

## Parts Installation

4. Place the board WITHOUT IC's into an empty system bus slot and power up. In case of smoke, power down immediately and investigate.

5. With a VOM or scope, check the regulators for +5V (both of the 7805's), +12V, and -12V. The bottom pin of all four regulators is the output. Check for Vcc and ground on all IC's. Check for +12V on the 1791/8866 controller, the 2941 baud rate generator, and the 1458/4558 op amp. Check for -12V on the 1602 UART and the 1458/4558 op amp. Finally, check for -5V on the 2941 baud rate generator. If everything is OK, power down and proceed to the next step.

## IC INSERTION

If an IC insertion tool is not available, IC leads should be straightened a ROW at a time, not by the individual PIN. The edge of a straight sided table is an excellent device for this operation. Hold the IC by the plastic case, place one row of legs against a flat surface and push very slightly. Repeat with the opposite row. Continue this procedure until the legs of the IC can be inserted with minimum effort into its socket.

When inserting an IC into its socket, take care that you DO NOT BEND THE IC'S LEGS UNDERNEATH ITS PLASTIC PACK. This is an extremely common error and can escape even a fairly careful visual inspection.

If IC pins become bent under during insertion, use a long nose pliers to straighten them and try again. When removing an IC from its socket, use an IC remover, an IC test clip (another must for any electronics shop) or a miniature screw driver. DO NOT ATTEMPT TO REMOVE AN IC WITH YOUR FINGERS. You will bleed on severely bent pins.

Once all IC's have been inserted, re-check for bent pins. Then check twice for proper orientation. Upside down IC's are generally destroyed upon power up.

<p>IF FOR ANY REASON IT BECOMES NECESSARY TO REMOVE A COMPONENT WHICH HAS BEEN SOLDERED TO THE CIRCUIT BOARD, CLIP ALL LEADS BEFORE REMOVING. THIS WILL REDUCE THE CHANCE OF LIFTING PADS OFF TRACES.</p>
---

## Parts Installation

### POWER UP

If all previous checks have been performed, you are ready to put power to your fully populated board. In an empty system with power off, insert the Disk Jockey and power up. If the board smokes, power down and investigate. If not, measure the regulated voltages again.

If any voltages have been lost since powering up the bare board, power down and check for upside down IC's. Isolate the possible faulty chip or chips by powering down, removing a section of IC's, and powering up again. Continue this sequence until the faulty IC or IC's are found.

BE SURE NEVER TO INSERT OR REMOVE A BOARD WITH POWER ON! THIS MAY DAMAGE THE BOARD
---

This completes the initial check-out of your Disk Jockey. If there are any problems or questions regarding the operation of your Disk Jockey contact the service department of Morrow Designs, (415) 524-2104.

DJ/2D MODEL B MEMORY MAP

HEX ADDRESS	FUNCTION		OCTAL ADDRESS
E000-E3F7	ROM FIRMWARE		340:000-343:367
I/O REGISTERS			
	WHEN READ	WHEN WRITTEN	
E3F8	UART INVERTED DATA INPUT	UART INVERTED DATA OUTPUT	343:370
E3F9	UART INVERTED STATUS	DISK JOCKEY FUNCTION	343:371
E3FA	DISK JOCKEY STATUS	DRIVE CONTROL REGISTER	343:372
E3FB	NOT USED	NOT USED	343:373
E3FC	1791 CONTROLLER STATUS	1791 CONTROLLER COMMAND	343:374
E3FD	1791 TRACK REGISTER		343:375
E3FE	1791 SECTOR REGISTER		343:376
E3FF	1791 DATA REGISTER		343:377
E400-E7FF	RAM		344:000-347:377

## SOFTWARE LISTINGS

Morrow Designs makes no representations or warranties with respect to the contents hereof and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Futher, Morrow Designs reserves the right to revise this publication and to make changes from time to time in the content hereof without obligation of Morrow Designs to notify any person of such revision or changes.

```

*****
*
* Boot loader program for cp/m. The following code is
* loaded by the boot program on the Disk Jockey 2D. The
* 2D loads sector one of track zero into memory at
* ORIGIN+300H (the last page of ram on the controller)
* then jumps there. It is the responsibility of this code
* to load in the rest of cp/m.
*
*****

```

```

2900 = CPMORG EQU 2900H ;CPM STARTING ADDRESS
E000 = ORIGIN EQU 0E000H ;Disk Jockey starting address
E400 = RAM EQU ORIGIN+400H ;ram starting address (of 2D)
E640 = STACK EQU RAM+240H ;stack pointer starting address within ram
E009 = TKZERO EQU ORIGIN+110 ;track zero seek entry point
E00C = TRKSET EQU ORIGIN+140 ;entry for track seek
E00F = SETSEC EQU ORIGIN+170 ;entry point for sector set
E012 = SETDMA EQU ORIGIN+220 ;entry address for read/write beginning address
E015 = DREAD EQU ORIGIN+250 ;disk read entry point
E018 = DWRITE EQU ORIGIN+300 ;disk write routine address
E024 = DMAST EQU ORIGIN+440 ;disk read/write status routine

E700 ORG ORIGIN+700H

```

```

*****
*
* load: load in all the rest of cp/m and the cbios. There
* are only two ways to exit this code: 1) If an
* error occurs, a jump is made to the loader on the
* Disk Jockey 2D. 2) If everything works, a jump is
* made to the starting location of the cold boot in
* the cbios.
*
*****

```

```

E700 21003E LOAD LXI H,CPMORG+1500H ;starting location for cbios
E703 3140E6 LXI SP,STACK ;initialize the stack
E706 E5 PUSH H ;save jump address for return later
E707 01022E STADDR LXI B,2E02H ;reg B=sector count, reg C=starting sector
E70A C5 PUSH B ;save sector and count
E70B CD0FE0 CALL SETSEC ;set the sector to read
E70E CD09E0 CALL TKZERO ;home the drive
E711 210029 LXI H,CPMORG ;starting location for load
E714 44 LDLOOP MOV B,H ;put starting address in B&C
E715 4D MOV C,L
E716 CD12E0 CALL SETDMA ;set up starting load address
E719 060A MVI B,10 ;retry counter
E71B C5 RDLOOP PUSH B ;save retry count
E71C CD15E0 CALL DREAD ;read in the sector
E71F C1 POP B ;fetch retry count
E720 D22AE7 JNC RDGOOD ;take jump if read is ok.
E723 05 DCR B ;update retry counter
E724 C21BE7 JNZ RDLOOP ;try again if not ten errors
E727 C300E0 EXIT JMP ORIGIN ;start all over from the beginning
E72A C1 RDGOOD POP B ;refetch sector count and #
E72B 05 DCR B ;update the count
E72C C8 RZ ;GO TO CPM IF DONE
E72D 0C INR C ;COMPUTE NEW SECTOR (MOD 26)
E72E 3E1B MVI A,27 ;test if over 26
E730 B9 CMP C
E731 C236E7 JNZ OK ;take jump if sector < 27
E734 0E01 MVI C,1 ;start with sector 1 of next track
E736 C5 OK PUSH B ;save count and sector
E737 CC0CE0 CZ TRKSET ;conditionally set new track
E73A C1 POP B ;restore count and sector #
E73B C5 PUSH B ;save it again
E73C CD0FE0 CALL SETSEC ;set new sector
E73F CD24E0 CALL DMAST ;get load address
E742 218000 LXI H,2000 ;update te load address
E745 09 DAD B
E746 C314E7 JMP LDLOOP ;read next sector

```

```

*****
*
* save: write all of cpm and the cbios onto the disk.
* If an error occurs, the status returned by the
* 2D controller will be in location STACK-1.
*
*****

```

```

E749 2118E0 SAVE LXI H,DWRITE ;change load to write instead of read
E74C 221DE7 SHLD RDLOOP+2
E74F 215EE7 LXI H,ERROR ;change error return address
E752 2228E7 SHLD EXIT+1
E755 215BE7 LXI H,STALL ;get return address
E758 C303E7 JMP LOAD+3 ;go and do the write
E75B C35BE7 STALL JMP STALL ;stop here if everything ok !
E75E F5 ERROR PUSH PSW ;save status and flags
E75F C35FE7 ERROR1 JMP ERROR1 ;stop here on error.

```

```

*****
*
* intlz: write this cold boot loader program out to the
* disk.
*
*****

```

```

E762 3140E6 INTLZ LXI SP,STACK ;set up stack
E765 CD09E0 CALL TKZERO ;home the drive
E768 0100E7 LXI B,RAM+300H ;get starting address of this program
E76B CD12E0 CALL SETDMA ;set the write address
E76E 0E01 MVI C,1 ;set the sector to write
E770 CD0FE0 CALL SETSEC
E773 CD18E0 CALL DWRITE ;write this program out
E776 DA5EE7 JC ERROR
E779 C379E7 DONE JMP DONE ;stop here

```

```

*****
*
* CBIOS DRIVERS FOR CPM
*
* Currently the cbios is set up for a 16K cpm, to make a
* larger system, change the value of CPM.
*
*****

```

```

2900 = CPM EQU 2900H ;cp/m beginning load address
3106 = ENTRY EQU CPM+806H ;cp/m entrance point
0004 = CDISK EQU 4 ;current disk storage location
0003 = IOBYTE EQU 3H ;iobyte storage location

```

```

*****
*
* Iobyte allows selection of different I/O devices. It
* can be initialized in any way by changing the equate
* below.
* Initial iobyte is currently defined as :
* console = tty
* reader = tty
* punch = tty
* list = tty
*
*****

```

```

0000 = INTIOBY EQU 0 ;initial iobyte,

```

```

*****
*
* The following equates reference the disk jockey/2d
* controller board. If your controller is non-standard
* then all the equates can be changed by re-assigning the
* value of ORIGIN to be the starting address of your
* controller.
*
*****

```

```

E000 = ORIGIN EQU 0E000H ;disk jockey/2d beginning address
E003 = INPUT EQU ORIGIN+3 ;serial input routine
E006 = OUTPUT EQU ORIGIN+6 ;serial output routine
E009 = TKZERO EQU ORIGIN+9H ;track zero seek routine
E00C = SEEK EQU ORIGIN+0CH ;regular track seek routine
E00F = SECTOR EQU ORIGIN+0FH ;set sector routine
E012 = DMA EQU ORIGIN+12H ;read/write beginning address set
E015 = DISKR EQU ORIGIN+15H ;disk read routine
E018 = DISKW EQU ORIGIN+18H ;disk write routine
E01B = SELECT EQU ORIGIN+1BH ;disk selection routine
E021 = TSTAT EQU ORIGIN+21H ;serial device status routine
E640 = STACK EQU ORIGIN+640H ;disk jockey/2d ram area for boot only
0099 = SEKERR EQU 99H ;seek error bit mask
00FF = RWERR EQU 0FFH ;read/write error bit mask
000D = ACR EQU 0DH ;carriage return
000A = ALF EQU 0AH ;line feed
E006 = COTTY EQU OUTPUT ;default character output
E003 = CITTY EQU INPUT ;default character input

```

```

*****
*
* The jump table immediately below must not be altered.
* It is ok to make the jumps to other address, but the
* function performed must be the same.
*
*****

```

```

3E00          ORG      CPM+1500H

3E00 C32D3E   START   JMP      BOOT      ;cold boot
3E03 C3603E           JMP      WBOOT    ;warm boot
3E06 C3C03E           JMP      CONST    ;console status
3E09 C3CC3E           JMP      CONIN     ;console input
3E0C C3DE3E   CPOUT  JMP      CONOUT   ;console output
3E0F C3F93E           JMP      LIST      ;list output
3E12 C3EE3E           JMP      PUNCH     ;punch output
3E15 C3E43E           JMP      READER    ;reader input
3E18 C3713E           JMP      HOME      ;track zero home
3E1B C31BE0           JMP      SELECT    ;disk selection
3E1E C39B3E           JMP      SETTRK   ;track seek
3E21 C30FE0           JMP      SECTOR    ;sector select
3E24 C312E0           JMP      DMA       ;read/write address select
3E27 C3A13E           JMP      READ      ;disk read
3E2A C3BA3E           JMP      WRITE     ;disk write

```

```

*****
*
* boot: load in all of cpm and then
*       jump there. Initialize iobyte.
*
*****

```

```

3E2D 3140E6  BOOT   LXI   SP,STACK      ;initial stack
3E30 3E00    MVI   A,INTIOBY      ;initialize iobyte
3E32 320300  STA   IOBYTE
3E35 21643F  LXI   H,PROMPT      ;print signon message
3E38 CD0E3E  CALL  MESSG
3E3B AF      XRA   A              ;select disk A
3E3C 320400  STA   CDISK
3E3F 018000  GOCPM  LXI   B,80H     ;set up default disk buffer
3E42 CD12E0  CALL  DMA
3E45 3EC3    MVI   A,0C3H       ;put jump instruction to warm boot at 0
3E47 320000  STA   0
3E4A 21033E  LXI   H,START+3
3E4D 220100  SHLD  1
3E50 320500  STA   5              ;put jump to cpm entry at 5
3E53 210631  LXI   H,ENTRY
3E56 220600  SHLD  6
3E59 3A0400  LDA   CDISK         ;jump to cpm with current disk in C
3E5C 4F      MOV   C,A
3E5D C30029  JMP   CPM

```

```

*****
*
* warm boot: load in all of cpm except the cbios. Then
* enter cpm.
*
*****

```

```

3E60 3140E6  WBOOT  LXI   SP,STACK      ;initialize the stack
3E63 AF      XRA   A              ;select drive A
3E64 4F      MOV   C,A
3E65 CD1BE0  CALL  SELECT
3E68 01022A  LXI   B,2A02H      ;sector count and beginning sector
3E6B CD0AE7  CALL  ORIGIN+70AH   ;call the cold start loader
3E6E C33F3E  JMP   GOCPM         ;now enter cpm

```

```

*****
*
* Home: move the head to track zero.
*
*****

```

```

3E71 CD09E0  HOME   CALL  TKZERO    ;call the disk jockey/2d
3E74 0E99    SEEK1  MVI   C,SEKERR   ;non relevent error mask

```

```

*****
*
* doerrs: returns if no error. Otherwise prints an appro-
* priate error messgae, and returns to cpm with an error
* indication.
*
*****

```

```

3E76 DA7B3E  DOERRS JC      DOERR1    ;test if error
3E79 AF      RWOK   XRA   A          ;return if ok
3E7A C9      RET
3E7B A1      DOERR1 ANA   C          ;strip off unwanted errors
3E7C 0E08    MVI   C,8           ;error counter
3E7E 217A3F  LXI   H,MSGTBL      ;beginning address of messages
3E81 5E      DOLOOP MOV   E,M'         ;get error address in D&E
3E82 23      INX   H
3E83 56      MOV   D,M
3E84 23      INX   H
3E85 1F      RAR
3E86 DA8D3E  JC      MESSGA       ;check if this bit is the error
3E89 0D      DCR   C             ;yes, exit after printig error
3E8A F2813E  JP      DOLOOP       ;no error, update the count down
                        ;continue if not found

```

```

*
* if fall through then unknown error
*

```

```

3E8D EB      MESSGA XCHG          ;put message address into H&L

```



```

*****
*
* messg: print the message pointed to by H&L and termin-
* ated by a 0FFH byte.
*
*****

3E8E 7E      MESSG  MOV    A,M      ;get character
3E8F A7      ANA    A          ;test for end
3E90 F8      RM
3E91 E5      PUSH   H          ;save address
3E92 4F      MOV    C,A        ;prep for console output
3E93 CD0C3E  CALL   CPOUT      ;output it
3E96 E1      POP    H          ;restore pointer
3E97 23      INX    H          ;bump to next character
3E98 C38E3E  JMP    MESSG      ;continue until end

*****
*
* settrk: call the disk jockey/2d to seek then exit by
* testing for errors.
*
*****

3E9B CD0CE0  SETTRK CALL   SEEK
3E9E C3743E  JMP    SEEK1

*****
*
* read: read one sector from the disk. Try ten times on
* errors, before returning an error condition.
*
*****

3EA1 2115E0  READ   LXI    H,DISKR ;put disk read address into repeat loop
3EA4 22AB3E  RDWR  SHLD   RW+1
3EA7 060A    MVI   B,10      ;retry counter
3EA9 C5      RDWRL PUSH   B
3EAA CD0000  RW    CALL   0      ;actually call disk read/write
3EAD C1      POP   B
3EAE D2793E  JNC   RWOK      ;exit if succesful
3EB1 05      DCR   B          ;test error count
3EB2 C2A93E  JNZ   RDWRL     ;continue if not zero
3EB5 0EFF    MVI   C,RWERR  ;read/write error bit mask
3EB7 C3763E  JMP   DOERRS   ;print the appropriate error message

*****
*
* write: write data onto the disk, also try ten times
* before reporting an error.
*
*****

3EBA 2118E0  WRITE  LXI    H,DISKW
3EBD C3A43E  JMP   RDWR

*****
*
* const: get the status for the currently assigned console
* device. The console device can be gotten from
* iobyte, then a jump to the correct console status
* routine is performed.
*
*****

3EC0 212C3F  CONST  LXI    H,CSTBLE ;beginning of jump table
3EC3 C3CF3E  JMP   CONIN1 ;select correct jump

*****
*
* csreader: if the console is assigned to the reader then
* a jump will be made here, where another jump
* will occur to the correct reader status.
*
*****

3EC6 21343F  CSREADR LXI   H,CSRTBLE ;beginning of reader status table
3EC9 C3E73E  JMP   READERA

```

```

*****
*
* conin: take the correct jump for the console input
* routine. The jump is based on the two least sig-
* nificant bits of iobyte.
*
*****

3ECC 21043F  CONIN  LXI      H,CITBLE      ;beginning of character input table
*
* entry at conin1 will decode the two least significant bits
* of iobyte. This is used by conin,conout, and const.
*

3ECF 3A0300  CONIN1 LDA      IOBYTE
3ED2 17      RAL

*
* entry at seldev will form an offset into the table pointed
* to by H&L and then pick up the address and jump there.
*

3ED3 E606    SELDEV ANI      6H      ;strip off unwanted bits
3ED5 1600    MVI      D,0      ;form offset
3ED7 5F      MOV      E,A
3ED8 19      DAD      D      ;add offset
3ED9 7E      MOV      A,M      ;pick up high byte
3EDA 23      INX      H
3EDB 66      MOV      H,M      ;pick up low byte
3EDC 6F      MOV      L,A      ;form address
3EDD E9      PCHL     ;go there !

*****
*
* conout: take the proper branch address based on the two
* least significant bits of iobyte.
*
*****

3EDE 210C3F  CONOUT LXI      H,COTBLE      ;beginning of the character out table
3EE1 C3CF3E  JMP      CONIN1 ;do the decode

*****
*
* reader: select the correct reader device for input. The
* reader is selected from bits 2 and 3 of iobyte.
*
*****

3EE4 21243F  READER LXI      H,RTBLE ;beginning of reader input table
*
* entry at readera will decode bits 2 & 3 of iobyte, used
* by csreader.
*

3EE7 3A0300  READERA LDA     IOBYTE
*
* entry at reader1 will shift the bits into position, used
* by list and punch.
*

3EEA 1F      READR1 RAR
3EEB C3D33E  JMP      SELDEV

*****
*
* punch: select the correct punch device. The section
* comes from bits 4&5 of iobyte.
*
*****

3EEE 211C3F  PUNCH  LXI      H,PTBLE ;beginning of punch table
3EF1 3A0300  LDA      IOBYTE
*
* entry at pnchl rotates bits a little more in prep for
* seldev, used by list.
*

3EF4 1F      PNCH1  RAR
3EF5 1F      RAR
3EF6 C3EA3E  JMP      READR1

```

```

*****
*
* list: select a list device based on bits 6&7 of iobyte
*
*****
3EF9 21143F LIST LXI H,LTBLE ;beginning of the list device routines
3EFC 3A0300 LDA IOBYTE
3EFF 1F RAR
3F00 1F RAR
3F01 C3F43E JMP PNCH1

*****
*
* If customizing I/O routines is being performed, the
* table below should be modified to reflect the changes.
* all I/O devices are decoded out of iobyte and the jump
* is taken from the following tables.
*
*****

*
* console input table
*
3F04 03E0 CITBLE DW CTTY ;input from tty (currently assigned by intioby,input from 2d)
3F06 473F DW CICRT ;input from crt (currently SWITCHBOARD serial port 1)
3F08 E43E DW READER ;input from reader (depends on reader selection)
3F0A 473F DW CIUC1 ;input from user console 1 (currently SWITCHBOARD serial port 1)

*
* console output table
*
3F0C 06E0 COTBLE DW CTTY ;output to tty (currently assigned by intioby,output to 2d)
3F0E 3C3F DW COCRT ;output to crt (currently SWITCHBOARD serial port 1)
3F10 F93E DW LIST ;output to list device (depends on bits 6&7 of iobyte)
3F12 3C3F DW COUC1 ;output to user console 1 (currently SWITCHBOARD serial port 1)

*
* list device table
*
3F14 06E0 LTBLE DW CTTY ;output to tty (currently assigned by intioby,output to 2d)
3F16 3C3F DW COCRT ;output to crt (currently SWITCHBOARD serial port 1)
3F18 3C3F DW COLPT ;output to line printer (currently SWITCHBOARD serial port 1)
3F1A 3C3F DW COUL1 ;output to user line printer 1 (currently SWITCHBOARD serial port 1)

*
* punch device table
*
3F1C 06E0 PTBLE DW CTTY ;output to the tty (currently assigned by intioby,output to 2d)
3F1E 3C3F DW COPTP ;output to paper tape punch (currently SWITCHBOARD serial port 1)
3F20 3C3F DW COUPL ;output to user punch 1 (currently SWITCHBOARD serial port 1)
3F22 3C3F DW COUP2 ;output to user punch 2 (currntly SWITCHBOARD serial port 1)

*
* reader device input table
*
3F24 03E0 RTBLE DW CTTY ;input from tty (currently assigned by intioby, input from 2d)
3F26 473F DW CIPTR ;input from paper tape reader (currently SWITCHBOARD serial port 1)
3F28 473F DW CIUR1 ;input from user reader 1 (currently SWITCHBOARD serial port 1)
3F2A 473F DW CIUR2 ;input from user reader 2 (currently SWITCHBOARD serial port 1)

*
* console status table
*
3F2C 533F CSTBLE DW CSTTY ;status of tty (currently assigned by intioby, ststus from 2d)
3F2E 5B3F DW CSCRT ;status from crt (currently SWITCHBOARD serial port 1)
3F30 C63E DW CSREADR ;status from reader (depends on reader device)
3F32 5B3F DW CSUC1 ;status from user console 1 (currently SWITCHBOARD serial port 1)

*
* status fromreader device
*
3F34 533F CSRTBLE DW CSTTY ;status from tty (currently assigned by intioby, status of 2d)
3F36 5B3F DW CSPTR ;status from paper tape reader (currently SWITCHBOARD serial port 1)
3F38 5B3F DW CSUR1 ;status from user reader 1 (currently SWITCHBOARD serial port 1)
3F3A 5B3F DW CSUR2 ;status of user reader 2 (currently SWITCHBOARD serial port 1)

```

```

*****
*
* The following equates set output device to output to
* the SWITCHBOARD serial port 1.
*
*****

```

```

3F3C = COCRT EQU $ ;output from crt
3F3C = COUC1 EQU $ ;output from user console 1
3F3C = COUL1 EQU $ ;output from user line printer 1
3F3C = COPTP EQU $ ;output from paper tape punch
3F3C = COUP1 EQU $ ;output from user punch 1
3F3C = COUP2 EQU $ ;output from user punch 2
3F3C DB02 COLPT IN 2 ;output from line printer, get status
3F3E E680 ANI 80H ;wait until ok to send
3F40 CA3C3F JZ COLPT
3F43 79 MOV A,C ;output the character
3F44 D301 OUT 1
3F46 C9 RET

```

```

*****
*
* The following equates set the input from the devices to
* come from the SWITCHBOARD serial port 1
*
*****

```

```

3F47 = CIUC1 EQU $ ;input from user console 1
3F47 = CICRT EQU $ ;input from crt
3F47 = CIUR1 EQU $ ;input from user reader 1
3F47 = CIUR2 EQU $ ;input from user reader 2
3F47 DB02 CIPTR IN 2 ;input from paper tape reader, get status
3F49 E640 ANI 40H ;wait for character
3F4B CA473F JZ CIPTR
3F4E DB01 IN 1
3F50 E67F ANI 7FH ;strip off the parity
3F52 C9 RET

```

```

*****
*
* console status routines, test if a character has arrived
*
*****

```

```

3F53 CD21E0 CTTY CALL TSTAT ;status from disk jockey 2d
3F56 3E00 STAT MVI A,0 ;prep for zero return
3F58 C0 RNZ ;nothing found
3F59 3D DCR A ;return with 0FFH
3F5A C9 RET

```

```

*****
*
* The following equates cause the devices to get status
* from the SWITCHBOARD serial port 1.
*
*****

```

```

3F5B = CSUR1 EQU $ ;status of user reader 1
3F5B = CSUR2 EQU $ ;status of user reader 2
3F5B = CSPTR EQU $ ;status of paper tape reader
3F5B = CSUC1 EQU $ ;status of user console 1
3F5B DB02 CSCRT IN 2 ;status from crt, get status
3F5D E640 ANI 40H ;strip of data ready bit
3F5F EE40 XRI 40H ;make correct polarity
3F61 C3563F JMP STAT ;return proper indication

```

```

*****
*
* The following messages could be put out by the cbios.
*
*****

```

```

3F64 0D0A PROMPT DB ACR,ALF ;prompt message - "16K CP/M VERS 1.4"
3F66 31364B20 DB '16K '
3F6A 43502F4D DB 'CP/M'
3F6E 20564552 DB ' VER'
3F72 5320312E DB 'S 1.'
3F76 34 DB '4'
3F77 0D0A DB ACR,ALF
3F79 FF DB 0FFH

```

\*  
\* error message table  
\*

3F7A 8C3F	MSGTBL	DW	ILLDATA ;illegal data
3F7C 983F		DW	DATAREQ ;data request
3F7E A33F		DW	DATALOS ;data lost
3F80 AF3F		DW	CRCERR ;crc error
3F82 8B3F		DW	ILLSEC ;illegal sector
3F84 CF3F		DW	ILLDMA ;illegal dma
3F86 DA3F		DW	WRITPRO ;write protected
3F88 E53F		DW	NOTRDY ;not ready
3F8A F13F		DW	UNKNOWN ;unknown error
3F8C 0D0A	ILLDATA	DB	ACR,ALF
3F8E 494C474C20		DB	'ILGL DATA'
3F97 FF		DB	0FFH
3F98 0D0A	DATAREQ	DB	ACR,ALF
3F9A 4441544120		DB	'DATA REQ'
3FA2 FF		DB	0FFH
3FA3 0D0A	DATALOS	DB	ACR,ALF
3FA5 4441544120		DB	'DATA LOST'
3FAE FF		DB	0FFH
3FAF 0D0A	CRCERR	DB	ACR,ALF
3FB1 4352432045		DB	'CRC ERROR'
3FBA FF		DB	0FFH
3FBB 0D0A	ILLSEC	DB	ACR,ALF
3FBD 494C474C20		DB	'ILGL SECTOR/TRACK'
3FCE FF		DB	0FFH
3FCF 0D0A	ILLDMA	DB	ACR,ALF
3FD1 494C474C20		DB	'ILGL DMA'
3FD9 FF		DB	0FFH
3FDA 0D0A	WRITPRO	DB	ACR,ALF
3FDC 5752542050		DB	'WRT PROT'
3FE4 FF		DB	0FFH
3FE5 0D0A	NOTRDY	DB	ACR,ALF
3FE7 4E4F542052		DB	'NOT READY'
3FF0 FF		DB	0FFH
3FF1 0D0A	UNKNOWN	DB	ACR,ALF
3FF3 554E4B4F57		DB	'UNKOWN ERROR'
3FFF FF		DB	0FFH

340:000		1	AORG	OE000H	
		2			
340:000	340:000	3	ORIGIN	EQU	340:000Q
		4			
340:000	343:336	5	BEGINS	EQU	ORIGIN+3:336Q
340:000	344:000	6	RAM	EQU	ORIGIN+4:000Q
340:000	343:370	7	IO	EQU	ORIGIN+3:370Q
340:000	343:370	8	UDATA	EQU	IO
340:000	343:371	9	DREG	EQU	IO+1
340:000	343:371	10	USTAT	EQU	IO+1
340:000	343:372	11	DCMD	EQU	IO+2
340:000	343:372	12	DSTAT	EQU	IO+2
340:000	343:373	13	CSTALL	EQU	IO+3
340:000	343:374	14	CMDREG	EQU	IO+4
340:000	343:374	15	CSTAT	EQU	IO+4
340:000	343:375	16	TRKREG	EQU	IO+5
340:000	343:376	17	SECREG	EQU	IO+6
340:000	343:377	18	DATREG	EQU	IO+7
		19			
340:000	000:001	20	LIGHT	EQU	1
340:000	000:001	21	HEAD	EQU	1
340:000	000:001	22	DENSITY	EQU	1
340:000	000:004	23	ISTAT	EQU	4
340:000	000:004	24	INTRQ	EQU	4
340:000	000:004	25	TZERO	EQU	4
340:000	000:004	26	LOAD	EQU	4
340:000	000:006	27	ULOAD	EQU	6
340:000	000:010	28	OSTAT	EQU	10Q
340:000	000:010	29	DSIDE	EQU	10Q
340:000	000:011	30	NOLITE	EQU	11Q
340:000	000:011	31	DCRINT	EQU	11Q
340:000	000:011	32	HCMD	EQU	11Q
340:000	000:020	33	INDEX	EQU	20Q
340:000	000:022	34	WINDXD	EQU	22Q
340:000	000:030	35	SKCMD	EQU	30Q
340:000	000:032	36	RINDXD	EQU	32Q
340:000	000:035	37	SVCMD	EQU	35Q
340:000	000:100	38	WPROT	EQU	100Q
340:000	000:100	39	ACCESS	EQU	100Q
340:000	000:200	40	RSTBIT	EQU	200Q
340:000	000:200	41	READY	EQU	200Q
340:000	000:210	42	RDCMD	EQU	210Q
340:000	000:250	43	WRCMD	EQU	250Q
340:000	000:300	44	STBITS	EQU	300Q
340:000	000:304	45	RACMD	EQU	304Q
340:000	000:320	46	CLRCMD	EQU	320Q
		47			
		48	*NP		

340:000	303	151	340	49	DBOOT	JMP	BOOT	
340:003	303	351	340	50	TERMIN	JMP	CIN	
340:006	303	332	340	51	TRMOUT	JMP	COUT	
340:011	303	132	341	52	TKZERO	JMP	HOME	
340:014	303	213	341	53	TRKSET	JMP	SEEK	
340:017	303	201	341	54	SETSEC	JMP	SECSET	
340:022	303	103	341	55	SETDMA	JMP	DMA	
340:025	303	335	341	56	DREAD	JMP	READ	
340:030	303	274	341	57	DWRITE	JMP	WRITE	
340:033	303	074	341	58	SELDRV	JMP	DRIVE	
340:036	303	370	340	59	TPANIC	JMP	CPAN	
340:041	303	003	341	60	TSTAT	JMP	TMSTAT	
340:044	303	064	341	61	DMAST	JMP	DMSTAT	
340:047	303	011	341	62	STATUS	JMP	DISKST	
340:052	303	305	340	63	DSKERR	JMP	LERROR	
340:055	303	263	343	64	SETDEN	JMP	DENFIX	
340:060	303	345	343	65	SETSID	JMP	SIDEX	
				66				
340:063	000	066		67		DS	66Q	
				68				
340:151				69	BOOT			
340:151	061	372	347	70	LXI	SP,TRACK+1	initialize SP	
340:154	315	322	343	71	CALL	TIMOUT	poc/reset timeout	
340:157	041	001	000	72	LXI	H,1		
340:162	345			73	PUSH	H	track 0, sector 1	
340:163	056	011		74	MVI	L,DCRINT	set up the	
340:165	345			75	PUSH	H	-side select	
340:166	046	377		76	MVI	H,377Q	-and initial	
340:170	345			77	PUSH	H	-drive	
340:171	345			78	PUSH	H	-parameters	
340:172	345			79	PUSH	H		
340:173	345			80	PUSH	H		
340:174	041	010	000	81	LXI	H,10Q	initialize	
340:177	345			82	PUSH	H	-tzflag & cdisk	
340:200	056	176		83	MVI	L,176Q	initialize	
340:202	345			84	PUSH	H	-disk & drvsel	
340:203	056	010		85	MVI	L,10Q	initialize	
340:205	345			86	PUSH	H	-hdflag & dsflag	
340:206	046	030		87	MVI	H,30Q	initialize	
340:210	345			88	PUSH	H	-timer constant	
340:211	076	177		89	MVI	A,177Q	start 1791	
340:213	062	371	343	90	STA	DREG		
340:216	076	320		91	MVI	A,CLRCMD	1791 reset	
340:220	062	374	343	92	STA	CMDREG		
340:223				93	LDHEAD			
340:223	257			94	XRA	A	load the head	
340:224	315	033	343	95	CALL	HDCHK	-and test for	
340:227	322	245	340	96	JNC	DOOROK	-drive ready	
340:232	076	001		97	MVI	A,LIGHT	turn on the	
340:234	062	366	347	98	STA	DCREG	-error LED	
340:237	315	322	343	99	CALL	TIMOUT	timeout to	
340:242	303	223	340	100	JMP	LDHEAD	-close drive door	
				101	*NP			

340:245			102	DOOROK		
340:245	076	011	103		MVI A,NOLITE	turn off the
340:247	062	366	347	104	STA DCREG	-error LED
340:252	315	226	343	105	CALL MEASUR	head load time
340:255	301			106	POP B	adjust the stack
340:256	001	000	347	107	LXI B,RAM+300H	DMA addr
340:261	305			108	PUSH B	initialize
340:262	325			109	PUSH D	-dmaadr & timer
340:263	041	000	000	110	LXI H,0	initialize
340:266	345			111	PUSH H	-error counts
340:267	000			112	NOP .	debug instruction
340:270	305			113	PUSH B	boot address
340:271	006	014		114	MVI B,12	number of retrys
340:273				115		
340:273	305			116		
340:274	315	335	341	117	PUSH B	save the retry no.
340:277	301			118	CALL READ	read boot sector
340:300	320			119	POP B	restor retry no.
340:301	005			120	RNC .	successful read?
340:302	302	273	340	121	DCR B	no! - count down
340:305				122	JNZ LDLOOP	-and try again
340:305	016	011		123		
340:307	021	303	242	124	MVI C,11Q	
340:312				125	LXI D,242:303Q	
340:312	033			126		
340:313	172			127	DCX D	
340:314	263			128	MOV A,D	
340:315	302	312	340	129	ORA E	
340:320	076	010		130	JNZ LLELOOP	
340:322	251			131	MVI A,10Q	blink
340:323	117			132	XRA C	-the LED at
340:324	062	372	343	133	MOV C,A	-top of the
340:327	303	307	340	134	STA DCMD	-circuit board
				135	JMP LERROR+2	
340:332				136		
340:332	072	371	343	137		
340:335	346	010		138	LDA USTAT	get UART status
340:337	302	332	340	139	ANI OSTAT	output ready mask
340:342	171			140	JNZ COUT	test buffer empty
340:343	057			141	MOV A,C	character data
340:344	062	370	343	142	CMA .	negative logic bus
340:347	057			143	STA UDATA	send data to UART
340:350	311			144	CMA .	make positive
				145	RET	
340:351				146		
340:351	072	371	343	147		
340:354	346	004		148	LDA USTAT	get UART status
340:356	302	351	340	149	ANI ISTAT	input ready mask
340:361	072	370	343	150	JNZ CIN	wait for input
340:364	057			151	LDA UDATA	get the character
340:365	346	177		152	CMA .	adjust for negative bus
340:367	311			153	ANI 177Q	trim to 7 bits
				154	RET	
340:370				155		
340:370	072	371	343	156		
340:373	346	004		157	LDA USTAT	get UART status
340:375	300			158	ANI ISTAT	input ready mask
340:376	315	351	340	159	RNZ .	test for data
341:001	271			160	CALL CIN	get character
341:002	311			161	CMP C	test for panic chtr
				162	RET	
					*NP	



341:003			163	TMSTAT		
341:003	072	371	343	164	LDA	USTAT get UART status
341:006	346	004		165	ANI	ISTAT input ready mask
341:010	311			166	RET	
			167			
341:011			168	DISKST		
341:011	041	375	343	169	LXI	H,TRKREG most recent
341:014	116			170	MOV	C,M -track to C
341:015	043			171	INX	H most recent
341:016	106			172	MOV	B,M -sector to B
341:017	072	366	347	173	LDA	DCREG get current
341:022	057			174	CMA	. -density in
341:023	346	001		175	ANI	1 -the msb
341:025	017			176	RRC	. -position
341:026	127			177	MOV	D,A save in D
341:027	072	367	347	178	LDA	SIDE put the
341:032	007			179	RLC	. -most recent
341:033	007			180	RLC	. -side select
341:034	007			181	RLC	. -in bit positin
341:035	262			182	ORA	D -6 and merge
341:036	127			183	MOV	D,A save in D
341:037	072	350	347	184	LDA	DSFLAG get the
341:042	356	010		185	XRI	DSIDE -most recent
341:044	027			186	RAL	. -double sided
341:045	027			187	RAL	. -status and place
341:046	202			188	ADD	D -in bit position
341:047	127			189	MOV	D,A -5 and merge
341:050	072	375	347	190	LDA	SECLEN get the
341:053	027			191	RAL	. -sector length
341:054	027			192	RAL	. -code bits in
341:055	262			193	ORA	D -positions 2 & 3
341:056	127			194	MOV	D,A -and merge
341:057	072	354	347	195	LDA	CDISK get the current
341:062	202			196	ADD	D -disk no. in bit
341:063	311			197	RET	. -positions 0 & 1
			198			
341:064			199	DMSTAT		
341:064	345			200	PUSH	H save the HL pair
341:065	052	346	347	201	LHLD	DMAADR move the
341:070	104			202	MOV	B,H -DMA address to
341:071	115			203	MOV	C,L -the BC pair
341:072	341			204	POP	H recover HL
341:073	311			205	RET	
			206			
341:074			207	DRIVE		
341:074	171			208	MOV	A,C drive select
341:075	346	003		209	ANI	3 -values must be
341:077	062	353	347	210	STA	DISK -between zero
341:102	311			211	RET	. -and three
			212	*NP		

341:103				213	DMA			
341:103	041	000	034	214		LXI	H,-RAM	test the
341:106	011			215		DAD	B	-DMA address
341:107	332	124	341	216		JC	DMASET	-for conflict
341:112	041	010	040	217		LXI	H,8-ORIGIN	
341:115	011			218		DAD	B	-with the I/O
341:116	322	124	341	219		JNC	DMASET	-on the DJ/2D
341:121	076	020		220		MVI	A,20Q	-controller
341:123	311			221		RET		
341:124				222	DMASET			
341:124	140			223		MOV	H,B	store the
341:125	151			224		MOV	L,C	-BC pair
341:126	042	346	347	225		SHLD	DMAADR	
341:131	311			226		RET		
				227				
341:132				228	HOME			
341:132	315	343	342	229		CALL	HDLOAD	load the head
341:135	330			230		RC	.	not ready error
341:136	315	160	341	231		CALL	HENTRY	move the head
341:141	365			232		PUSH	PSW	save status
341:142	237			233		SBB	A	update the
341:143	062	371	347	234		STA	TRACK	-track
341:146	062	375	343	235		STA	TRKREG	-registers
341:151	257			236		XRA	A	set the not
341:152	062	355	347	237		STA	TZFLAG	-verified flag
341:155	303	043	342	238		JMP	LEAVE+1	unload the head
341:160				239	HENTRY			
341:160	257			240		XRA	A	set the force
341:161	062	351	347	241		STA	HDFLAG	-verify flag
341:164	041	000	000	242		LXI	H,0	timeout constant
341:167	076	011		243		MVI	A,HCMD	move the head
341:171	315	142	343	244		CALL	CENTRY	to track 0
341:174	346	004		245		ANI	TZERO	track zero bit
341:176	300			246		RNZ		
341:177	067			247		STC	.	error flag
341:200	311			248		RET		
				249				
341:201				250	SECSET			
341:201	257			251		XRA	A	test for
341:202	261			252		ORA	C	-zero value
341:203	067			253		STC	.	error flag
341:204	310			254		RZ	.	error return
341:205	346	037		255		ANI	37Q	trim & clear cry
341:207	062	370	347	256		STA	SECTOR	
341:212	311			257		RET		
				258				
341:213				259	SEEK			
341:213	171			260		MOV	A,C	test for
341:214	376	115		261		CPI	77	-track
341:216	077			262		CMC	.	-too large
341:217	330			263		RC		
341:220	062	371	347	264		STA	TRACK	
341:223	311			265		RET		
				266	*NP			

341:224				267	ISSUE		
341:224	062	343	347	268		STA ECOUNT+1	update count
341:227	315	226	343	269		CALL MEASUR	find the index
341:232	016	001		270		MVI C,1	start w/sector 1
341:234				271	ISLOOP		
341:234	171			272		MOV A,C	initialize the
341:235	062	376	343	273		STA SECREG	-sector register
341:240	072	370	347	274		LDA SECTOR	test for
341:243	271			275		CMP C	-target sector
341:244	310			276		RZ	
341:245	076	210		277		MVI A,RDCMD	do a fake
341:247	315	135	343	278		CALL COMAND	-read command
341:252	332	040	342	279		JC PLEAVE	abort on error
341:255	014			280		INR C	increment sector no.
341:256	303	234	341	281		JMP ISLOOP	
				282			
341:261				283	COMNDP		
341:261	062	374	343	284		STA CMDREG	start the operation
341:264	110			285		MOV C,B	initialize block count
341:265	021	377	343	286		LXI D,DATREG	data register
341:270	052	346	347	287		LHLD DMAADR	transfer address
341:273	311			288		RET	
				289			
341:274				290	WRITE		
341:274	315	063	342	291		CALL PREP	prepare for write
341:277	332	042	342	292		JC LEAVE	abort operation
341:302				293	WREENTRY		
341:302	076	250		294		MVI A,WRCMD	start a write
341:304	315	261	341	295		CALL COMNDP	
341:307				296	WRLOOP		
341:307	176			297		MOV A,M	load 1st byte of data
341:310	043			298		INX H	advance pointer
341:311	022			299		STAX D	write 1st byte of data
341:312	176			300		MOV A,M	load 2nd byte of data
341:313	043			301		INX H	advance pointer
341:314	022			302		STAX D	write 2nd byte of data
341:315	176			303		MOV A,M	load 3rd byte of data
341:316	043			304		INX H	advance pointer
341:317	022			305		STAX D	write 3rd byte of data
341:320	015			306		DCR C	reduce block count
341:321	176			307		MOV A,M	load 4th byte of data
341:322	043			308		INX H	advance pointer
341:323	022			309		STAX D	write 4th byte of data
341:324	302	307	341	310		JNZ WRLOOP	write next 4 bytes
341:327	041	302	341	311		LXI H,WREENTRY	return entry addr
341:332	303	373	341	312		JMP CBUSY	
				313	*NP		

341:335				314	READ	
341:335	315	063	342	315	CALL PREP	prepare for read
341:340	332	042	342	316	JC LEAVE	abort operation
341:343				317	RDENTRY	
341:343	076	210		318	MVI A,RDCMD	start a read
341:345	315	261	341	319	CALL COMNDP	
341:350				320	RDLOOP	
341:350	032			321	LDAX D	read 1st byte
341:351	167			322	MOV M,A	store 1st byte
341:352	043			323	INX H	advance pointer
341:353	032			324	LDAX D	read 2nd byte
341:354	167			325	MOV M,A	store 2nd byte
341:355	043			326	INX H	advance pointer
341:356	032			327	LDAX D	read 3rd byte
341:357	167			328	MOV M,A	store 3rd byte
341:360	043			329	INX H	advance pointer
341:361	015			330	DCR C	reduce block count
341:362	032			331	LDAX D	read 4th byte
341:363	167			332	MOV M,A	store 4th byte
341:364	043			333	INX H	advance pointer
341:365	302	350	341	334	JNZ RDLOOP	read next 4 bytes
341:370	041	343	341	335	LXI H,RDENTRY	return entry addr
				336		
341:373				337	CBUSY	
341:373	345			338	PUSH H	save return
341:374	041	374	343	339	LXI H,CSTAT	wait for 1791
341:377	315	154	343	340	CALL BUSY	-to finish command
342:002	346	137		341	ANI 137Q	error bit mask
342:004	312	041	342	342	JZ LEAVE-1	no error
342:007	376	020		343	CPI 20Q	premature interrupt
342:011	302	040	342	344	JNZ PLEAVE	other error type
342:014	072	342	347	345	LDA ECOUNT	decrement error
342:017	075			346	DCR A	-count number 1
342:020	372	027	342	347	JM STEST	hard interrupt error
342:023	062	342	347	348	STA ECOUNT	update count
342:026	311			349	RET .	do operation over
342:027				350	STEST	
342:027	072	343	347	351	LDA ECOUNT+1	decrement error
342:032	075			352	DCR A	-count number 2
342:033	362	224	341	353	JP ISSUE	issue a command
342:036	076	020		354	MVI A,20Q	irrecoverable error!
342:040				355	PLEAVE	
342:040	067			356	STC .	error flag
342:041	341			357	POP H	adjust the stack
342:042				358	LEAVE	
342:042	365			359	PUSH PSW	save the status
342:043	072	366	347	360	LDA DCREG	control bits
342:046	356	004		361	XRI LOAD	toggle the
342:050	062	372	343	362	STA DCMD	-head load bit
342:053	072	352	347	363	LDA DRVSEL	enable access to
342:056	062	371	343	364	STA DREG	-the data register
342:061	361			365	POP PSW	recover the status
342:062	311			366	RET	
				367	*NP	

342:063				368	PREP		
342:063	315	343	342	369		CALL HDLOAD	load the head
342:066	330			370		RC .	test for drive ready
342:067	072	375	343	371		LDA TRKREG	get old track
342:072	074			372		INR A	test for head
342:073	314	160	341	373		CZ HENTRY	-not calibrated
342:076	330			374		RC .	seek error?
342:077	041	375	343	375		LXI H,TRKREG	old track
342:102	072	371	347	376		LDA TRACK	new track
342:105	276			377		CMP M	test for head motion
342:106	043			378		INX H	advance to the
342:107	043			379		INX H	-data register
342:110	167			380		MOV M,A	save new track
342:111	171			381		MOV A,C	turn off data reg
342:112	062	371	343	382		STA DREG	-access control bit
342:115	312	152	342	383		JZ TVERFY	test for seek
342:120	257			384		XRA A	force a read
342:121	062	351	347	385		STA HDFLAG	-header operation
342:124	072	372	343	386		LDA DSTAT	get the double
342:127	346	010		387		ANI DSIDE	-sided flag
342:131	062	350	347	388		STA DSFLAG	save for status
342:134	037			389		RAR .	shift for
342:135	037			390		RAR .	-3/6 ms step
342:136	037			391		RAR .	-rate constant
342:137	306	030		392		ADI SKCMD	do a
342:141	041	000	000	393		LXI H,0	-seek
342:144	315	142	343	394		CALL CENTRY	-operation
342:147	332	216	342	395		JC SERROR	seek error?
				396			
342:152				397	TVERFY		
342:152	072	351	347	398		LDA HDFLAG	get the force
342:155	267			399		ORA A	-verify hdr flag
342:156	302	271	342	400		JNZ CHKSEC	no seek & head OK
342:161	006	002		401		MVI B,2	verify retry count
342:163				402	SLOOP		
342:163	076	035		403		MVI A,SVCMD	do a verify
342:165	315	135	343	404		CALL COMAND	-command
342:170	346	231		405		ANI 231Q	error bit mask
342:172	127			406		MOV D,A	save
342:173	312	225	342	407		JZ RDHDR	no error!
342:176	072	366	347	408		LDA DCREG	density control
342:201	356	001		409		XRI DENSITY	flip the density
342:203	062	366	347	410		STA DCREG	update and
342:206	062	372	343	411		STA DCMD	-change density
342:211	005			412		DCR B	decrement retry
342:212	302	163	342	413		JNZ SLOOP	-count & test
342:215	172			414		MOV A,D	restore error bits
342:216				415	SERROR		
342:216	067			416		STC .	error flag
342:217	365			417		PUSH PSW	save errors
342:220	315	160	341	418		CALL HENTRY	seek to trk 0
342:223	361			419		POP PSW	recover errors
342:224	311			420		RET	
				421	*NP		

342:225				422	RDHDR		
342:225	006	012		423		MVI B,12Q	number of retrys
342:227				424	RHLOOP		
342:227	021	377	343	425		LXI D,DATREG	data register
342:232	041	372	347	426		LXI H,TRACK+1	data pointer
342:235	076	304		427		MVI A,RACMD	start a read
342:237	062	374	343	428		STA CMDREG	-header operation
342:242				429	RHL1		
342:242	032			430		LDAX D	get disk data
342:243	167			431		MOV M,A	store in mem
342:244	054			432		INR L	advance pointer
342:245	302	242	342	433		JNZ RHL1	test end of page
342:250	041	374	343	434		LXI H,CSTAT	wait for 1791
342:253	315	154	343	435		CALL BUSY	-to finish cmd
342:256	267			436		ORA A	test for errors
342:257	312	271	342	437		JZ CHKSEC	transfer OK?
342:262	005			438		DCR B	no! - test for
342:263	302	227	342	439		JNZ RHLOOP	-hard error
342:266	303	216	342	440		JMP SERROR	recalibrate
342:271				441	CHKSEC		
342:271	072	375	347	442		LDA SECLEN	get the sector
342:274	117			443		MOV C,A	-size and setup
342:275	006	000		444		MVI B,0	-the table offset
342:277	041	337	342	445		LXI H,STABLE	sector table
342:302	011			446		DAD B	sector size pntr
342:303	072	370	347	447		LDA SECTOR	get the sector
342:306	107			448		MOV B,A	-and save in B
342:307	206			449		ADD M	compare w/table
342:310	076	020		450		MVI A,20Q	error flag
342:312	330			451		RC	error return
342:313	170			452		MOV A,B	initialize 1791
342:314	062	376	343	453		STA SECREG	-sector register
342:317	076	040		454		MVI A,40Q	128 byte sector
342:321	041	005	005	455		LXI H,5:005Q	initialize
342:324	042	342	347	456		SHLD ECOUNT	-error counts
				457			
342:327				458	SZLOOP		
342:327	015			459		DCR C	reduce size count
342:330	107			460		MOV B,A	sector size to B
342:331	370			461		RM	return on minus
342:332	027			462		RAL	double the count
342:333	267			463		ORA A	clear the carry
342:334	303	327	342	464		JMP SZLOOP	
				465			
342:337				466	STABLE		
342:337	345			467		DB 345Q	26 sector diskettes
342:340	345			468		DB 345Q	26 sector diskettes
342:341	360			469		DB 360Q	15 sector diskettes
342:342	367			470		DB 367Q	8 sector diskettes
				471	*NP		



343:033			513	HDCHK		
343:033	041	372	343	514	LXI	H,DSTAT test for
343:036	246			515	ANA	M -head loaded
343:037	062	351	347	516	STA	HDFLAG save the head
343:042	365			517	PUSH	PSW -loaded status
343:043	072	352	347	518	LDA	DRVSEL get current drive
343:046	117			519	MOV	C,A save
343:047	072	367	347	520	LDA	SIDE get current side
343:052	057			521	CMA	. -and merge
343:053	241			522	ANA	C -with drive select
343:054	062	371	343	523	STA	DREG select drive & side
343:057	356	100		524	XRI	ACCESS toggle access bit
343:061	117			525	MOV	C,A save for PREP routine
343:062	072	366	347	526	LDA	DCREG den & head cntl bits
343:065	107			527	MOV	B,A save
343:066	072	371	347	528	LDA	TRACK get the new track
343:071	326	001		529	SUI	1 force single
343:073	237			530	SBB	A -density
343:074	075			531	DCR	A -if track = 0
343:075	057			532	CMA	. compliment
343:076	260			533	ORA	B merge w/control bits
343:077	167			534	MOV	M,A load head & set density
343:100	361			535	POP	PSW head load status
343:101	302	117	343	536	JNZ	RDYCHK conditionally
343:104	345			537	PUSH	H -wait for head
343:105	052	344	347	538	LHLD	TIMER -load time out
343:110				539		TLOOP
343:110	053			540	DCX	H count down
343:111	174			541	MOV	A,H -40 ms for
343:112	265			542	ORA	L -head load
343:113	302	110	343	543	JNZ	TLOOP -time out
343:116	341			544	POP	H
343:117				545		RDYCHK
343:117	176			546	MOV	A,M test for
343:120	346	200		547	ANI	READY -drive ready
343:122	300			548	RNZ	
343:123				549		UNLOAD
343:123	072	366	347	550	LDA	DCREG force a
343:126	366	006		551	ORI	ULOAD -head
343:130	167			552	MOV	M,A -unload
343:131	076	200		553	MVI	A,READY set drive
343:133	067			554	STC	. -not ready
343:134	311			555	RET	. -error flag
				556		
343:135				557		COMAND
343:135	052	344	347	558	LHLD	TIMER get index count
343:140	051			559	DAD	H -and multiply
343:141	051			560	DAD	H -by four
343:142				561		CENTRY
343:142	353			562	XCHG	. save in D-E pair
343:143	041	374	343	563	LXI	H,CSTAT issue command
343:146	167			564	MOV	M,A -to the 1791
343:147				565		NBUSY
343:147	176			566	MOV	A,M wait
343:150	037			567	RAR	. -for the
343:151	322	147	343	568	JNC	NBUSY -busy flag
				569		*NP



343:154				570	BUSY		
343:154	176			571		MOV A,M	test for
343:155	037			572		RAR .	-device busy
343:156	176			573		MOV A,M	restore status
343:157	320			574		RNC .	return if not busy
343:160	303	166	343	575		JMP PATCH+3	jump around patch
343:163				576	PATCH		
343:163	303	343	342	577		JMP HDLOAD	patch for old ATE
343:166	033			578		DCX D	test for
343:167	172			579		MOV A,D	-two disk
343:170	263			580		ORA E	-revolutions
343:171	302	154	343	581		JNZ BUSY	47 machine cycles
343:174	136			582		MOV E,M	get error code
343:175	345			583		PUSH H	save cmd address
343:176	043			584		INX H	track register
343:177	126			585		MOV D,M	save present track
343:200	072	352	347	586		LDA DRVSEL	control bits
343:203	356	200		587		XRI RSTBIT	reset the 1791
343:205	062	371	343	588		STA DREG	-controller to
343:210	356	300		589		XRI STBITS	-clear the
343:212	343			590		XTHL .	-command busy
343:213	062	371	343	591		STA DREG	-fault condition
343:216	066	320		592		MVI M,CLRCMD	force interrupt
343:220	343			593		XTHL .	restore the
343:221	162			594		MOV M,D	-the track reg
343:222	341			595		POP H	restore the stack
343:223	173			596		MOV A,E	error code to A
343:224	067			597		STC .	-error flag
343:225	311			598		RET	
				599			
343:226				600	MEASUR		
343:226	021	000	000	601		LXI D,O	initialize count
343:231	041	372	343	602		LXI H,DSTAT	status port
343:234	016	020		603		MVI C,INDEX	index bit flag
343:236				604	INDXLO		
343:236	176			605		MOV A,M	wait for
343:237	241			606		ANA C	-index
343:240	312	236	343	607		JZ INDXLO	-pulse high
343:243				608	INDXHI		
343:243	176			609		MOV A,M	wait for
343:244	241			610		ANA C	-index
343:245	302	243	343	611		JNZ INDXHI	-pulse low
343:250				612	INDXCT		
343:250	023			613		INX D	advance count
343:251	343			614		XTHL .	four dummy
343:252	343			615		XTHL .	-instructions
343:253	343			616		XTHL .	-to lengthen
343:254	343			617		XTHL .	-the delay
343:255	176			618		MOV A,M	wait for
343:256	241			619		ANA C	-the index
343:257	312	250	343	620		JZ INDXCT	-to go high
343:262	311			621		RET .	98 machine cycles
				622	*NP		

343:263			623	DENFIX		
343:263	171		624	MOV	A,C	trim the
343:264	346	001	625	ANI	1	-excess bits
343:266	057		626	CMA	.	compliment and
343:267	107		627	MOV	B,A	-save in B
343:270	041	353 347	628	LXI	H,DISK	new disk ptr
343:273	136		629	MOV	E,M	get disk no.
343:274	026	000	630	MVI	D,O	offset addr
343:276	043		631	INX	H	current disk ptr
343:277	176		632	MOV	A,M	move to ACC
343:300	253		633	XRA	E	cmpr old w/new
343:301	365		634	PUSH	PSW	save status
343:302	043		635	INX	H	disk table
343:303	043		636	INX	H	-address
343:304	031		637	DAD	D	add the
343:305	031		638	DAD	D	-offset
343:306	176		639	MOV	A,M	get parameters
343:307	366	001	640	ORI	1	mask off density
343:311	240		641	ANA	B	set new density
343:312	167		642	MOV	M,A	update parameters
343:313	361		643	POP	PSW	test new=old?
343:314	300		644	RNZ		
343:315	176		645	MOV	A,M	updata CDISK
343:316	062	366 347	646	STA	DCREG	-also
343:321	311		647	RET		
			648			
343:322			649	TIMOUT		
343:322	041	000 000	650	LXI	H,O	time-out delay
343:325			651	TILOOP		
343:325	053		652	DCX	H	decrement count
343:326	174		653	MOV	A,H	test for delay
343:327	265		654	ORA	L	-count equal zero
343:330	343		655	XTHL	.	long NOP
343:331	343		656	XTHL	.	-instruction
343:332	302	325 343	657	JNZ	TILOOP	
343:335	311		658	RET		
			659			
343:336			660	SBEGIN		
343:336	345		661	PUSH	H	
343:337	041	342 343	662	LXI	H,DSTALL	
343:342			663	DSTALL		
343:342	351		664	PCHL		
343:343	341		665	POP	H	
343:344	311		666	RET		
			667			
343:345			668	SIDEFX		
343:345	171		669	MOV	A,C	get the side bit
343:346	346	001	670	ANI	1	trim the excess
343:350	027		671	RAL	.	move the bit
343:351	027		672	RAL	.	-to the side
343:352	027		673	RAL	.	-select bit
343:353	027		674	RAL	.	-position
343:354	062	367 347	675	STA	SIDE	save side bit
343:357	311		676	RET		
			677	*NP		