AMON USERS MANUAL

Amon Version 2.8

28 March 2022

Martin Eberhard

Revision History

Manual Revision	Firmware Revision	Author	Revision
28 Aug 2016	2.1	M. Eberhard	First released version
3 Oct 2016	2.2	M. Eberhard	Add IN and OT commands
10 Oct 2016	2.3	M. Eberhard	Better error messages, etc.
13 Oct 2016	2.4	M. Eberhard	Add GO record to AD command. Option to not GO on AL command.
11 Jun 2020	2.5	M. Eberhard	Add 2732 support, with 2 new commands: ‘MT’ and ‘?’
26 Mar 2022	2.7	M. Eberhard	Fix MT command to correctly avoid stuff in Amon’s RAM page. Prevent commands’ stack use from messing up RAM code and variables. Print ‘Memory OK’ if no errors. Allow ESC to abort commands.
28 Mar 2022	2.8	M. Eberhard	‘OK’ message after MT and VE commands. Flush Transfer Port buffer more thoroughly on AL.

TABLE OF CONTENTS

Introduction 1
RAM Requirements 1
EPROM Requirements 1
I/O Ports 1
Amon Commands 2
  Memory Commands 2
    CO <Source> <Dest> <Count> [<Repeat>] (Copy Memory) 2
    DU [<Addr> [<Count>]] (Dump Memory) 2
    EN [<Addr>] (Enter Memory Data) 2
    EX [<Addr> [<Option>]] (Execute) 2
    FI [<Value> [<Addr> [<Count>]]] (Fill Memory) 3
    MT <Addr> <Count> {4K-Byte EPROM only} 3
    SE <Addr> <Element1> [<Element2> [..<ElementN>]] (Search) .. 3
    VE <Addr1> <Addr2> <Count> (Verify Memory) 3
  Transfer Commands 4
    AD <Addr> <Count> [<Go>](Dump Memory as Altair Binary File) 4
    AL [<0/1>] (Load and Execute Altair Binary File) 4
    HD <Addr> <Count> [<Offset>](Dump Memory as Intel Hex) 4
    HL [<Offset>] (Load Intel Hex into memory) 5
    TE [<ExitChr>] (Terminal Mode) 5
    TP [<0-7>] (Set Transfer Port) 6
  I/O Commands 6
    IN <Port> (Input from port) 6
    OT <Port> <Data> (Output to port) 6
  Other Commands 6
    ? Print Help Screen {4K-Byte EPROM only} 6
    BO (Boot from Altair Floppy Diskette) 6
    HB [<0/1>] (Boot from Altair hard disk) 7
    TT <0/1> (Set Terminal Type) 7
Entry Points 8
  F800h: Amon Monitor 8
  FC00h: Boot from Altair Hard Disk (HDBL) 8
  FE00h: Boot from Altair Tape (MBL) 8
  FF00h: Boot Altair Floppy Disk (DBL & MDBL) 9

Example: Programming an EPROM with a Cromemco Bytesaver..... 9
Altair Absolute Binary File Format..... 11

AMON

Full Featured ROM Monitor
For an Altair 8800 with an 88-2SIOJP or an 88-2SIO

INTRODUCTION

Amon is a full-featured ROM-based monitor for an Altair 8800 with my own 88-2SIOJP, an Altair 88-2SIO, or an Altair 88-UIO. (The 88-UIO does not have a second serial port, so the default Transfer Port will not work.)

Amon provides commands for manipulating memory, transferring (uploading and downloading) memory in Altair Absolute Binary format and Intel Hex format, as well as booting from any Altair boot device (paper tape, cassette tape, 8" floppy disks, minidisks, or from an Altair Datakeeper hard disk).

Amon can also be used to program EPROMs, using a memory-based EPROM programmer such as any of the Cromemco Bytesavers.

RAM REQUIREMENTS

Amon requires most of the highest 256-byte page of contiguous RAM for stack space, variables, buffers, and for relocated code. During initialization, Amon will search and find this page. The address of Amon's RAM page is printed immediately following the sign-on banner.

EPROM REQUIREMENTS

Amon can be assembled to run in a 2K-byte (2716) EPROM, or in a 4K-byte (2732) EPROM. When assembled for a 4K-byte EPROM, two additional commands are available: '?' (Help) and 'MT' (memory test).

I/O PORTS

Amon uses the 88-2SIOJP's (or 88-2SIO's) Port 0 as its console. It can use this board's Port 1 for its "Transfer Port", as well as any of the other standard Altair serial or parallel ports. The Transfer Port is initialized to be the 88-2SIOJP's (or 88-2SIO's) Port 1.

The Transfer Port is used as the source or destination for any of the five Transfer Commands.

AMON COMMANDS

Amon commands may be typed at the Amon prompt, '>'. Commands are executed once you type the Return key. You can correct typing mistakes with the DEL or the BACKSPACE key.

All parameters are 4 hex digits (16-bits), unless otherwise noted. Additional upper hex digits are ignored, and leading zeros are not required.

You can abort most commands with Control-C. If Amon is in a 4K-Byte EPROM, then Escape will also abort most commands.

MEMORY COMMANDS

CO <SOURCE> <DEST> <COUNT> [<REPEAT>] (COPY MEMORY)

Copies <Count> bytes memory starting at address <Source> to memory starting at address <Dest>. Optionally, repeats the copy <Repeat> times. (Max value for <Repeat> is FF for 255 passes.)

A period is printed on the Console for each completed pass through the copy, unless <Repeat>=1 (the default).

The CO command verifies the copy when done, using the VE command.

This command can be used to program an EPROM with (for example) a Cromemco Bytesaver board. See example below.

DU [<ADDR> [<COUNT>]] (DUMP MEMORY)

Dumps <Count> bytes memory on the Console in hexadecimal, starting at <Addr>, which defaults to 0. If no <Count> is specified, then dump all 65K bytes of memory.

Press the space bar to pause and restart the dump.

EN [<ADDR>] (ENTER MEMORY DATA)

Allows you to enter 2-digit hex data into memory starting at <Addr>, using a space or Return as a separator between bytes. Type Return on a blank line to exit. If no address is provided, then the starting address will be 0.

CONTROL-C aborts without saving the current line of data.

EX [<ADDR> [<OPTION>]] (EXECUTE)

Calls <Addr>, which defaults to 0. A RET instruction will return to the monitor, if the stack remains intact and the PROM has not been disabled (with an IN FFh instruction).

If <Option> = 1 (or any odd number), then Amon will input from port FFh prior to executing the requested code. This will disable the Amon PROM on an 88-2SIOJP board (with its

ED switch closed) and enable any other memory that occupies the top 2K-bytes of memory (starting at F800h).

For example, if your Altair has both an 88-2SIOJP and MITS's ROM Basic Module (88-RMB) board installed, then the top 2K-bytes of the 88-RMB will be disabled, and Amon will occupy this memory space, until an IN from port FFh is executed. Start ROM Basic (at address C000) from Amon this way, to disable the Amon PROM and enable the top 2K-bytes of ROM Basic:

```
EX C000 1
```

FI [<VALUE> [<ADDR> [<COUNT>]]] (FILL MEMORY)

Fills <Count> bytes of memory, starting at <Addr>, with <Value>, which is a 2-digit hex value. <Value> and <Addr> default to 0. <Count> defaults to all of memory, wrapping around if necessary. The fill stops after either <Count> bytes have been filled or the fill reaches the RAM pages used by Amon.

Note that FI with no arguments will clear all memory below Amon's RAM page.

MT <ADDR> <COUNT> {4K-BYTE EPROM ONLY}

Test <Count> bytes of memory, starting at the specified address. Report all errors on the console. A dot is printed on the console at the completion of each of 29 passes through memory. If no errors, then MT will print 'OK' when done. This test will skip over locations used by Amon in its RAM page. Use control-C to abort the test.

SE <ADDR> <ELEMENT1> [<ELEMENT2> [..<ELEMENTN>]] (SEARCH)

Search memory, starting at the specified address, for the specified sequence of elements, where each element is either a 2-digit hexadecimal number or a text string within single-quotes. Example: (Try this to find a string in the Amon PROM.)

```
SE 0 'M. Eberhard' 0D 0A 'RAM:'
```

This will print the address of the beginning of the sequence if it is found. You will be given a chance to continue searching for another instance of the sequence, if the sequence is found.

VE <ADDR1> <ADDR2> <COUNT> (VERIFY MEMORY)

Compares a block of memory starting at <Addr1> that is <Count> bytes long, to an equal-sized block of memory starting at <Addr2>. Differences are reported on the

Console, with the address and data from the first data block, followed by the data found in the second block. If no mismatches, then VE will print 'OK' when done.

TRANSFER COMMANDS

All of these commands use the specified Transfer Port as either the source or destination for data.

AD <ADDR> <COUNT> [<GO>](DUMP MEMORY AS ALTAIR BINARY FILE)

Dumps <Count> bytes of memory to the Transfer Port, starting at <Addr>, in Altair Absolute Binary format.

If a Go Address <GO> is provided, then the dump will be terminated with a GO record that contains this address. Otherwise, no GO record will be included.

AL [<0/1>] (LOAD AND EXECUTE ALTAIR BINARY FILE)

Loads an Altair Absolute Binary file via the Transfer Port and optionally jumps to its GO address. Amon will input from port FFh prior to executing the loaded code, to disable Amon's PROM, freeing all 65K of memory space for RAM (assuming the ED switch on the 88-2SIOJP is closed).

You can abort a load with CONTROL-C. If your Altair Binary File does not end with a GO Record, then you will need to type CONTROL-C to return to Amon when the load is done.

If no parameter is typed (or the parameter is not 0) then a GO record will cause execution at the GO address. If the optional parameter is 0 then a GO record will cause the GO address to be printed on the console and control returned to the monitor.

The Altair binary loader will terminate and print an error message for any of the following reasons. The error message will include a single-character error code and the 16-bit (4 hex digit) memory address associated with the error.

Error Code	Error Type	Explanation
C	Checksum Error	Record checksum is incorrect
M	Memory Error	Write to memory failed
O	Overwrite Error	Attempt to overwrite AMON's RAM

HD <ADDR> <COUNT> [<OFFSET>](DUMP MEMORY AS INTEL HEX)

Dumps <Count> bytes of memory to the Transfer Port, starting at <Addr>, in Intel Hex format. The optional <Offset> is added to the address of each record.

HL [<Offset>] (Load Intel Hex into memory)

Loads an Intel Hex file from the Transfer Port into memory at the addresses specified in the hex file. If <Offset> is specified, then it is added to the record addresses.

A period is printed on the Console for each record.

The HL command will terminate and print an error message for any of the following reasons. The error message will include a single-character error code and the 16-bit (4 hex digit) memory address associated with the error.

Error Code	Error Type	Explanation
C	Checksum Error	Hex record checksum is incorrect
H	Hex Error	Bad hex digit in input
M	Memory Error	Write to memory failed
O	Overwrite Error	Attempt to overwrite Amon's RAM

Note that if a Hex Error is detected during the first four bytes of a hex record then the address in the error message will be meaningless. Otherwise, the address printed is the memory address associated with the failure (which includes the offset, if one was provided).

Loading terminates normally with any record that has 0 data bytes. You can also abort the load by typing CONTROL-C. If the load terminates normally then the total number of records loaded will be displayed on the console.

The maximum baud rate when loading a file with HL is 19200 baud. (Characters will be dropped at higher baud rates.)

TE [<ExitChr>] (Terminal Mode)

Enters Terminal Mode: console keyboard data goes to Transfer Port, and Transfer Port data goes to the Console. (Use this command to verify a Transfer Port connection.)

<ExitChr> specifies the Exit Character, a control character that defaults to CONTROL-C. Control characters may be entered without the CONTROL. For example, you may type Z instead of CONTROL-Z. (Note: if you type CONTROL-C as as <ExitChr>, the TE command will immediately abort.)

Type the Exit Character to exit Terminal Mode.

TP [<0-7>] (Set Transfer Port)

The Transfer Port is the port used for transferring Intel hex files with the AD, AL, HD, HL, and TE commands.

TP Value	Port	Port Address
0	88-2SIOJP Port 0 (2 stop bits)	10h,11h
1	88-2SIOJP Port 0 (2 stop bits)	10h,11h
2	88-SIO	00h,01h
3	88-ACR	06h,07h
4	88-4PIO Port 0	20h,21h
5	88-PIO	04h,05h
6	88-2SIOJP Port 1 (2 stop bits)	12h,13h
7	CompuPro Interfacer 1, Channel B	02h,03h

(TP 7 is a spare location that can be used for a custom port, with reassembly of Amon.)

I/O Commands

IN <Port> (Input from port)

The specified input port is read, and the result printed on the console. Note that if the ED switch is closed on the 88-2SIOJP then an “IN FF” will disable the AMON EPROM, and the software will most likely crash.

OT <Port> <Data> (Output to port)

The specified data is written to the specified output port.

Other Commands

? Print Help Screen {4K-Byte EPROM only}

Print help for Amon commands. This is a 2-page display: the display will stop after the first page, and will continue after typing any key.

BO (Boot from Altair Floppy Diskette)

This will boot from either an Altair 88-DCDD 8” diskette or from an Altair 88-MDS Minidisk, automatically determining which type of floppy drive is installed. Amon will input from port FFh prior to executing the loaded code, to disable Amon’s PROM, freeing all 65K of memory space for RAM (assuming the ED switch on the 88-2SIOJP is closed).

The floppy disk boot code will retry any sector with a checksum error up to 16 times before giving up. Booting

will terminate and print an error message for any of the following reasons. The error message will include a single-character error code and the 16-bit (4 hex digit) memory address associated with the error.

Error Code	Error Type	Explanation
C	Checksum Error	Sector checksum is incorrect after 16 retries
M	Memory Error	Write to memory failed
O	Overwrite Error	Attempt to overwrite Amon's RAM

HB [<0/1>] (Boot from Altair hard disk)

Boot from Altair Datakeeper hard disk subsystem. 'HB 0' boots from the removable cartridge (default), and 'HB 1' boots from the fixed platter. Amon will input from port FFh prior to executing the loaded code, to disable Amon's PROM, freeing all 65K of memory space for RAM (assuming the ED switch on the 88-2SIOJP is closed).

The hard disk boot code will terminate and print an error message if it gets an error from the Datakeeper disk controller. The error message will contain the 8-bit (2 hex digit) error code from the Datakeeper controller, and the 16-bit (4 hex digit) disk command that caused the error. See the Datakeeper documentation for interpretation of these error components.

TT <0/1> (Set Terminal Type)

TT 0 (or just TT) specifies a terminal that can backspace. TT 1 specifies a terminal (such as a Teletype) that cannot backspace. This command just affects how backspaces that you type are presented. When TT 0 is selected (the default), a backspace or delete key will cause the cursor to back up, erasing the previous character typed. When TT 1 is selected, a backspace or delete key will cause the previous character to be displayed between two slashes, indicating that this character has been deleted.

Entry Points

Amon has four different entry points. You can set up the 88-2SIOJP to jump to any of these at reset, using SW1 and the JS switch. (See the 88-2SIOJP manual.)

F800h: Amon Monitor

Entry at F800h invokes the monitor, as described in the previous section.

FC00h: Boot from Altair Hard Disk (HDBL)

Entry at FC00h boots from the removable cartridge of an Altair Datakeeper hard disk subsystem. Upon successful load, HDBL code will input from the Altair's sense switches (port FFh) prior to executing the loaded code, to disable Amon's PROM, freeing all 65K of memory space for RAM (if the ED switch on the 88-2SIOJP is closed).

If loading from the hard disk fails, an error message will be printed, and control will pass to the monitor. See the HB command for a description of the error messages.

FE00h: Boot from Altair Tape (MBL)

Entry at FE00h boots from either an Altair paper tape or cassette tape. This is exactly the same as invoking MITS's MBL loader PROM. The MBL code will input from the Altair's sense switches (port FFh) to determine the boot device. This input will also disable Amon's PROM, freeing all 65K of memory space for RAM (if the ED switch on the 88-2SIOJP is closed). The boot device is specified by three of the Altair's sense switches, as follows:

A10	A9	A8	Boot Device
0	0	0	88-2SIO Port 0
0	0	1	88-2SIO Port 0
0	1	0	88-SIO
0	1	1	88-ACR
1	0	0	88-4PIO Port 0
1	0	1	88-PIO
1	1	0	88-2SIO Port 1
1	1	1	88-2SIO Port 0 (Custom port)

Because the MBL code reads from the Altair's sense switches prior to loading, the monitor may be disabled when an error is detected. For this reason, if an error is detected then the error code will be printed out continuously on the console and also stored at address 0000. The memory address associated with the error will be stored at addresses 0001 and 0002.

FF00H: BOOT ALTAIR FLOPPY DISK (DBL & MDBL)

Entry at FF00h boots from either an Altair 88-DCDD 8" floppy disk or from an Altair 88-MDS minidisk. This is equivalent to MITS's DBL and MDBL boot PROMs, with the added functionality of automatically detecting which kind of drive is attached. (This is exactly the same as my own CDBL Combo-Disk Boot Loader.) The CDBL code will input from the Altair's sense switches (port FFh) prior to executing the loaded code, to disable Amon's PROM, freeing all 65K of memory space for RAM (if the ED switch on the 88-2SIOJP is closed).

If an error occurs during loading from the floppy disk, then an error message will be printed, and control will pass to the monitor. See the BO command for a description of the error messages.

EXAMPLE: PROGRAMMING AN EPROM WITH A CROMEMCO BYTESAVER

As an example, suppose:

1. We have a Cromemco 8K Bytesaver board, which occupies addresses E000h through FFFFh[1]
2. We have assembled code whose target address is E400h (which is Socket 1 in this 8K Bytesaver)
3. We actually use the Bytesaver's Socket 2 (starting at E800h) for programming EPROMS.[2]
4. We will use 400h bytes of RAM, starting at 1000h, as a buffer
5. We plan to load the hex file via Port 1 of an 88-2SIOJP

Step 1: Select 88-2SIOJP's Port 1 as the Transfer Port

```
>TP 6
```

(You can verify that the Transfer Port is working by using the TE command.)

Step 2: Load the Intel Hex file into the RAM buffer:

The Intel Hex file that was generated by our assembler has address fields starting at E400. The address offset to our buffer is calculated as follows:

```
1000h - E400h = -D400h
```

To create a negative hex number, compliment, and add one:

```
-D400h = 2BFFh+1 = 2C00h
```

[1] An 8K Bytesaver has eight sockets, each of which can read or program a 2708 EPROM.

[2] We might do this because we have a ZIF socket installed in the 8K Bytesaver's Socket 2 (or any other socket).

Load the Intel Hex file with this offset:

```
>HL 2C00

{Send the Intel Hex file to the Transfer port}
```

The file should now be in RAM, starting at 1000h. You can see it using the Memory Dump command:

```
>DU 1000 400
```

Step 3: Program the EPROM

The 8K Bytesaver uses 2708 EPROMs, which have 400h bytes of data, and require 60 (3Ch) programming passes on a Cromemco 8K Bytesaver.

Note that Cromemco recommends removing the Programming Diodes on the 8K Bytesaver, for any EPROM sockets that contain code that you don't want to overwrite accidentally. Make sure that the socket that you plan to use for programming has its Programming Diode installed. (These diodes are just above the sockets, near pin 24 - see the 8K Bytesaver manual.)

To program and verify our EPROM:

1. Insert a blank EPROM in 8K Bytesaver Socket 2
2. Turn on the red programming switch on the 8K Bytesaver
3. Issue a Copy command:

```
>CO 1000 E800 400 3C
```

Programming will take about 35 seconds. When done, the EPROM will be verified, and any mismatches will be reported on the Console.

4. Turn off the red programming switch on the 8K Bytesaver.

Step 4: Move the EPROM to its target socket

Remove the EPROM from Socket 2 and insert it in Socket 1.

Alternatively, we could have just put the EPROM in the 8K Bytesaver's Socket 1 in the first place (assuming that Socket 1 has its Programming Diode installed), and programmed it there:

```
>CO 1000 E400 400 32
```

ALTAIR ABSOLUTE BINARY FILE FORMAT

An Altair 'Absolute Binary file' on tape has up to four sections, which may be separated by any number of nulls. These sections are:

1. The Leader, which comprises 2 or more identical bytes, the value of which is the length of the Checksum Loader. If there is no Checksum Loader then the Leader will be nulls.
2. The (optional) Checksum Loader, which is a program that is normally used to load the subsequent sections. This Loader is written backwards on the tape.
3. Zero or more Load Records, each structured as follows:

   byte 0: Sync byte = 3Ch (identifies a Load Record)

   byte 1: NN = number of data bytes in the Load Record

   byte 2: LL = load address low byte

   byte 3: HH = load address high byte

   bytes 4-NN+3: NN data bytes to store at HHLL, NN>0

   byte NN+4: CC = checksum of bytes 2 through NN+3

4. The GO record, structured as follows

   byte 0: Sync byte = 78H (identifies the GO record)

   byte 1: LL = low byte of go address

   byte 2: HH = high byte of go address

Altair file Leaders and Checksum Loaders are specific to both the version of the particular software and the memory size. For example, the Checksum Loader for 4K Basic 3.2 is different than the Checksum Loader for 8K Basic 3.2, and both the Leader and Checksum Loader for 8K Basic 3.2 are different than those for 8K Basic 4.0.

Amon’s AL command avoids problems with the different Checksum Loaders by ignoring the Checksum Loader on the tape, and loading the Load Records directly.

Note that many Altair programs (such as all versions of Basic) require the front pane Sense Switches to be set to indicate which port to use as the Terminal (console). Note also that these switch settings differ from version to version of Basic. Be sure to check the documentation for the particular version you are loading.