Memec 3SMB MicroBlaze LCD Reference Design

July 10, 2006
Version 8.1

Overview

This document describes a simple reference design that uses the Xilinx MicroBlaze™ soft processor core running at 66 MHz along with processor peripheral cores to provide interfaces to the I/O devices located on the Memec Spartan-3 MB MicroBlaze development board. The MicroBlaze development board I/O devices used in this application consist of a UART, four User LEDs, and a 2x16 LCD character display.

Block Diagram

The following figure shows a high-level block diagram for this reference design followed by a brief description of each sub-section. The design consists of:

- MicroBlaze soft processor core
- One GPIO Output Ports (used to drive 4 User LEDs)
- Two GPIO Input Ports (used to read the on-board 8-position DIP Switch and two push-button switches)
- One UART Ports (RS232)

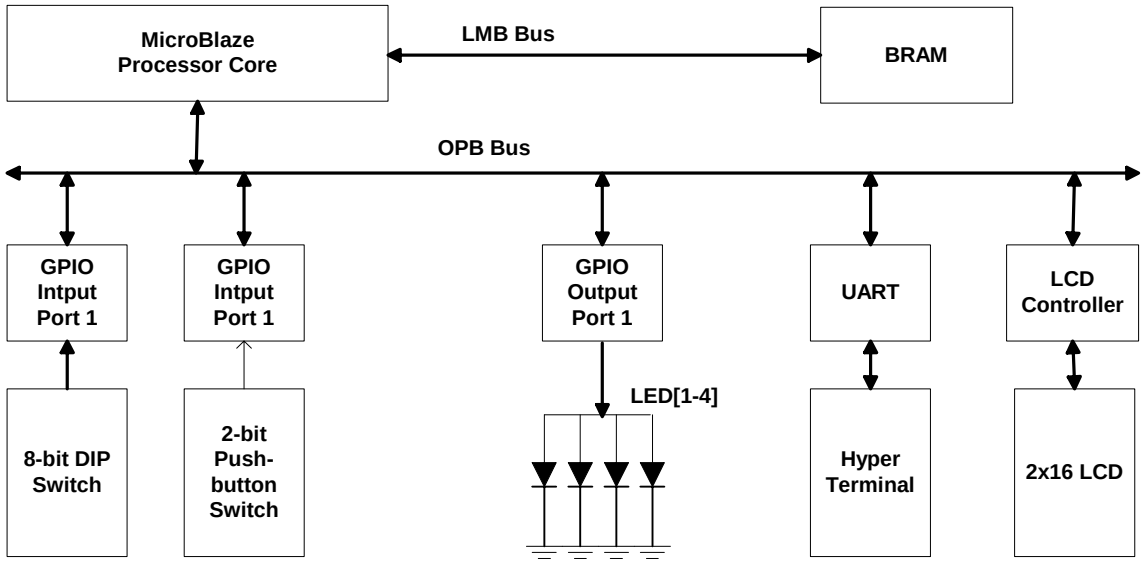

Figure 1 - LCD Design Block Diagram

Memory Map

The MicroBlaze processor provides 4G of address space that can be used to access memory and I/O devices that reside on the processor bus. For this reference design, the

I/O devices located on the MicroBlaze development board and the processor memory are memory mapped as shown in Figure 2.

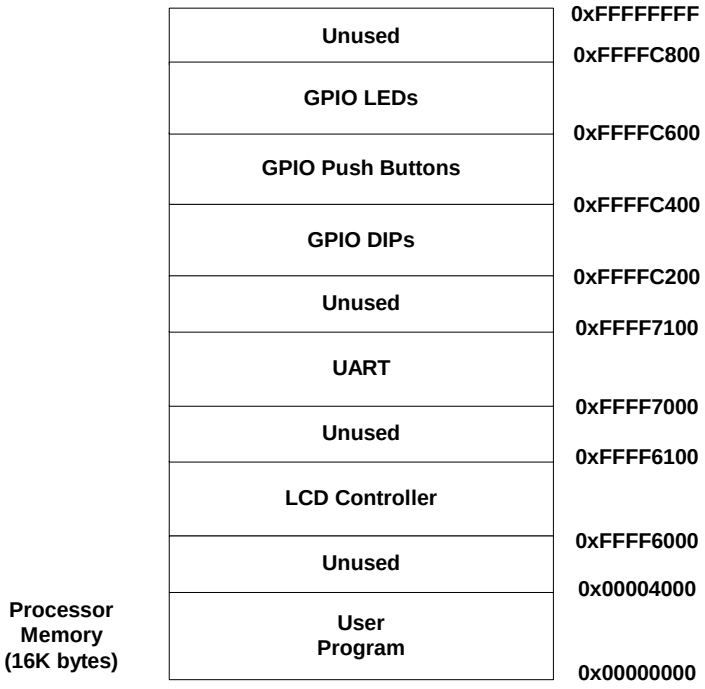

Figure 2 - Memory Map

Data and Code Memory

The data and code for this application will reside in the on-chip BLOCKRAM. The Spartan-3 MB MicroBlaze development board is designed with the XC3S1500-4FG676C Xilinx FPGA. The XC3S1500 contains 32 BLOCKRAMs at 2Kbytes per block. A XC3S1500 MicroBlaze design can have a maximum of 64Kbytes on-chip processor memory. This reference design uses 8 BLOCKRAMs to provide 16Kbytes of memory to the processor.

GPIO Leds

This 4-bit output port is used to drive four User LEDs. These four LEDs are accessed by writing a value to the memory space 0xFFFFC600 – 0xFFFFC7FF. For this reference design, the user program rights a sequencing patter to the LEDs.

GPIO DIPs

This 8-bit input port is used to read the 8-position DIP Switch located on the board. The DIP Switch is accessed by reading a byte from the memory space 0xFFFFC200 – 0xFFFFC3FF. The user program does not use the DIPs.

GPIO Push Buttons

This 2-bit input port is used to read a two push-button switches located on the board. The push-button switch is accessed by reading a byte from the memory space 0xFFFFC600 – 0xFFFFC7FF. The user program does not use the DIPs.

RS232 UART Port

The RS232 UART port for this reference design is used as the STDIN/STDOUT port. The RS232 UART is accessed by writing to or reading from the memory space 0xFFFF7000 – 0xFFFF70FF.

LCD Controller

The LCD Controller sends several different messages to the LCD Display.

Experiment Setup

Software

The recommended software setup for this reference design is:

- Windows2000 or WindowsXP
- Xilinx ISE 8.1i (Foundation or BaseX) with latest Service Pack[1]
- Xilinx EDK 8.1 with latest Service Pack[1]

Hardware

The hardware setup used by this reference design includes:

- Computer with a recommended minimum of 1GB RAM and 1 GB Virtual Memory[2]
- Memec Spartan-3 MB Development Kit
- Platform USB Cable or JTAG Programming Cable IV
- Serial Cable

[1] Latest Service Packs are available at www.support.xilinx.com/swupdate
[2] Refer to the *ISE 8.1i Release Notes and Installation Guide* http://toolbox.xilinx.com/docsan/xilinx8/books/docs/irn/irn.pdf

3S1500MB Board Setup

Refer to Figure 3 for jumper locations.

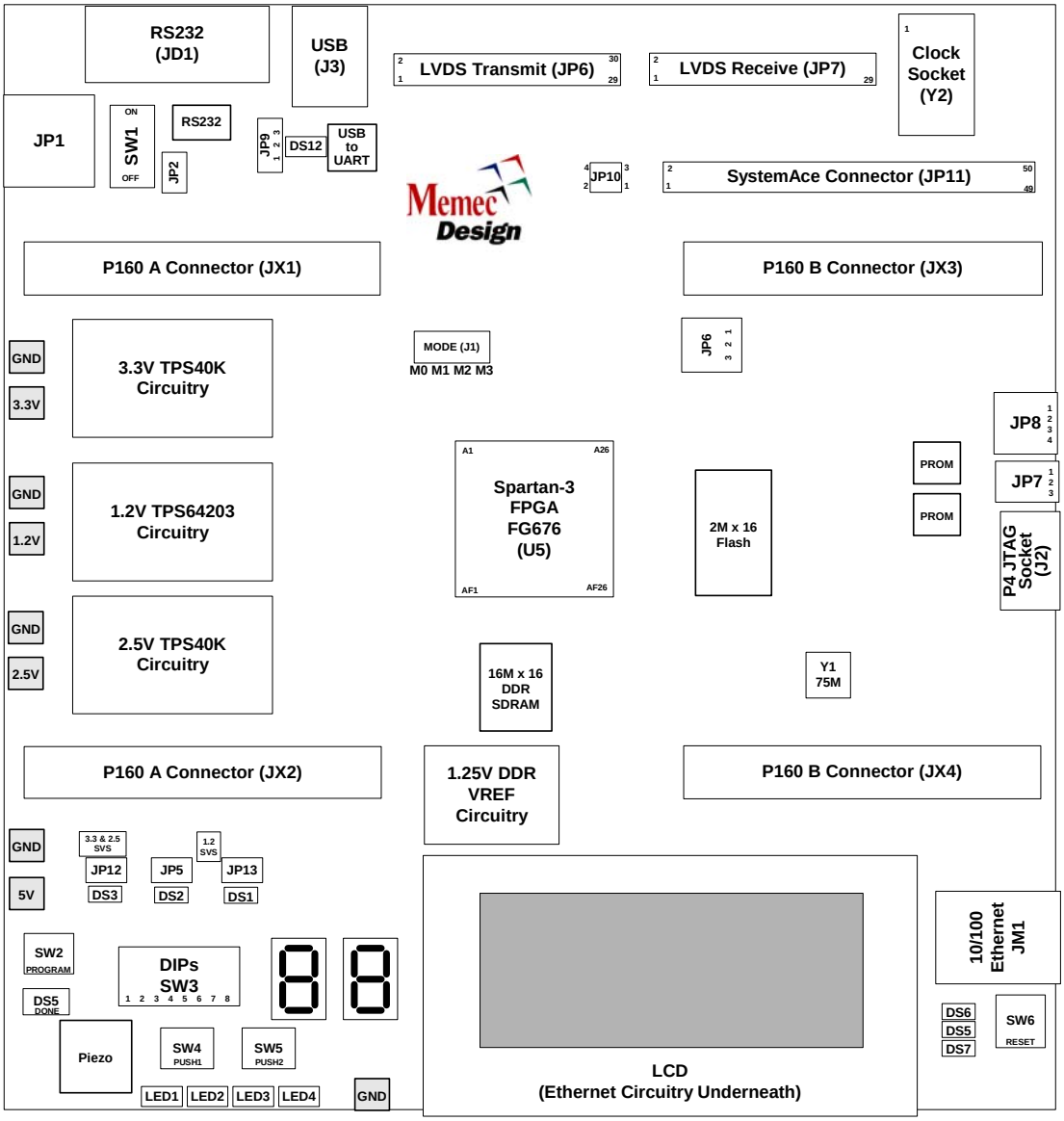

Figure 3 – Memec Spartan-3 MB Jumper Locations

The Memec Spartan-3 MB board should be configured as follows:

1. Uninstall all MODE jumper on J1.
2. Install a jumper on JP9 in the “BOARD” position (pins 1-2).
3. Install a jumper on JP10, pins 1-3.
4. Install two jumpers on JP8.
5. Install a jumper on JP7 in the “PROM ENABLE” position (pins 1-2).
6. Install a jumper on JP6 in the 3.3V position (pins 1-2).

7. No other jumpers should be installed.
8. Connect a straight through RS232 cable to the board DB-9 connector (JD1) and the serial port of the PC.
9. Verify the Power switch (SW1) is in the OFF position.
10. Connect the AC/DC adapter to JP1.

Experiments

Implementing the Project to Create a Bitstream

1. Start XPS and select **File→Open Project**
2. Browse to Memec_3SMB_MicroBlaze_Basic\3SMB_LCD_EDK81. Select system.xmp, and select Open. You should see the following XPS GUI.

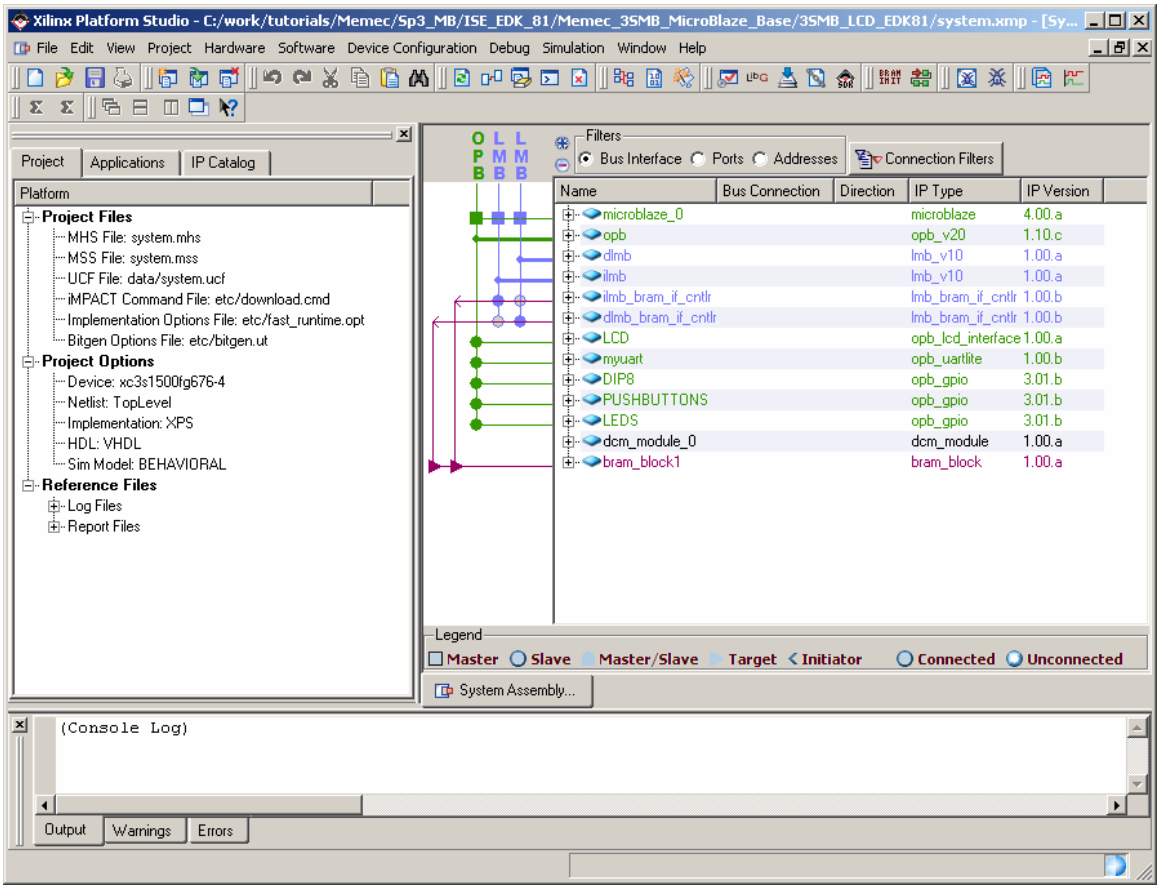

Figure 4 – LCD Project Open in XPS

1. Generate libraries (**Software → Generate Libraries and BSPs**)
2. Compile program sources (**Software → Build All User Applications**)
3. Generate Netlist (**Hardware → Generate Netlist**)
4. Generate Bitstream (**Hardware → Generate Bitstream**)

5. Update the bitstream with the program (**Device Configuration → Update Bitstream**)

Starting a HyperTerminal Session

6. Start a HyperTerminal session by double-clicking **com1_19200_8n1n.ht**. Or to start HyperTerminal manually, follow steps 7 through 10:
7. **Start→Programs→Accessories→Communications→HyperTerminal**

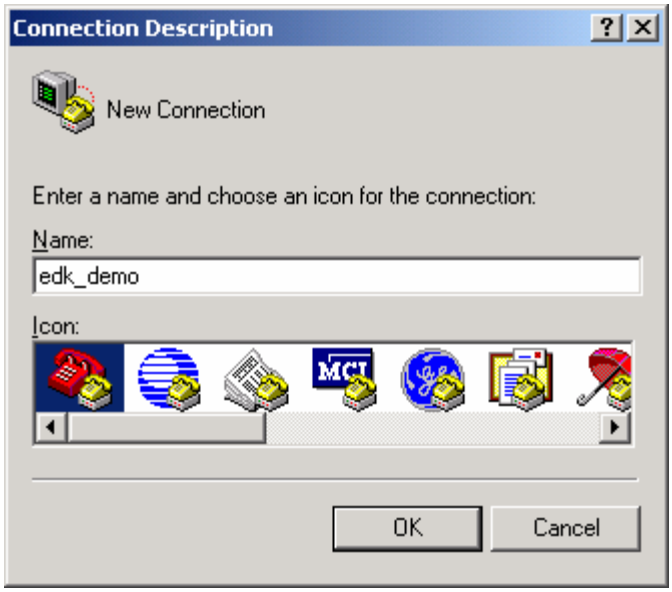

Figure 5 – HyperTerminal Connection Name

8. Enter edk_demo in the "Name" field and select OK.

Figure 6 – HyperTerminal COM

9. Select "COM1" from "Connect using" drop down menu and select OK.

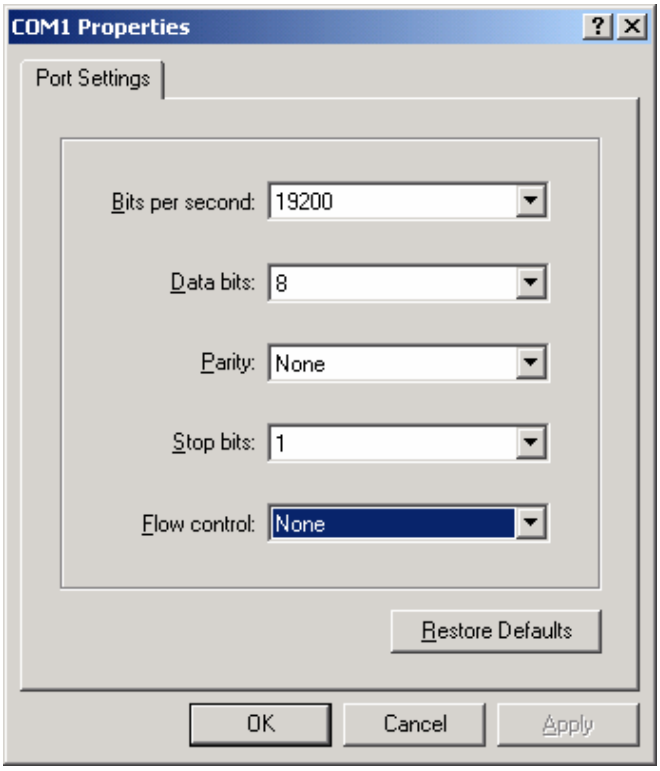

Figure 7 – HyperTerminal Settings

10. Enter the above port settings and then select OK. You should see the following HyperTerminal window.

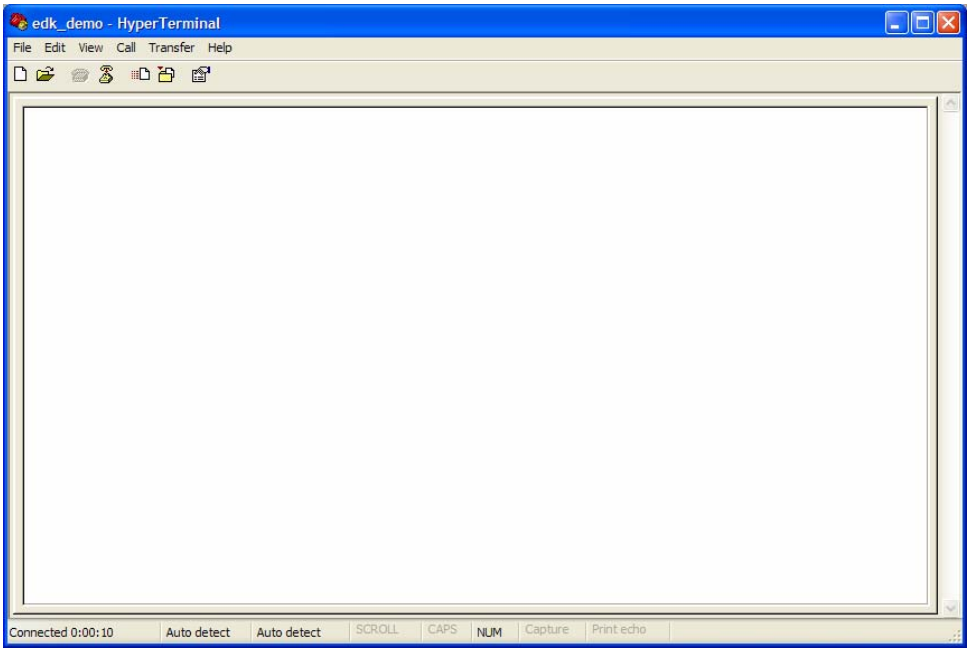

Figure 8 – HyperTerminal Launched

Running The Application Program

11. Slide the power switch to the ON position.
12. Verify the three power LEDs labeled 1.2V (DS2), 2.5V (DS1), and 3.3V (DS6) turn on.
13. Go to the XPS GUI and download the bitstream into the FPGA (**Device Configuration → Download Bitstream**). Upon completion of the download, the HyperTerminal appears as shown in Figure 9.

The following is observed on the Spartan-3 MB board:

- LCD alternates between three different messages:

```
>>> MEMEC <<<
 Spartan-3 MB

*** XILINX ***
Delivers 90nm!

Memec & Xilinx =
 A Winning Team!
```

- In between each LCD change, the four LEDs toggle a different pattern

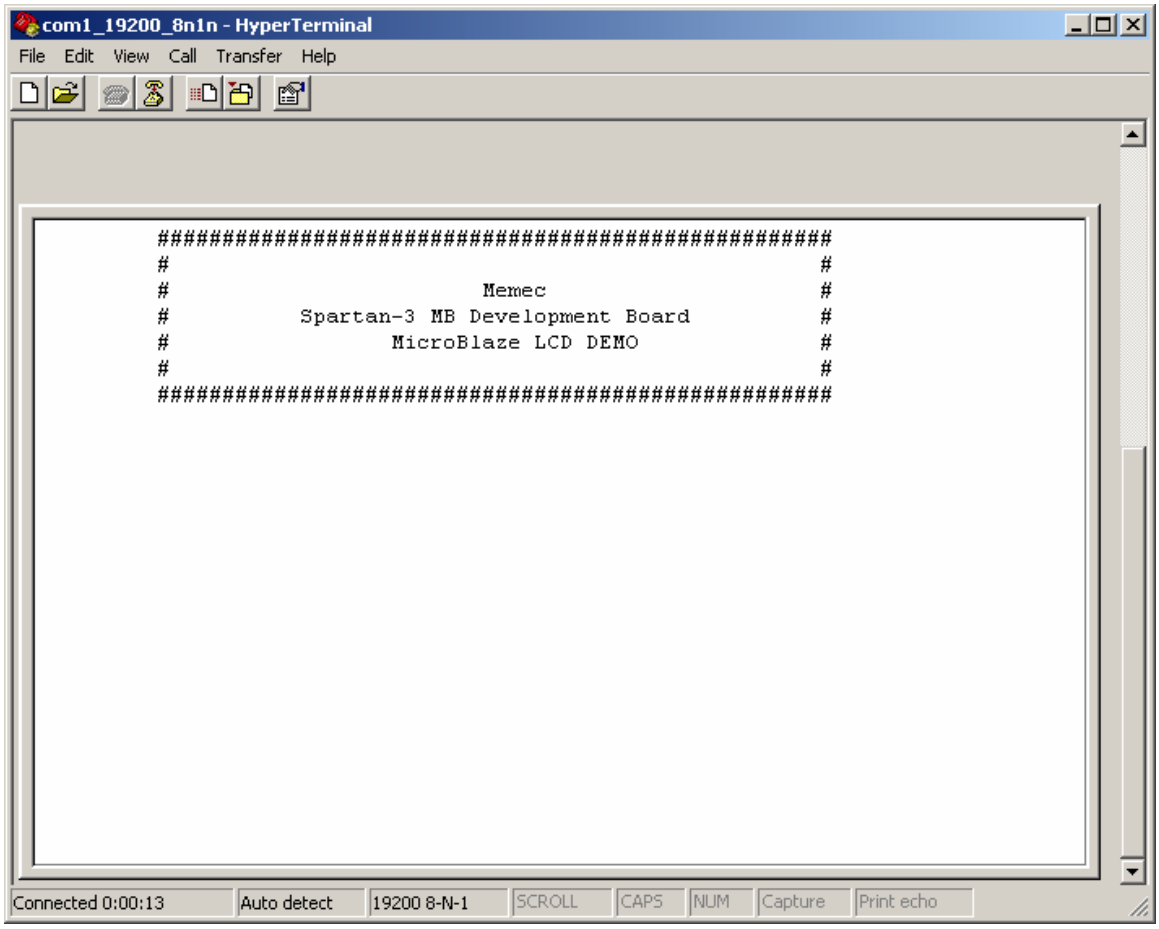

Figure 9 – LED HyperTerminal Results

Revision History

Date	Version	Revision
12/07/04	6.3	Initial Memec release, EDK 6.3
09/28/05	7.1	Updated to EDK 7.1
07/10/06	8.1	Updated to EDK 8.1