

# CME-555

**Single Board Computer for the Freescale MPC555 Microcontroller**

---

## USERS MANUAL

**Axiom Manufacturing Inc.**

2813 Industrial Lane  
Garland, TX 75041  
972 – 926-9303  
FAX 972-926-6063

email: [Sales@axman.com](mailto:Sales@axman.com)  
web: [www.axman.com](http://www.axman.com)

# CONTENTS

<b>1</b>	<b>Cautionary Notes .....</b>	<b>2</b>
<b>2</b>	<b>Terminology.....</b>	<b>2</b>
<b>3</b>	<b>Features .....</b>	<b>3</b>
<b>4</b>	<b>Getting Started .....</b>	<b>4</b>
4.1	SOFTWARE DEVELOPMENT.....	4
4.2	TERMINAL SETTINGS.....	5
4.3	TUTORIAL .....	5
4.3.1	Using the Debug Monitor.....	5
4.3.2	Programming External Flash Memory .....	6
4.3.3	Programming the MPC555 On-Chip CMFI Flash Memory .....	7
4.4	USING A BDM.....	8
4.4.1	Configuring the MPC555 Registers .....	8
4.4.2	Connecting and Resetting the BDM .....	9
4.4.3	Using the Metrowerks CodeWarrior BDM Software .....	9
4.4.4	Using the SDS Single Step BDM Software .....	10
<b>5</b>	<b>Hardware Specifications .....</b>	<b>10</b>
<b>6</b>	<b>Memory Map .....</b>	<b>11</b>
<b>7</b>	<b>Jumpers and Switches .....</b>	<b>12</b>
7.1	EXTERNAL HARD RESET CONFIGURATION OPTIONS. ....	12
7.1.1	Config Switch .....	12
7.1.2	Mode Switch 1 .....	12
7.1.3	Mode Switch 2 .....	13
7.2	MEMORY DEVICE SELECTION AND CONFIGURATION. ....	14
7.2.1	RAM-SEL Jumper.....	14
7.2.2	FLSH-SEL Jumper .....	14
7.2.3	M-SEL Jumper.....	14
7.2.4	MEM_OPT Jumpers.....	15
7.2.5	MEM_VOLT Option Jumper .....	16
7.3	LCD-PORT AND KEYPAD / KEY-PORT JUMPERS.....	16
7.3.1	CS3 Base Memory Map.....	16
7.3.2	LCD Display CONTRAST .....	16
7.3.3	LCD_PORT Power Polarity Select – JP10.....	16
7.3.4	KEY-PORT and LCD_PORT Enable – JP11 .....	16
7.3.5	KEY-PORT Interrupt Enable – JP12 .....	16
7.4	MPC555 OPTION JUMPERS .....	17
7.4.1	Cut-away Jumpers.....	17
7.5	COM-1 AND COM-2 SERIAL PORT OPTIONS.....	17
7.5.1	JP5 – TPUA Channel 15 to COM_1 .....	17
7.5.2	JP6 – TPUB Channel 15 to COM_2 .....	17
7.5.3	JP7 – TPUA Channel 14 to COM_1 .....	17
7.5.4	JP8 – SCI2 RXD to COM_2 .....	17
7.5.5	JP9 – TPUB Channel 14 to COM_2 .....	17
7.6	USER LED INDICATORS, SWITCHES, AND POTENTIOMETER.....	18
	<b>TROUBLESHOOTING.....</b>	<b>19</b>

## 1 Cautionary Notes

- 1) Electrostatic Discharge (ESD) prevention measures should be applied whenever handling this product. ESD damage is not a warranty repair item.
- 2) Axiom Manufacturing reserves the right to make changes without further notice to any products to improve reliability, function or design. Axiom Manufacturing does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under patent rights or the rights of others.
- 3) EMC Information on the CME-0555 board:
  - a) When indicated, this product as shipped from the factory with associated power supplies and cables, has been tested and meets with requirements of CE and the FCC as a **CLASS A** product.
  - b) This product is designed and intended for use as a development platform for hardware or software in an educational / professional laboratory or as a component in a larger system.
  - c) In a domestic environment this product may cause radio interference in which case the user may be required to take adequate prevention measures.
  - d) Attaching additional wiring to this product or modifying the products operation from the factory default as shipped may effect its performance and also cause interference with other apparatus in the immediate vicinity. If such interference is detected, suitable mitigating measures should be taken.

## 2 Terminology

This development board applies option selection jumpers. Terminology for application of the option jumpers is as follows:

Jumper on, in, or installed = jumper is a plastic shunt that fits across 2 pins and the shunt is installed so that the 2 pins are connected with the shunt.

Jumper off, out, or idle = jumper or shunt is installed so that only 1 pin holds the shunt, no 2 pins are connected, or jumper is removed. It is recommended that the jumpers be placed idle by installing on 1 pin so they will not be lost.

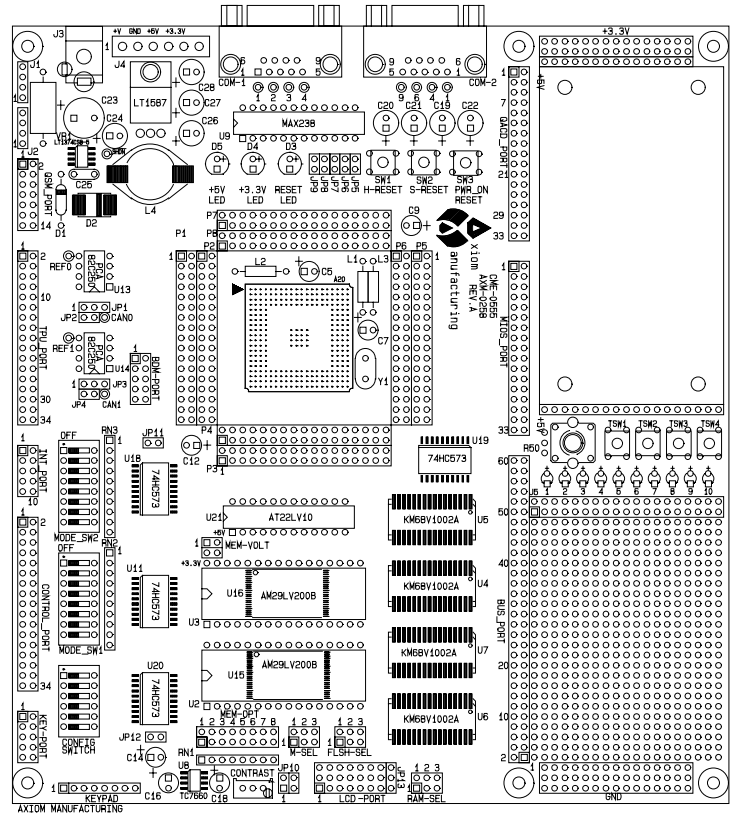
This development board applies option selections that require a soldering tool to install or remove. This type connection places an equivalent Jumper Installed type option. Applying the connection can be performed by installing a 0 ohm resistor component or small wire between the option pads. See the Option sections for more details.

### 3 Features

The CME0555 is a fully configured development system for the Freescale MPC555 PowerPC Microcontroller. The system provides all the features and memory to develop expanded or single-chip MPC555 applications. Provided with Axiom Utility Monitor operating in the external EPROM that provides a serial connection. Development board provides external 128K x 32 Fast SRAM (512K bytes) and 128K x 32 Flash EPROM (512K bytes) for application support. Kit includes serial cable, 9V US Wall Power Adapter, printed hardware manual, MPC5xx support CD with example source codes, GNU C compiler, utility support software, and technical manuals.

#### Features:

- MPC555 with Floating Point Unit
  - 40MHz operation w/ 4Mhz crystal reference
  - 28K bytes Ram
  - 448K byte Flash
  - 2x TPU
  - 2x QADC
  - 2x CAN
  - 2x SCI
  - SPI
  - PWM and timers
- Standard fixed memory:
  - 128K x 32 Fast SRAM (12ns)
  - 128K x 32 Flash EPROM (120ns)
- Two Configurable 32pin memory sockets for
  - 32K to 2MByte PROM and 32K to 512KByte SRAM
- 32 Bit Bus support with 8 bit memory bank control
- COM1 - SCI1 w/ RS232 type DB9-S Connection
- COM2 - SCI2 w/ RS232 type DB9-P Connection
- 2 TouCAN Ports w/ transceivers (PCA82C250)
- LCD Interface Connectors w/ Contrast Adjust, Memory Mapped (80 or 160 character)
- Keypad Interface Connector, 16 Key
- 60 pin BUS\_PORT with 32 data and 24 address
- 34 pin CONTROL\_PORT Connector with Bus Controls
- 14 pin QSM\_PORT with serial port I/O
- 34 pin MIOS\_PORT with PWM and parallel I/O
- 34 pin TPU\_PORT with Timer Module I/O
- 34 pin QADC\_PORT with Analog I/O
- 10 pin INT\_PORT with Interrupt I/O
- Power\_Port with Power supply access
- BDM Port - Debug Development connector
- All I/O in .1 grid to maximize use
- Large Prototyping and breadboardAreas
- 6 to 25VDC Input to 5V @ 3A Switching Supply and 3.3V @ 1.5A Power Supply
- Typical Operating Power: 400ma @ 40MHz
- Easy configuration with Mode Switches and Option Jumpers
- Size: 7 x 7.5 inch.



The Axiom development system provides for low cost software testing with the use of the Monitor in the on-board EPROM. Operation allows the user to load code in the On-Board RAM or Flash, Execute application, and display or modify registers or memory. After code is operational the user may relocate the code and reprogram the MPC555 Internal Flash EEPROM or the fixed flash for dedicated operation of new software. No additional hardware or software is required to operate. For high level debug, extensive tools are available for the PowerPC core and the Debug Port is available to connect a background debugger.

## 4 Getting Started

See the **README.TXT** file on the CD provided with this board for all the documentation and support files provided.

This section assumes that you have just received your board from the manufacturer. If this is not the case then jumpers and switches may have been changed so that the board may not function as expected. In this case, see the "Jumpers and Switches" section of this manual and return everything to "default" positions before proceeding. Also make sure the switches are set to work properly with the EPROM's installed. MODE\_SW1 position 3 must be ON, MODE\_SW2 position 1 must be OFF and M-SEL must be on 1.

To get started quickly, perform the following test now to make sure the board is working correctly:

1. Connect one end of the supplied 9-pin serial cable to a free COM port on your PC. Connect the other end of the cable to the COM-1 port on the board.
2. Run a standard ANSI terminal communications program set to 9600 baud, N,8,1. Any terminal program will work, including the simple terminal that comes with MS Windows. The AXIDE terminal software is provided on the support CD. This software will provide an easy method to communicate and upload files to the CME0555 board. AXIDE does not provide a target setting for the CME0555 board and only operates as a terminal with this target.
3. Apply power to the board by plugging in the wall plug power supply transformer that came with the system.
4. If everything is working properly, should see the main utilities menu, similar to the following:

```
-----  
AXIOM CME-555 UTILITIES      2.0  
-----
```

1. Debug Monitor
2. On-Chip CMFI Flash Programming
3. External Flash Programming
4. Test Hardware

Select:

Your board is now ready to use! If you do not see the monitor prompt, press then release the RESET button on the board. If still no go, or if the text is garbage, see the **TROUBLESHOOTING** section of this manual.

### 4.1 Software Development

The example monitor software initializes the clock to run this board at 40 MHz on power-up. The default MPC555 clock will be 20Mhz at Reset. You can set the clock by changing the PLL Register in your software.

Software development on the CME555 is best performed using a BDM tool connected to the BDM-PORT connector. This provides real-time access to all hardware, peripherals and memory on the board. BDM software is also available for high-level source code debugging. Contact the manufacturer for more information.

If a BDM is not available, software development can be done using the included simple monitor program to upload your code to ram and execute it or program it into the EPROM's.

In either case, software is usually uploaded to RAM and executed during development, then programmed into FLASH EEPROM to execute when power is applied.

## 4.2 Terminal Settings

The utility software provided uses the serial port connected to your PC running a terminal program as described under "Getting Started". While you can use almost any commercial or free serial communications terminal interface, it is important that you configure it to work properly with this board. The following settings are important:

1. BPS: 9600, Data Bits 8, Parity none, Stop Bits 1
2. Flow Control should be set to HARDWARE or NONE.
3. Emulation should be ANSI
4. When uploading files to the board, be sure to send in Text mode. AXIDE software will expect text files for uploads.
5. When loading or programming FLASH memory, you must apply ONE of the following:
  - a. 2ms line pace delay after each line for RAM loading.
  - b. 80ms line pace delay after each line for internal MPC555 flash or 10ms for external flash.
  - c. Wait for echo character following each line, AXIDE UPLOAD "wait for reply character" option.

## 4.3 Tutorial

The following brief tutorial was created to help you become familiar with the software development process on this board quickly. For more information, consult the online documentation of your BDM software.

### 4.3.1 Using the Debug Monitor

This board ships with a simple monitor utility programmed into the U2 and U3 EPROM devices. See "Getting Started" for startup instructions. Choose menu item 1 to launch the monitor.

Using this program and any serial communications terminal program, you can display and modify memory and registers on the board from any PC. This is done by typing commands into the terminal program on the PC. Type help and hit <enter> for a complete list of available commands.

You can experiment with some of the commands like reading (MD) and modifying (MM) memory. Make sure you're modifying memory that is mapped to valid addresses or the monitor program will "hang" or throw an exception. You may then have to RESET the board to get the monitor back. Internal RAM starts at 0x3F9800. External RAM starts at 0x800000 (as shipped with the RAM-SEL jumper set to 2). The monitor program itself uses internal RAM from 0x3F9800 - 0x3F9FFF, so if you modify anything in this area while the monitor is running it may malfunction.

You can also load and execute a program from memory using this utility. A simple "hello world" program is provided for you on the CD as an example. Follow these steps to load and execute it from internal RAM using the monitor program:

1. At the monitor prompt > type **DL** and hit <enter>.
2. Select the send text file (or upload) command from your terminal program and locate the file named "**HELLO\_R.S19**" included on the support CD. Send this file to the board.
3. When finished uploading, type **go 3FA004** at the prompt. You should see the phrase "Hello World!" echoed back to the screen, which is all that this simple test program does, each time you hit a key.
4. To exit from this test program, hit the RESET button on the board.

This example program was compiled with the Diab Data compiler. The source files and scripts used are included on the support CD as follows:

HELLO.C	Main source code file
SERIAL.C	Serial port I/O and ASCII text conversion routines
HELLO_R.DLD	Diab Linker script - specifies the programs memory organization
HELLO_R.BAT	DOS batch file used to build the program

### 4.3.2 Programming External Flash Memory

After testing your program running out of RAM you will probably want to program it into EEPROM so that it starts whenever power is applied to the board. To do this you must first change the starting address of your program to match that of the memory device being programmed. This is usually done in assembly language using the ORG statement, or by MAK or Project file arguments to the LINKER if using a compiler.

As an example, a version of the "hello world" program is included on the support CD that has been relocated to start at address 0x400000. This will be the starting address assigned to external onboard flash memory, once we move some jumpers and reset the board.

This version is called **"HELLO\_F"**. It uses the same HELLO.C and SERIAL.C source code as the RAM version, with the following unique files:

INIT.S	Example Boot initialization assembly source file
HELLO_F.DLD	Diab Linker script - specifies the programs memory organization
HELLO_F.BAT	DOS batch file used to build the program

The output files produced from this build are called HELLO\_F.S19 and HELLO\_F.ELF.

Follow these steps to program the "hello world" example to start on power-up instead of the monitor program:

1. Make sure the M-SEL jumper is on 1, FLSH-SEL jumper is on 3 and RAM-SEL jumper is on 2. This will configure the board to match the default Memory Map.
2. Make sure MODE\_SW1 switch 3 is ON and press RESET if you do not see the utilities menu (see Getting Started).
3. Select External Flash Programming from the menu.
4. Select **2** for External Flash (U15/U16) - CS2.
5. Select **E** to Erase the Flash Memory. This should take less than 10 seconds.
6. Select **P** to Program from a file -> Flash CS. This will prompt you to upload the file to be programmed.
7. Select the send text file (or upload) command from your terminal program and locate the file named **"HELLO\_F.S19"** included on the support CD and send this file to the board.
8. After all lines from the file have been programmed, the words programmed are displayed along with any errors, followed by the menu again. Now the "hello world" program is at address 0xC00000 (the current location of onboard flash), however it is linked to be run from 0x400000, since this is the address the program assigns to the CS0 base register at runtime.  
  
**Explanation:** Since the monitor EEPROMS in U2/3 use the same Chip Select 0 that we wish to use for the "hello world" program (by the setting of the M-SEL jumper), we must do an offset program into a different chip select. Setting the FLSH-SEL jumper to 3 configures the onboard flash to Chip Select 2. The monitor program will offset the program to that chip select base address (0xC00000) temporarily during programming.
9. Remove power from the board and move the MODE\_SW1 switch 3 to the OFF position. This will configure the board to 32-bit mode, which is required for the onboard flash memory to boot.
10. Remove the M-SEL jumper and move the FLSH-SEL jumper to position 1.
11. Re-apply power to the board. You should see the "hello world" message on your terminal screen every time you press a key.
12. To return to the Utilities Menu, remove power, move the MODE\_SW1 switch 3 back to ON, move the M-SEL jumper back to 1 and the FLSH-SEL jumper back to 3. Reapply power and you should see the menu again.

Although this "Hello World" example is a simple program, you can use the same procedure for programming your own application into external flash memory using the CME555 development board.

To develop more complex applications, with source level debugging, you will no doubt require a debugger tool. An excellent choice would be a Background Debug Module (BDM) which is supported using the BDM-PORT on this board. See the "Using a BDM" section of this manual for more information.

### 4.3.3 Programming the MPC555 On-Chip CMFI Flash Memory

You can also write your program to the MPC555 internal flash memory so that it starts whenever power is applied to the board. To use this memory, the starting address of your program should be 0x100, which is the MPC555 reset vector. You must also initialize ALL CPU registers that you will be using, since your program will be running the show from power-up.

A version of the "Hello World" program which does this for you is provided on the support CD. This version is called "**HELLO\_A**". It uses the same HELLO.C and SERIAL.C source code as the RAM version, with the following unique files:

INITC.S	Example Boot initialization assembly source file
HELLO_A.DLD	Diab Linker script - specifies the programs memory organization
HELLO_A.BAT	DOS batch file used to build the program

The output files produced from this build are called HELLO\_A.S19 and HELLO\_A.ELF. To program the internal flash memory, do the following:

1. Config Switch positions 4 and 5 should both be ON.
2. If a BDM is connected to the board - disconnected it.
3. Reset the board to get the Utilities menu (see Getting Started for more information).
4. Select On-Chip CMFI Flash Programming.
5. Set Config Switch position 6 - ON to apply programming voltage.
6. Select **E** to erase the internal flash memory
7. Select **H** to program the Hard Reset Configuration Word. Press **C** to confirm this. The value 0x00006000 will be written, which will allow this chip to boot on this board.
8. Select **P** to program the flash. Send the file named **HELLO\_A.S19** to the board using your terminal program (be sure to send in text mode if given the option). This CMFI programming software also needs at least 15ms pacing delays between each line sent, or optionally you can have your terminal program wait for an echo character before sending the next line, since the utilities echo a '.' character after each line is programmed.
9. The programming utility will program each data word as it is received. When finished, you will be returned to the programming menu. **NOTE** - if you have trouble here, see the previous section "Terminal Settings".
10. **IMPORTANT** – Reset the Config Switch position 6 back OFF to protect the MPC555 Flash.

To make your new program start on power-up or reset, simply move switch 1 on the MODE\_SW2 bank to the ON position. This will enable the internal flash memory and if you cycle power or press RESET, you should see the "Hello World" message every time you hit a key.

To return to the Utilities software, flip MODE\_SW2 switch 1 back OFF and press RESET.

Following this procedure you can program any application into the MPC555 internal flash memory.



## 4.4 Using a BDM

### 4.4.1 Configuring the MPC555 Registers

A BDM is a real-time debugger that allows you to download to and modify both internal and external memory and peripheral registers on the board from a completely user controlled state. The BDM software writes to a number of MPC555 configuration registers before your application software is loaded, after it resets and establishes initial communication with the board.. The values that these registers are set to is the source of most BDM debugging problems.

BDM software saves these initialization values in a configuration file. The SDS software uses a file with the CFG extension under its \CMD directory. For this board, the file name is "MPC555.CFG". If you make any changes to these registers under the SDS Debug window, they are saved in this file.

The Metrowerks CodeWarrior BDM software uses a "BDM/JTAG Configuration File" listed under "EPPC Target Settings" for register configuration. This also is a text file and can be modified in a text editor.

In addition to modifying register values in these configuration files, some BDM programs provide a convenient interface you can use to modify individual registers. You should become familiar with this method since it provides a quick and easy way to make a change to a single bit value then test the results of the change.

The following table lists the recommended default values of the MPC555 configuration registers to debug and execute software on the CME555 development board with a BDM. If you have trouble loading or executing the example software, you should first check these values against the current settings in your BDM software. The memory configuration below assumes the CME555 board is set to 32-bit mode (MODE\_SW1 #3 OFF), the RAM\_SEL jumper is on 1, the FLSH-SEL jumper is on 3 and the M-SEL jumper is removed. Note that these are not the same jumper setting as the board is shipped (with 16-bit M-SEL booting the monitor on external EPROM) but rather this configuration would be a valid setting for debugging out of external RAM for example.

ADDRESS	NAME	VALUE
SPR	638	0xFFFF00800
SPR	560	0x00000000
0x2fc000	SIUMCR	0x00000000
0x2fc004	SYPCR	0x0000FF88
0x2fc100	BR0	0x00400003
0x2fc104	OR0	0xffff80012
0x2fc108	BR1	0x00800003
0x2fc10c	OR1	0xffff80002
0x2fc110	BR2	0x00c00003
0x2fc114	OR2	0xfffe0022
0x2fc118	BR3	0x01000403
0x2fc11c	OR3	0xfffe0FF1
0x2fc140	DMBR	0x00000000
0x2fc144	DMOR	0x00000000

ADDRESS	NAME	VALUE
0x2fc280	SCCR	0x00010000
0x2fc284	PLPRCR	0x00400000
0x300000	DPT	0x0000
0x300004	RAM	0xfffa0
0x305014	PORTQS	0x0000
0x305016	PQSPAR/DDRQS	0x0000
0x306100	MPIOSMDR	0x0000
0x306102	MPIOSMDDR	0x0000
0x306800	MPIOS1TPCR	0x0000
0x380000	SRAMMCR	0x0000
0x2fc024	SGPIODT1	0x00000000
0x2fc028	SGPIODT2	0x00000000
0x2fc02c	SGPIOCR	0x00000000
0x2fc030	EMCR	0x00000000
0x307f80	UMCR	0x00000000

#### 4.4.2 Connecting and Resetting the BDM

The CME555 has been tested thoroughly using a Macraigor Systems Inc. "wiggler" BDM. Since this development board simply brings out the BDM Port connector from the MPC555 controller, any Freescale MPC BDM standard device should work.

You should connect the BDM to the board in the following order. If the BDM module is powered and connected to the board before the board is powered on, the microcontroller may not reset properly. The following instructions assume you are using a parallel type BDM.

- Connect the supplied parallel cable to your PC
- Connect the parallel cable to the BDM
- Apply power to the BDM
- Apply power to the CME555 development board.
- Connect the BDM to the BDM\_PORT on the board, which is located near the microcontroller. Pin 1 of the BDM\_PORT connector must line up with the RED Stripe along the side of the BDM cable.
- Press then release the PWR\_ON RESET button. The red RESET LED should flash ON briefly, indicating that the board reset properly.

You should now be able to communicate with the CME555 using compatible BDM software. The following sections describe methods used to debug a simple program using two BDM interface programs the manufacturer has tested with this board.

#### 4.4.3 Using the Metrowerks CodeWarrior BDM Software

The Metrowerks debugger software can be executed from within the CodeWarrior compiler IDE after you load your software project. You must configure the CodeWarrior IDE to work with your BDM and the CME555 development board. First, load your Project file and compile it. Be sure you use valid memory addresses for this board, set under the Linker settings. Change the project settings for the debugger as follows:

1. Set the Target Processor to Freescale 555
2. Set the Protocol to BDM - MSI Wiggler, or to whatever your BDM module supports.
3. Change the BDM/JTAG Configuration File to the ax555ini.txt file included on the CME555 support CD. This is simply a text file listing of register address that the BDM will modify and their values, described previously in this manual.
4. Change the device to Wiggler (or other device you have) and the Parallel port to match that of your PC
5. The other options can be left as is. If you have trouble connecting, change the Launch Action property to "Connect". Also increasing the Speed selection can help. Never make this 0 however. Verify the BDM port on your pc and the board. See the Metrowerks debugger documentation for further help.

When finished, select the Save button and close the project options window. Enable the debugger under the Project window then select Debug to launch the debugger. This will connect to the BDM, write the register values in the configuration file and download your program to the board.

You now should see the debugger window, which contains source, disassembly, stack and variables. There are commands available to single step, breakpoint, execute, etc. under the Debug menu. The final test is to execute and test your program. If you configured the software to just "Connect" as the Launch Action, it will break at the start of the Main() function, in which case you will have to select Run again.

See the Metrowerks CodeWarrior documentation for more information.

#### 4.4.4 Using the SDS Single Step BDM Software

When the SDS Single Step software starts, you should see the debug status window. All the settings in this window are automatically saved when you exit the program. Follow these steps to configure the software to debug the "hello world" program used in the "Using the Debug Monitor" section of this manual.

1. Select the **Connection** tab and make sure the settings match those used to connect the BDM to your PC. Set the delay to 0, unless you're having connection problems.
2. Select the **Processor** tab and choose MPC555 with no coprocessor.
3. Select the **Target Configuration** tab. The Category should be General, the Processor should be set to MPC555. Each of the MPC555 registers listed in this window are set by the BDM whenever it connects to the development board. It is important that these values match your hardware/software configuration in order to debug your hardware/software. If you're not sure what these registers must be set to, double-click on each of them and verify they match with the tables on the previous page.

**NOTE:** A file is included on the CD called "MPC555.CFG" that has these setting already entered. If you copy it to the Single Step software \CMD directory it will use those setting instead. Unfortunately, this file will be overwritten if you change anything in the Target Configuration window.

4. Select the **File** tab. The "debug without a file" box should be unchecked. Select Browse and locate the file named "HELLO\_R.ELF" that was included on the CME555 support CD.
5. Press the OK button at the bottom of the debug window.

The software will first create its own debugging files based on the ELF file and try to write them to your hard drive. It will then attempt to connect to the development board through the BDM. If this does not work, disconnect the BDM and power from the board, re-apply power to the board, press the PWR\_ON RESET button, then re-connect the BDM to the board. You can get the debug window back from the File menu.

Once connection with the board is established, the software will start downloading the program onto the board. When finished downloading, it will attempt to execute it until the main() function is reached. You can disable this option under the Options tab. When it is finished, close the debug status window.

You should now see the HELLO\_R program in the debug window, along with other useful information. You can single step, examine and modify memory, registers, etc. The final test is to execute the program. If you bring up a terminal program in the background and press the Go button (green light) it will send the hello message to terminal screen (assuming you have the serial cable connected).

See the SDS Single Step documentation for more information.

## 5 Hardware Specifications

<b>Oscillator</b>	4.0 MHz Reference Crystal
<b>Clock</b>	40 MHz Maximum, 20MHz default
<b>Operating temperature</b>	0°C to +70°C
<b>Power requirement</b>	7 - 20V @ 120 ma Typical

## 6 Memory Map

Following is the **Default** memory map for this board as configured by the monitor utilities in U2/U3 EPROMS. The internal map is detailed further in the MPC555 Users Manual. Chip Selects 0-2 can be changed by your software to map the external memory any way you like, however CS3 is dedicated to the peripherals (Keypad / LCD).

### Internal Memory

0000 0000	CMFI Internal Flash Memory Array
0006 FFFF	
0007 0000	
002F BFFF	Reserved
002F C000	
003F 97FF	On-chip Control and Status Registers (see MPC555 Documentation)
003F 9800	
003F FFFF	Internal SRAM

### External Memory

0040 0000	40 0000 - 40 FFFF for 64K device 40 0000 - 47 FFFF for 512K device	<b>CS0</b> - External EPROM - 16 bit M-SEL devices U2-U3 (shipped with monitor program here)
007F FFFF		
0080 0000	<b>CS1</b> - External SRAM - 32 bit RAM-SEL devices U4-U7 512K bytes at 80 0000 - 87 FFFF on this board	
00BF FFFF		
00C0 0000	<b>CS2</b> - External Flash Memory - 32 bit FLSH-SEL devices U15-U16 512K bytes at C0 0000 - C7 FFFF on this board  these are located on the board underneath the external EPROM	
00FF FFFF		
0100 0000	<b>CS3</b> - External Peripheral Devices (see the CS3 map under KEYPAD / LCD section of this manual for details)	
013F FFFF		
0140 0000	Un-Mapped Address Space	
FFFF FFFF		

## 7 Jumpers and Switches

### 7.1 External Hard Reset Configuration Options.

**Mode Switch 1, Mode Switch 2** and the **Config Switch** provide the external Hard Reset Configuration Word when enabled. The switches provide a logic 0 or low level when off and a logic 1 or high level when on. The configuration is only presented to the data bus during Hard Reset if External Configuration is enabled with Config Switch 4. The Hard Reset Configuration Word options not set by Mode Switches 1 or 2 are all defaulted to logic 0 by pull-down resistors R24 through R31. In the unlikely event that user operation requires modification of one of the defaulted low levels, a jumper wire to +3.3V can be installed on the bottom of the board. For more information on the MPC555 Reset configuration refer to Chapter 7 in the MPC555 technical manual.

#### 7.1.1 Config Switch

Position	Reset Function	Default Position
1	MODCK1 – clock select	Off
2	MODCK2 – clock select = 4 MHz Oscillator	On
3	MODCK3 – clock select	Off
4	External Hard Reset Word enable	On
5	EPEE signal High – on-chip program enable	Off
6	VPP power enabled	Off

**Config Switch 4** ON enables the External Reset Configuration Word. When enabled, a Hard Reset signal will enable the External Reset Configuration Word from Mode Switch 1 and 2 on to the Data Bus for input by the MPC555. Switching this OFF allows alternate configuration methods.

**Config Switch 5** ON makes the EPEE signal valid high which is necessary to program the on-chip CMF flash.

The MPC555 Vpp supply to the internal Flash EPROM Memory module is connected to a circuit supplying +3.3VDC for normal operation. **Config Switch 6** ON provides Vpp supply connection to +5V DC for programming the internal Flash EPROM Memory module. This switch should remain OFF when not programming to protect the internal Flash EPROM.

#### 7.1.2 Mode Switch 1

Position	Reset Function	Default Position
1	Initial interrupt prefix (IP), on = 1	Off
2	External Boot (BDIS), Disabled = 1	Off
3	Boot Port size (BPS0), on = 16-bit boot port, off = 32-bit	On
4	Debug pins configuration (DBGCO) * see note 1	Off
5	External Bus division factor (EBDF1), on = clock / 2	Off
6	Single Chip Mode Select (SC0:1) * see note 2	Off
7	Single Chip Mode Select (SC0:1) * see note 2	Off
8	Exception table relocation enable (ETRE), on = relocated	Off

Note:

- Switch 3 off makes VFLS0, VFLS1, BI\*, BG\*, BR\*, BB\* signals active.  
Switch 3 on makes VFLS0, VFLS1, STS\*, VF0, VF1, VF2 signals active.
- 

6	7	
Off	Off	Extended 32 bit data bus
Off	On	Extended 16 bit data bus
On	Off	Single chip with show cycles ( not recommended on CME0555 board)
On	On	Single chip with no external bus

**7.1.3 Mode Switch 2**

Position	Reset Function	Default
1	Internal Flash enable (FLEN), Off = internal flash removed from memory map	Off
2	Enable Compression (EN_COMP), On = compression on (MPC556 only)	Off
3	Exception Compression Enable (EN_COMP)	Off
4	Reserved, do not enable.	Off
5	Internal register space select (ISB) off:off:off = 0:0:0, register space at 0x00000000	Off
6		Off
7		Off
8	Dual mapping enable (DME), Off = dual mapping disabled	Off

Notes for Manual for External Hard reset word options with Mode switches 1 and 2

- EARB bit is forced low in External Hard Reset Word, internal arbitration.
- BDRV bit is forced low in external hardreset word, Bus Drive = full strength.
- Boot Port size of 8 bits is not available on the CME-0555.
- DBG0 bit 1 is forced to low in the external hard reset word, Debug pins IWP0/VFLS0 and IWP1/VFLS1 are only available as VFLS0 and VFLS1. See MODE SW1 position 4 for other Debug pin options.
- DBPC bit is forced low in the external hard reset word, DSDI, DSDO, and DSCK debug signals enabled.
- ATWC bit in External Hard Reset Word is forced low, WE0-3/BE0-3 signals enabled.
- EBDF bit 0 in the external hard reset configuration word is forced low, see MODE Switch 1 position 5.
- PRPM bit in the external hard reset word is forced low, Peripheral Mode should not be used on the CME-0555 board.

## 7.2 Memory Device Selection and Configuration.

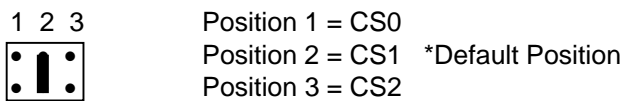
The CME0555 board provides three external memory banks that provide:

- 128K x 32bit Fast Static Ram (U4 - U7)
- 128K x 32bit Flash EPROM (U15 - U16)
- Memory Option Sockets for 16bit wide memory (U2 - U3)

Each memory bank can be configured individually to operate from the MPC555 chip selects CS0, CS1 or CS2. Caution should be used not to place more than one memory bank on the same chip select.

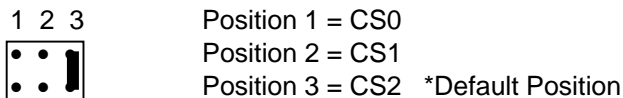
### 7.2.1 RAM-SEL Jumper

The RAM-SEL jumper selects which chip select accesses the on-board Fast Static Ram memory bank (U4 - U7). If no jumper is installed then the memory bank is disabled and idle. This memory bank is 128K x 32 bits (512K bytes) and operates with zero wait states. This memory bank may be useful for emulating program operation prior to programming into internal or external flash memory. Absolute addressing of the memory bank is determined by programming the chip select register.



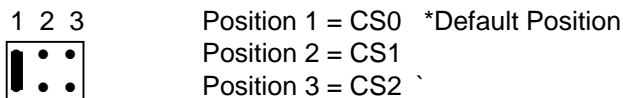
### 7.2.2 FLISH-SEL Jumper

The FLISH-SEL jumper selects which chip select accesses the on-board Flash EPROM memory bank (U15 - U16). If no jumper is installed then the memory bank is disabled and idle. This memory bank is 128K x 32 bits (512K bytes) and operates with 1 wait state at 20MHz. The memory bank is available to the user to provide additional program code or data space and optional Boot start memory space. Absolute addressing of the memory bank is determined by programming the chip select register.



### 7.2.3 M-SEL Jumper

The MEM-SEL jumper selects which chip select accesses the on-board Option Memory EPROM bank (U2 - U3). If no jumper is installed then the memory bank is disabled and idle. This memory bank is 16 bits wide and must have the same device type installed in both sockets. Devices can be optioned up to 256KBytes each or 512KBytes total memory space. The memory bank may be used for Boot Start program code from Power On Reset for the board. If used for Boot start program, Mode Switch 1 position 3 should be ON to select 16 bit wide bus for boot start. The board ships with a simple Monitor program in EPROM installed in these sockets. Absolute addressing of the memory bank is determined by programming the chip select register.

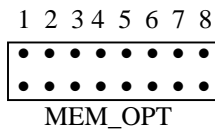


#### NOTE:

- CS0 is the external Boot Memory device
- CS3 is used by the Keypad and LCD devices

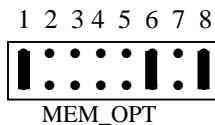
## 7.2.4 MEM\_OPT Jumpers

The MEM\_OPT jumper block selects the **TYPE** of devices installed in the 32 pin U2 and U3 optional memory sockets. RAM, Flash, EEPROM, and EPROM may be installed. Following is a list of the jumper signal options and a chart showing standard jumper positions for typical devices. Note that U2/3 device pins without an option jumper installed are pulled to a high level. The option jumpers are setup in pairs for device pins so that **Only One** jumper is installed for each pair 1/2, 3/4, 5/6, 7/8, for a maximum of 4 jumpers.



- Position 1 = A16 to U2/3 pin 3
- Position 2 = A15 to U2/3 pin 3
- Position 3 = A15 to U2/3 pin 31
- Position 4 = WE0 to U2/3 pin 31 \* removed for write protection
- Position 5 = A16 to U2 pin 29
- Position 6 = WE0 to U2 pin 29 \* removed for write protection
- Position 7 = A16 to U3 pin 29
- Position 8 = WE1 to U3 pin 29 \* removed for write protection

### 32K x 8 Standard Low Power SRAM and EEPROM Jumper Options

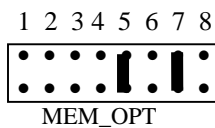


Standard device types: 62256, 55257, 28256

Manufacturer device types: KM62256, TC55257, AT28C256, and others

**Note:** Positions 6 and 8 can be removed to write protect devices

### 32K x 8 Standard EPROM Jumper Options



Standard device type: 27256

Manufacturer device types: NM27C256 and others

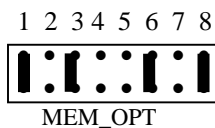
### 64K x 8 Standard EPROM Jumper Options



Standard device type: 27512

Manufacturer device types: NM27C512 and others

### 128K x 8 Standard Low Power SRAM and EEPROM Jumper Options

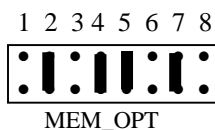


Standard device types: 681000, 551001, 28010

Manufacturer device types: KM681000, TC551001, AT28C010 and others

**Note:** Positions 6 and 8 can be removed to write protect devices

### 256K x 8 Standard Flash EPROM Jumper Options

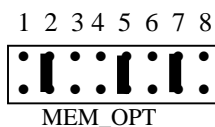


Standard device type: 29010

Manufacturer device types: AM29F010 and others

**Note:** 1. Position 4 can be removed to write protect device.  
2. Devices must be written with 16 bit half word data for proper alignment.

### 256K x 8 Standard EPROM Jumper Options



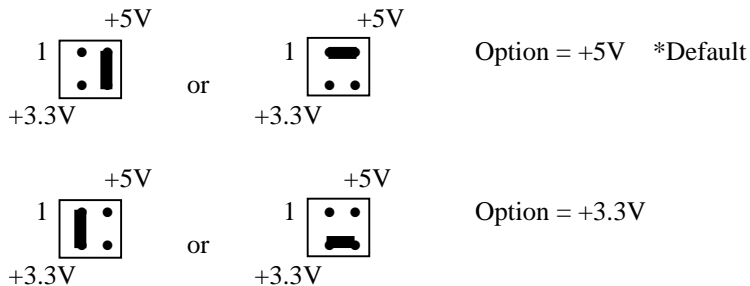
Standard device type: 27C010

Manufacturer device types: NM27C010 and others



### 7.2.5 MEM\_VOLT Option Jumper

The Memory Voltage Option jumper allows selection of 5V or 3.3V for Vcc to the memory devices installed in the U2 and U3 MEM\_OPT sockets. The jumper is set to the voltage required for the correct operation of the devices installed. **Only one jumper can be installed for this option!**



## 7.3 LCD-PORT and KEYPAD / KEY-PORT Jumpers

The LCD Port provides a versatile connector to attach 80 character display modules, 160 character display modules, and some graphics display modules with embedded drivers. The Programmable Logic Device U21 provides address decode and LCD select control signals for the LCD and Keypad ports. The 2 ports are controlled by chip select CS3 when enabled by JP12.

**USE CAUTION:** when connecting your LCD to the LCD-PORT - make sure the power polarity (JP10) and signals match the end of the cable from the LCD module.

The memory map offset from the CS3 base address is as follows:

### 7.3.1 CS3 Base Memory Map

0x0000	LCD_PORT, Command Register 1 for up to 80 character display.
0x0001	LCD_PORT, Display Data Register 1 for up to 80 character display.
0x0002	LCD_PORT, Command Register 2 for graphic or 160 character display.
0x0003	LCD_PORT, Display Data Register 2 for graphic or 160 character display.
0x0004	Keypad / KEY_PORT, key data . Read Only.
0x0005	Keypad / KEY_PORT, key data . Read Only.
0x0006-	
0xFFFF	not used

### 7.3.2 LCD Display CONTRAST

The contrast adjustment allows a contrast Vee voltage to be presented to the LCD\_PORT of -5V to +5V DC.

### 7.3.3 LCD\_PORT Power Polarity Select – JP10

JP10 allows selection of the display power pin polarity on the LCD\_PORT. Depending on the type and location of your IDC connector on the Display Module, the power connections may need to be reversed. Care should be used to verify proper connection and signal matching at the IDC Cable Connector and LCD\_PORT.

### 7.3.4 KEY-PORT and LCD\_PORT Enable – JP11

JP11 when installed enables both the KEY-PORT's and LCD\_PORT for use on chip select CS3. Removing the jumper will disable and idle the KEY-PORT's and LCD\_PORT, and will allow CS3 to be used for other purposes.

### 7.3.5 KEY-PORT Interrupt Enable – JP12

JP12 installed provides IRQ4 to be used as a Key input interrupt service. Removing JP12 disables the key input use of IRQ4. This jumper should be removed with JP11 as it is possible that Key input can still occur even though the KEY\_PORT is removed from the memory map.

## **7.4 MPC555 Option Jumpers**

### **7.4.1 Cut-away Jumpers**

Cut-away jumpers are provided for rarely used changes to the MPC555 module power sources and to allow for easy attachment of a modification wire from the new source. Each cut-away is numbered on the board and in the schematic. Following is a description of what each cut-away provides:

- 1 Isolate MPC555 VDDSRAM (J4) from +3.3V supply.
- 2 Isolate MPC555 VRL (A7) from the common Ground (Vss).
- 3 Isolate MPC555 VRH (B7) from +5V supply.
- 4 Isolate MPC555 KAPWR from +3.3V supply.

**Note:** VRH and VRL signals can be found on the QADC\_PORT connector also. Care should be taken so that an external reference applied to VRH or VRL, if required, is isolated from the main supplies by cutting the associated cut-away.

## **7.5 COM-1 and COM-2 Serial Port Options**

COM-1 and COM-2 provide serial connection to the MPC555 internal serial ports SCI1 and SCI2 respectfully. SCI1 is connected to the U9 RS232 level translator and SCI2 can be optioned to attach to that translator. For additional Transmit / Receive serial channels or RTS / CTS flow control signals, MPC555 TPUA channels 14 and 15 and TPUB channels 14 and 15 can be applied individually. Following are the jumper options for the various connections:

### **7.5.1 JP5 – TPUA Channel 15 to COM\_1**

JP5 installed allows TPUA channel 15 to be used as either the COM-1 RTS signal or as an additional serial channel presented at COM\_1 DB9S connector pin 8.

### **7.5.2 JP6 – TPUB Channel 15 to COM\_2**

JP6 installed allows TPUB channel 15 to be used as either the COM-2 RTS signal or as an additional serial channel presented at COM-2 DB9P connector pin 7.

### **7.5.3 JP7 – TPUA Channel 14 to COM\_1**

JP7 installed allows TPUA channel 14 to be used as either the COM-1 CTS signal or as an additional serial channel presented at COM-1 DB9S connector pin 7.

### **7.5.4 JP8 – SCI2 RXD to COM\_2**

JP8 installed allows SCI2 RXD to be received from the COM-2 DB9P connector pin 2.

### **7.5.5 JP9 – TPUB Channel 14 to COM\_2**

JP9 installed allows TPUB channel 14 to be used as either the COM-2 CTS signal or as an additional serial channel presented at COM-2 DB9P connector pin 8.

Outgoing RS-232 RTS handshake lines to both COM-1 and COM-2 are defaulted to active levels without associated jumper options installed. This will prevent loss of communication due to an unknown level on a flow control signal. COM-1 and COM-2 provide default flow control signaling returned to the host. These connection on pins 1, 4, and 6 can be isolated and the associated tie points can provide different signals via the DB9 connectors.

## 7.6 User LED Indicators, Switches, and Potentiometer

User LED indicators 1 – 10, Test Switches SW1 – 4, and Potentiometer R50 are provided in the CME0555 prototyping area for optional use during development. Option header J5 provides fixed connections to certain I/O ports of the MPC555 or access to the user components by other signals. To apply other signals to the to the user components, the jumper options must be removed on J5 or possible I/O port conflicts may occur.

### J5 Option Header:

The J5 option header provides fixed connections to MPC555 I/O ports or if the option shunt is removed, access for other signals to drive the user components. MPC555 I/O port signals on J5 pins 1, 3, 5, ... are connected to associated User component on J5 pins 2, 4, 6, ... respectfully by option jumper shunts when the board is delivered form Axiom. J5 pin 1 is located in the lower left corner of the J5 header when viewed similar to the feature drawing.

### J5 Connections:

MPC555 Port	J5 Odd Pin	J5 Even Pin	User Component
QADC - VRL	1	2	R50 POT -
QADC_A – AN55	3	4	R50 POT OUT
QADC – VRH	5	6	R50 POT +
A_TPUCH2	7	8	LED 1
A_TPUCH3	9	10	LED 2
A_TPUCH4	11	12	LED 3
A_TPUCH5	13	14	LED 4
A_TPUCH6	15	16	LED 5
A_TPUCH7	17	18	LED 6
A_TPUCH8	19	20	LED 7
A_TPUCH9	21	22	LED 8
A_TPUCH11	23	24	LED 9
A_TPUCH12	25	26	LED 10
B_TPUCH2	27	28	SW 1
B_TPUCH3	29	30	SW 2
B_TPUCH4	31	32	SW 3
B_TPUCH5	33	34	SW 4
A_TPUCH0	35	36	MPC555 B_TPUCH0
A_TPUCH1	37	38	MPC555 B_TPUCH1

### R50 Potentiometer Operation

R50 fully counterclockwise will set output to J5 pin 2 value. Fully clockwise will set output value to J5 pin 6 value. R50 positions between fully clockwise or counterclockwise will output a value between the J5 pin 2 and 6 values. With the J5 options for pin 1 – 6 installed, a voltage of 0 to 5V will be presented to the MPC555 QADC AN55 analog input channel.

### SW1, 2, 3, and 4 Operation

User Switches 1 – 4 outputs on J5 pins 28, 30, 32, and 34 respectfully provide a pull-up value of +5V or logic 1 while the switches are idle. Depressing a switch will cause its respective output value to go to 0 volts or logic 0. With J5 option installed for the user switches, the respective MPC555 B\_TPU channels can be used to input the switch logic values.

### LED Indicator 1 – 10 Operation

User indicators LED1 – 10 inputs on J5 pins 8, 10, 12, 14, 16, 18, 20, 22, 24, and 26 respectfully provide a visible indication when a voltage of 3.3 to 5V or logic high is applied. With J5 option installed for the Led indicators, the MPC555 A\_TPU channel outputs when logic high will enable the respective LED indicator.

**Refer to CME0555 (AXM0258) Schematic sheet 3 for connection diagram.**

## TROUBLESHOOTING

1. If trying to boot from external EPROM devices (U2 and U3) be sure the following are all true:
  - a) CONFIG switch 2 and 4 must be ON
  - b) MODE\_SW1 switch 3 must be ON
  - c) MODE\_SW2 switch 1 must be OFF
  - d) The M-SEL jumper should be set to 1 (cs0). The FLSH-SEL and RAM-SEL jumpers must NOT be set to 1
2. If trying to boot from onboard FLASH devices (U15 and U16) be sure the following are all true:
  - a) CONFIG switch 2 and 4 must be ON
  - b) MODE\_SW1 switch 3 must be OFF
  - c) MODE\_SW2 switch 1 must be OFF
  - d) The FLSH-SEL jumper should be set to 1 (cs0). The M-SEL and RAM-SEL jumpers must NOT be set to 1
3. BDM hardware can cause a small amount of power to be supplied to the micro while connected, and thus prevent the processor from resetting properly. For this reason, you may need to disconnect the BDM device before applying power to the board. The RESET LED should flash briefly when the PWR\_ON RESET switch is pressed and released - indicating a proper reset condition. The BDM can then be connected to the board and should communicate properly.
4. On CME-555 boards with a PB-0555 piggyback device, make sure it is properly seated by pressing down firmly on the center of the microcontroller.
5. Be certain that your software is located at a correct address corresponding to the device you're loading it into. The chip select jumpers on the board (M-SEL, FLSH-SEL and RAM-SEL) should be set to match the corresponding chip select registers (BR0-3, OR0-3) in your software.
6. The external EPROM devices (U2 and U3) are 16-bit devices and any software on them should be configured as such. For example, the 32-bit version of the sample monitor program writes 0x400003 to the CS0 Base address register, while the 16-bit version writes 0x400803.
7. When connecting a serial cable to the board, make sure you're using a "strait-through" serial cable (such as the one provided).
8. If connecting your PC to the COM-2 port on the board, you will need to use a NULL MODEM serial adapter for serial I/O to work properly.
9. If you get an exception error or if your program "hangs" or appears to jump out into unused memory, it is usually caused by trying to access "un-mapped" or improperly configured memory addresses. Look at the Memory Map page of this manual for the default memory map as set by the monitor program. Memory device location and address range is all configured under your software control at run-time (or the BDM configuration registers if using a BDM). These registers include the BR0 - BR3 registers as well as the OR0 - OR3 registers. Consult the MPC555 manual for more information on these registers.
10. On power-up, all memory addresses are mapped to addresses 0x00000000 - 0x00400000, and mirrored at each successive 0x00400000 memory blocks. If booting externally (from CS0) your software must be located at a valid address when it changes the BR0 register. For example: The MPC555 resets at address 0x0100. The monitor starts at the first 40 0000 block following the internal block (see memory map). When the MPC555 resets, the code in the CS0 device - in this case the monitor - is mirrored to every 40 0000 block in the map. This allows the monitor to jump to the \_\_start address + 40 0000. The monitor initialization code then re-maps CS0 to address 40 0000 and everything works ok.
11. If you want to re-map one of the external memory devices (flash for example) to the 0x0000 address space you may have an address conflict with the internal CMFI flash memory. To make this work you must disable the onchip flash (see the FLEN bit of the IMMR register in the MPC555 documentation). You can then relocate your code with interrupt handlers at 0x0, running out of external memory. Writing 0xFFFF0000 to the SPR638 register in your BDM configuration file does this so you can debug external memory mapped at the 0x0 address space.