

## Altair Minidisk CP/M 2.2b Version 2.3

This version of CP/M 2.2 runs on the Altair Minidisk floppy drive. As of version 2.0, the same BIOS is used for both the Altair 8" drive and the Mindisk and a conditional assembly switch selects the drive type. Version 2.0 is a track buffered BIOS and improves performance 20%-75% compared to previous versions of CP/M for the Minidisk.

There are four drives configured in the BIOS with 1K blocks and 32 directory entries. 62K on each drive is available for CP/M.

Tracks 0-3 are formatted the same as tracks 0-5 on 8 inch floppies (same as BASIC, Lifeboat CP/M, and Burcon CP/M). This is the format expected by the disk boot loader ROM. These tracks contain the boot image of CP/M and are not used by CP/M for file storage.

Offset	Description
0	Track number + 80h
1-2	Length of boot loader 0100h (00h, 01h)
3-130	128 byte CP/M sector
131	0FFh stop byte
132	Checksum of bytes 3-130 with no carry
133-136	Not used

Tracks 4-34 are formatted the same as tracks 6-76 on 8" floppies (same as BASIC, Lifeboat, and Burcon CP/M). These tracks are used by CP/M for directory and file storage.

Offset	Description
0	Track number + 80h
1	Sector number
2	BASIC file number (not used)
3	BASIC bytes written (not used)
4	Checksum (all data except 0,1,135,136)
5-6	BASIC group pointer (not used)
7-134	128 byte payload (CP/M sector)
135	0FFh stop byte
136	Not used

Location of the MODE, IOBYTE and SNDNULL bytes match the locations used for 8" CP/M and Hard Disk CP/M for the Altair:

Item	Offset from start of BIOS
MODE	0559 (25D9 in memory after MOVCPM)
IOBYTE	055A (25DA in memory after MOVCPM)
SNDNULL	055B (25DB in memory after MOVCPM)

## IOBYTE Implementation

Minidisk CP/M 2.2b implements CP/M's IOBYTE feature which allows redirection of logical CP/M devices to a variety of physical devices. IOBYTE implementation is required by some programs (e.g., Kermit) for proper operation. The possible logical-to-physical device assignments are shown below:

<b>CON</b> device (bits 1,0): 00 - TTY on 88-SIO at 0/1 01 - CRT on 88-2SIO at 10h/11h* 10 - BAT (indirect through RDR and LST devices) 11 - UC1 on 88-2SIO at 12h/13h	<b>PUN</b> device (bits 5,4): 00 - TTY on 88-SIO at 0/1 01 - PTP on 88-2SIO at 10h/11h* 10 - UP1 on cassette port (SIO 6/7) 11 - UP2 on 88-2SIO at 12h/13h
<b>RDR</b> device (bits 3,2): 00 - TTY on 88-SIO at 0/1 01 - PTR on 88-2SIO at 10h/11h* 10 - UR1 on cassette port (SIO 6/7) 11 - UR2 on 88-2SIO at 12h/13h	<b>LST</b> device (bits 7,6): 00 - TTY on 88-SIO at 0/1 01 - CRT on 88-2SIO at 10h/11h* 10 - LPT on 88-LPC at 2/3 11 - UL1 on 88-2SIO at 12h/13h

\* = Default IOBYTE

## Patching the IOBYTE

The STAT utility in CP/M is typically used to make IOBYTE device assignments as required. These assignments are temporary and the defaults shown above are restored whenever CP/M is cold started. To permanently modify the IOBYTE, the default IOBYTE in the CP/M boot image must be patched.

To change the default IOBYTE that CP/M loads on cold start, the MOVCPM5 file can be patched or memory can be modified after running MOVCPM5, but prior to writing the image to disk using SYSGEN. Patching the IOBYTE would be necessary, for example, to force CP/M to use the serial port on an SIO board as the console upon cold start instead of the 2SIO serial port.

The IOBYTE is at location 0x25DA in memory after running MOVCPM5 or loading MOVCPM5.COM into memory using DDT. Each of the four logical devices occupies a two bit field in the IOBYTE. Each two bit field, in turn, selects one of four physical devices for the logical device. The bit field locations and values are shown in the device mapping table above.

The default IOBYTE in binary is 01010101 (55h). Note that this is a value of 1 in each of the four logical device fields. As an example, to force the CON, RDR and PUN devices to use an SIO serial board port at I/O address 0/1 and an 88-LPC line printer controller as the LST device, the IOBYTE should be set to binary 10000000 (80h).

## Using MOVCPM5

MOVCPM5 is used to create an image of CP/M configured for the amount of RAM in a system. The SYSGEN program is used to write that image to the boot sectors of a disk. Following are a few examples of using MOVCPM5:

```
A>MOVCPM5           Create CP/M sized for the amount of RAM in this system
A>MOVCPM5 48       Create CP/M sized for 48K of RAM
A>MOVCPM5 56       Create CP/M sized for 56K of RAM
```

When MOVCPM5 completes, the CP/M image is in memory. The very next step **MUST** be to write the image to the boot sectors on a disk (run SYSGEN), or to save the image as a file to disk (use the SAVE command as detailed in messages from MOVCPM5).

To write the image to the boot sectors of a disk, run the SYSGEN program. A typical SYSGEN session after executing MOVCPM5 is as follows:

```
A>SYSGEN
ALTAIR MINI-DISK SYSGEN Ver 2.0
Source drive name (or RETURN to skip) - press return to use the CP/M image in memory
Destination drive name (or RETURN to reboot) - type drive letter, e.g., A (not echoed)
Destination on A, then type RETURN - press RETURN to write CP/M to the specified drive
Function Complete
Destination drive name (or RETURN to reboot) - if desired, write CP/M to another floppy or to a
different drive. If finished and a disk just writ-
ten is in drive A, reboot the computer from the
front panel instead of pressing return.
```