

eZapple Machine Language Monitor

Old-timers (like me) will recognize the name “eZapple” and many of the single-letter commands described below as deriving from Technical Design Labs famous Zapple monitor. In fact, eZapple is a port of my own extensively modified Zapple monitor that I run on an original Imsai 8080 system.

Here is a summary of eZapple's single letter commands:

There are 4 commands to load or save files on the SD card or a PC acting as a terminal:

H: HEXLOAD	Load a Hex file from the PC into memory converting it into a binary file
X: XMODEM	Load a file from the PC into memory using the Xmodem protocol
L: LOAD	Load a file into memory from the SD card
S: SAVE	Save a file from memory to the SD card

There are 8 commands to manipulate memory and I/O ports:

C: CHANGE	Change and/or examine individual bytes in memory
D: DISP	Display a block of memory as both Hex and ASCII
F: FILL	Fill a block of memory with a constant
M: MOVE	Move a block of memory from one location to another
Q: QUERY	Read or write any of the eZ80's I/O ports
T: TYPE	Display a block of memory as ASCII text
V: VERIFY	Verify that two blocks of memory are identical
W: WHERE	Search memory for a specified sequence of bytes

There are 2 commands that work together to enable program debugging:

G: GOTO	Execute a program that's been loaded into memory and/or debug it using breakpoints
R: REGDISP	Display the contents of the eZ80's registers

There are also 3 miscellaneous commands:

ESC:	Exit program if ESC (Escape key) is pressed
E: ERROR	Display a '*' and return to monitor if there is an input error
U: USER	Jump to location 040100 and execute a new command or user program located there

The GOTO ('G') command allows you to debug running programs. You can use the LOAD command to load a program into memory. If you then type G<start address>[CR] , where [CR] means a carriage return, the program will just be executed, and will return to eZapple if it ends with a RET instruction.

If you type `G<start address>,<break address>[CR]` the program will execute up to the breakpoint, halt, display the contents of all the registers, and return to the eZapple prompt. Breakpoints must be set at the first byte of an instruction. To see where to set them, you need to look at the list file that was created when the program was assembled. To go on to another breakpoint, just type `G,< break address>[CR]`, and the program will restart from where it left off. By doing this repeatedly, you can single step through your program if you wish.

When the program stops at a breakpoint, you can examine and modify memory using the 'C' and 'D' commands, or even change the program or the register contents (since they're stored at a fixed location). Do this at your own risk!. Then resume execution again using `G,< break address>[CR]`.

eZapple uses UART1 for its user interface, so the Agon needs to be connected to a PC's serial port with the PC running a serial terminal program.

The easiest way to make the connection is to get a USB to TTL Serial 3.3V UART Converter Cable that uses an FTDI chip and has 2.45mm Dupont Headers on each wire. These are readily available from Amazon for about \$15 (I'm using the DSD TECH SH-U09G USB to TTL Serial Cable). Make sure it uses 3.3V TTL, NOT 5V TTL.

You only need to hook up 3 wires to the pins of the Agon's GPIO header. The connections are:

Adapter TxD --- Agon GPIO_PC1 (pin 18) This is the receive data (RxD) pin for UART1

Adapter RxD --- Agon GPIO_PC0 (pin 17) This is the transmit data (TxD) pin for UART1

Adapter GND --- Agon GPIO Ground (GND) Either pin 3 or pin 5 can be used.

NOTE: Agon versions other than the AgonLight2 may have a different pinout on the header. See <https://agonconsole8.github.io/agon-docs/GPIO/>

The eZapple monitor runs in 24-bit (ADL) mode and has two pieces: ZapLoader.bin and eZapple.bin. Zaploader runs as a MOS command, automatically loads the actual monitor at address 04F000h, and then jumps to the start of the monitor's code. Both pieces must be saved in the mos directory, just like other MOS commands. This can be done using either Jeroen Venema's hexload program or directly onto your SD card from a PC.

I strongly recommend using the (free) Tera Term program as your serial terminal as it works seamlessly with the monitor's HEXLOAD and XMODEM commands. For example, bring up Tera Term, and set the baud rate to 38400. Then boot the Agon, and type ZapLoader into MOS. eZapple's sign-on message will appear:

```
eZ80 Zapple
SPL=04EFD0
>
```

Now enter 'H' and a carriage return. The monitor will then wait for you to send it a hex file. In Tera Term's menu bar select 'File-->Send File...' and pick the hex file you want to send from your directory. Click the 'Open' button, and the file will be sent to the Agon and converted to a binary file.

Similarly, for XModem transfers, select 'File-->Transfer-->XMODEM-->Send...'

The file sent to the agon can then be saved on the SD card using eZapple's SAVE command.