Micro11

68HC11 Monitor Program

User's Manual

by
Grant Beattie

Micro11 Version 1.03
Manual Revision - May 1995

Introduction

A *monitor* is a simple operating system which provides I/O facilities and the ability to load, run and debug user programs. Micro11 is a monitor program for use with the 68HC11 Development Board. It is resident in a 27128A EPROM and in conjuction with a terminal or personal computer provides the following services:

- Downloading user programs from a PC in either Motorola or Intel formats
- Memory examine/modify facilities (including EEPROM)
- Program run facilities
- Integrated breakpoint, trace and CPU register examine/change window
- CPU vector display and on-board ROM self-test

Micro11 is a menu driven monitor which uses VT-100 (ANSI) command codes to provide a modern user interface over the RS-232 serial link.

Operating Environment

To take full advantage of Micro11's features the following hardware is required as part of the 68HC11 Development Board:

- Micro11 must be located at C000 - FFFF
- Micro11 uses RAM in the range of 7F00 - 7FFF and Micro11 expects to load the stack pointer.
- The HC11 must use a 4.9152MHz crystal
- PA3/OC5 must be connected to XIRQ (only used for trace function)

A terminal or PC running a VT-100 compatible communications program (such as PROCOMM) is also required. Because Micro11 makes extensive use of VT-100 command codes, it is recommended that the communication link be at 4800 baud or higher.

Micro11 Main Menu

All menu choices are made by selecting one of the **highlighted** characters present (usually the first letter of a choice). Address and data input is in HEX. Upper and lower case entry is accepted for both menu and data input. Other characters will be rejected and the user must re-enter the information.

Esc will typically abort the current command and return control back to the menu. **Backspace** is supported for correcting address and data entry.

Upon powering up Micro11 a greeting panel will appear once. Pressing any key will bring up the Main Menu. Pressing hardware RESET will take the user back to the Main Menu (not the greeting). The only way to re-display the opening greeting is by executing the Options/Cold Reset command or by cycling the power to the 68HC11 Development Board.

The Main Menu:

Micro11 - 68HC11 Monitor Program

Main Menu

Download	Motorola or Intel Hex File
Memory	Dump, Examine, Fill, Modify
Options	Cold Boot, Vectors, ROM Test
Run	Program or Subroutine
Window	Trace/Breakpoint Window

Choosing D, M, O, R or W will bring up the appropriate sub-menu. The following pages explain the operation of every Micro11 command. All commands which appear under a given sub-menu are discussed as a group.

Download Menu

Pressing **D** from the Main Menu selects the download sub-menu. User programs in either Motorola S-Record or Intel Hex format may be downloaded from a PC.

Download Menu

Motorola S-Record
Intel Hex File
Default Download File Name
Current filename: 341lab8a.s19

Default Filename (Download/Default)

This feature is only applicable to PC's equipped with recent versions of PCPLUS. When the user requests a file upload on the PC, PCPLUS will examine the current PC screen for some text separated by a period. When it prompts you for a file to upload, it uses the text it found as a default (if it found any). The purpose behind this function on the 68HC11 is to give the user the ability to place text on the screen. Thus when uploading a file repeatedly from a PC, it is only necessary to enter the file name once.

In the example below, the user has entered 341LAB8a.S19. Whatever name you choose, you must have a period. Drive names will be rejected by PCPLUS therefore only 11 characters plus the period are permitted.

Download Menu

Procomm looks on the screen for text
to use as a filename.

Use the form FILENAME.EXT : 341lab8a.S19

Motorola (Download/Motorola)

Motorola S19 files are uploaded from the PC (and downloaded to the 68HC11) using this command. After selecting **M** the following menu appears:

Download Menu

Motorola S-Record
Intel Hex File
Default Download File Name
Current filename: 341lab8a.s19

68HC11 Waiting for File . . .

The 68HC11 will wait forever for the file. Press PgUp on the PC and send the S19 file as a RAW ASCII file. When the transfer is complete, control will return to Micro11. To abort the transfer press RESET on the 68HC11 board or press Esc on the PC. The default filename is optional.

Intel (Download/Intel)

Intel Hex files are uploaded from the PC (and downloaded to the 68HC11) using this command. After selecting **I** the following menu appears:

Download Menu

Motorola S-Record
Intel Hex File
Default Download File Name
Current filename: 341lab8a.hex

68HC11 Waiting for File . . .

The 68HC11 will wait forever for the file. Press PgUp on the PC and send the hex file as a RAW ASCII file. When the transfer is complete, control will return to Micro11. To abort the transfer press RESET on the 68HC11 board or press Esc on the PC. The default filename is optional.

Memory Menu

Pressing **M** from the Main Menu brings up the memory sub-menu:

Memory Menu

Dump Display Memory Block
Examine Individual Locations
Fill/Copy Memory Block
Modify Memory/Register Locations

Dump (Memory/Dump)

Displays 256 bytes of RAM or ROM in hex and ASCII.
After choosing memory dump you will be prompted for a starting address:

Display Memory Block

Enter Starting Address: 0000_

After entering a 4 digit starting address press **Enter**. To view the NEXT 256 locations press the space bar. Pressing any other key returns control to the Main Menu.

Examine (Memory/Examine)

Used to examine individual memory or register locations.
After choosing memory examine you will be prompted for a starting address:

```
Examine Individual Locations

<ADDRESS>     for specific location
<N> <RET>     for next address
<RETURN>      to exit

Address: C000   Value: FC
Address: _
```

After entering a 4 digit address press **Enter**. To view the NEXT location press **N**, **Enter**. To view a different address, simply enter that address. To quit, press **Enter** or **Esc**.

Be aware that examining some types of I/O can have side-effects. Try examining 9000. Since the LED is a "write-only" device, examining it may give unpredictable results depending on whether or not it was decoded using R/W.

Modify (Memory/Modify)

Used to modify individual memory or register locations.
After choosing memory modify you will be prompted for a starting address:

```
Modify Memory/Register Location(s)

<ADDRESS>     for specific location
<N> <RET>     for next address
<RETURN>      to exit

Address: 9000   Value: 21          Done
Address: _
```

After entering a 4 digit address press **Enter** and then enter a 2 digit byte value.
To modify the NEXT location press **N**, **Enter**. To modify a different address, simply enter that address. To quit, press **Enter** or **Esc**.

Modifying EPROM will have no effect.

Fill/Copy Menu

Pressing **F** from the Memory Menu brings up the Fill/Copy sub-sub-menu. Since fill and copy are seldom used they have been given a separate menu underneath the memory sub-menu. Both fill and copy are intended to operate on large blocks of RAM.

Fill/Copy Menu

Fill Memory Block
Copy Memory Block

Fill (Memory/Fill/Fill)

Fills a block of RAM specified by a starting and ending address with a constant hex value.

Fill Memory Block

Enter Starting Address: 0000
Enter Ending Address: 00FF
Fill Value: 12

Done!

After the operation is complete control returns to the Main Menu. The operation can be aborted by pressing **Esc**.

Filling EPROM will have no effect.

Copy (Memory/Fill/Copy)

Copies a block of RAM specified by a starting and ending address to a new (target) location.

Copy Memory Block

Enter Starting Address: 2000
Enter Ending Address: 20FF
Enter Target Address: 2100

Done!

After the operation is complete control returns to the Main Menu. The operation can be aborted by pressing **Esc**.

Options Menu

Several miscellaneous commands have been gathered together under the Options sub-menu:

Options Menu

Cold Reset
Vector Display
EEPROM Operations
Rom Test (CRC)

Cold Reset (Options/Cold Reset)

Reinitializes the monitor program which deletes the default download filename and resets the status of the Trace/Breakpoint function. This serves the same purpose as cycling the power to the 68HC11 except that the contents of ram are not disturbed. The opening greeting is displayed.

Vector Display (Options/Vector Display)

Shows the current target addresses for each of the interrupt vectors. The interrupt vectors are maintained in RAM (except for RESET) so that they may be modified by the user while the Micro11 EPROM is in place. Micro11 uses an illegal opcode (41) for breakpoints and XIRQ for the trace function. The vectors for XIRQ and Illegal Opcode are written to the appropriate locations in ram when the user enters the Trace/Breakpoint window. Therefore if you have written your own XIRQ and/or Illegal Opcode interrupt service routines it is important that you NEVER enter the Trace/Breakpoint facility.

Before ever entering the Trace/Breakpoint Menu:

Vector Display

RESET(7ffe): **C000**

SCI Interrupt	(7FD6): **0000**	SPI Interrupt	(7FD8): **0000**
Pulse I/P Edge	(7FDA): **0000**	Pulse Overflow	(7FDC): **0000**
Timer Overflow	(7FDE): **0000**	O/P Compare 5	(7FE0): **0000**
O/P Compare 4	(7FE2): **0000**	O/P Compare 3	(7FE4): **0000**
O/P Compare 2	(7FE6): **0000**	O/P Compare 1	(7FE8): **0000**
I/P Capture 3	(7FEA): **0000**	I/P Capture 2	(7FEC): **0000**
I/P Capture 1	(7FEE): **0000**	Real Time Int	(7FF0): **0000**
IRQ Interrupt	(7FF2): **0000**	XIRQ Interrupt	(7FF4): **0000**
SWI Interrupt	(7FF6): **0000**	Illegal Opcode	(7FF8): **0000**
COP Watchdog	(7FFA): **0000**	Clock Monitor	(7FFC): **0000**

After entering and leaving Trace/Breakpoint Window:

Vector Display

RESET(7ffe): **C000**

SCI Interrupt	(7FD6): **0000**	SPI Interrupt	(7FD8): **0000**
Pulse I/P Edge	(7FDA): **0000**	Pulse Overflow	(7FDC): **0000**
Timer Overflow	(7FDE): **0000**	O/P Compare 5	(7FE0): **0000**
O/P Compare 4	(7FE2): **0000**	O/P Compare 3	(7FE4): **0000**
O/P Compare 2	(7FE6): **0000**	O/P Compare 1	(7FE8): **0000**
I/P Capture 3	(7FEA): **0000**	I/P Capture 2	(7FEC): **0000**
I/P Capture 1	(7FEE): **0000**	Real Time Int	(7FF0): **0000**
IRQ Interrupt	(7FF2): **0000**	XIRQ Interrupt	(7FF4): **D037**
SWI Interrupt	(7FF6): **0000**	Illegal Opcode	(7FF8): **CFAE**
COP Watchdog	(7FFA): **0000**	Clock Monitor	(7FFC): **0000**

There is a special "hook" that allows your program to run right out of RESET. This is to allow user programs in SRAM to run once immediately from RESET to gain access to the 64~ time protected registers. Briefly, what you must do is place the two-byte address 0x1040 at address 0x7FFE. Following the next RESET, execution will start at 1040 instead of 0xC000. See your instructor for more details.

Rom Test (Options/Rom Test)

The on-board rom test prompts the user for a starting and ending address over which to perform the test. The test is a Cyclic Redundancy Check (Kontron Algorithm).

After accepting the second address the cursor moves down one line and the test proceeds. When running the test over the range C000 - FFFF the test execution time is approximately 10 seconds.

Cyclic Redundancy Check

Enter Starting Address: C000
Enter Ending Address: FFFF
EPROM CRC Checksum: B48C

Press any key to continue

EEPROM Menu

Pressing **E** from the Options Menu brings up the EEPROM Operations sub-sub-menu. Since EEPROM operations are seldom used they have been given a separate menu underneath the Options sub-menu. Also, since EEPROM is "read-mostly" memory, procedures such as writing/erasing EEPROM locations should be carefully considered.

NOTES:

- EEPROM memory consists of B600 - B7FF and the CONFIG byte at 103F.
- **Beware! Modifying the contents of 103F could render your HC11 chip useless!**
- Micro11 uses EEPROM location B7FF to store the user's baud rate. The erased state of this byte (FF) will produce a baud rate of 9600.

EEPROM Menu

Erase EEPROM byte
Bulk erase EEPROM
Write EEPROM byte
Set SCI baud rate

Erase EEPROM byte (Options/EEPROM/Erase)

Used to erase any single EEPROM byte. For best EEPROM life expectancy, Motorola recommends that any byte to be written should be blank first (FF). This command can be used for that purpose. After choosing Erase EEPROM byte, you will be prompted for the address of the byte to erase:

Enter Address to ERASE: B600

Done!

Bulk Erase EEPROM (Options/EEPROM/Bulk)

Used for bulk erasing the entire 512 byte EEPROM array (but not the config byte). After choosing Bulk Erase EEPROM, you will be asked to confirm your intention:

Bulk ERASE all 512 bytes (Y/N)? _

Done!

Write EEPROM byte (Options/EEPROM/Write)

Used to write any single EEPROM byte. For best EEPROM life expectancy, Motorola recommends that any byte to be written should be blank first (FF). The Erase EEPROM byte command can be used for that purpose. After choosing Write EEPROM byte, you will be prompted for the address of the byte and then the value to write:

Enter Address to WRITE: B600
Vaule to Write: 47

Done!

Set SCI baud rate (Options/EEPROM/SCI)

The user's default baud rate is maintained in EEPROM memory location B7FF. The erased state of this location is FF and will result in a baud rate of 9600. The various baud rates are derived from the 4.9152MHz system clock and the various choices are:

SCI Baud Rate Menu

1 . . . 19.2 kbaud
2 . . . 9600 baud
3 . . . 4800 baud
4 . . . 2400 baud
5 . . . 1200 baud

Selection?_

Run Menu

Selecting R from the Main Menu brings up the Run Menu. Complete user programs or individual subroutines can be executed from the Run Menu.

Run Menu

Program Run
Subroutine Run

Program Run (Run/Program)

Complete execution is handed over to the user program. The stack pointer is managed by Micro11 and need not be initialized in the user program. If you intend to use breakpoints do not initialize the stack pointer in your program. Breakpoints will be trapped by Micro11 and control will be turned over to the Trace/Breakpoint Menu.

Program Run

Enter Starting Address: 1040_

Subroutine Run (Run/Subroutine)

Complete execution is handed over to the user subroutine in a manner similar to Run Program. Prior to the subroutine execution, a return address is pushed on the system stack. When the user subroutine executes RTS, the return address is pulled into the PC and control is returned to Micro11.

Subroutine Run

Enter Starting Address: _

Trace/Breakpoint Window

Breakpoint, trace (single-step) and CPU register examine/modify functions have been collected into a common interface called the Trace/Breakpoint Window. Within this window there are two distinct areas. The CPU WINDOW is used for the management of the CPU registers and for controlling program execution (step or run). The BREAKPOINT WINDOW is used for setting and removing breakpoints.

The Trace/Breakpoint Window:

*** 68HC11 CPU WINDOW ***

```
                    TEST
PC=  1040 >         00 00 00 00
S =  7F6A >         00 00 00 00
Y =  1040 >         00 00 00 00
X =  1040 >         00 00 00 00

A =  00 00= B
                     N  Z  V C
     CCR = 00 = 0  0  0  0
```

P,Y,X,A,B ..... Modify CPU Registers
T ..... Trace One Instruction
G ..... Go to next RTS
R ..... Run Program from PC
Esc ..... Exit to Main Menu

N ..... Set Multiple Trace Number
M ..... Execute Multiple Trace
01 ..... Current Multiple Trace Count (hex)

*** BREAKPOINT WINDOW ***

```
Breakpoint #1 = [redacted]
Breakpoint #2 = [redacted]
Breakpoint #3 = [redacted]
Breakpoint #4 = [redacted]
```

Clear Breakpoint
Set Breakpoint

Selection?_

Modifying Registers

Pressing P, X, Y, A, or B will place the cursor in the appropriate CPU register. You must type 2 or 4 digits as required (simply editing the value displayed will not work). There is no way of modifying the contents of the condition code register. In all cases backspace and escape are supported.

Each of the pointer registers has a 5 byte data area associated with it. The data bytes shown are 5 bytes in memory starting at the address specified in the pointer. Note that using a pointer to point to an I/O device can have undesirable side-effects in the CPU window since the I/O device will be read every time the CPU window is redrawn.

If you have unused pointer registers, you can use them to display any memory you wish. Therefore, you can display any ram data without having to exit to Micro11's memory display function.

The contents of the CPU window are refreshed after every program step and can be refreshed by pressing the space bar. Note that the CPU window is NOT refreshed when breakpoints are inserted and deleted.

Tracing One Instruction

To single-step (trace) through one instruction, simply press T. After the instruction is complete, the CPU window will be refreshed. Do NOT attempt to trace through Output Compare 5 routines as both you and Micro11 will be competing for these resources and Micro11 may crash.

Holding T down will cause a continual series of traces.

Important information for advanced programmers:

Starting with Version 1.03, it is possible to trace through programs which have interrupts occuring in the background. An example of this would be a main program that runs concurrently with several Output Compares which are interrupt based. It will be possible to trace through the main program since Micro11 allows for nested interrupts. You should avoid using OC5 for the same reasons given above. Also, very high frequency interrupts may tax Micro11's ability to service all pending interrupts. If your interrupt rate is too high it will appear as though tracing is "stuck" on one instruction.

Tracing More than One Instruction

If you wish to step through larger pieces of code than one instruction, a Multiple Trace Count **N**umber can be specified. After specifying the number (the default is one instruction) executing a **M**ultiple Trace will cause that number of instructions to be executed. Since the number is specified in HEX, the range is 1 to 255 instructions.

The number of steps is always visible in the CPU window as the "Current Multiple Trace Count".

Running a Program

Running a program in the CPU Window is identical to running a program in the Micro11 Run/Program function (see page 13). The only difference is that you will not be prompted for a program counter value. Execution will begin at the PC value visible on the screen. Control is completely turned over to the user program.

Go to next RTS

If you wish to execute to the end of a subroutine you may use the **G** command. Execution will NOT occur in real time since Micro11 inspects every opcode before it is executed to determine if the opcode is RTS.

If the opcode is RTS, control is returned to the CPU window. If it is not, it is executed normally.

Setting Breakpoints

Micro11 supports up to four breakpoints. BREAKPOINTS MUST BE SET AT **OPCODES!** When a breakpoint is set, the target opcode is replaced by an illegal opcode (41). Thus when the CPU attempts to execute the illegal opcode, control is returned to Micro11 AND THE BREAKPOINT IS **REMOVED**. The status of all 4 breakpoints is indicated on the screen.

When a breakpoint is set the CPU Window is NOT redrawn. If you wish to redraw the CPU Window (to look at a breakpoint in memory for instance) press the space bar.

To insert a breakpoint simply press **S** followed by a number (1 to 4) and the address of the target opcode.

Clearing Breakpoints

Since breakpoints are cleared (removed) when they are encountered, one normally does not need to clear them. If you wish to clear a breakpoint, press C followed by the number of the breakpoint you wish to clear. The illegal opcode (41) will be removed and the original opcode restored.

NOTE: If a user breakpoint exists and then the user presses RESET, the breakpoint opcode (41) will be left stranded in the user code. Re-download your program or fix the opcode by hand.