

Altair CP/M 2.2b ver 2.3

Two versions of CP/M 2.2 were available for the Altair 8800 back in the early 1980's – a version released by the Burcon computer store out of Houston, Texas, and a version from Lifeboat Associates. The Burcon version appears to be Lifeboat CP/M 1.4 with the minimal set of patches required to make the BIOS work with CP/M 2.2. The Lifeboat CP/M 2.2 version is a full featured release typical of Lifeboat CP/M distributions.

CP/M 2.2b is a modern release of CP/M 2.2 that uses track buffering to provide substantial speed improvement versus the Burcon and Lifeboat releases, and also adds a few additional features:

- 1) Compared to original CP/M, version 2.2b loads CP/M programs in less than half the time. Any program with substantial disk I/O will run 20%-75% faster. No changes were made to the sector skew used with Burcon and Lifeboat CP/M disks, so version 2.2b can interchangeably read and write Burcon and Lifeboat disks.
- 2) Version 2.2b provides a full implementation of CP/M's IOBYTE feature which allows redirection of logical CP/M devices to a variety of physical devices. This allows use of ports in addition to just the first 2SIO port and allows programs like Kermit, which requires IOBYTE, to run properly.
- 3) To better support a Teletype as the console, version 2.2b automatically transmits a null after C/R when it detects a Teletype as the console during cold boot.
- 4) Writes to disk are verified by default. This is done with almost no performance penalty.

Version 2.0 (11/12/2017) is a complete re-write from scratch of CP/M 2.2b. Previous versions were created using Burcon CP/M as the starting point which contains a lot of convoluted code I was never happy with. This update also adds some new features listed below. Version 2.0 increases performance yet another 5%-20% over previous versions of CP/M 2.2b.

- 1) Works for either the Altair 8" or Altair Mini-Disk with conditional assembly.
- 2) Disk select timeout (8" floppy only).
- 3) Individual sectors can be bad within a track and the remaining sectors are still accessible.
- 4) To improve speed, checksum calculation and stop byte validation are done on the fly as the sector is read. This is the only Altair software I know of that does this.
- 5) Smarter and faster error detection and recovery: Up to 12 retries per sector for all 32 sectors on a track and a possible track restore and re-seek all complete in less than three seconds.
- 6) Write verify is enabled by default but can be enabled/disabled with the fWRTVFY bit in the MODE byte (same bit as used with Burcon CP/M 2.2).

IOBYTE Implementation

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:

CON 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	PUN 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
RDR 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	LST 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 MOVCPM8 file can be patched or memory can be modified after running MOVCPM8, 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 25DAh in memory after running MOVCPM8 or loading MOVCPM8.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).

Updating to Version 2.2b

Installing or updating to CP/M version 2.2b is as simple as running MOVCPM8 (located in the same directory as this file) followed by SYSGEN. You can upload MOVCPM8.COM to a CP/M disk using the PCGET utility.

Additional programs in this directory include a fast format utility (AFORMAT) and a fast disk copy utility (ACOPY) both written by Martin Eberhard. If you have FORMAT already present on your CP/M disk, you'll want to replace it with AFORMAT as FORMAT has a minor bug that is magnified by version 2.2b, and AFORMAT is dramatically faster.

Finally, a complete disk image can be written using the PC2Flop utility. Disk images with a 24K and 56K CP/M build are provided. The 24K image can be loaded onto most any machine and MOVPCM8 and SYSGEN used to size CP/M for your machine. MOVPCM8 and SYSGEN are on both disk images. Altair Clone users can also load the disk image using the Configuration Monitor.

Using MOVPCM8

MOVPCM8 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 MOVPCM8:

```
A>MOVPCM8           Create CP/M sized for the amount of RAM in this system
A>MOVPCM8 48        Create CP/M sized for 48K of RAM
A>MOVPCM8 56        Create CP/M sized for 56K of RAM
```

When MOVPCM8 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 MOVPCM8).

To write the image to the boot sectors of a disk, run the SYSGEN program. A typical SYSGEN session after executing MOVPCM8 is as follows:

```
A>SYSGEN
SYSGEN Ver 1.2
For MITS DISK CP/M
```

```
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 written is in drive A, reboot the computer from the
front panel instead of pressing return.
```