Mini RoboMind Instructions

A first test:

The Mini RoboMind should be ready to go when you receive it. However, you will need a special type of serial cable to connect it to your computer. If you have a programming cable for the 68HC11 or 68HC12, it'll work fine. Otherwise, you can order a kit or an assembled version from Kevin Ross (http://www.nwlink.com/~kevinro/products.html). Start HyperTerminal (or any other ASCII terminal program) and set it to COM1, 9600 baud, 8 data bits, 1 stop bit and no parity. Use Xon/Xoff protocol. Locate the battery cable, plug it into the three-prong connector on the board and attach a nine-volt battery. You should see "Hello World!" on your terminal program and the debugger prompt. You might try modifying this program, compiling it, linking it and downloading it (see below for details) before you write your own program from scratch.

Using a 5-volt supply instead of a 9-volt battery:

If you want to use a regulated 5-volt supply instead of a 9-volt battery, make a power connector with ground in the center and +5 volts in the left socket (reference the board with the power connector at the bottom, looking at the board from above). Move the jumper located just above the power connector from the center and right pins to the center and left pins, then apply power.

Mini RoboMind memory map:

Function	Start Address	End Address	Notes
RAM (8-bit, slow)	$000000[1]	$07FFFF for 32k	$01FFFF-128k $07FFFF-512k
Flash	$080000[2]	$0BFFFF	256K, uses CSBOOT and CS6
A/D 0 to 7	$F00000	$F00007	Uses CS1
LCD	$F00800	$F00801	Uses CS9 and CS10/ECLK[3]
TPU RAM (16-bit, fast)	$FFE800	$FFEFFF	2K
SIM Registers	$FFF000	$FFFFFF	MM = 1
TPU Registers	$FFFE00	$FFFE28	Original TPU, not TPU 2 Mask G
LEDs	$FFFA41		Red = bit5, Green = bit4 Active Low

[1] CPU32bug uses $00000000 to $00002FFF in RAM
[2] CPU32bug uses $00080000 to $0008FFFF in Flash
[3]CS10 uses E-clock generation mode.

Adjustable Clock Speed:

The frequency control bits ($FFFA04 bits 8 - 13) are set to %111111 and the prescalers (W and X) are both set to 0 so the startup clock speed is 32.768 kHz * 256 = 8.388688 MHz. Setting W and X to 1 and Y to 23 (%010111) will result in a 25.16582 MHz clock and setting Y to 15 (%001111) will result in a 16.77722 MHz clock. DO NOT set a clock faster than your CPU is rated for! Also, if you change the clock speed, the SCI baud rate will also change so terminal communications will need to be adjusted.

See the SIM manual for more information on adjusting your clock speed; there is a sequence that should be followed. Also, do not expect to slow the processor down for serial communications; it takes several thousand cycles for the clock to stabilize and there is no way to tell when it's stable (the SLOCK bit doesn't work).

To load the C/C++ compiler on your computer:

1. Drag the "xgcc" folder from the CD-ROM to your C: drive's root directory.
2. Open a DOS prompt and enter the command: attrib -r \xgcc*.* /s
3. Add the following lines to your C:\AUTOEXEC.BAT file:
   set djgpp=c:\djgpp\djgpp.env
   set path=%path%;c:\xgcc\bin
   set path=%path%;c:\xgcc
   set path=%path%;c:\xgcc\68k\2_8_1
4. Run the AUTOEXEC.BAT file (or reboot).

To compile a C/C++ program:

After the compiler is setup, copy the following files to a folder where your program will reside:

RM_CRT0.O
rm_rom.ld
stddef.h

Edit your program however you like and save it with the customary .h, .c or .cpp extensions. To compile your program, type *cc filename.c* from the DOS prompt. To link your program, *type ll destname source1.o source2.o source3.o* and so forth. Two files will be created; destname.txt and destname.s19. Both are identical, download either one to the board.

SBasic:

Karl Lunt has adapted his SBasic compiler for the 68000 family. It can be downloaded from:

http://www.seanet.com/~karllunt/sbasic.htm

(note: the author of this document does not use SBasic and cannot supply directions for using it. If someone who is using SBasic would supply general directions for getting started, I'll include them in this document.)

To download an S19 file:

From CPU332bug, type 'EF' and press "Enter" to erase the flash. From HyperTerminal, select "Transfer" then "Send Text File". Select the file you want to send and it should go. Press "Enter" twice to get back to the debugger.

Port E, F and C:

Port E is not used for bus control on the Mini RoboMind so it is available for I/O. Data can be read or written to either $FFFA11 or $FFFA13 (bits 0 - 7). $FFFA15 (bits 0 - 7) hold the data direction registers (1 = output) and $FFFA17 (bits 0 - 7) holds the pin assignment register, which will be all zeros since the port is not used for bus control. Port F is mapped similarly to Port E, but starting at $FFFA19. Port F, by default, is not configured for interrupts. Port C is configured to be used as chip selects, not general purpose I/O.

Documentation:

Although 68332 manuals are supplied on the CD-ROM for the Mini RoboMind, you may want to have the book versions. The document numbers for the various manuals are:

Book	**Document ID**
MC68332 User's Manual	MC68332UM/AD
Time Processor Unit	TPURM/AD
System Integration Module	SIMRM/AD
CPU32 Central Processor Unit	CPU32RM/AD
General Purpose Timer	GPTRM/AD
Queued Serial Module	QSMRM/AD

They can be ordered free of charge from Motorola. The on-line order form is at:

http://ebus.mot-sps.com/collateral/M951446327730collateral.html