02/11/05

PB-0555

Freescale MPC555 PowerPC Microcontroller Board

xiom
anufacturing
™

2813 Industrial Ln. 2813 Industrial Ln. • Garland, TX 75041 • (972) 926-9303
FAX (972) 926-6063
email: Sales@axman.com • web: www.axman.com

CONTENTS

Features

The PB-0555 is a Freescale MPC555 PowerPC Microcontroller module with I/O pin headers, oscillator circuit, power supply, and RS232 COM port. The MPC555 high integration processor architecture has been designed for high-performance and low-cost embedded control applications. A version of the PB-0555 can be used on the CME0555 Development Board, in your own application hardware or as a single chip stand alone controller.

Board Features:

- Simple Monitor and CMFI Programmer in MPC555 Flash (self protected)
- Single Chip Operation
- All I/O and Bus to Header Connectors
- 4MHz Reference Crystal Oscillator
- RS232 Level Serial Port
- +7-20V Input Power Supply
- Prototype Area

MPC555 Features:

- Power PC core with Floating Point Unit Processor
- 40MHz Operation, software selectable
- 448K Bytes Flash EEPROM on chip
- 26K Bytes SRAM on chip with battery backup Supply support
- 5V External Interface I/O system with
  - 24 address lines, 32 data lines,
  - 4 programmable chip selects
- Queued Serial Multi-Channel Module with
  - 2 SCI UART channels w/ 16Byte FIFOs,
  - Queued Serial Peripheral Interface
- 2 TPU3 Time Processor Units with
  - 16 Channels with I/O per Unit
  - 9 Pre-programmed functions
- 2 Queued 10Bit Analog to Digital Converters w/Up to 16 input channels each,
  - 10us conversion time, 64 result registers,
  - Selectable modes and data type
- 2 Controller Area Network (CAN2.0B) Modules w/
  - TouCAN protocol 2.0 A and B,
  - 16 message buffers with masking,
  - Time-stamping and Bus Wake-up
- 18 Channel Modular I/O System (MIOS1) w/
  - 8 PWM Output Channels,
  - Two 16Bit Counter Modules,
  - 2 General Purpose 8 bit I/O Ports,
  - 10 Double Action Submodules
- Background Debug Module and JTAG
- DC to 40 MHz Operation
- Low Voltage Operation (3.3V)

GETTING STARTED

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

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 port connector 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. See Terminal Settings in this manual for more information.
3. Apply power to the board by plugging in the 9V power transformer wall plug that came with the system.
4. If everything is working properly, you should see the utilities menu on in your terminal, similar to the following:

```
   ---------------------------------
   AXIOM PB-555 UTILITIES        1.0
   ---------------------------------

1. Debug Monitor
2. On-Chip CMF Flash Programming

Select:
```

Your board is now ready to use. If you do not see the menu, be sure the PROG_EN jumper is installed then press and release the RESET button on the board.

Software Development

Utility software is programmed into the first 2 blocks of internal CMFI flash memory of the MPC555 chip on the board. This software initializes the clock to run this board at 20 MHz on power-up. The board has been tested and does run at speeds up to 40 MHz however, which you can set by changing the PLL Register in your software.

Software development on the PB-555 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 package is not available, software development can be done using the included simple monitor utility to upload your code to ram and execute it during development, then programmed into the CMFI flash to test (see memory map).

TUTORIAL

The following tutorial sections were created to help you become familiar with the utilities and support software provided with this board.

Using the Monitor Utility

This board ships with a simple debug monitor utility programmed into the first 64K bytes of the CMFI flash memory. See "Getting Started" for startup instructions.

Using this monitor program and any serial communications terminal, you can display and modify memory and registers from any PC, as well as upload and execute programs. 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 modify only user ram or register memory that is mapped to valid addresses or the monitor program will "hang" or throw an exception. If this happens simply press RESET to get the monitor back. Internal SRAM starts at 0x3F9800, but the monitor uses the first 800 hex bytes, so address 0x3FA000 - 0x3FFFFF are available for you to use under the monitor, along with DPTRAM.

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

1. Select 1 on the main menu. 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 software disk. Send this file to the board.

3. If your terminal is set to Echo characters sent, you should see each line of the file sent. The last line of the file contains the START address of the hello program, which is 3FA004.

4. 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.

5. 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 utility disk 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

Using a BDM

A BDM real-time debugger allows you to download to and modify memory and peripheral registers on the board in a completely user controlled state. You must configure the BDM software correctly to recognize this board. In your BDM software set the target processor setting to MPC555.

The BDM software writes to a number of CPU registers before your application software is loaded, after it resets and establishes initial communication with the board. The values these registers are set to is the source of many BDM debugging problems.

BDM software saves these initialization values in a configuration file. The SDS software, for example, uses a file with the CFG extension under it's program \CMD directory (MPC555.CFG for this board). You can change the values written to these registers by manually editing this file or in the SDS Debug window, under the Target Configuration tab.

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

Example configuration scripts for BDM software using this board are included on the CD under the \BDM directory.

If you have trouble loading or executing the example software, you should first check these values against the current settings in your BDM software. Remove the BDM from the board then apply power to the board. Now try to re-connect the BDM to the BDM-PORT connector. Be sure the red stripe on the bdm pod cable lines up with pin 1 on the bdm connector.

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
3. Emulation should be ANSI
4. When uploading files to the board, be sure to send in Text mode.
5. When programming CMFI memory, you must use ONE of the following:
    a. 15 millisecond pace delay after each line
    b. wait for echo character following each line

Programming the CMFI Flash Memory

You can write your application program to the MPC555 internal flash memory so that it starts whenever power is applied to the board. The starting address of your program should be 0x10100, which is the reset vector remapped by the utilities software. You must also initialize ALL of the CPU registers that you will be using, since your program will be running the show from power-up.

A version of the example "Hello World" program which does this for you is provided on the utilities disk. This version is called HELLO_A. It uses the same HELLO.C and SERIAL.C source code, but uses the following unique files:

INITA.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 flash memory, do the following:

1. The EX-CFG jumper should be OFF and PROG_EN should 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 2. for On-Chip programming

5. Install the VPP_EN jumper to apply programming voltage.

6. Select E. to erase the user space in flash memory

7. 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).

8. 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".

9. IMPORTANT -- Remove the VPP_EN jumper to protect the MPC555 Flash.

To test your program, select T from the menu. You should see the "Hello World!" message each time you hit a key. To make your program start on power-up or reset, simply remove the PROG_EN jumper and press reset or cycle power. You should see the program running instead of the utilities menu.

To return to the Utilities software, restore the PROG_EN jumper and press RESET.

Although this is a simple example, you can use the same procedure for programming your own application using the utilities software.

MEMORY MAP

Following is the memory map for the PB-0555 board as configured by the utility software installed in internal memory:

0000 0000	**Monitor Utilities** – CMFI flash memory debug monitor and CMFI programming tools	64K
0001 0000	**User Program Memory** – CMFI flash memory can be used for non-volatile program and data storage The utilities also re-map interrupt vectors here starting at 0x10100 for the RESET vector and continue to 0x12000	380k
0007 0000	unused	
002F C000	Control and Status Registers see MPC555 Users Manual - Appendix A	
0030 2000	**User RAM** – DPTRAM Array see MPC555 Users Manual - Section 18	6K
0030 3800	Control and Status Registers see MPC555 Users Manual - Appendix A	
003F 9800	**Monitor Utilities RAM** – SRAM Array used by monitor and CMFI programming tools	2K
003F A000	**User RAM** – SRAM Array see MPC555 Users Manual - Section 20	24K
0040 0000	unused	

Hardware Specifications

Oscillator	4.0 MHz Crystal
Clock Maximum Operation	40 MHz Maximum
Operating temperature	0°C to +70°C
Power requirement	8 - 25V @ 120 ma Typical

Jumper Options

PROG-EN (Auto Start) Jumper

The PROG-EN jumper provides 2 functions: it provides write protection to the CMFI flash memory banks when removed and is used by the utilities to auto-start user applications. With the jumper installed (default) the MPC555 CMFI flash memory can be programmed by the utilities. Removing the jumper will disable MPC555 CMFI flash programming. The jumper OFF position also allows user applications programmed into the flash to be started on RESET or Power-up. See "Programming the CMFI Flash Memory" for more information.

This jumper is ON by default.

EX_CFG Jumper

Enables / Disables External Reset Configuration. When installed this jumper will ground the RSTCONF line, forcing the MPC555 to read the Reset Configuration from the Data Bus instead of the internal configuration word. Removing this jumper allows the MPC555 to default to the internal Reset configuration word programmed in the CMFI Flash memory (0x00006000).

This jumper is OFF by default.

VPP_EN Jumper

The MPC555 Vpp supply to the internal CMFI Flash Memory module is connected to a circuit supplying +3.3VDC for normal operation. VPP_EN installed provides Vpp supply connection to +5V DC for programming the internal CMFI Flash Memory module. This jumper should be removed when not programming to protect the internal flash memory during power cycles.

This jumper is OFF by default.