

MIT S Programming Package II

The MIT S Programming Package II allows development of 8080 assembly language applications on the Altair 8800 using just paper tape or cassette for mass storage. The package consists of an editor, assembler, debugger and a monitor that allows execution of these programs as well as the programs you may develop.

To use the programming package, the monitor program is loaded from paper tape or cassette in the same manner as loading BASIC. Once loaded, the monitor is then used to load the editor, assembler, debugger or user programs.

The monitor loads from paper tape or cassette using the same bootstrap loader as would be used for BASIC version 3.2. Sense switch settings are the same with the addition of A9 functionality:

- A9 up – the monitor does NOT use serial input interrupts
- A9 down – the monitor uses serial input interrupts

By enabling interrupts, a Ctrl-C can be typed to return control from a program directly back to the monitor. The programming package assumes the 88-VI/RTC board is present and enables real-time clock interrupts from the board, yet doesn't provide an interrupt handler for the RTC. In order to run with interrupts enabled, interrupts from the RTC on the 88-VI/RTC must be disabled and interrupts from the console serial port (serial port 1) must be set to vector "7." This can be done in firmware versions 1.5 or greater using the "Interrupt Vector Assignment" option in the "Admin" menu of the Configuration Monitor.

Memory Allocation

- The monitor is about 2.5K in length and uses RAM up through 0x0A3F
- The editor is about 2K in length and resides just above the monitor at 0xA40 – 0x11B1, followed by the default 2K edit buffer space from 0x11B2 – 0x19B1.
- Two versions of the assembler are available. Each are about 3K in length. ASM (assembler version 1) loads at the same address as the editor (0xA40). This is inconvenient for the iterative cycle of edit, assemble and test. As an alternative, AM2 (assembler version 2) loads just above the editor at 0x1350 – 0x1D78. The assembler's symbol table grows up from 0x1E06. Note that the default location of the edit buffer conflicts with the load address of AM2 as the edit buffer grows. Before using the editor for longer programs, the location of the edit buffer should be moved to a free area in memory by using the monitor DEP command to patch the buffer start address into octal locations 5124-5125 and the buffer end address+1 into octal locations 5530-5531.
- When using the AM2 assembler, a program loaded at 0x2000 leaves about 500 bytes of symbol table space (0x1E06 – 0x1FFF). This is enough for small demo programs of 75 lines or less. Otherwise, a higher starting address should be used for the target program
- The debugger is about 2K in length and overlays the editor at 0xA40 – 0x133F. The debugger and AM2 assembler can both reside in memory at the same time. This makes it easy to patch programs with the assembler while debugging the program.

Loading the Monitor, Editor and Assembler via Cassette

- 1) If using Tera-Term for the console, set "Serial Port" to 7 bits plus space parity. In the "Keyboard" menu, check the "Backspace key" checkbox under "Transmit DEL by:" Turn on caps lock. Note that delete displays an underscore each time it is pressed as with older versions of BASIC, but underscore does not work as a delete.
- 2) Toggle in the bootstrap loader for the ACR. The bootstrap loaders are listed in the Programming System II manual. The bootstrap loader is the same loader used to load 4K BASIC version 3.2
- 3) Set the front panel switches for the ACR and the console being used (e.g., A15 up for cassette, A11 and A10 up for 2SIO with 1 stop bit). The sense switch settings match those used for loading BASIC version 3.2. Note these are different than used with the 4.0 and newer loaders.
- 4) Raise A9 for no interrupts, leave down for interrupts. (Allows Ctrl-C out of programs back to the monitor).
- 5) Boot the monitor tape. The monitor prompt is two spaces and "?"
- 6) STOP! Do not type any commands to see "what happens." If what is typed is not a command, the monitor tries loading a program of that name from the cassette and hangs there until you provide that program. If the ABS device has been set to audio-cassette (AC), then Ctrl-C will return to the monitor prompt if interrupts were enabled during the boot process. Otherwise, follow the instructions below to restart the monitor from the front panel.
- 7) Assign the program load device to the cassette, type : "OPN ABS,AC<cr>"
- 8) Load the editor, type: "EDT<cr>" and then send the editor .bin file through the tape port. When loaded, the editor's prompt "*" is displayed. Type "E<cr>" to return the monitor.
- 9) Move the editor's buffer location into high memory so the assembler can reside in memory at the same time as the editor. For larger programs, more than the default 2K of buffer space will probably be needed as well. Here are settings for an 8K buffer:
Type "DEP 5124<cr>" and enter "0<cr>" then "100<cr>" then ctrl-z.
This specifies the 16 bit octal address 40000 (0x4000) for buffer start.

Type "DEP 5530<cr>" and enter "0<cr>" then "140<cr>" then ctrl-z.
This specifies the 16 bit octal address 60000 (0x6000) for buffer end.
- 10) Load the assembler, type: "AM2<cr>" and then send the assembler version 2 .bin file through the tape port. When loaded, the assembler prompt is "**ASM*<cr>". Type "EOA<cr>" (end of assembly) to return to the monitor.

The computer is now ready to iteratively edit, assemble and run/test a program.

Using the Editor and Assembler

- 1) To start the editor, type: "EDT<cr>" This starts the editor and clears the edit buffer. To subsequently re-edit a program, type "EDT(R)<cr>" This leaves the existing source code in memory. If the "(R)" is left off, the program source will be erased.
- 2) At the editor prompt type "I" for insert. Ctrl-Z exits the insert mode. "P" prints (displays) the file with line numbers. "W" displays the file without line numbers. "D line[,line]" deletes line(s). "R line" replaces a line. "I line" inserts after the specified line. "I" by itself inserts before the first line. "E" exits the editor and returns control to the monitor.
- 3) End programs with:
 BEG start label
 END program name
 EOA
Where "start label" is the program entry point, "program name" is a three character program name. "EOA" means end of assembly.
- 4) Assign source file input to come from the edit buffer instead of a device and declare file type of ASCII (text file): Type, "OPN FIL,EB,A<cr>" This only has to be done once during a session (or if the FIL device was subsequently assigned to a different device during the session).
- 5) Run the assembler: Type "AM2<cr>"
- 6) Tell the assembler to take input from a file: Type "FILE<cr>" The assembler will run and show errors and undefined symbols. Note: The "Undefined Symbols" heading is displayed even when there are no undefined symbols. Control is returned to the monitor.
- 7) Run the program by jumping to the starting address: Type "JMP xxxxxx" where xxxxxx is the program starting address in octal.
- 8) You can enter the program into the monitor's program table as follows: After a successful assembly, re-enter the assembler and preserve symbols: Type "AM2(P)<cr>"
- 9) Type "RUN name<cr>" where name is the three character name for the program. The program will run. From here on, the program can be run by typing the "name" specified at the monitor prompt. This step does not have to be repeated after subsequent assemblies if the entry address has not changed.

Loading and Saving Files

- 1) Type "NUL 4" to include four nulls after each source file line when writing files.
- 2) Assign the FIL device to the audio cassette and specify ASCII files: Type "OPN FIL,AC,A<cr>"
- 3) In the editor, type "L<cr>" to load a source file. Type "S<cr>" to save a source file.
- 4) After loading a source file into the editor, you can re-assign FIL to the edit buffer for use as the assembler input: Type "OPN FIL,EB,A<cr>"
- 5) When you want to save an edited file after the edit and assemble process, be sure to set the FIL device back to AC before using the "S" command from within the editor.
- 6) Optionally, you can take assembler source file input from the audio cassette. After assigning FIL to the AC (see step 2), in the assembler, type "FILE name<cr>" where "name" is the three character source file name for the assembler to read. The source file is then read directly from the audio cassette instead of the edit buffer.

Assembler Bug

When the AM2 assembler (probably ASM as well) checks for symbol table overflow, a JM instruction is used instead of a JC instruction. This works fine until the ORG or ORR statement for a user's program is 32K or more above the start of the symbol table. The symbol table starts at 0x1E06, so ORG or ORR statements at or above 0x9E06 (117006 octal) generate a symbol table overflow error at the first symbol since the msb of the comparison result (0x9E06 minus 0x1E06) is set.

This can be fixed in AM2 by patching the JM instruction at 0x17AD (13655Q) from 0xFA to 0xDA (372Q to 332Q). This changes the JM opcode to a JC opcode.

Additional Notes

When using the edit buffer as the source for the assembler, the source file must be "rewound" by editing the source file in between successive assemblies.

If using a terminal emulator and serial port instead of a real cassette or paper tape, 4800 baud is the maximum baud rate for loading binary programs from the monitor. When reading source files (into the editor or for input to the assembler), you may need to run at 1200 baud instead. The editor and assembler tend to have problems even with the NUL 4 setting in place.

To restart the monitor, stop the machine, set all switches off except A6 (i.e., set address of octal 100). Raise examine, set front panel switches back the way they were at load time, then depress run.