Cerebot 32MX4 ™ Microchip TCP/IP Stack™ Reference Design

Revision: January 19, 2011
Author: Aaron Odell, Jesse Adams

www.digilentinc.com
1300 NE Henley Court, Suite 3
Pullman, WA 99163
(509) 334 6306 Voice | (509) 334 6300 Fax

Overview

The Cerebot 32MX4 Microchip TCP/IP Stack™ Reference Design was written to provide an example of how to use Microchip's TCP/IP Stack™ using the Digilent Cerebot 32MX4 microcontroller board and the PmodNIC. This design will run an interactive website hosted on the Cerebot 32MX4. To follow along in this design, the reader will need to download the Microchip Applications Library, which includes the TCP/IP Stack (v. 5.31 at the time of this writing) from Microchip's website. Install the Microchip Applications Library to C:\Microchip Solutions on your hard drive. The line numbers listed in this reference design are specific to version 5.31 of the TCP/IP Stack Software. Future versions of the software may have slightly differing line numbers, but the figures included in this document should provide a good idea of approximately where code modifications were made.

This reference design requires the Cerebot 32MX4 and the PmodNIC. Optionally, the PmodLED can be used for a better example of the web application running.

This reference design will guide the reader through modifying the TCPIP Demo that is located in the C:\Microchip Solutions\TCPIP Demo App\ directory of the system. Before installing the TCP/IP Stack on the system, the reader should download and install Microchip's MPLAB IDE and the MPLAB C32 C Compiler. This will provide the development environment as well as the USB drivers needed by the Cerebot 32MX4's on-board debugger. Instructions for installing these software packages can be found on Microchip's website accompanying the installation files.

The modifications required to get the TCPIP Demo App to run on the Digilent Cerebot 32MX4 board are summarized as follows:

- Configure the MPLAB project for the appropriate device.
- Add hardware specific definitions to HardwareProfile.h
- Add HardwareProfile Cerebot32MX4PmodNIC.h to the Alternative configurations folder
- Run the TCPIP Configuration Wizard to do a bulk edit to the TCPIPConfig.h file
- Manually finish editing the TCPIPConfig.h file to allow for static IP addressing (instead of dynamic addressing via DHCP).
- Connect to board via Debugger->Select Tool->PIC32 Starter Kit
- Run the demo

Functional Description

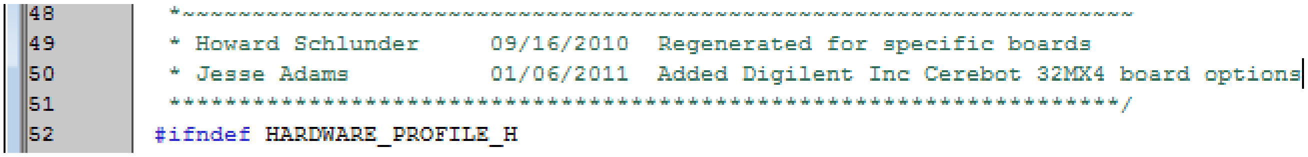
To get started with the design, navigate to C:\Microchip Solutions\TCPIP Demo App and open the TCPIP Demo App - C32 - PIC32_USB_DM320003_1_ENC28J60.mcp file. This will cause MPLAB to open up the project for editing and debugging. We will use the TCPIP Demo App - C32 - PIC32_USB_DM320003_1_ENC28J60.mcw workspace file to navigate around the project for editing.

MPLAB Device Configuration

In MPLAB, go to Configure > Select Device. From the dropdown menu, choose the PIC32MX**460**F512L.

HardwareProfile.h

Open up the HardwareProfile.h file. Since there is not yet a default for our board configuration, we must add this to the file. Generally, when editing source code in a project, it is good to make a note of it in the top header block as done here in Figure 1.



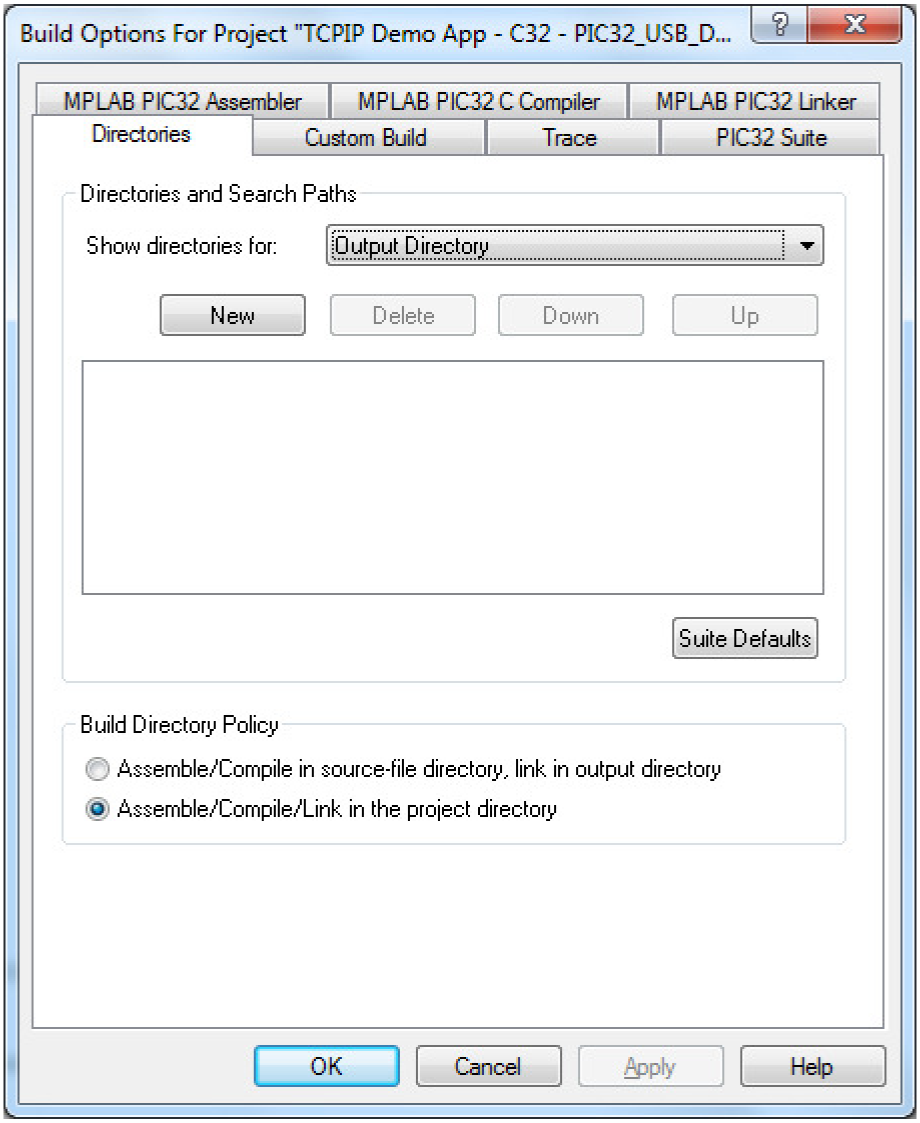
```
 *~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
 * Howard Schlunder     09/16/2010  Regenerated for specific boards
 * Jesse Adams          01/06/2011  Added Digilent Inc Cerebot 32MX4 board options
 *******************************************************************/
#ifndef HARDWARE_PROFILE_H
```

Figure 1

Now go to line 61 and insert the following code into the HardwareProfile.h file by copying and pasting it into the MPLAB editor:

```
#elif defined (Cerebot32MX4PmodNIC)
       // Cerebot 32MX4 board, PIC32MX460F512L, external PmodNIC, ENC28J60
       #include "Alternative Configurations/HardwareProfile Cerebot32MX4PmodNIC.h"
```

As noted in the introductory comment block, it is also necessary to modify the macro used. To do so, click on Project -> Build Options -> Project. This will open the dialogue box shown in figure 2.



Build Options For Project "TCPIP Demo App - C32 - PIC32_USB_D...

Directories and Search Paths

Show directories for: Output Directory

Build Directory Policy

○ Assemble/Compile in source-file directory, link in output directory

● Assemble/Compile/Link in the project directory

Figure 2

In both the “MPLAB PIC32 Assembler” and “MPLAB PIC32 C Compiler” tabs, click the “Add…” button and add “Cerebot32MX4PmodNIC” as a preprocessor macro. Then remove the current preprocessor macro “PIC32_USB_DM320003_1_ENC28J60” from both locations. Finally, click the “OK” button to save the changes.

HardwareProfile Cerebot32MX4PmodNIC.h

Now add HardwareProfile Cerebot32MX4PmodNIC.h which was included in the reference design folder to the Alternative Configurations folder, located at C:\Microchip Solutions\TCPIP Demo App\Alternative Configurations.

TCPIPConfig.h

In TCPIPConfig.h, it is necessary to add the new hardware profile to the file. As in figure 1, it is good to make a note of the changes in the header. Go to line 123, and add the following line:

```
defined(Cerebot32MX4PmodNIC) || \
```

Initial Configuration of TCPIPConfig ENC28J60 PIC32 Starter Kits.h

The TCPIPConfig ENC28J60 PIC32 Starter Kits.h file declares what functions will be present in the demonstration application. The Cerebot 32MX4 cannot support many of these applications because it lacks external memory on the board. In the interests of providing the simplest example application running the Microchip TCP/IP Stack, it is necessary to remove as much of the added features as possible. In order to modify the TCPIPConfig ENC28J60 PIC32 Starter Kits.h file, it is best to run the Microchip TCPIP Configuration Wizard application. To do this, click on the Windows Menu bar and navigate **start**->**Programs**->**Microchip**->**TCPIP Stack**->**TCPIP Configuration Wizard**.

When the application opens, browse to **C:\Microchip Solutions\TCPIP Demo App\AlternativeConfigurations\TCPIPConfig ENC28J60 PIC32 Starter Kits.h** and click **Next**.

In the screen that follows, only select the **Web Server** check box, uncheck all other options and click **Next**.

Uncheck all of the options on the following screen and click **Next**.

Leave the Host Name and Default MAC Address the same and click **Next**.

Uncheck all of the options that appear on the right-hand side of the screen and click **Next**.

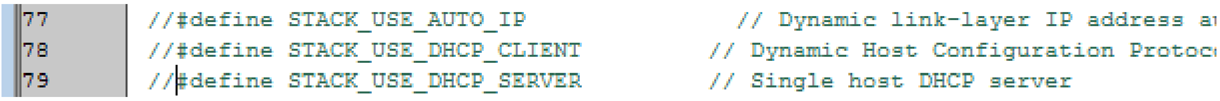
Select the “Internal Program Memory” radio button and click **Next** and finally click **Finish**.

Static IP Address Assignment in TCPIPConfig ENC28J60 PIC32 Starter Kits.h

The IP address of the board is required to view the web page in a browser. It is easier to statically assign the IP address than it is to find out what was assigned to it when using DHCP. For this reason, this design uses the option to statically set the IP address found in TCPIPConfig ENC28J60 PIC32 Starter Kits.h. If your design requires a static IP address, then modify TCPIPConfig ENC28J60 PIC32 Starter Kits.h following these two steps:

Comment out the following lines as shown in Figure 3:

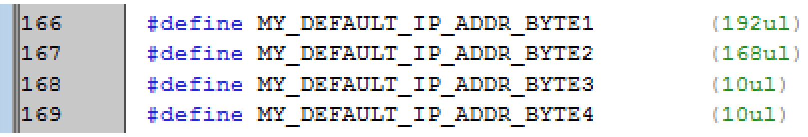
- #define STACK_USE_AUTO_IP
- #define STACK_USE_DHCP_CLIENT
- #define STACK_USE_DHCP_SERVER



```
//#define STACK_USE_AUTO_IP               // Dynamic link-layer IP address a
//#define STACK_USE_DHCP_CLIENT         // Dynamic Host Configuration Protoc
//#define STACK_USE_DHCP_SERVER         // Single host DHCP server
```

Figure 3

Set the desired static IP address for the board in lines 165 through 168. If the board is going to be sharing a network where DHCP is used, keep the first three sets of numbers the same as the network and assign the final number to a value outside of the DHCP pool. Generally DHCP assigns devices numbers between 100 and 255, so assign the board a number below 100. Use Figure 4 as an example.

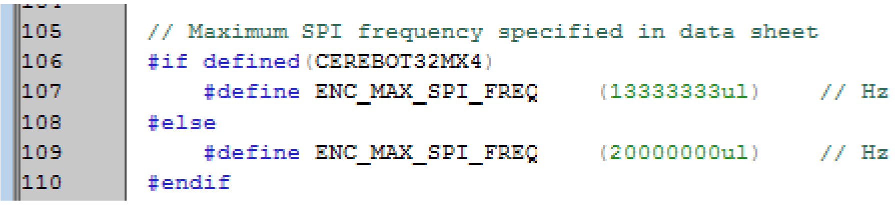


```
166    #define MY_DEFAULT_IP_ADDR_BYTE1        (192ul)
167    #define MY_DEFAULT_IP_ADDR_BYTE2        (168ul)
168    #define MY_DEFAULT_IP_ADDR_BYTE3        (10ul)
169    #define MY_DEFAULT_IP_ADDR_BYTE4        (10ul)
```

Figure 4

Fix SPI Clock Speed

The PmodNIC cannot run at 20MHz. In the ENC28J60.c (located in the TCPIP Stack subfolder of the source folder) driver that accompanies the TCPIP Library, there is a line that sets the SPI clock speed. You need to add a condition that modifies the maximum SPI frequency to 13.33 MHz for the Cerebot32MX4. To do this, find the line that defines ENC_MAX_SPI_FREQ, and modify it as shown in Figure 5.



```
// Maximum SPI frequency specified in data sheet
#if defined(CEREBOT32MX4)
    #define ENC_MAX_SPI_FREQ    (13333333ul)    // Hz
#else
    #define ENC_MAX_SPI_FREQ    (20000000ul)    // Hz
#endif
```

Figure 5

Attaching Pmods

The application requires the PmodNIC to work and optionally uses the PmodLED to show additional visual output used by the web interface. Plug the PmodNIC into port JB which is where the SPI2 pins are located on the Cerebot 32MX4 and optionally plug the PmodLED into the lower pins (pin 7 through 12) of port JK. Pins 1 through 4 of port JK are shared with the onboard LEDs.

Select Debugging Tool

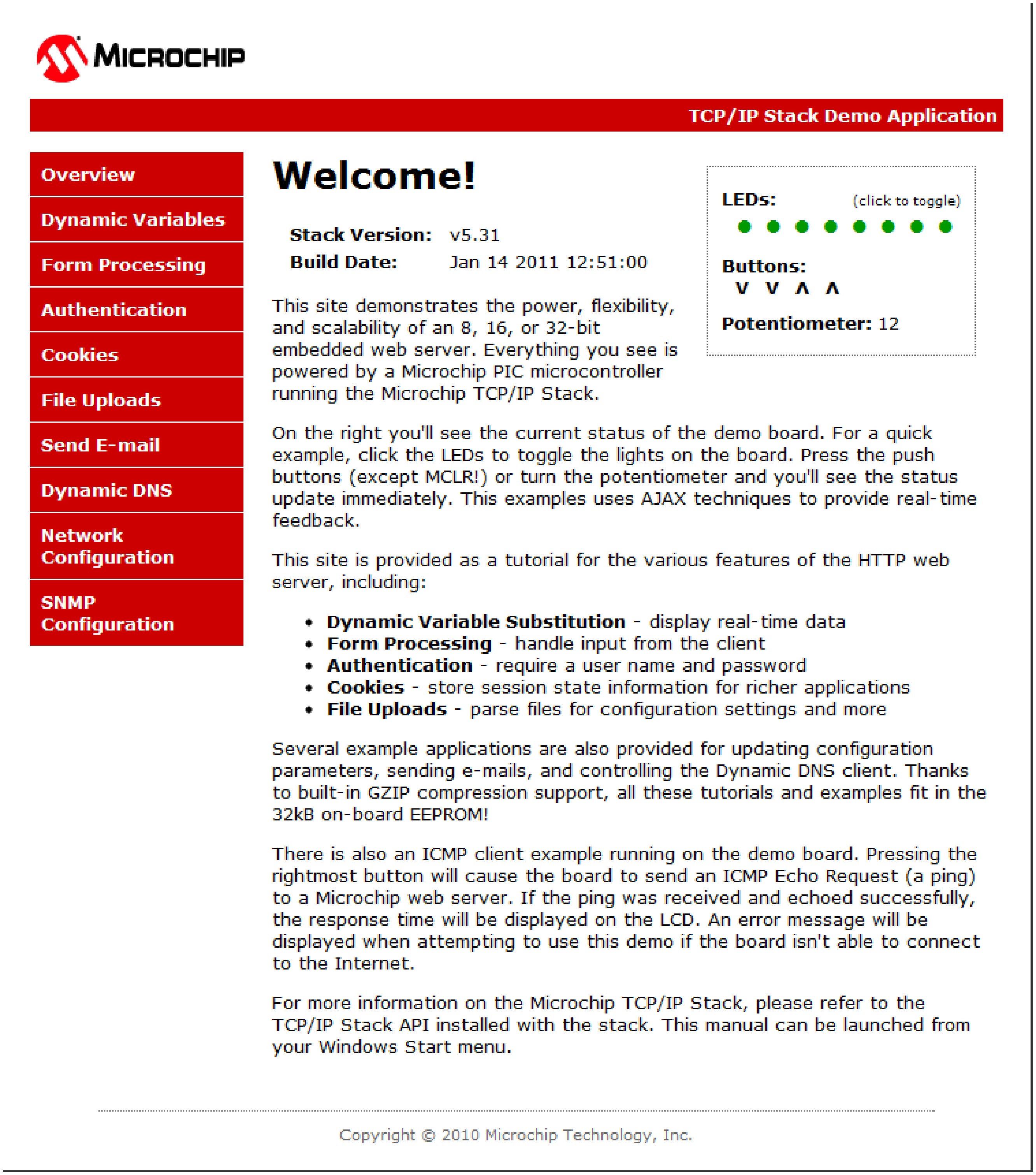
Click on the **Debugger** Menu option followed by **Select Tool** and finally select the **PIC32 Starter Kit** to connect to the Cerebot 32MX4. The Output window should pop up indicating that the PIC32 Starter Kit was found.

Compiling and Running Program

The PmodNIC and Cerebot 32MX4 should be connected to the same network as the PC, ideally through a shared switch.

All of the source editing has now been completed. Click on the **Debugger** menu item and select **Run**. MPLAB will prompt you to build the project. Click **Yes**. Repeat the previous step and MPLAB will prompt that the Cerebot 32MX4 needs to be programmed. Click **Yes** to program and run the application on the board.

Once the program is running, view the Microchip embedded website using a web browser. In the address bar, type in the IP address of the board and connect. In this example, the web site is located at 192.168.10.10 as was defined in TCPIPConfig ENC28J60 PIC32 Starter Kits.h, lines 166 through 169. If successful, the web browser should display the page as shown in Figure 6.



Welcome!

Stack Version: v5.31
Build Date: Jan 14 2011 12:51:00

Potentiometer: 12

This site demonstrates the power, flexibility, and scalability of an 8, 16, or 32-bit embedded web server. Everything you see is powered by a Microchip PIC microcontroller running the Microchip TCP/IP Stack.

On the right you'll see the current status of the demo board. For a quick example, click the LEDs to toggle the lights on the board. Press the push buttons (except MCLR!) or turn the potentiometer and you'll see the status update immediately. This examples uses AJAX techniques to provide real-time feedback.

This site is provided as a tutorial for the various features of the HTTP web server, including:

- **Dynamic Variable Substitution** - display real-time data
- **Form Processing** - handle input from the client
- **Authentication** - require a user name and password
- **Cookies** - store session state information for richer applications
- **File Uploads** - parse files for configuration settings and more

Several example applications are also provided for updating configuration parameters, sending e-mails, and controlling the Dynamic DNS client. Thanks to built-in GZIP compression support, all these tutorials and examples fit in the 32kB on-board EEPROM!

There is also an ICMP client example running on the demo board. Pressing the rightmost button will cause the board to send an ICMP Echo Request (a ping) to a Microchip web server. If the ping was received and echoed successfully, the response time will be displayed on the LCD. An error message will be displayed when attempting to use this demo if the board isn't able to connect to the Internet.

For more information on the Microchip TCP/IP Stack, please refer to the TCP/IP Stack API installed with the stack. This manual can be launched from your Windows Start menu.

Copyright © 2010 Microchip Technology, Inc.

Figure 6