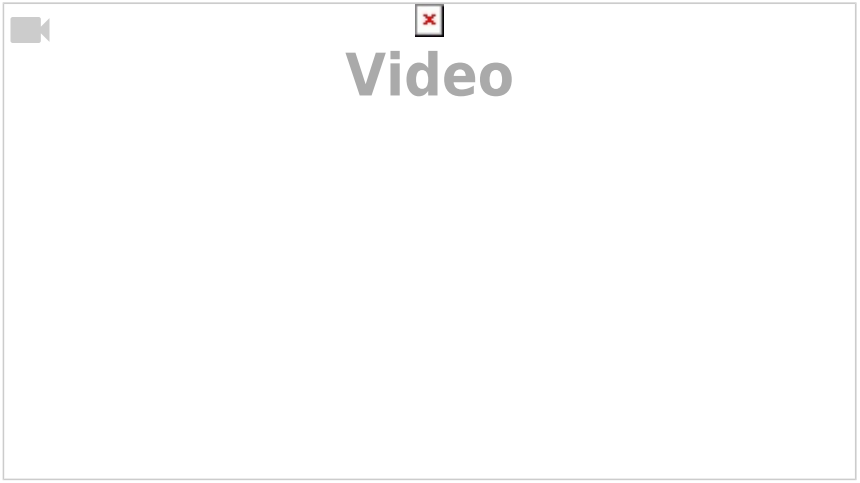
HEXMON



wozmon is a tiny but elegant monitor program Steve Wozniak wrote for the original Apple-1 computer in 1976. It lets you examine memory, change memory, and run machine-language programs directly from the keyboard. This tutorial is designed to teach you how to use HEXMON, which is a very similar program.

All addresses and data values are in **hexadecimal** (no $ prefix needed). Press **RETURN** (or Enter) after each line to execute it. The prompt for the Apple-1 was a backslash \, sometimes shown as a @ in emulators/replicas. The prompt was changed to a * for the Apple II and that is the convention that HEXMON uses.

Starting the Monitor

- Power on the SD-8516 / VC-3.
- After you hear the chime, you can type `MON`.
- You should see *. The system is now ready for input.

1. Examining Memory (Display contents)

Single location: Type the 4-digit address and press RETURN Example:

```
0030
```

This might respond `0030: A9` i.e. it shows the current byte at address $0030.

Range of locations: Use the dot operator `.` between start and end address.

Example:

```
200.2FF
```

This dumps bytes from $0200 to $02FF, 8 bytes per line with address prefix and printable display.

2. Writing to Memory

Use the entry operator `:` after an address, followed by data bytes. Bytes are stored starting at the location you have specified.

Write-One

Example:

```
1000:A9
```

This will store $A9 into memory location $1000.

Write-List Example:

```
200: A9 00 20 EF FF
```

Stores those 5 bytes starting at address $0200.

3. Running a Program

Type the start address followed by R (Run).

Example:

```
1000R
```

This will call the program at $1000 and begin executing code. If the program issues a RET, it will return to HEXMON. You could then quit HEXMON normally by typing Q.

It's worth noting you don't need HEXMON to run a program. If a program is assembled starting at $030100, typing SYS by itself on a line will run it. Otherwise you can type SYS `$(address)` at the command prompt to run a machine language program.

Example Program

Example “Hello World” program:

```
C000: 00 34 10 C0 00 00 20 66
C008: 86 05 00 20 64 86 05 85
C010: 48 45 4C 4C 4F 20 57 4F
C018: 52 4C 44 21 00 00 00 00
```

Type this in and then type **`C000R`** to run the program!

Full Listing

You can also enter code from full disassembly listings. Just type in the values on the left, as shown above in 'example program'. The other parts of the disassembly are just for your reference and to help you learn what the program does.

```
 ADDR    BYTES          INSTRUCTION          COMMENT
------------------------------------------------------------------------------
----
$C000:   00 34          LDBLX @msg           ; Load pointer to string into
BLX
         10 C0 00                            ; (Bank 0, address $C0 10)
$C005:   00 20 66       LDAH $66             ; IO_PRINT_STR function
$C008:   86 05          INT $05              ; BASIC services library
$C00A:   00 20 64       LDAH $64             ; IO_NEWLINE function
$C00D:   86 05          INT $05              ; Call BASIC services
$C00F:   85             RET                  ; Return to caller
$C010:                  msg:                 ; String data label (equates to
$C000 above)
$C010:   48 45 4C 4C    .bytes "HELLO        ; H E L L
$C014:   4F 20 57 4F           WORLD!", 0    ; O   W O
$C018:   52 4C 44 21                         ; R L D !
$C01C:   00                                  ; null terminator (zero
terminated string)
```

Just like before, you can run the program by typing in **C000R** (run program at start location C000).

The RET ($85) at $C00F will return control to the monitor program.

4. LOAD, SAVE and PUBLISH

HEXMON is equipped with three commands that help you load, save and publish machine language programs.

Command	Function
L	Load a machine language file using the address in the file header.
####L	Load a machine language file explicitly at the address given. This ignores the address in the program header.
####.####S	Save machine code in range to file. You can use this to save your programs to disk.
####.####P	Publish. This will create a file with a publishable listing including checksum data.

The L and S commands deal with binary data. The P command will produce a publishable listing like this:

```
  $C000: 00 34 10 C0 00 00 20 66 :8A
  $C008: 86 05 00 20 64 86 05 85 :1F
  $C010: 48 45 4C 4C 4F 20 57 4F :3A
```

```
$C018: 52 4C 44 21 00 00 00 00 :03
$C020: 00 :00
```

When you enter the program, you can skip the $ and the :byte checksum, or you can type them in (your choice). You can also substitute a comma , in place of a colon for the checksum.

You can also enter data like this:

```
C000:003410C000002066
```

However, doing it this way will not verify the checksum. If you enter the program incorrectly there will be no way for you to automatically find the error.

Summary of Main Commands

Command format	Purpose	Example
————————	————————————	————————
addr	Show one byte	C100
start.end	Show range	E000.E0FF
addr:data data data...	Store bytes starting at addr	1000:EA A9 00` | | addr1.addr2S | Save binary data | C000.C020S | | addr1.addr2P | Create publishable listing | C000.C020P | | L | Load file at header address | L | | addrL | Load file at the given address | C000L`

Now you have everything you need to know to use HEXMON!

Many people still use it today because of its simplicity and elegance.

NEXT STEPS

If you want to have some fun and try more example programs that can be loaded via HEXMON, please see: Assembly Programs.

If you are looking for a tutorial on how to write Assembly Language programs yourself, you can refer to:

- SD-8516 Assembly Language (crash course)
- SD-8516 Programmer's Reference Guide (full tutorial + resources)

Have fun exploring SD-8516 machine code!