N8VEM Test Monitor Facility

The CP/M Single Board Z80 Computer boots up to a Test Monitor as an aid to getting the full system up and running. It also provides a tool for digging around in the memory space to see where things are located. There are several versions of this code around which use different command letters and different baud rates. The most recent distributed version is V09.

Recent versions of the EPROM code are set up to run at 38400 baud, while older versions use 9600 baud. Garbled text on the screen is the clue to reset the terminal baud rate.

The executable for this monitor (dbgmon-g.asm) was built for the TASM assembler using a DOS machine. The sign-on message is followed by a '>' prompt. This is the current Andrew Lynch PROM not the RomWBW version. All monitor commands are single letters (either upper or lower case), and all data is entered in hexadecimal format (0-9, A-F).

A quick guide to using the DBGMON monitor program follows:

C - Boots CP/M 2.2C from RAM drive created by EPROM startup. This command will exit the monitor and jump to the cold boot location created by the loader (default $D400). There is a GM.com executable that will print ‘Hello World!’ to the terminal and then revert to the Test Monitor. A version of XModem 5.0 is loaded on drive A: for transferring files back and forth via the terminal subsystem (which will disappear when power is removed from the SBC in the absence of a battery backed memory facility).

D $xxxx $yyyy - Dump memory from hex location $xxxx to $yyyy on the screen as lines of 16 hexadecimal bytes with their ASCII equivalents (if within a set range, else a '.' is printed). A good tool to see where code is located, check for version id, obtain details for chip configurations and execution paths.

E $xxxx - Examine memory location $xxxx. This routine will print the hex address $xxxx and the memory contents found at that location. There are four options beyond this: Press the Space bar to increment to the next location and show its contents, Press 'P' to program the byte at the current RAM location to a new value, Press 'R' to jump to the current location and run a program at that address, or Press 'ESC' to return to the Monitor prompt.
F $xxxx $yyyy $zz - Fill memory from hex $xxxx to $yyyy with a single value of $zz over the full range. The Dump command can be used to confirm that the Fill completed as expected. A good way to zero out memory areas before writing machine data for debug purposes.

H ABC.hex - Load an Intel Hex format file via the terminal program. The terminal emulator program should be configured to give a delay at the end of each line to allow the monitor enough time to parse the line and move the data to memory. Keep in mind that this will be a transient effort as memory is not battery backed and no disk facility is available.

I $pp - Input data from port $pp and display to the screen. This command is used to read values from hardware I/O ports and display the contents in hexadecimal. I/O facilities and memory areas use parallel and separate instruction handling.

K - Echo any key-presses from the terminal. Press 'ESC' key to quit. This facility provides that any key stroke sent to the computer will be echoed back to the terminal. File downloads will be echoed as well while this facility is 'on'.

M $xxxx $yyyy $zzzz - Move hex memory block $xxxx to $yyyy to memory starting at hex location $zzzz. Care should be taken to insure that there is enough memory at the destination so that code does not get over-written or memory wrapped around.

O $pp $xx - Output data byte $xx to port $pp. This command is used to send hexadecimal values to hardware I/O ports to verify their operation and is the sequel to the I operation. Use clip leaded LEDs to confirm the data written.

S - Save memory block to RTC/NVRAM. Note you must have used the 'T' command previously. This facility provides the means for setting the state of the time clock on the SBC. There are some utility programs available to set and configure the time and time systems (see note below). The CP/M version of the RTC does not support a time function.

T $xxxx - Copy contents of RTC/NVRAM into memory at $xxxx. This will copy all of the DS1302 registers into a clear 48 byte block. Use the 'E' and 'P' memory edit commands to examine and alter the values. When done, the 'S' command will restore the new values into the DS1302. The table below describes the data offsets and their contents:
OFFSET CONTENTS
0 Seconds
1 Minutes
2 Hours
3 Date
4 Month
5 Day of week
6 Year
7 Write protect
8 Trickle charger
10 NVRAM buffer location 0
: " "
2E " " 31

'ESC' - releases current mode and returns control back to the command prompt.

NOTES:

The Dbug Monitor allows examination of all memory locations, however, it is not possible to write to memory locations below $8000 hex since that lower area contains the boot ROM, which must be programmed rather than written.

The SBC in default state has no battery backed RAM or disk store facility so programs that are downloaded or stored in the RAM Disk facility are transient only. There is a battery facility via the expansion buss that can be implemented or persistent storage facilities added to the base system.

There is an RTC utility on the A: disk of the ROM CP/M system which provides better facilities to manipulate the Real Time Clock than the Test Monitor. Use C to load CP/M and then run RTC to see the options list.