TEST FDC FOR READY
0424'   07        ;           JRNC    DATAIN ;IF NOT READY, WAIT
0425'   300B     ;           RLC       ;TEST FDC DIRECTION
0427'   DB8F     ;           JRNC    FDCERR  ;IF WRONG DIRECTION, DIAGNOSE
0429'   C9        ;           IN    FDCDAT   ;GET FDC DATA BYTE
                ;           RET           ;DONE
042A'   4F        ;
042B'   DB8E     ;OUTRDY: MOV   C,A       ;SAVE OUTPUT BYTE
042D'   07        ;..RW:  IN    FDCST   ;GET FDC STATUS
042E'   30FB     ;           RLC       ;TEST FDC FOR READY
0430'   07        ;           JRNC    ..RW   ;IF NOT READY, WAIT
0431'   D0        ;           RLC       ;TEST FDC DIRECTION
                ;           RNC       ;IF DIRECTION CORRECT, DONE
0432'   CD 0000:11 ;
0435'   87        ;FDCERR: CALL  DMS#     ;SOUND BELL
0436'   CD 0000:12 ;           .ASCIS [ABEL]
0439'   CD 0000:11 ;           CALL  CONSO#   ;SHIFT CONSOLE TO ERROR LINE
043C'   464443204572 ;           CALL  DMS#     ;DISPLAY ERROR MESSAGE
0445'   C3 0445'  ;           .ASCIS "FDC Error"
                ;           JMP     .       ;HALT
0448'   21 0080   ;
044B'   DD7E12   ;CALCSS: LXI   H,128    ;GET 128 BYTE SECTOR LENGTH
044E'   3D        ;           MOV   A,SECSIZ(X) ;GET SECTOR SIZE
044F'   F8        ;..SL:  DCR   A       ;DECREMENT SECTOR SIZE
0450'   29        ;           RM    ;IF UNDERFLOW, DONE
0451'   18FB     ;           DAD   H       ;ELSE, SHIFT SECTOR SIZE LEFT
                ;           JMPR  ..SL   ;CONTINUE
0453'   CD 0469'  ;
0456'   11 0000:13 ;GETXLT: CALL  GETDST   ;GET DST ADDRESS
0459'   19        ;           LXI   D,XLTBL#  ;GET OFFSET TO TRANSLATION TABLE
045A'   5E        ;           DAD   D       ;CALC TRANSLATION TABLE ADDRESS
045B'   23        ;           MOV   E,M     ;GET TRANSLATION TABLE ADDRESS
045C'   56        ;           INX   H
045D'   EB        ;           MOV   D,M
                ;           XCHG   ;TRANSLATION TABLE ADDRESS TO HL-REG
045E'   7C        ;           MOV   A,H
045F'   B5        ;           ORA   L       ;TRANSLATION REQUIRED?
0460'   C9        ;           RET           ;DONE
0461'   CD 0469'  ;
0464'   11 0000:14 ;GETTCA: CALL  GETDST   ;GET DST ADDRESS
0467'   19        ;           LXI   D,TYPCOD# ;GET OFFSET TO DISK TYPE CODE
0468'   C9        ;           DAD   D       ;CALC DISK TYPE CODE ADDRESS
                ;           RET           ;DONE
0469'   DD6E0C   ;GETDST: MOV   L,PDRDST(X) ;GET PD REQUEST DST ADDRESS

```

3K401 - TURBODOS OPERATING SYSTEM IMS FLOPPY DISK DRIVER
 COPYRIGHT (C) 1981, SOFTWARE 2000, INC.

```

046C' DD660D MOV H,PDRDST+1(X)
046F' C9 RET ;DONE

;
0470' EDTB 0012" FATAL: LSPD RETSP ;RESTORE STACK POINTER
0474' 3EFF MVI A,OFFH ;RETURN ERROR CODE
0476' C9 RET ;DONE

;
0000" .LOC .DATA.# ;LOCATE IN DATA AREA

;
0000" DMXSPH: ;MUTUAL EXCLUSION SEMAPHORE
0000" 0001 .WORD 1 ;SEMAPHORE COUNT
0002" 0002" ..DMXH: .WORD ..DMXH ;SEMAPHORE P/D HEAD
0004" 0002" .WORD ..DMXH

;
0006" DWTSPH: ;DISK WAIT SEMAPHORE
0006" 0000 .WORD 0 ;SEMAPHORE COUNT
0008" 0008" ..DWTH: .WORD ..DWTH ;SEMAPHORE P/D HEAD
000A" 0008" .WORD ..DWTH

;
000C" 00 TRYCNT: .BYTE 0 ;TRY COUNT
000D" 0000 CALTBL: .WORD 0 ;DRIVE CALIBRATED TABLE
000F" 00 FLAGS: .BYTE 0 ;FLAGS
0010" 00 IORWC: .BYTE 0 ;I/O READ/WRITE COMMAND
0011" 00 IODMAC: .BYTE 0 ;I/O DMA COMMAND
0012" 0000 RETSP: .WORD 0 ;ERROR RETURN STACK POINTER
0014" 00 RIDDSK: .BYTE 0 ;READ ID DISK
0015" 00 CURSEC: .BYTE 0 ;CURRENT SECTOR NUMBER
0016" 0000 CURADR: .WORD 0 ;CURRENT DMA ADDRESS
0018" 00 CURSC: .BYTE 0 ;CURRENT SECTOR COUNT
0019" 00 IOERR: .BYTE 0 ;I/O ERROR STATUS

;
001A" IDINFO: ;SECTOR ID INFO LIST
001A" 00 CYL: .BYTE 0 ;DISK CYLINDER NUMBER
001B" 00 HEAD: .BYTE 0 ;DISK HEAD NUMBER
001C" 00 REC: .BYTE 0 ;DISK RECORD NUMBER
001D" 00 SIZE: .BYTE 0 ;DISK SECTOR SIZE
001E" 00 EOT: .BYTE 0 ;END OF TRACK SECTOR NUMBER
001F" 00 GPL: .BYTE 0 ;DISK GAP 3 SIZE
0020" 00 DTL: .BYTE 0 ;DISK SECTOR SIZE WHEN SIZE=0

;
0021" RESULT: ;RESULT PHASE LIST
0021" 00 ST0: .BYTE 0 ;STATUS REGISTER 0
0022" 00 ST1: .BYTE 0 ;STATUS REGISTER 1
0023" 00 ST2: .BYTE 0 ;STATUS REGISTER 2
0024" 00 RCYL: .BYTE 0 ;DISK CYLINDER NUMBER
0025" 00 RHEAD: .BYTE 0 ;DISK HEAD NUMBER
0026" 00 RREC: .BYTE 0 ;DISK RECORD NUMBER
0027" 00 RSIZE: .BYTE 0 ;DISK SECTOR SIZE
0028" 00 MAINST: .BYTE 0 ;MAIN STATUS REGISTER
0029" 00 ST3: .BYTE 0 ;STATUS REGISTER 3

;
.END

```

DSKFMT - TURBODOS OPERATING SYSTEM DRIVE SPECIFICATION TABLES
 COPYRIGHT (C) 1981, SOFTWARE 2000, INC.

```

;
; COPYRIGHT (C) 1981, SOFTWARE 2000, INC.
;
; AUTHORS: RONALD E. RAIKES
;          MICHAEL D. BUSCH
;
; VERSION: 09/08/81
;
; IDENT DSKFMT ;MODULE ID
;
; INSERT DREQUATE ;DRIVER SYMBOLIC EQUIVALENCES
;
0002 TSD = 2 ;TWO-SIDED DISK BIT (TYPE CODE)
0003 DDD = 3 ;DOUBLE DENSITY DISK BIT (TYPE CODE)
;
0004 MINI = 4 ;MINI-FLOPPY DISK BIT (TYPE CODE)
;
0000' .LOC .PROG.# ;LOCATE IN PROGRAM AREA
;
; 1024 BYTE SECTOR, DOUBLE-DENSITY, TWO-SIDED
;
0000' 0011' DSTBLS: .WORD .+DSTL ;DISK SPEC TABLE LINK POINTER
0002' 04 .BYTE 4 ;BLOCK SIZE
0003' 0268 .WORD (77*(16*(1<3)))/(1<4) ;NUMBER OF BLOCKS
0005' 04 .BYTE 4 ;NUMBER OF DIRECTORY BLOCKS
0006' 03 .BYTE 3 ;PHYSICAL SECTOR SIZE (2^N*128)
0007' 0010 .WORD 16 ;PHYSICAL SECTORS PER TRACK
0009' 004D .WORD 77 ;PHYSICAL TRACKS PER DISK
000B' 0000 .WORD 0 ;NUMBER OF RESERVED TRACKS
000D' 0000 .WORD 0 ;TRANSLATION TABLE ADDRESS
000F' 0F .BYTE 1<DDD!1<TSD!3 ;DISK TYPE CODE
0010' 35 .BYTE 35H ;GAP LENGTH
;
; 1024 BYTE SECTOR, DOUBLE-DENSITY, TWO-SIDED (MINI)
;
; .WORD .+DSTL ;DISK SPEC TABLE LINK POINTER
; .BYTE 4 ;BLOCK SIZE
; .WORD (40*(10*(1<3)))/(1<4) ;NUMBER OF BLOCKS
; .BYTE 2 ;NUMBER OF DIRECTORY BLOCKS
; .BYTE 3 ;PHYSICAL SECTOR SIZE (2^N*128)
; .WORD 10 ;PHYSICAL SECTORS PER TRACK
; .WORD 40 ;PHYSICAL TRACKS PER DISK
; .WORD 0 ;NUMBER OF RESERVED TRACKS
; .WORD 0 ;TRANSLATION TABLE ADDRESS
; .BYTE 1<MINI!1<DDD!1<TSD!3 ;DISK TYPE CODE
; .BYTE 35H ;GAP LENGTH
;
; 1024 BYTE SECTOR, DOUBLE-DENSITY, ONE-SIDED
;
0011' 0022' .WORD .+DSTL ;DISK SPEC TABLE LINK POINTER
0013' 04 .BYTE 4 ;BLOCK SIZE
0014' 0134 .WORD (77*(8*(1<3)))/(1<4) ;NUMBER OF BLOCKS
0016' 03 .BYTE 3 ;NUMBER OF DIRECTORY BLOCKS
0017' 03 .BYTE 3 ;PHYSICAL SECTOR SIZE (2^N*128)

```


DSKFMT - TURBODOS OPERATING SYSTEM DRIVE SPECIFICATION TABLES
 COPYRIGHT (C) 1981, SOFTWARE 2000, INC.

0027'	03	.BYTE	3	;NUMBER OF DIRECTORY BLOCKS
0028'	02	.BYTE	2	;PHYSICAL SECTOR SIZE (2^N*128)
0029'	0010	.WORD	16	;PHYSICAL SECTORS PER TRACK
002B'	004D	.WORD	77	;PHYSICAL TRACKS PER DISK
002D'	0000	.WORD	0	;RESERVED TRACKS
002F'	0000	.WORD	0	;TRANSLATION TABLE ADDRESS
0031'	06	.BYTE	1<TSD12	;DISK TYPE CODE
0032'	1B	.BYTE	1BH	;GAP LENGTH

512 BYTE SECTOR, SINGLE-DENSITY, ONE-SIDED

0033'	0044'	.WORD	+.DSTL	;DISK SPEC TABLE LINK POINTER
0035'	04	.BYTE	4	;BLOCK SIZE
0036'	009A	.WORD	(77*(8*(1<2)))/(1<4)	;NUMBER OF BLOCKS
0038'	02	.BYTE	2	;NUMBER OF DIRECTORY BLOCKS
0039'	02	.BYTE	2	;PHYSICAL SECTOR SIZE (2^N*128)
003A'	0008	.WORD	8	;PHYSICAL SECTORS PER TRACK
003C'	004D	.WORD	77	;PHYSICAL TRACKS PER DISK
003E'	0000	.WORD	0	;RESERVED TRACKS
0040'	0000	.WORD	0	;TRANSLATION TABLE ADDRESS
0042'	02	.BYTE	2	;DISK TYPE CODE
0043'	1B	.BYTE	1BH	;GAP LENGTH

256 BYTE SECTOR, DOUBLE-DENSITY, TWO-SIDED

.WORD	+.DSTL	;DISK SPEC TABLE LINK POINTER
.BYTE	4	;BLOCK SHZET
.WORD	(77*(52*(1<1)))/(1<4)	;NUMBER OF BLOCKS
.BYTE	4	;NUMBER OF DIRECTORY BLOCKS
.BYTE	1	;PHYSICAL SECTOR SIZE (2^N*128)
.WORD	52	;PHYSICAL SECTORS PER TRACK
.WORD	77	;PHYSICAL TRACKS PER DISK
.WORD	0	;RESERVED TRACKS
.WORD	0	;TRANSLATION TABLE ADDRESS
.BYTE	1<DDD11<TSD11	;DISK TYPE CODE
.BYTE	0EH	;GAP LENGTH

256 BYTE SECTOR, DOUBLE-DENSITY, ONE-SIDED

.WORD	+.DSTL	;DISK SPEC TABLE LINK POINTER
.BYTE	4	;BLOCK SIZE
.WORD	(77*(26*(1<1)))/(1<4)	;NUMBER OF BLOCKS
.BYTE	2	;NUMBER OF DIRECTORY BLOCKS
.BYTE	1	;PHYSICAL SECTOR SIZE (2^N*128)
.WORD	26	;PHYSICAL SECTORS PER TRACK
.WORD	77	;PHYSICAL TRACKS PER DISK
.WORD	0	;RESERVED TRACKS
.WORD	0	;TRANSLATION TABLE ADDRESS
.BYTE	1<DDD11	;DISK TYPE CODE
.BYTE	0EH	;GAP LENGTH

256 BYTE SECTOR, SINGLE-DENSITY, TWO-SIDED

.WORD	+.DSTL	;DISK SPEC TABLE LINK POINTER
-------	--------	-------------------------------

SKFMT - TURBODOS OPERATING SYSTEM DRIVE SPECIFICATION TABLES
 COPYRIGHT (C) 1981, SOFTWARE 2000, INC.

```

;
; .BYTE 4 ;BLOCK SIZE
; .WORD (77*(30*(1<1)))/(1<4) ;NUMBER OF BLOCKS
; .BYTE 3 ;NUMBER OF DIRECTORY BLOCKS
; .BYTE 1 ;PHYSICAL SECTOR SIZE (2^N*128)
; .WORD 30 ;PHYSICAL SECTORS PER TRACK
; .WORD 77 ;PHYSICAL TRACKS PER DISK
; .WORD 0 ;RESERVED TRACKS
; .WORD 0 ;TRANSLATION TABLE ADDRESS
; .BYTE 1<TSD!1 ;DISK TYPE CODE
; .BYTE 0EH ;GAP LENGTH
;
; 256 BYTE SECTOR, SINGLE-DENSITY, ONE-SIDED
;
; .WORD .+DSTL ;DISK SPEC TABLE LINK POINTER
; .BYTE 4 ;BLOCK SIZE
; .WORD (77*(15*(1<1)))/(1<4) ;NUMBER OF BLOCKS
; .BYTE 2 ;NUMBER OF DIRECTORY BLOCKS
; .BYTE 1 ;PHYSICAL SECTOR SIZE (2^N*128)
; .WORD 15 ;PHYSICAL SECTORS PER TRACK
; .WORD 77 ;PHYSICAL TRACKS PER DISK
; .WORD 0 ;RESERVED TRACKS
; .WORD 0 ;TRANSLATION TABLE ADDRESS
; .BYTE 1 ;DISK TYPE CODE
; .BYTE 0EH ;GAP LENGTH
;
; 128 BYTE SECTOR, SINGLE-DENSITY, TWO-SIDED (OLD)
;
; .WORD .+DSTL ;DISK SPEC TABLE LINK POINTER
; .BYTE 4 ;BLOCK SIZE
; .WORD (76*(52*(1<0)))/(1<4) ;NUMBER OF BLOCKS
; .BYTE 2 ;NUMBER OF DIRECTORY BLOCKS
; .BYTE 0 ;PHYSICAL SECTOR SIZE (2^N*128)
; .WORD 52 ;PHYSICAL SECTORS PER TRACK
; .WORD 77 ;PHYSICAL TRACKS PER DISK
; .WORD 1 ;RESERVED TRACKS
; .WORD 0 ;TRANSLATION TABLE ADDRESS
; .BYTE 1<TSD ;DISK TYPE CODE
; .BYTE 7 ;GAP LENGTH
;
; 128 BYTE SECTOR, SINGLE-DENSITY, TWO-SIDED
;
; .WORD .+DSTL ;DISK SPEC TABLE LINK POINTER
; .BYTE 4 ;BLOCK SIZE
; .WORD (77*(52*(1<0)))/(1<4) ;NUMBER OF BLOCKS
; .BYTE 2 ;NUMBER OF DIRECTORY BLOCKS
; .BYTE 0 ;PHYSICAL SECTOR SIZE (2^N*128)
; .WORD 52 ;PHYSICAL SECTORS PER TRACK
; .WORD 77 ;PHYSICAL TRACKS PER DISK
; .WORD 0 ;RESERVED TRACKS
; .WORD 0 ;TRANSLATION TABLE ADDRESS
; .BYTE 1<TSD ;DISK TYPE CODE
; .BYTE 7 ;GAP LENGTH
;
; 128 BYTE SECTOR, SINGLE-DENSITY, ONE-SIDED

```

DSKFMT - TURBODOS OPERATING SYSTEM DRIVE SPECIFICATION TABLES
 COPYRIGHT (C) 1981, SOFTWARE 2000, INC.

```

;
0044' 0000 DSTA: .WORD 0 ;DISK SPEC TABLE LINK POINTER
0046' 03 DSTB: .BYTE 3 ;BLOCK SIZE
0047' 00F3 .WORD (75*(26*(1<0)))/(1<3) ;NUMBER OF BLOCKS
0049' 02 .BYTE 2 ;NUMBER OF DIRECTORY BLOCKS
004A' 00 .BYTE 0 ;PHYSICAL SECTOR SIZE (2^N*128)
004B' 001A .WORD 26 ;PHYSICAL SECTORS PER TRACK
004D' 004D .WORD 77 ;PHYSICAL TRACKS PER DISK
004F' 0002 .WORD 2 ;RESERVED TRACKS
;
000B XLTBL =: .-DSTB ;TRANSLATION TABLE ADDRESS OFFSET
;
0051' 0055' .WORD TRTBL ;TRANSLATION TABLE ADDRESS
;
000F DTCO =: .-DSTA ;DISK TYPE CODE OFFSET
000D TYPCOD =: .-DSTB ;DISK TYPE CODE OFFSET
;
0053' 00 .BYTE 0 ;DISK TYPE CODE
;
000E GAPLEN =: .-DSTB ;GAP LENGTH OFFSET
;
0054' 07 .BYTE 7 ;GAP LENGTH
;
0011 DSTL = .-DSTA ;DISK SPEC TABLE LENGTH
;
; SINGLE-DENSITY/SINGLE-SIDED SECTOR TRANSLATION TABLE
;
0055' 00060C121804 TRTBL: .BYTE 0,6,12,18,24,4,10,16,22
005E' 02080E140107 .BYTE 2,8,14,20,1,7,13,19,25
0067' 050B11170309 .BYTE 5,11,17,23,3,9,15,21
;
.END
    
```

TC442 - TURBODOS OPERATING SYSTEM IMS REAL TIME CLOCK ROUTINES
 COPYRIGHT (C) 1981 BY SOFTWARE 2000, INC.

```

;
; COPYRIGHT (C) 1981 BY SOFTWARE 2000, INC.
;
; AUTHORS: RONALD E. RAIKES
;          MICHAEL D. BUSCH
;
; VERSION: 09/08/81
;
; IDENT   RTC442           ;MODULE ID
;
; INSERT  DREQUATE        ;DRIVER SYMBOLIC EQUIVALENCES
0010      IOBASE = 10H      ;SERIAL/PARALLEL I/O PORT BASE
;
0016      TIM2   = IOBASE+06H ;TIMER 2 DATA REGISTER
0017      TIMCTL = IOBASE+07H ;TIMER CONTROL REGISTER
0018      SINTE  = IOBASE+08H ;SERIAL INTERRUPT ENABLE REGISTER
0019      T2RES  = IOBASE+09H ;TIMER 2 INTERRUPT RESET
;
0001      RTCENA = 1        ;REAL TIME CLOCK ENABLE BIT
;
00B6      T2CMD  = 0B6H     ;TIMER 2 COMMAND
;
0000"     .LOC   .DATA.#    ;LOCATE IN DATA AREA
0000"     00      TICCNT: .BYTE 0 ;TICK COUNTER
;
0000'     .LOC   .PROG.#    ;LOCATE IN PROGRAM AREA
;
0000'     002A   .WORD  NITLEN+2 ;INITIALIZATION CODE LENGTH
;
0002'     3EC3   RTCNIT::MVI  A, JMP ;INIT RTC INTERRUPT VECTOR ADDR
0004'     32 0008 STA      1*8
0007'     21 002A' LXI      H, RTCISR
000A'     22 0009 SHLD    (1*8)+1
000D'     3EB6   MVI      A, T2CMD ;GET TIMER 2 COMMAND
000F'     D317   OUT      TIMCTL ;SELECT TIMER 2
0011'     21 0000:04 LXI    H, RTCCNT# ;GET RTC COUNTER VALUE
0014'     7D     MOV      A, L ;GET LSB OF TIMER VALUE
0015'     D316   OUT      TIM2 ;OUTPUT IT TO TIMER 2 DATA REGISTER
0017'     7C     MOV      A, H ;GET MSB OF TIMER VALUE
0018'     D316   OUT      TIM2 ;OUTPUT IT TO TIMER 2 DATA REGISTER
001A'     21 0000:05 LXI    H, INTMSK# ;GET INTERRUPT MASK
001D'     CBCE   SET      1, M ;SET RTC INTERRUPT ENABLE BIT
001F'     3A 0000:05 LDA    INTMSK ;GET INTERRUPT MASK
0022'     D318   OUT      SINTE ;ENABLE RTC INTERRUPT MASK
0024'     21 0002' LXI    H, RTCNIT ;GET INITIALIZATION CODE ADDRESS
0027'     C3 0000:06 JMP    DEALOC# ;DE-ALLOCATE INITIALIZATION CODE
;
0028      NITLEN = .-RTCNIT ;INITIALIZATION CODE LENGTH
;
002A'     ED73 0000:07 RTCISR: SSPD INTSP# ;SAVE STACK POINTER
002E'     31 0000:08 LXI    SP, INTSTK# ;SET UP AUX STACK POINTER
0031'     F5     PUSH   PSW ;SAVE REGISTERS

```

RTC442 - TURBODOS OPERATING SYSTEM IMS REAL TIME CLOCK ROUTINES
 COPYRIGHT (C) 1981 BY SOFTWARE 2000, INC.

```

0032r  C5          PUSH    B
0033r  D5          PUSH    D
0034r  E5          PUSH    H
0035r  D319       OUT     T2RES ;RESET RTC INTERRUPT
0037r  21 0000r  LXI     H,TICCNT ;GET TICK COUNTER
003Ar  34          INR     M ;INCREMENT TICK COUNTER
003Br  7E          MOV     A,M ;GET TICK COUNT
003Cr  01 0000:09 LXI     B,TICSEC# ;GET NUMBER OF TICKS PER SECOND
003Fr  B9          CMP     C ;SECONDS COUNT REACHED?
0040r  3805       JRC     ..NSEC ;IF NOT, CONTINUE
0042r  3600       MVI     M,0 ;ELSE, RESET TICK COUNTER
0044r  CD 0000:0A CALL   RTCSEC# ;SERVICE REAL TIME CLOCK MANAGER
0047r  CD 0000:0B ..NSEC: CALL   DLYTIC# ;SERVICE DISPATCHER DELAY MANAGER
004Ar  E1          POP     H ;RESTORE REGISTERS
004Br  D1          POP     D
004Cr  C1          POP     B
004Dr  F1          POP     PSW
004Er  EDTB 0000:07 LSPD   INTSP# ;RESTORE STACK POINTER
0052r  C3 0000:0C JMP     ISRIT# ;CONTINUE
    
```

;

 .END

UPD401 - TURBODOS OPERATING SYSTEM BOOT PROM DRIVER FOR IMS 401
 COPYRIGHT (C) 1981, SOFTWARE 2000, INC.

```

;
; COPYRIGHT (C) 1981, SOFTWARE 2000, INC.
;
; AUTHORS: RONALD E. RAIKES
;          MICHAEL D. BUSCH
;
; VERSION: 07/21/81
;
; IDENT BPD401 ;MODULE ID
;
; INSERT EQUATE ;O/S SYMBOLIC EQUIVALENCES
;
0080 RAM == TBUF ;WORKING STORAGE ADDRESS
0040 RAMLEN = 64 ;WORKING STORAGE LENGTH
;
0082 CH1DMA = 82H ;CHANNEL 1 DMA REGISTER (FDC)
0083 CH1TC = 83H ;CHANNEL 1 TERMINAL COUNT (FDC)
0088 DMACTL = 88H ;DMA COMMAND AND STATUS REGISTERS
008A DSKSEL = 8AH ;DISK SELECT PORT
008C DSKCTL = 8CH ;STATUS AND INT MASK (BOARD)
008E FDCST = 8EH ;DISK CONTROLLER STATUS (uPD-765)
008F FDCDAT = 8FH ;DISK CONTROLLER DATA (uPD-765)
;
0042 CH1ENA = 42H ;DMA CHANNEL 1 ENABLE COMMAND
0000 DMAVFX = 00H ;DMA VERIFY COMMAND
0040 DMARD = 40H ;DMA READ COMMAND
0080 DMAWR = 80H ;DMA WRITE COMMAND
;
0003 FDCSFY = 03H ;FDC SPECIFY COMMAND
0004 FDCSDS = 04H ;FDC SENSE DRIVE STATUS COMMAND
0007 FDCRCL = 07H ;FDC RECALIBRATE COMMAND
0008 FDCSIS = 08H ;FDC SENSE INTERRUPT STATUS COMMAND
000A FDCRID = 0AH ;FDC READ ID COMMAND
000F FDCSK = 0FH ;FDC SEEK COMMAND
0085 FDCWR = 85H ;FDC WRITE COMMAND
0086 FDCRD = 86H ;FDC READ COMMAND
;
0000 DSKENI = 0 ;DISK CONTROLLER ENABLE INTERRUPTS
0007 DSKDLC = 7 ;DISK CONTROLLER DELAY COMPLETE
;
0006 FDCMFM = 6 ;FDC DOUBLE-DENSITY BIT
;
0004 FDCBSY = 4 ;FDC BUSY STATUS
0005 FDCSE = 5 ;FDC SEEK END
0006 FDCOUT = 6 ;FDC OUTPUT MODE
0007 FDCRDY = 7 ;FDC READY FOR DATA
;
00C0 SRT5 = (16-4)<4 ;5 INCH FDD STEP RATE (4 MS-MINI)
00A0 SRT8S = (16-6)<4 ;8 INCH FDD STEP RATE (6 MS-SHUGART)
;
00D0 SRT8R = (16-3)<4 ;8 INCH FDD STEP RATE (3 MS-REMEX)
00F0 SRT8P = (16-1)<4 ;8 INCH FDD STEP RATE (1 MS-PERSCI)
;
0024 HLT = 18*2 ;FDD HEAD LOAD TIME (36 MS)

```

BPD401 - TURBODOS OPERATING SYSTEM BOOT PROM DRIVER FOR IMS 401
 COPYRIGHT (C) 1981, SOFTWARE 2000, INC.

```

0001          HUT          = 1          ;FDD HEAD UNLOAD TIME (16 MS)
;
0003          STONR       = 3          ;STATUS REGISTER 0 NOT READY
0004          STOEC       = 4          ;STATUS REGISTER 0 EQUIP CHECK
0005          STOSE       = 5          ;STATUS REGISTER 0 SEEK END
;
0000          ST1MA       = 0          ;STATUS REGISTER 1 MISSING ADDR MK
0001          ST1NW       = 1          ;STATUS REGISTER 1 NOT WRITABLE
0002          ST1ND       = 2          ;STATUS REGISTER 1 NO DATA
0004          ST1OR       = 4          ;STATUS REGISTER 1 OVER RUN
0005          ST1DE       = 5          ;STATUS REGISTER 1 DATA ERROR
;
0003          ST3TS       = 3          ;STATUS REGISTER 3 TWO-SIDED
0004          ST3TO       = 4          ;STATUS REGISTER 3 TRACK 0
0005          ST3RDY      = 5          ;STATUS REGISTER 3 READY
0006          ST3WP       = 6          ;STATUS REGISTER 3 WRITE PROTECTED
;
0002          TSD         = 2          ;TWO-SIDED DISK BIT (TYPE CODE)
0003          DDD         = 3          ;DOUBLE DENSITY DISK BIT (TYPE CODE)
;
000A          MAXTRY      = 10         ;MAX TRY COUNT
;
00C0          .LOC        RAM+RAMLEN  ;LOCATE IN WORKING STORAGE AREA
;
00C0          IODSK:      .BLKB       1          ;DISK NUMBER
00C1          IOTRK:      .BLKW       1          ;TRACK NUMBER
00C3          IOSEC:      .BLKW       1          ;SECTOR NUMBER
00C5          IODMA:      .BLKW       1          ;DMA ADDRESS
00C7          ST3REG:     .BLKB       1          ;STATUS REGISTER 3
00C8          TRYCNT:     .BLKB       1          ;TRY COUNT
;
00C9          DSKNFO:     ;DISK TYPE INFORMATION
00C9          BLKSIZ:     .BLKB       1          ;BLOCK SIZE
00CA          NMBLKS:     .BLKW       1          ;NUMBER OF BLOCKS
00CC          NMBDIR:     .BLKB       1          ;NUMBER OF DIRECTORY BLOCKS
00CD          SECSIZ:     .BLKB       1          ;PHYSICAL SECTOR SIZE (2^N*128)
00CE          SECTRK:     .BLKW       1          ;PHYSICAL SECTORS PER TRACK
00D0          TRKDSK:     .BLKW       1          ;PHYSICAL TRACKS PER DISK
00D2          RESTRK:     .BLKW       1          ;NUMBER OF RESERVED TRACKS
00D4          XLTBL:      .BLKW       1          ;TRANSLATION TABLE ADDRESS
00D6          TYPCOD:     .BLKB       1          ;DISK TYPE CODE
00D7          GAPLEN:     .BLKB       1          ;GAP LENGTH
00DF          DNFOL      = .-DSKNFO    ;DISK INFO LENGTH
;
0000          .LOC        .PROG.#     ;LOCATE IN PROGRAM AREA
;
0000'         3E03        INIT::      MVI      A,FDSCFY  ;GET FDC SPECIFY COMMAND
0002'         CD 01B3'    CALL      DATOUT  ;OUTPUT FDC SPECIFY COMMAND
0005'         3EA1        MVI      A,SRTSS!HUT  ;GET STEP RATE/HEAD UNLTD TIME
0007'         CD 01B3'    CALL      DATOUT  ;OUTPUT STEP RATE/HEAD UNLTD TIME
000A'         3E24        MVI      A,HLT      ;GET HEAD LOAD TIME/NON-DMA BIT
000C'         CD 01B3'    CALL      DATOUT  ;OUTPUT HEAD LOAD TIME/NON-DMA BIT
000F'         21 0100    LXI      H,TPA     ;GET LOAD BASE ADDRESS

```

RPD401 - TURBODOS OPERATING SYSTEM BOOT PROM DRIVER FOR IMS 401
 COPYRIGHT (C) 1981, SOFTWARE 2000, INC.

```

0012'   C9                RET                ;DONE
;
0013'   FE04             ;SELECT:=CPI      4      ;TEST FOR VALID DRIVE
0015'   3061             JRNC              ..NR   ;IF INVALID DRIVE, CONTINUE
0017'   32 00C0         STA              IODSK  ;ELSE, SET DISK NUMBER
001A'   4F              MOV              C,A    ;DISK NUMBER TO C-REG
001B'   DB8C           IN                DSKCTL ;GET DISK CONTROLLER STATUS
001D'   0F             RRC              ;EXTRACT SELECTED DRIVE
001E'   E603          ANI              3
0020'   B9             CMP              C      ;DRIVE ALREADY SELECTED?
0021'   2803          JRZ              ..DS   ;IF SO, CONTINUE
0023'   79             MOV              A,C    ;ELSE, GET DISK NUMBER
0024'   D38A          OUT              DSKSEL ;SELECT DISK NUMBER
0026'   01 0014       ..DS:           LXI      B,20 ;DELAY 10 MILLISECONDS
0029'   10FE         ..DLY:         DJNZ     C
002B'   0D             DCR              C
002C'   20FB          JRNZ             ..DLY
002E'   3E04          MVI              A,FDCSDS ;GET FDC SENSE DRIVE STATUS CMD
0030'   CD 01B3'      CALL             DATOUT ;OUTPUT COMMAND TO FDC
0033'   3A 00C0         LDA              IODSK  ;GET DISK NUMBER
0036'   CD 01B3'      CALL             DATOUT ;OUTPUT IT TO FDC
0039'   CD 01A7'      CALL             DATAIN ;GET STATUS REGISTER 3
003C'   CB6F          BIT              ST3RDY,A ;DRIVE READY?
003E'   2838          JRZ              ..NR   ;IF NOT READY, CONTINUE
0040'   32 00C7         STA              ST3REG ;ELSE, SAVE STATUS REGISTER 3
0043'   CD 0165'      CALL             RECAL  ;RECALIBRATE DRIVE
0046'   2030          JRNZ             ..NR   ;IF ERRORS, CONTINUE
0048'   0E00          MVI              C,0    ;ELSE, GET INITIAL TYPE VALUE
004A'   21 00C7         LXI              H,ST3REG ;GET STATUS REGISTER 3
004D'   CB5E          BIT              ST3TS,M ;ONE-SIDED DISK?
004F'   2802          JRZ              ..OSD  ;YES
0051'   CBD1          SET              TSD,C   ;SET TWO-SIDED DISK BIT
0053'   3E0A         ..OSD:         MVI              A,FDCRID ;GET FDC READ ID COMMAND (SD)
0055'   CD 0089'      CALL             ..RID  ;ATTEMPT TO READ SINGLE-DENSITY
0058'   2809          JRZ              ..TPC  ;IF SINGLE-DENSITY, DONE
005A'   3E4A          MVI              A,FDCRID ;GET READ ID CMD (DD)
005C'   CD 0089'      CALL             ..RID  ;ATTEMPT TO READ DOUBLE-DENSITY
005F'   2017          JRNZ             ..NR   ;IF NOT DOUBLE-DENSITY, DONE
0061'   CBD9          SET              DDD,C   ;SET DOUBLE-DENSITY DISK BIT
0063'   B1             ..TPC:         ORA              C      ;ADD SECTOR SIZE TO TYPE CODE
0064'   4F             MOV              C,A    ;SAVE TYPE CODE IN C-REG
0065'   11 0000:04     LXI              D,DSTBLS# ;GET DST BASE ADDRESS
0068'   79             ..SL:           MOV              A,C    ;GET DISK TYPE CODE
0069'   21 0000:05     LXI              H,DTCO#  ;GET OFFSET TO DISK TYPE CODE
006C'   19             DAD              D      ;CALC DISK TYPE CODE ADDRESS
006D'   BE             CMP              M      ;DST FOUND?
006E'   EB             XCHG             ;DST ADDRESS TO HL-REG
006F'   2809          JRZ              ..DSTF ;IF DST FOUND, CONTINUE
0071'   5E             MOV              E,M    ;ELSE, GET NEXT DST ADDRESS
0072'   23             INX              H
0073'   56             MOV              D,M
0074'   7A             MOV              A,D
0075'   B3             ORA              E      ;END OF LIST?
0076'   20F0          JRNZ             ..SL   ;IF NOT, CONTINUE

```

BPD401 - TURBODOS OPERATING SYSTEM BOOT PROM DRIVER FOR IMS 401
 COPYRIGHT (C) 1981, SOFTWARE 2000, INC.

```

0078' AF          ..NR: XRA      A      ;SET RETURN CODE=0
0079' C9          RET          ;DONE
007A' 23          ..DSTF: INX     H      ;ADVANCE PAST LINK POINTER
007B' 23          INX     H
007C' E5          PUSH     H      ;SAVE DST ADDRESS
007D' 11 00C9     LXI     D,DSKNFO ;GET DISK INFO WORK AREA
0080' 01 000F     LXI     B,DNFOL  ;GET DISK INFO LENGTH
0083' EDB0       LDIR    ;COPY DST INTO WORK AREA
0085' E1         POP      ;RESTORE DST ADDRESS
0086' 3EFF       MVI     A,OFFH ;SET RETURN CODE=OFFH
0088' C9         RET          ;DONE
0089' C5          ..RID: PUSH     B      ;SAVE BC-REG
008A' CD 01B3'   CALL    DATOUT  ;OUTPUT COMMAND TO FDC
008D' 3A 00C0     LDA     IODSK   ;GET DISK NUMBER
0090' CD 01B3'   CALL    DATOUT  ;OUTPUT IT TO FDC
0093' CD 0170'   CALL    WTINT   ;WAIT FOR INTERRUPT
0096' 78         MOV     A,B      ;RETURN SECTOR SIZE
0097' C1         POP      B      ;RESTORE BC-REG
0098' C9         RET          ;DONE

;
0099' ED43 00C1  ;READ:: SBCD   IOTRK  ;SAVE TRACK NUMBER
009D' ED53 00C3  SDED   IOSEC  ;SAVE SECTOR NUMBER
00A1' 22 00C5    SHLD  IODMA  ;SAVE DMA ADDRESS
00A4' 21 00C8    LXI     H,TRYCNT ;GET TRY COUNT
00A7' 360A       MVI     M,MAXTRY ;INITIALIZE TRY COUNT
00A9' CD 0152'   ..RR:  CALL   SEEK   ;SEEK TO REQUESTED TRACK
00AC' C2 0143'   JNZ     ..ERR   ;IF ERRORS, CONTINUE
00AF' AF        XRA     A
00B0' D388       OUT    DMACTL  ;RESET DMA CONTROLLER
00B2' 21 0080    LXI     H,128   ;GET SECTOR SIZE=0 SECTOR LENGTH
00B5' 3A 00CD    LDA     SECSIZ ;GET PHYSICAL SECTOR SIZE
00B8' B7         ORA     A      ;PHYSICAL SECTOR SIZE=0?
00B9' 2804       JRZ     ..NO1  ;IF SO, CONTINUE
00BB' 29         ..SL:  DAD     H      ;ELSE, SHIFT HL-REG LEFT
00BC' 3D         DCR     A      ;SECTOR SIZE TIMES
00BD' 20FC       JRNZ   ..SL
00BF' 2B         ..NO1: DCX     H      ;COUNT -1 FOR 8257
00C0' 7D         MOV     A,L     ;GET LSB OF TERMINAL COUNT
00C1' D383       OUT    CH1TC  ;OUTPUT LSB OF TERMINAL COUNT
00C3' 7C         MOV     A,H     ;GET MSB OF TERMINAL COUNT
00C4' F640       ORI     DMARD   ;ADD DMA READ COMMAND
00C6' D383       OUT    CH1TC  ;OUTPUT MSB OF TERMINAL COUNT
00C8' 2A 00C5    LHL    IODMA  ;GET DMA ADDRESS
00CB' 7D         MOV     A,L     ;GET LSB OF DMA ADDRESS
00CC' D382       OUT    CH1DMA ;OUTPUT LSB OF DMA ADDRESS
00CE' 7C         MOV     A,H     ;GET MSB OF DMA ADDRESS
00CF' D382       OUT    CH1DMA ;OUTPUT MSB OF DMA ADDRESS
00D1' 3E42       MVI     A,CH1ENA ;GET CHANNEL 1 ENABLE COMMAND
00D3' D388       OUT    DMACTL  ;ENABLE DMA CONTROLLER
00D5' 3E86       MVI     A,FDCRD ;GET FDC READ COMMAND
00D7' 21 00D6    LXI     H,TYPCOD ;GET DISK TYPE CODE
00DA' CB5E       BIT     DDD,M   ;SINGLE DENSITY DISK?
00DC' 2802       JRZ     ..SD   ;IF SO, CONTINUE
00DE' CBF7       SET    FDCMFM,A ;ELSE, SET FDC MFM BIT

```


PD401 - TURBODOS OPERATING SYSTEM BOOT PROM DRIVER FOR IMS 401
 COPYRIGHT (C) 1981, SOFTWARE 2000, INC.

```

00E0'   CD 01B3'   ..SD:   CALL   DATOUT   ;OUTPUT FDC READ COMMAND
00E3'   3A 00C3   LDA     IOSEC   ;GET SECTOR NUMBER
00E6'   5F         MOV     E,A     ;SECTOR NUMBER TO E-REG
00E7'   2A 00D4   LHL    XLTBL   ;GET TRANSLATION TABLE ADDRESS
00EA'   7C         MOV     A,H
00EB'   B5         ORA     L        ;SECTOR TRANSLATION REQUIRED?
00EC'   2804     JRZ    ..NI     ;IF NOT, CONTINUE
00EE'   1600     MVI    D,0     ;ELSE, MAKE SECTOR DOUBLE LENGTH
00F0'   19         DAD    D        ;INDEX INTO TRANSLATION TABLE
00F1'   5E         MOV     E,M     ;GET TRANSLATED SECTOR NUMBER
00F2'   1C         ..NI:   INR     E        ;CONVERT SECTOR TO BASE 1
00F3'   3A 00CE   LDA     SECTRK ;GET NUMBER OF SECTORS/TRACK
00F6'   21 00D6   LXI    H,TYP   ;GET DISK TYPE CODE ADDRESS
00F9'   CB56     BIT     TSD,M   ;TWO SIDED DISK?
00FB'   2802     JRZ    ..SSD   ;IF NOT, CONTINUE
00FD'   CB3F     SRLR   A        ;ELSE, CALC NUMBER OF SECTORS/SIDE
00FF'   57         ..SSD:  MOV    D,A     ;SAVE NUMBER OF SECTORS/SIDE
0100'   0600     MVI    B,0     ;PRESET FOR FRONT SIDE
0102'   BB         CMP     E        ;FRONT SIDE OF DISK?
0103'   3004     JRNC   ..FS1   ;IF SO, CONTINUE
0105'   7B         MOV     A,E     ;ELSE, GET SECTOR NUMBER
0106'   92         SUB    D        ;SUBTRACT ONE SIDES WORTH
0107'   5F         MOV     E,A     ;SECTOR NUMBER TO C-REG
0108'   04         INR    B        ;SET HEAD NUMBER=1
0109'   3A 00C0   ..FS1:  LDA     IODSK  ;GET DISK NUMBER
010C'   04         INR    B
010D'   05         DCR    B        ;HEAD=0?
010E'   2802     JRZ    ..FS2   ;IF SO, CONTINUE
0110'   CBD7     SET    2,A     ;ELSE, SET HEAD BIT
0112'   CD 01B3'   ..FS2:  CALL   DATOUT  ;OUTPUT UNIT NUMBER
0115'   3A 00C1   LDA     IOTRK  ;GET TRACK NUMBER
0118'   CD 01B3'   CALL   DATOUT  ;OUTPUT TRACK NUMBER
011B'   78         MOV     A,B     ;GET HEAD NUMBER
011C'   CD 01B3'   CALL   DATOUT  ;OUTPUT HEAD NUMBER
011F'   7B         MOV     A,E     ;GET SECTOR NUMBER
0120'   CD 01B3'   CALL   DATOUT  ;OUTPUT SECTOR NUMBER
0123'   3A 00CD   LDA     SECSIZ ;GET SECTOR SIZE
0126'   F5         PUSH   PSW     ;SAVE SECTOR SIZE
0127'   CD 01B3'   CALL   DATOUT  ;OUTPUT SECTOR SIZE
012A'   7A         MOV     A,D     ;GET EOT
012B'   CD 01B3'   CALL   DATOUT  ;OUTPUT EOT
012E'   3A 00D7   LDA     GAPLEN ;GET GAP LENGTH
0131'   CD 01B3'   CALL   DATOUT  ;OUTPUT GAP LENGTH
0134'   F1         POP    PSW     ;RESTORE SECTOR SIZE
0135'   B7         ORA     A        ;SECTOR SIZE=0?
0136'   3E80     MVI    A,128  ;PRESET DTL=128
0138'   2802     JRZ    ..NO   ;IF SECTOR SIZE=0, CONTINUE
013A'   3EFF     MVI    A,OFFH ;ELSE, DTL=OFFH
013C'   CD 01B3'   ..NO:   CALL   DATOUT  ;OUTPUT DTL
013F'   CD 0170'   CALL   WTINT   ;WAIT FOR INTERRUPT
0142'   C8         RZ          ;IF NO ERRORS, DONE
0143'   CD 0165'   ..ERR:  CALL   RECAL   ;RECALIBRATE DRIVE
0146'   2007     JRNZ   ..X    ;IF ERRORS, CONTINUE
0148'   21 00C8   LXI    H,TRYC ;ELSE, GET TRY COUNT

```

BPD401 - TURBODOS OPERATING SYSTEM BOOT PROM DRIVER FOR IMS 401
 COPYRIGHT (C) 1981, SOFTWARE 2000, INC.

```

014B'   35           DCR     M           ;DECREMENT TRY COUNT
014C'   C2 00A9'   JNZ     ..RR          ;IF COUNT NOT EXH, TRY AGAIN
014F'   3EFF       MVI     A,OFFH      ;ELSE, SET RETURN CODE=OFFH
0151'   C9         RET                    ;DONE

;
0152'   3E0F       ;SEEK: MVI     A,FDCKSK ;SET FDC SEEK COMMAND
0154'   CD 01B3'   CALL    DATOUT  ;OUTPUT FDC SEEK COMMAND
0157'   3A 00C0   LDA     IODSK   ;GET DISK NUMBER
015A'   CD 01B3'   CALL    DATOUT  ;OUTPUT DISK NUMBER
015D'   3A 00C1   LDA     IOTRK   ;GET TRACK NUMBER
0160'   CD 01B3'   CALL    DATOUT  ;OUTPUT TRACK NUMBER
0163'   180B      JMPR   WTINT   ;WAIT FOR INTERRUPT

;
0165'   3E07       ;RECAL: MVI     A,FDRCRL ;SET FDC RECALIBRATE COMMAND
0167'   CD 01B3'   CALL    DATOUT  ;OUTPUT FDC RECALIBRATE COMMAND
016A'   3A 00C0   LDA     IODSK   ;GET DISK NUMBER
016D'   CD 01B3'   CALL    DATOUT  ;OUTPUT DISK NUMBER

;
0170'   DB8C       WTINT: IN     DSKCTL  ;GET DISK CONTROLLER STATUS
0172'   0F         RRC                    ;TEST FOR FDC INTERRUPT
0173'   30FB       JRNC   WTINT   ;IF NO INTERRUPT, WAIT
0175'   DB8E       ..RQML: IN     FDCST   ;GET FDC STATUS
0177'   07         RLC                    ;FDC READY TO COMMUNICATE?
0178'   30FB       JRNC   ..RQML  ;IF NOT, WAIT
017A'   07         RLC                    ;TEST FDC DIRECTION
017B'   3818       JRC     ..RW     ;IF FDC OUTPUT AVAILABLE, PROCESS
017D'   3E08       MVI     A,FDCKSK ;GET SENSE INTERRUPT STATUS CMD
017F'   D38F       OUT    FDCDAT  ;OUTPUT BYTE TO FDC DATA REGISTER
0181'   CD 01A7'   CALL    DATAIN  ;GET STATUS REGISTER 0
0184'   4F         MOV     C,A         ;SAVE IT IN C-REG
0185'   E6C0       ANI     0COH     ;EXTRACT COMPLETION STATUS
0187'   FE80       CPI     80H     ;INTERRUPT STACK EMPTY?
0189'   2818       JRZ     ..X       ;IF SO, DONE
018B'   CD 01A7'   CALL    DATAIN  ;GET PRESENT CYLINDER NUMBER
018E'   CB69       BIT     STOSE,C   ;READY LINE CHANGE STATE?
0190'   28E3       JRZ     ..RQML  ;IF SO, IGNORE
0192'   51         MOV     D,C       ;GET STATUS REGISTER 0 IN D-REG
0193'   18E0       JMPR   ..RQML  ;FLUSH ANY REMAINING INTERRUPTS
0195'   CD 01A7'   ..RW: CALL    DATAIN  ;GET STATUS REGISTER 0
0198'   57         MOV     D,A       ;TO D-REG
0199'   0606       MVI     B,6       ;B=LENGTH OF REMAINING RESULT PHASE
019B'   CD 01A7'   ..RL: CALL    DATAIN  ;GET RESULT BYTE FROM FDC
019E'   10FB       DJNZ   ..RL     ;READ ALL SEVEN BYTES
01A0'   47         MOV     B,A       ;SAVE SECTOR SIZE IN B-REG
01A1'   18D2       JMPR   ..RQML  ;FLUSH ANY REMAINING INTERRUPTS
01A3'   7A         ..X: MOV     A,D     ;GET STATUS REGISTER 0
01A4'   E6C0       ANI     0COH     ;EXTRACT COMPLETION STATUS
01A6'   C9         RET                    ;DONE

;
01A7'   DB8E       DATAIN: IN     FDCST  ;GET FDC STATUS
01A9'   07         RLC                    ;TEST FDC FOR READY
01AA'   30FB       JRNC   DATAIN  ;IF NOT READY, WAIT
01AC'   07         RLC                    ;TEST FDC DIRECTION
01AD'   D2 0000:06 JNC     .BEG.#  ;IF WRONG DIRECTION, CONTINUE

```

BPD401 - TURBODOS OPERATING SYSTEM BOOT PROM DRIVER FOR IMS 401
 COPYRIGHT (C) 1981, SOFTWARE 2000, INC.

```

01B0'  DB8F          IN      FDCDAT  ;GET FDC DATA BYTE
01B2'  C9           RET      ;DONE

;
01B3'  4F          ;DATOUT: MOV   C,A    ;SAVE OUTPUT BYTE
01B4'  DB8E       ;..RW:  IN    FDCST  ;GET FDC STATUS
01B6'  07         ;       RLC    ;TEST FDC FOR READY
01B7'  30FB      ;       JRNC  ;..RW  ;IF NOT READY, WAIT
01B9'  07         ;       RLC    ;TEST FDC DIRECTION
01BA'  DA 0000:06 ;       JC   .BEG.# ;IF WRONG DIRECTION, CONTINUE
01BD'  79         ;       MOV   A,C    ;RESTORE OUTPUT BUTE
01BE'  D38F      ;       OUT   FDCDAT ;OUTPUT BYTE TO FDC DATA REGISTER
01C0'  C9         ;       RET      ;DONE

;
01C1'  AF          ;XFER:: XRA   A
01C2'  32 0080    ;       STA  TBUF  ;MAKE DEFAULT BUFFER EMPTY
01C5'  C3 0100    ;       JMP  TPA   ;TRANSFER TO O/S LOADER

;
.END

```

REPORT - TELETYPE OPERATING SYSTEM BOOT FROM DRIVE FOR THE 1040
SYSTEM (S) 1040, SOFTWARE 1000, INC.

ADDRESS	DATA	DESCRIPTION	ADDRESS	DATA	DESCRIPTION
0000	0000	START OF PROGRAM	0000	0000	START OF PROGRAM
0001	0001	INITIALIZE REGISTERS	0001	0001	INITIALIZE REGISTERS
0002	0002	LOAD OPERATING SYSTEM	0002	0002	LOAD OPERATING SYSTEM
0003	0003	TEST FOR READY	0003	0003	TEST FOR READY
0004	0004	IF NOT READY, WAIT	0004	0004	IF NOT READY, WAIT
0005	0005	TEST FOR INTERRUPT	0005	0005	TEST FOR INTERRUPT
0006	0006	IF INTERRUPT, STOP	0006	0006	IF INTERRUPT, STOP
0007	0007	RESTORE OUTPUT UNIT	0007	0007	RESTORE OUTPUT UNIT
0008	0008	OUTPUT DATA TO REGISTERS	0008	0008	OUTPUT DATA TO REGISTERS
0009	0009	DONE	0009	0009	DONE
0010	0010	TEST FOR INTERRUPT	0010	0010	TEST FOR INTERRUPT
0011	0011	IF INTERRUPT, STOP	0011	0011	IF INTERRUPT, STOP
0012	0012	RESTORE OUTPUT UNIT	0012	0012	RESTORE OUTPUT UNIT
0013	0013	OUTPUT DATA TO REGISTERS	0013	0013	OUTPUT DATA TO REGISTERS
0014	0014	DONE	0014	0014	DONE

END

Software 2000 inc.

P.O. Box 845 • Los Alamitos, California 90720 • 213/429-2317

acilmois 5000 inc.

1111 1111 1111 1111 1111 1111 1111 1111 1111 1111

