

MCF51CN128—Lab Tutorials 1 & 2 (sheet 1 of 2)

LAB 1

Security System, Freescale MQX™ RTOS for MCF51CN128

Introduction

This lab will guide you through the general use of the Freescale MQX™ operating system under the CodeWarrior™ integrated development environment and will familiarize you with the general compile and download process. This lab will simulate an Ethernet-enabled home security system that is accessed over telnet.

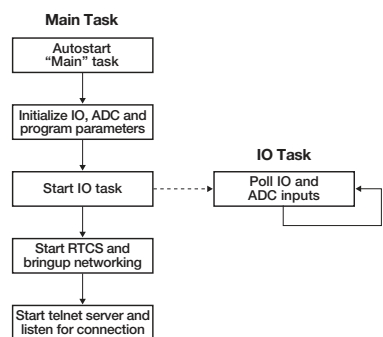


Figure: Software Flow Chart

Demonstrates

- MQX project in CodeWarrior IDE
- Project build, download and run in CodeWarrior IDE
- Accelerometer
- MQX shell
- MQX GPIO driver (push button and LEDs)
- MQX RTCS TCP/IP network stack
- Telnet server functionality

Step by Step Instructions

1. Install CodeWarrior for Microcontrollers 6.2 (30 day evaluation version available, Basic or higher required for MQX Lab tutorials) then install CodeWarrior 6.2.2 patch. Then open CodeWarrior.
2. Press the card edge connector of the MCU module into a slot on the Elevator—take care to match the primary connector card edges and plug them into a Functional Elevator. Do the same for the serial module.

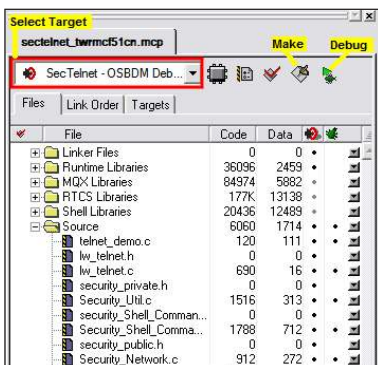


Figure 2: Project Loaded in the CodeWarrior IDE

A module may be placed into any slot on the Elevator, but it is recommended to put the MCU module on the top for easier access to the switches and LEDs.

Then press the Dummy Elevator onto the card edges labeled secondary.

3. Make the following connections from the MCF51CN128 Tower System to the computer. See Figure 1 on reverse side.
 - a. USB cable between the OSBDM debugger (J14 on the TWR-MCF51CN module) and a USB port on PC
 - b. Ethernet cable between the Ethernet port on the TWR-SER module and an Ethernet port on PC
4. The first time you connect the USB debugger cable to your PC, Windows will install a driver for the debugger. Follow the prompts to automatically detect and install the driver.
5. If you are running the board for the very first time, the telnet demo has already been loaded on the board. Skip to Step 9 to setup the network, and then skip to Step 16 to run the demo. If you would like to re-flash the board with the original demo code, then continue onto the next step.
6. If you did not install the MQX project in the default **C:\Program Files\Freescale\Freescale MQX 3.2** directory during installation, you must first recompile the MQX libraries. See the release notes for more information.
7. Open the lab project by selecting the File > Open menu, and opening: **C:\Program Files\Freescale\Freescale MQX 3.2\demo\security_telnet\codewarrior\sectelnet_twrncf51cn.mcp**

8. In the project pane, select the “SecTelnet - OSBDM Debug Int Flash” build target. See Figure 2.

9. The default IP address of the board is **169.254.3.3**. Typically, when you connect your computer directly to the board, the computer will default to an auto IP address on the same subnet as the board (169.254.x.x), therefore requiring no setup. You may see a message from Windows saying “Limited Connectivity” when doing this, but that is expected.

Note: The PC may take a few minutes to default to the auto IP address and make the connection. If you have trouble connecting, you may configure the IP address of the computer manually. Select **Start > Control Panel > Network Connections > Local Area Connection**. Open up the TCP/IP properties, Note your original TCP/IP settings, and then set your IP address to 169.254.3.4 and your subnet mask to 255.255.0.0.

10. Open the Security.h file in the CodeWarrior window. Double-click the file item located in the “Source” group in the CodeWarrior project tree. See Figure 2.
11. If you would like to change the default IP address of the board, locate the line of code starting with **#define ENET_IPADDR** and specify your target IP address and IP mask address by using the IPADDR macro. The default IP address is 169.254.3.3 and subnet mask is 255.255.0.0

```
#ifndef ENET_IPADDR
#define ENET_IPADDR IPADDR(169, 254, 3, 3)
#endif

#ifndef ENET_IPMASK
#define ENET_IPMASK IPADDR(255, 255, 0, 0)
#endif
```

Figure 3: MQX Source

12. Compile the project by pressing the F7 key or by clicking the Make icon on the project pane toolbar. This icon is identified in Figure 2.
13. Now start debugger by clicking the Debug icon. This icon is identified in Figure 2.
14. The V1 ColdFire debugger will appear with a dialog box asking if you would like to erase the flash and load it with the new software. Select OK.



Figure 4: Re-Flashing the chip

15. After the code has been flashed to the board, you will see the MQX entry-point function in

the code window. Hit the “Start” button as seen in Figure 5.

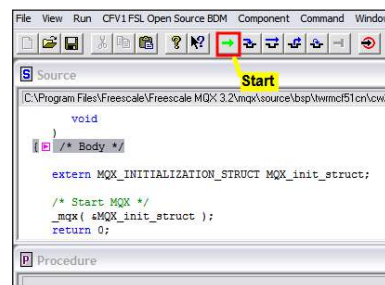


Figure 5: Debugger

16. Open a Command Prompt on the PC (Start > Programs > Accessories > Command Prompt)
17. At the prompt invoke a telnet session to the board by typing **telnet 169.254.3.3**. You will be connected to the Tower board via telnet.



Figure 6: Connect via telnet

18. Type **help** to see the list of available commands.

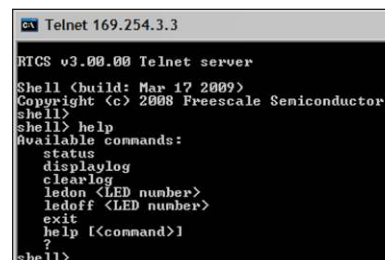


Figure 7: Telnet commands

19. A simple Security system has been implemented as the example application to demonstrate the features of the MQX RTOS. It detects button presses and movement of the tower system, keeps a log of the events and provides a user interface over telnet.

20. The LEDs on the MCU board represent the current state of the system.
 - When the tower system is held flat, LED1, LED2 and LED3 are controlled by the commands **ledon** and **ledoff** (ie **ledon 2**).
 - When the tower system is tilted, LED1, LED2, and LED3 represent how far the board is tilted in the Y-axis direction.

- LED4 represents the Door (SW2) or Window (SW3) status as OPEN or CLOSED

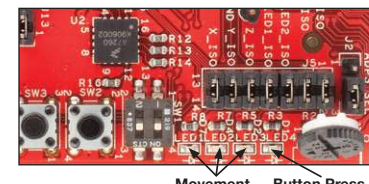


Figure 8: LEDs

21. Use the **status** command to see the current status of the board. It will display the current state of the switches, potentiometer and accelerometers, as well as the elapsed time since bootup.

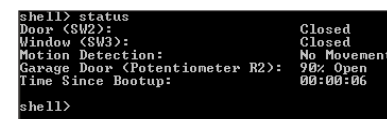


Figure 9: Status Command

22. Hold SW2 down, representing the door, with one hand, while typing **status** with the other and hit the Enter key. You will see that the door is Open.

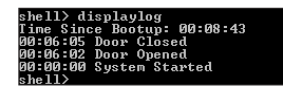


Figure 10: Display Log

23. Type **displaylog** to see the last 10 events that the tower system has detected. Clearlog will clear the history.

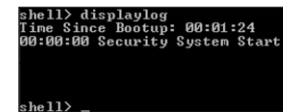


Figure 11: Display Log

24. Shake the Tower vigorously for a few seconds. Now type **displaylog** again.

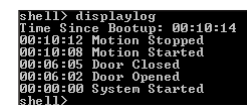


Figure 12: Display log

25. Use the **ledon** or **ledoff** commands to control the LED GPIO.



Figure 13: LED Commands

LAB 2

Web Security System, Freescale MQX RTOS for MCF51CN128

Introduction

This lab uses dynamic Web pages served by the TWR-MCF51CN-KIT to serve as a graphical user interface to the Security application.

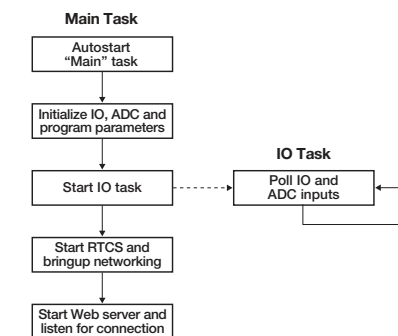


Figure: Software Flow Chart

Demonstrates

- MQX RTCS TCP/IP network stack
- HTTP Server functionality

Step by Step Instructions

1. Follow steps 1 through 4 in Lab 1 to setup the Tower System.
2. Open the Web Server Lab Project by selecting the File > Open menu, and opening: **C:\Program Files\Freescale\Freescale MQX 3.2\demo\security_webserver\codewarrior\secwebserver_twrncf51cn.mcp**
3. In the project pane, select “SecWebserver - OSBDM Debug Int. Flash” build target. See Figure 1.

continued on reverse side...

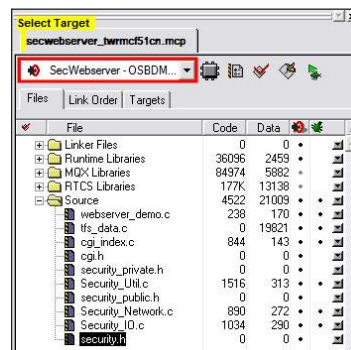

LAB 2
Web Security System, Freescale MQX RTOS for MCF51CN128 *continued*


Figure 1: Project Loaded in the CodeWarrior IDE

- The default IP address will be **169.254.3.3**. If you would like to change the IP address, open the file **security.h**, and follow the directions in Steps 9 to 11 of Lab 1.
- Compile and load project on the Tower board by following steps 12 through 15 in Lab 1.
- Open up an Internet browser and navigate to the target device address. In this case, **169.254.3.3**. Turn off any proxy settings that may be enabled for your browser.
- You should see the Web server welcome page in the browser window, as seen in *Figure 2*.
- Try pressing buttons and tilting the board. You will see the state of the inputs change on the Web page, as well as a running log of the actions performed.

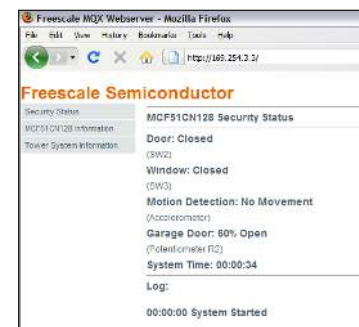


Figure 2: Web Interface

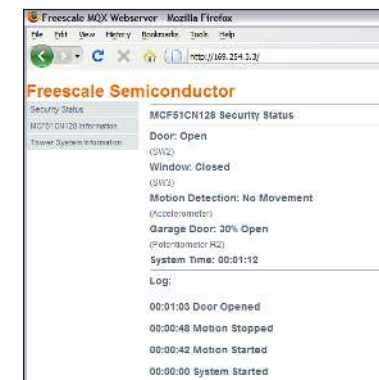
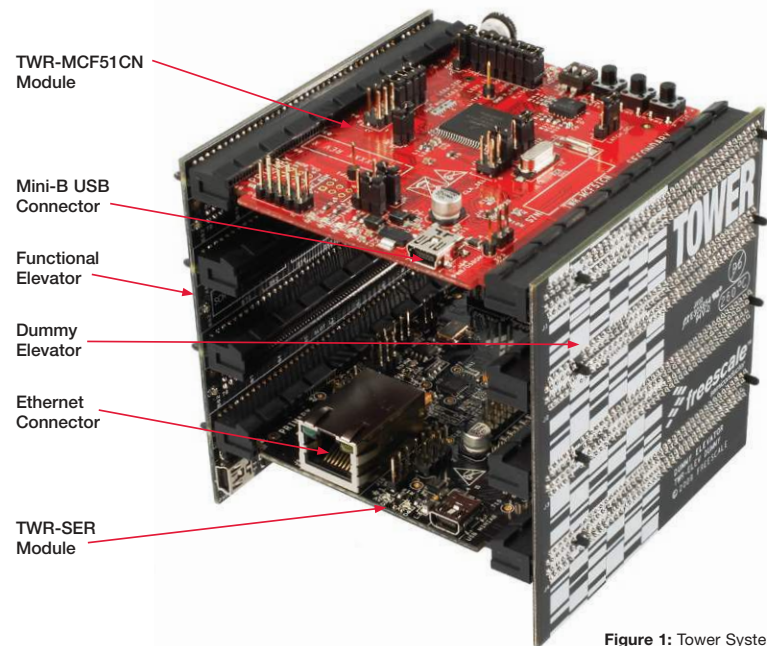
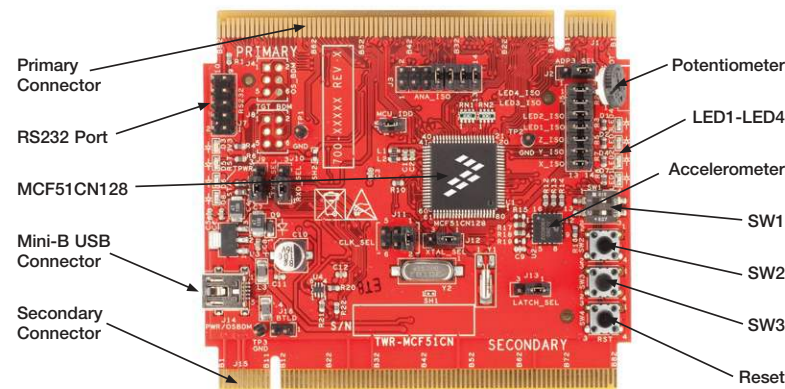


Figure 3: Web Interface

- Click on the links to the left to get more information on the MCF51CN128 microprocessor and the Tower System.
- The Web pages are stored in the `\demo\security_webserver\web_pages` directory. You can edit them with any text editor. Then double click on **Build_Webpages.bat** which calls **mkfts.exe**, and that script will convert the Web pages into a C array that is stored in the **tfsdata.c** file.
- Then recompile the project, and the new Web pages will appear when you re-run the code. The Web server's default page is **mqx.html**.

See more Lab Tutorials (3 & 4) on sheet 2, and at www.freescale.com/tower.

About the Tower System

Get to know the TWR-MCF51CN

TWR-MCF51CN-KIT Freescale Tower System

The TWR-MCF51CN module is part of the Freescale Tower System, a modular development platform that enables rapid prototyping and tool re-use through reconfigurable hardware. Take your design to the next level and begin constructing your Tower System today.

Learn More: For more information about Freescale products, please visit www.freescale.com/tower.

MCF51CN128—Lab Tutorials 3 & 4 (sheet 2 of 2)

LAB 3

Low-Power e-Mail-Enabled Security, Freescale MQX RTOS for MCF51CN128

Introduction

This lab demonstrates the MCF51CN128 Tower System by sending out an e-mail when one of the push buttons are pressed. It also updates itself to the current time using the SMTP protocol, and acquires an IP address off a network via DHCP. It will conserve power in Stop3 sleep mode and wake at the press of a button.

Please note that this lab requires basic knowledge about your network setup and e-mail configuration. These values will vary from network to network, and not all networks will have an e-mail server or SMTP server. Also some corporate networks are located behind a firewall or proxy, which might inhibit the functionality of this lab. Please contact your system administrator for more information on your network configuration.

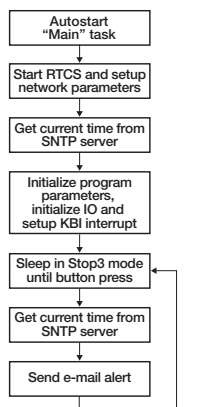


Figure: Software Flow Chart

Demonstrates

- MQX RTCS TCP/IP network stack
- DHCP functionality
- SMTP functionality

- DNS functionality
- Sockets

Step by Step Instructions

1. Install CodeWarrior for Microcontrollers 6.2 (30 day evaluation version available, Basic or higher needed to run MQX lab tutorials) then install CodeWarrior 6.2.2 patch. Then open CodeWarrior.
2. Construct the TWR-MCF51CN-KIT as instructed in Step 2 of Lab 1. Then make the following connections from the TWR-MCF51CN-KIT to the computer. See Figure 1 on reverse side.
 - a. USB cable between the OSBDM debugger (J14 on the TWR-MCF51CN module) and a USB port on PC
 - b. Ethernet cable between the TWR-SER module and an Ethernet port on an external network
 - c. Serial Port on the TWR-SER module to a Serial Port on PC (optional, cable not included)
3. The first time you connect the USB debugger cable to your PC, Windows will install a driver for the debugger. Follow the prompts to automatically detect and install the driver.
4. If you did not install the MQX project in the default C:\Program Files\Freescale\Freescale MQX 3.2\demo\security_email\codewarrior\secemail_twr_mcf51cn.mcp
5. Open the Lab Project by selecting the File > Open menu, and opening: C:\Program Files\Freescale\Freescale MQX 3.2\demo\security_email\codewarrior\secemail_twr_mcf51cn.mcp
6. In the project pane, select "SecEmail - OSBDM Debug Int. Flash" build target.
7. Open up the security.h file in the Sources group.

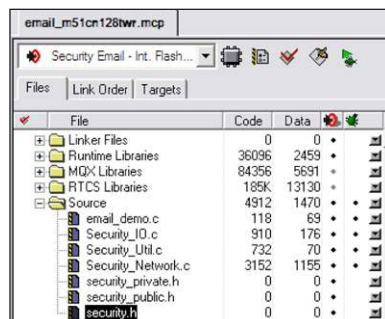


Figure 2: MQX Source Tree

8. The network settings need to be configured. Look for the line in security.h that begins with #define DEMOCFG_ENABLE_DHCP 1
9. If the network you connected to uses DHCP, skip to Step 10. However if the network you are connected to requires you use a static IP, change


```
#define DEMOCFG_ENABLE_DHCP 1 to #define DEMOCFG_ENABLE_DHCP 0
```

 Then change the other parameters shown in Figure 3 to match your network. This is only required if using a static IP:

```
#ifndef ENET_IPADDR
#define ENET_IPADDR IPADDR(192.168.1.114)
#endif

#ifndef ENET_IPMASK
#define ENET_IPMASK IPADDR(255.255.255.0)
#endif

#ifndef ENET_IPGATEWAY
#define ENET_IPGATEWAY IPADDR(192.168.1.1)
#endif

#ifndef ENET_IPDNS
#define ENET_IPDNS IPADDR(192.168.1.1)
#endif
```

Figure 3: Network Parameters

You can get those values by asking your network administrator. Alternatively, if your computer is already connected to the network via an Ethernet cable, open a Command Prompt on the PC (Start > All Programs > Accessories > Command Prompt). Then type ipconfig /all to get the information for your Local Area Connection to the Internet, as shown in Figure 4. Then unplug the Ethernet cable from your computer and plug it into the Ethernet port on the Tower System.



Figure 4: IPconfig on Simple Network

10. The GMT time can be configured on the MCF51CN128 by using the SMTP protocol. The SMTP server to contact is specified by a domain name which must be enclosed in quotation marks, as shown in Figure 5. The default SMTP address is the NIST time server. However if your external internet connection goes through a proxy or firewall, you must use a SMTP server located within your network. Contact your network administrator to find out the address for your network's SMTP server. If you cannot determine the SMTP server for your network, then disable the SMTP feature by setting DEMOCFG_ENABLE_SNTP to 0:


```
#define DEMOCFG_ENABLE_SNTP
```

```
#define SMTP_SERVER "time.nist.gov"
```

Figure 5: SMTP Parameters

11. Next, set up the e-mail configuration. If you are using Outlook Express or Thunderbird, you can find these settings in the configuration options. You might also find these settings on the help page for your network provider.

Note: e-Mail servers that requires encrypted authentication, usually via SSL (for example, Gmail, Hotmail, etc) will not work with this release of MQX

```
#define AUTH_USERNAME "username"
#define AUTH_PASSWORD "password"
```

Figure 6: e-Mail Parameters

12. The main e-mail settings are configured in the #defines listed in Figure 6. Ensure that all values are enclosed in quotation marks.
 - a. EMAIL_SERVER: The domain name of the SMTP server that you use for e-mail. It cannot be a SMTP server that uses SSL for authentication (for example Gmail or Hotmail). If your network is behind a firewall or proxy, you must use a SMTP server located within your network.
 - b. EMAIL_TO: e-Mail address to send the e-mail to. This can be any e-mail address.
 - c. EMAIL_FROM: Your official e-mail address for the SMTP server you are connecting to. You cannot use an alias or "friendly address" e-mail address. Many servers will reject connections that do not provide a real e-mail address on the assumption that it is spam.
13. If your SMTP server does not require authentication, then skip to step 16. However if it does require authentication, and does not require encryption, change


```
#define DEMOCFG_AUTH_REQUIRED 0 to #define DEMOCFG_AUTH_REQUIRED 1
```

 Then change the username and password parameters shown in Figure 7. The username and password is usually the same combination that you would use to access your e-mail.

```
#define AUTH_REQUIRED 0
#define AUTH_USERNAME "smtp_username"
#define AUTH_PASSWORD "smtp_password"
```

Figure 7: e-Mail Authentication

14. Now open up a serial connection using Hyperterminal of Lab 1. This is optional, as it will print out debug information that might be useful if there is an issue with your network connection.

15. Start HyperTerminal on the PC (Start menu>Programs>Accessories>Communications). Make a connection to the serial port that is connected to the board (usually will be COM1)



Figure 8: Connect to COM1

16. Set it for 115200 baud, no parity, 8 bits and click OK.

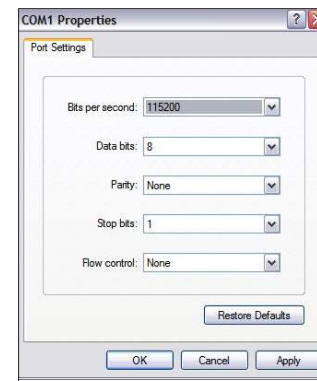


Figure 9: COM1 Properties

17. Compile, download, and run the application as was done in steps 12 to 15 of Lab 1.
18. Now, press a button on the Tower System. After you release it, an e-mail will be sent to the e-mail address you specified, stating the time you pressed it and how much time elapsed from the last event.

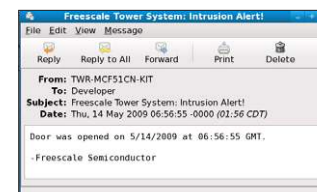


Figure 10: e-Mail

19. You can also send text messages via e-mail. For example, in the EMAIL_TO field, use 5125551234@tmomail.net to send a text message to a T-Mobile user with the phone number 512-555-1234. The domain name is specific to the cell phone service provider for that phone number. Contact your service provider for more information, or search online.

```
#define EMAIL_TO "5125551234@tmomail.net"
```

Figure 11: e-Mail-to-Text

LAB 4

Telnet to Serial Bridge, Freescale MQX RTOS for MCF51CN128

Introduction

This lab demonstrates how to create a bridge between a TCP/IP (telnet) connection and a serial line.

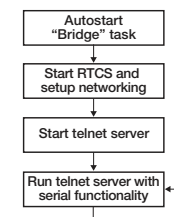


Figure: Software Flow Chart

Demonstrates

- MQX RTCS TCP/IP network stack
- Custom telnet server implementation
- Re-directing STDIN and STDOUT within a MQX task

Step by Step Instructions

1. Install CodeWarrior for Microcontrollers 6.2 (30 day evaluation version available, Basic or higher needed to run MQX Lab tutorials) then install CodeWarrior 6.2.2 patch. Then open CodeWarrior.

Continued on reverse side...



LAB 4

continued

- Construct the TWR-MCF51CN-KIT as instructed in Step 2 of Lab 1. Then make the following connections from the TWR-MCF51CN-KIT to the computer. See Figure 1 at the bottom of this page.
 - USB cable between the OSBDM debugger (J14 on the TWR-MCF51CN module) and an USB port on PC
 - Ethernet cable between the TWR-SER module and an Ethernet port on an external network
 - Serial Port on the TWR-SER module to a Serial Port on PC (cable not included)
- The first time you connect the MXQ debugger cable to your PC, Windows will install a driver for the debugger. Follow the prompts to automatically detect and install the driver.
- If you did not install the MXQ project in the default `C:\Program Files\Freescale\ Freescale MQX 3.2\demo\telnet2ser\codewarrior\telnet2ser_twrmc51cn.mcp` directory, you must first recompile the MXQ libraries to reflect the new path name. See the release notes for more information before continuing the lab.
- Open the Lab Project by selecting the File > Open menu, and opening: `C:\Program Files\Freescale\Freescale MQX 3.2\demo\telnet2ser\codewarrior\telnet2ser_twrmc51cn.mcp`
- In the project pane, select "Telnet2Ser - OSBDM Debug Int. Flash" build target.

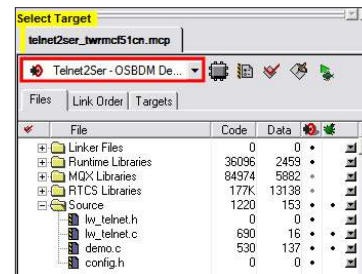


Figure 2: MXQ Source Tree

Note: The PC may take a few minutes to default to the auto IP address and make the connection. However, if you have trouble connecting, you may configure the IP address of the computer manually. Select Start > Control Panel > Network Connections > Local Area Connection. Open up the TCP/IP properties, Note your original TCP/IP settings, and then set your IP address to 169.254.3.4 and your subnet mask to 255.255.0.0.

- Open the config.h file in the CodeWarrior window. Double-click the file item located in the "Source" group in the CodeWarrior project tree. See Figure 2.
- If you would like to change the default IP address of the board, locate the line of code starting with `#define ENET_IPADDR` and specify your target IP address and IP mask address by using the IPADDR macro. The default IP address is **169.254.3.3** and subnet mask is **255.255.0.0**
- Compile and load the project on the tower board by following steps 12 through 15 in Lab 1.
- Open a Command Prompt on the PC (Start > All Programs > Accessories > Command Prompt). At the prompt invoke a telnet session to the board by typing **telnet 169.254.3.3** You are now connected to the MXQ shell via telnet.

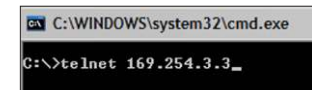


Figure 3: Connect via telnet

- Now open up a serial console by following steps 17 and 18 from Lab 3.
- The serial console and the telnet sessions should be "bridged." Type some characters into the telnet session, and you should see the characters appearing on the console terminal window. See Figure 4 for how it would appear.

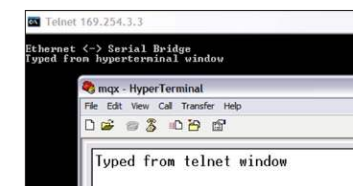


Figure 4: Ethernet to Serial Bridge

- Then try typing into the console terminal window, and you will see the characters appearing in the telnet session.

About the Tower System

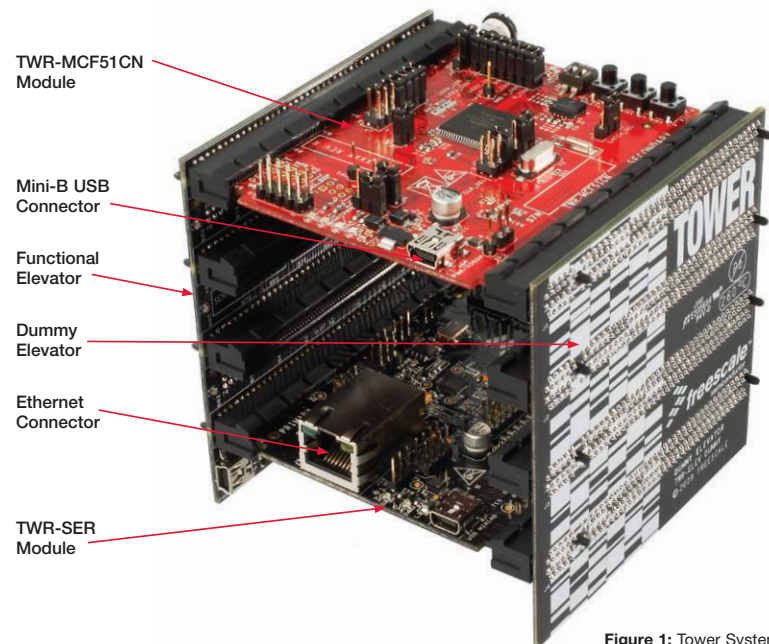
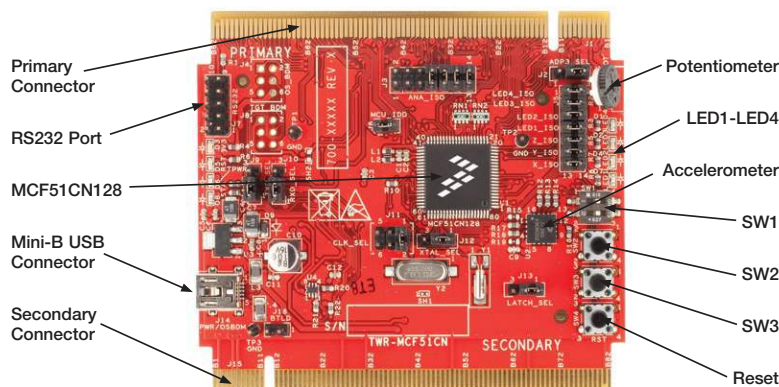


Figure 1: Tower System

Get to know the TWR-MCF51CN



TWR-MCF51CN-KIT Freescale Tower System

The TWR-MCF51CN module is part of the Freescale Tower System, a modular development platform that enables rapid prototyping and tool re-use through reconfigurable hardware. Take your design to the next level, and begin constructing your Tower System today.

Learn More: For more information about Freescale products, please visit www.freescale.com/tower.