Freescale Semiconductor
Application Note

Document Number: AN3231
Rev. 1.1, 10/2006

MC1319x, MC1320x, and MC1321x Demonstration Operation

Running SMAC Based Demonstration Operations

1 Introduction

This application note describes how to run the following Simple Media Access Controller (SMAC) based demonstration applications:

- Basic Packet Error Rate (PER)
- Wireless UART
- Accelerometer V 3.0
- Range
- Lighting
- Test Mode
- Repeater
- Simple Protocol Test Client
- Over the Air Programmer (OTAP)

For more details on running the Accelerometer application, refer to AN3232, *Accelerometer Demonstration With the 13213-SRB (Sensor Reference Board).*

For more details about the MC1319x, MC1320x and MC1321x devices, refer to the appropriate Reference Manuals and/or Data Sheets.

Contents

2 Introducing the SMAC

The SMAC is incorporated into the Freescale BeeKit Wireless Connectivity Toolkit. The incorporation of SMAC into BeeKit makes it easier for users to employ and customize the SMAC. BeeKit generates the project files and the project folder structure has changed from previous SMAC versions. Libraries are no longer used, everything in a project is in source code, and BeeKit generates a single target per project. Every BeeKit generated project runs transparently with CodeWarrior Special Edition

To use any of the existing applications available in SMAC, users must first generate the applications as projects in a BeeKit solution.

For more information about BeeKit, and BeeKit Projects and Solutions, refer to the *BeeKit Wireless Connectivity Toolkit User's Guide* (BKWCTKUG) and the BeeKit on-line help.

2.1 Supported Hardware

SMAC contains targets for the following Freescale hardware:

- Axiom-0308 with MC13192 RF Daughter Card
- 13192-SARD
- 13192-EVB
- 13213-NCB
- 13213-SRB
- Axiom-0308 with the MC1320x RF Daughter Card

The example provided in this section uses the Sensor Reference Board (SRB) as the hardware. Figure 1 shows the SRB after it was removed from its plastic case.

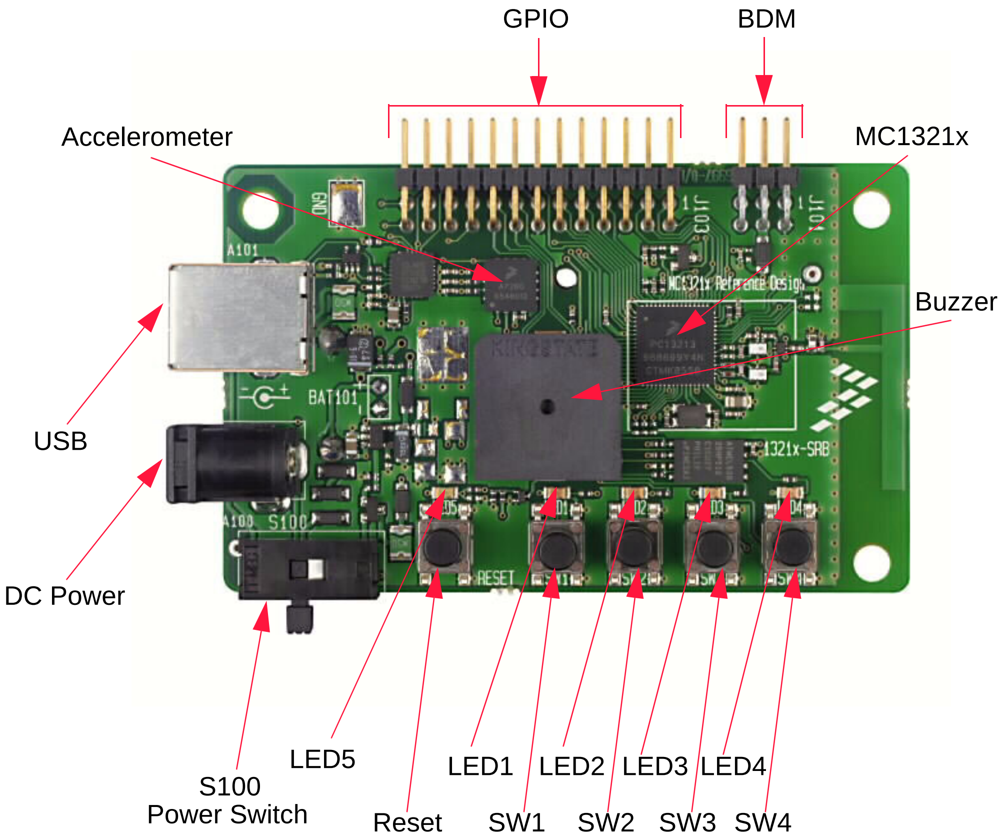

Figure 1. SRB Connectors, Buttons, Switches, and LEDs

2.2 SMAC Software

Be sure to download the latest version of the SMAC which is part of BeeKit, from the Freescale ZigBee web page at www.freescale.com/zigbee. Section 1, "Introduction" lists the currently available SMAC demonstration applications. Refer to the release notes for more information.

2.3 Generating an Application in BeeKit

This section shows how to generate an SMAC demonstration application using BeeKit.

1. Launch the BeeKit GUI and the BeeKit main window appears as shown in Figure 2.

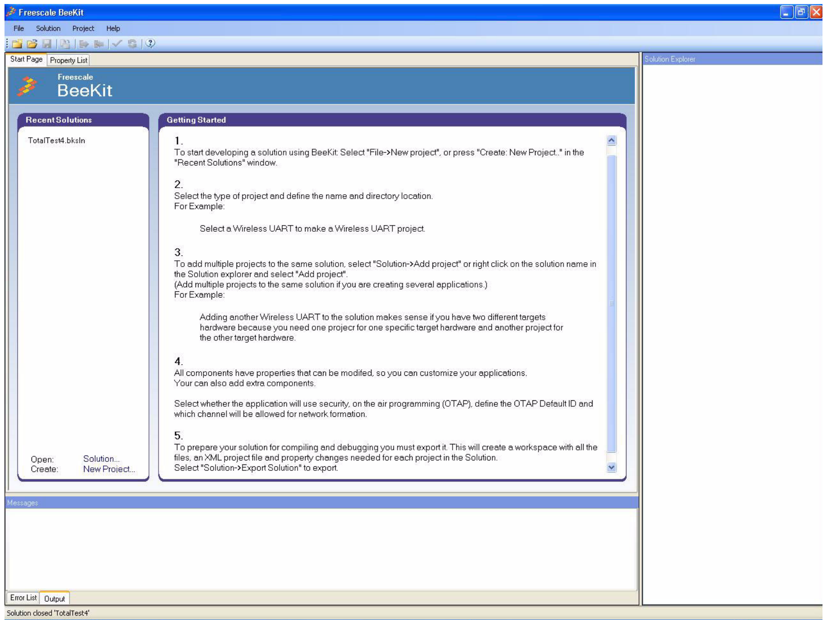

Figure 2. BeeKit Graphical User Interface

2. Select SMAC Codebase as shown in Figure 3.

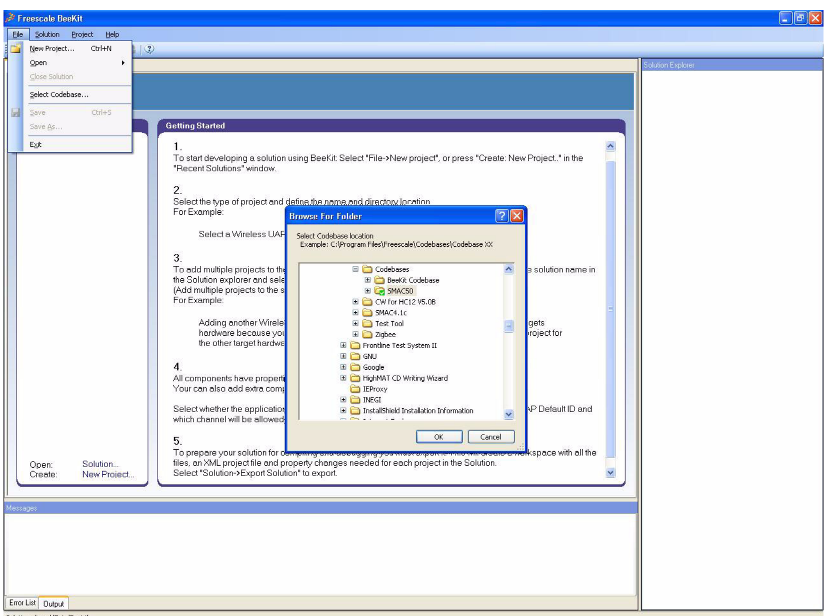

Figure 3. selecting SMAC Codebase

In BeeKit, a Codebase is a set of files and rules that permit users to generate the final applications. This step is important, because the SMAC Codebase is different than the Codebase of MAC and BeeStack.

3. Create a new project as shown in Figure 4.

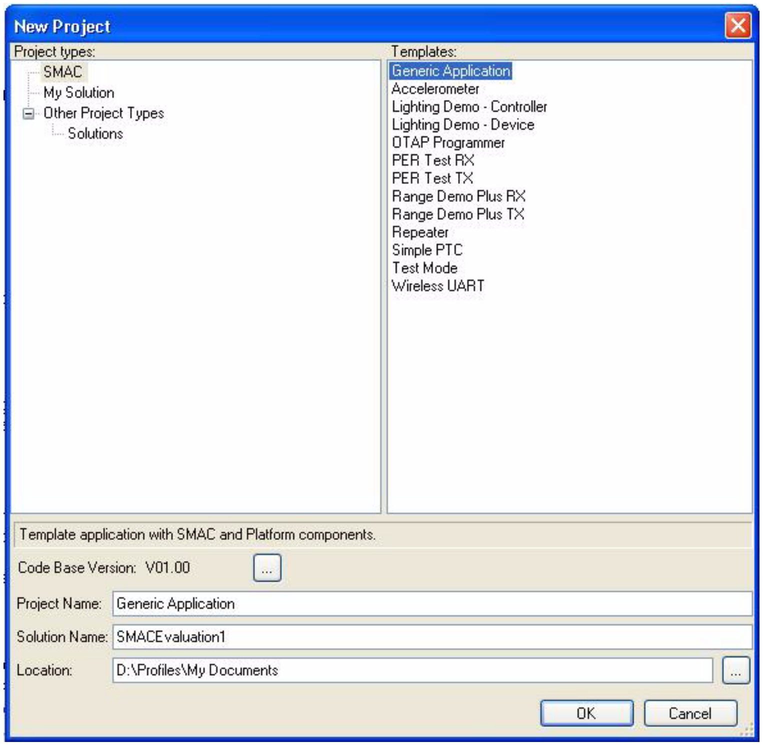

Figure 4. Creating a new project

If users create a new project and there is not a an existing solution, then users are also creating a solution. The solution must have a name. Specify a name for the solution.

4. Add other required projects for the evaluation as shown in Figure 5.

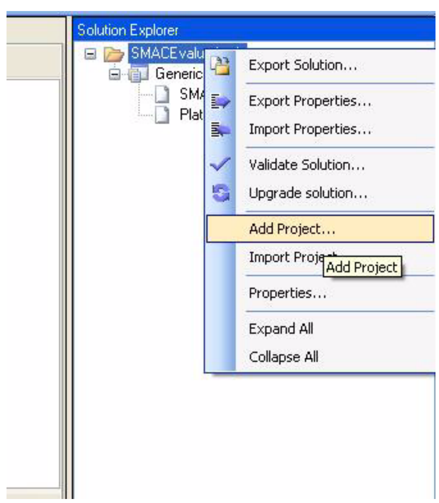

Figure 5. Adding a New Project to a Existing Solution

5. Modify the project properties as shown in Table 1.

Table 1. Software Component Properties

Component	Property	Description
SMAC	Security Enabled	Enables the security Module to automatically encrypt the information sent and received.
	OTAP Enabled	Enables the OTAP Module to allow over the air programming of the application
	Promiscuous Mode	If set to true, the SMAC Code Bytes are not sent prior to the message.
	OTAP Default ID	This is the default ID for the device.
	Security Type	This is the security engine used if the security enabled property is true. Currently, only the SIMPLE_XOR is implemented.
PLM	Target hardware	Allows board selection.
	LCD Enabled	Includes support for LCDs.
	Default SCI Port	Selects the SCI port. Freescale ZigBee boards usually have SCI Port 1 connected to a DB9 connector and SCI port 2 connected to a USB connector.
App	Default Channel	Default starting channel in the application.
	Output Power	Initial power output configuration.

6. Users must now validate the solution as shown in Figure 6.

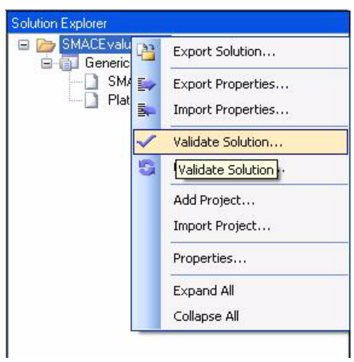

Figure 6. Validating a Solution

7. Users must now export the solution as shown in Figure 7.

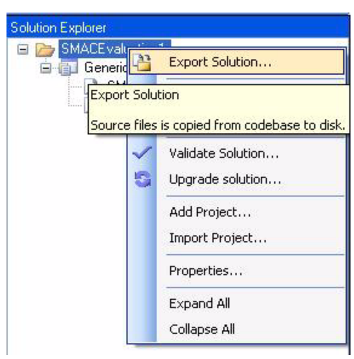

Figure 7. Exporting a Solution

2.4 Loading Applications Into a Board Using the BDM

After exporting from BeeKit, the projects are generated as a `.xml` file. Perform the following tasks to load the application to a board. This example uses the Basic PER Test.

1. As shown in Figure 8, import the `PER RX.xml` file, then save the project as `PER Rx.mcp`.

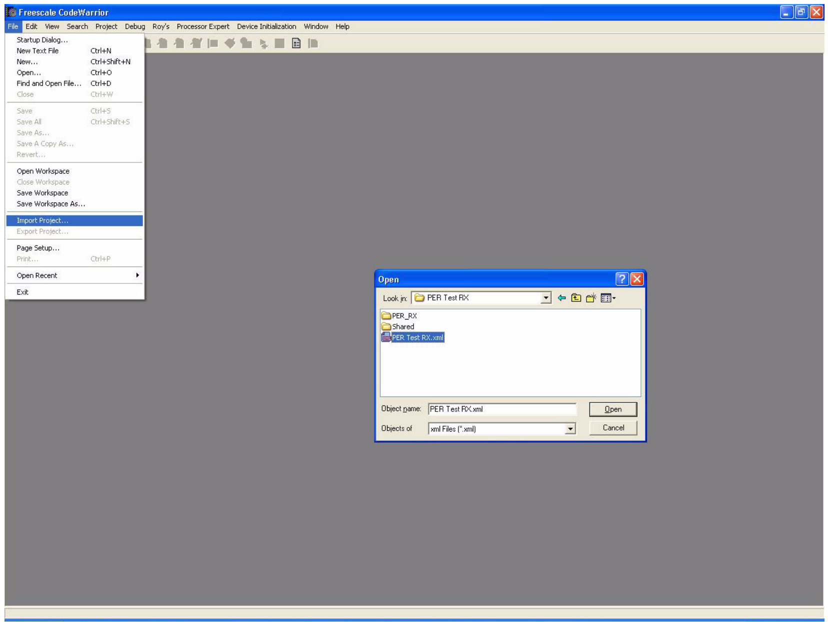

Figure 8. Importing a BeeKit .xml File

2. Compile the project by clicking the Compile button (Figure 9) which is located at the top of the Project Window as shown in Figure 10.

Figure 9. Compile Button

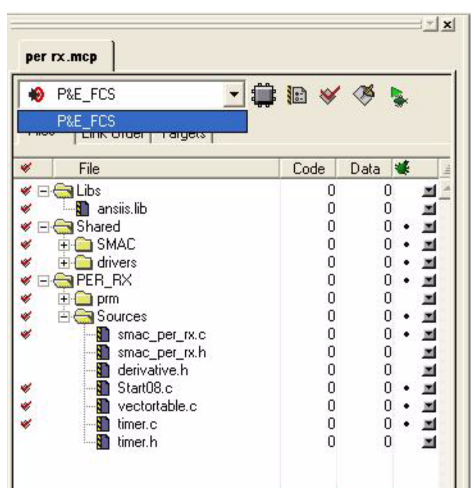

Figure 10. New Project Window

The structure of the project window displays various group files as follows:

- Libs — ANSI standard libraries are included
- Shared — In this group, the SMAC and the platform (PLM) source files are included.
- PER RX — This group contains the actual application source files as well as the linker file (`.prm`).

NOTE

Because of the integration into BeeKit, there is only one target, no `SMAC.lib` or `Target.lib` files, and no subprojects. This allows the use of a Special Edition license of CodeWarrior to transparently compile the projects.

The IDE builds the target if required and provides information messages, reports errors, and displays warnings as needed.

3. If no errors are encountered from the build process, connect the BDM to the target as shown in Figure 11.

   The blue LED on the P&E Multilink turns on if the connection to the PC is functioning.

   The yellow LED on the P&E Multilink turns on if the cable is properly connected to the target.

Figure 11. P&E Multilink Connected to an SRB

4. Click the Debug/Download button (Figure 12) which is located at the top of the `PER RX.mcp` Project Window (Figure 10).

Figure 12. Download Button

If the connection to the USB BDM debugger is in place and the polarity to the BDM interface is correct, the debugger tries to download the binary to the target. At that point, the ICD-Connection Manager window (Figure 13) appears where users have the ability to select the target interface.

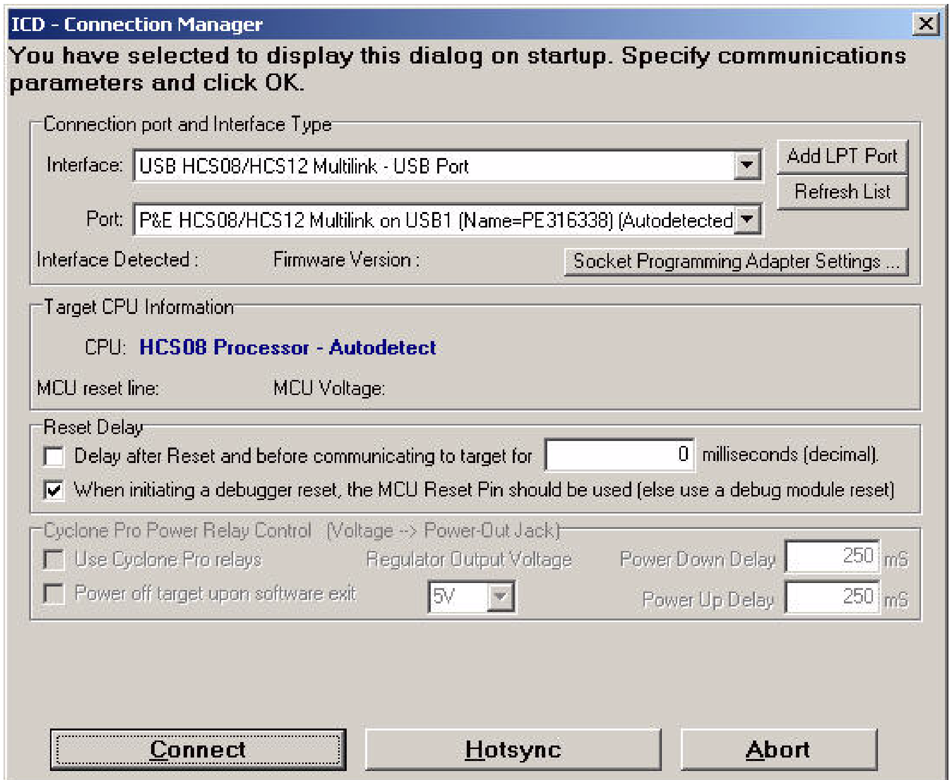

Figure 13. ICD Connection Manager Window (IDE)

5. The ICD Connection Manager window (Figure 13) shows the available connections to the debugger. Select the correct Interface and Port using the appropriate drop down menus and click the Connect button.
6. As shown in Figure 14, another window may appear with a request to erase the flash memory data on the target. Click the Yes button to Flash the board again.

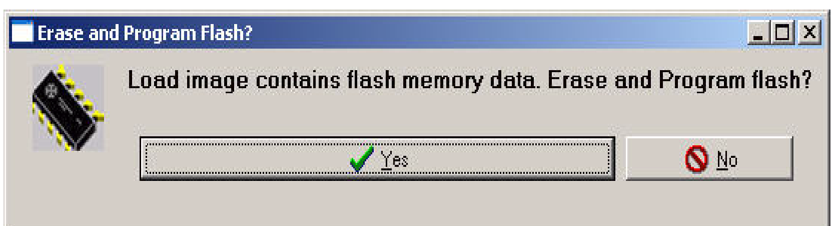

Figure 14. Erase and Program Flash Window

7. After the flash memory is programmed, the Default/Main debugger window appears as shown in Figure 15.

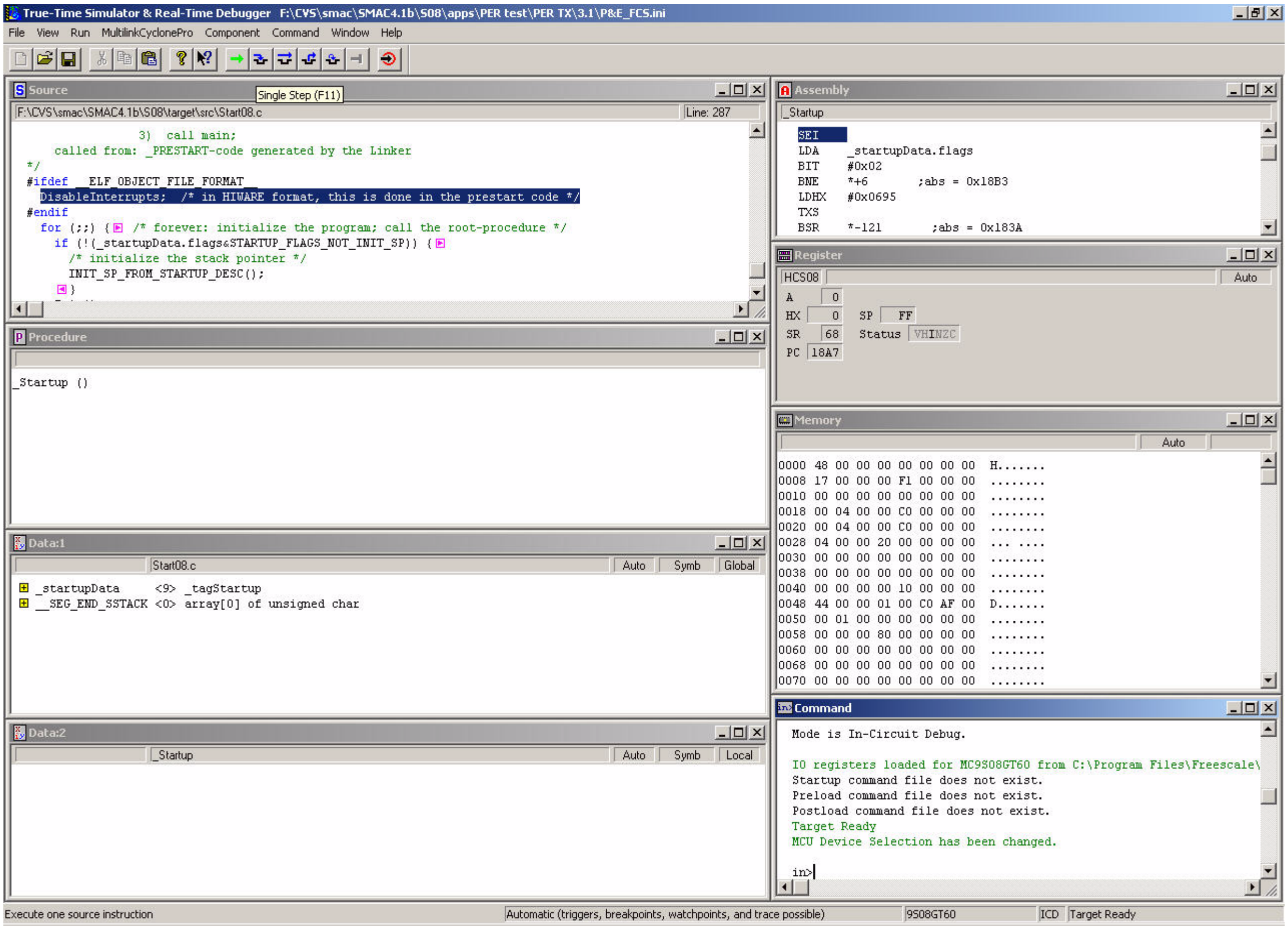

Figure 15. HiWave Debugger Main Window

8. To run the program, users have two options:
   a) Disconnect the BDM cable from the board and push the Reset button on the board once.
   b) Leave the BDM cable connected and click on the Run button in the HiWave Debugger Main window (Figure 16).

Figure 16. HiWave Debugger (Run Button)

The loading process described in this section applies to all currently available SMAC applications.

NOTE

If using the Freescale Embedded Bootloader instead of the BDM to load the SMAC applications, refer to the *802.15.4/ZigBee Embedded Bootloader Reference Manual* (802154EBRM) for more information about how to load bootloader enabled S19 files. The bootloader S19 files are built when compiling a project that was generated in BeeKit with the Bootloader property set to true. The S19 files are located in the bin folder of the mcp directory and have the `<project name>.s19` file name format.

2.5 IDE Considerations

SMAC does not currently support the device initialization feature of CodeWarrior. If a port to another MCU is required, users must employ the MCUs supported in BeeKit or the generated project must be ported manually.

3 Basic Packet Error Rate (PER) Test

3.1 Overview

The Basic Packet Error Rate (PER) test is a unidirectional test that sends 1000 packets from a transmitter to a receiver all on a single channel. The packet length is 18 bytes of data with 2 bytes of SMAC header for a total of 20 bytes of payload. The results of the packets received by the receiver can be monitored in two ways:

- Using a PC and a USB connection
- In a standalone mode by reading the count on the LEDs on the board

This test allows for basic PER testing using a PC.

3.2 Loading the Packet Error Rate Application

Prior to loading the Packet Error Rate application, the following applications must be generated using BeeKit using the steps shown in Section 2.3, “Generating an Application in BeeKit”.

- PER TX
- PER RX

1. Open the `PER TX.mcp` file as shown in Figure 17. Load the PER TX application into one board as shown in Section 2.4, “Loading Applications Into a Board Using the BDM”.

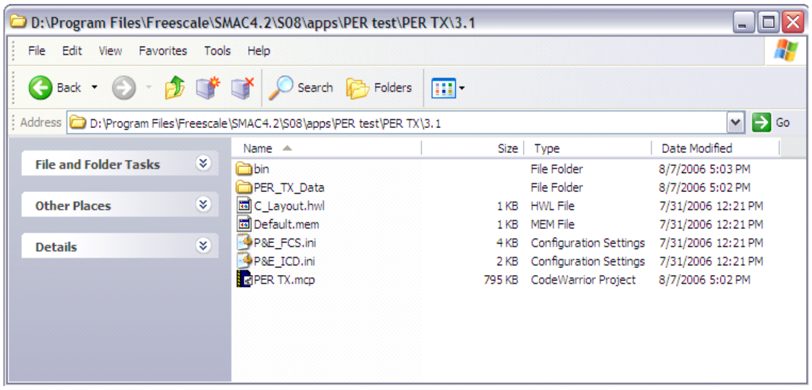

Figure 17. PER TX MCP File Location

3.3 Changing the Frequency of the Board

In there is an interference issue, or users need to simultaneously run more than one demonstration, the RF channel can be changed by pressing SW2 on both of the boards. Only four (4) frequencies are available:

- 2405 GHz
- 2430 GHz
- 2455 GHz
- 2480 GHz

For example, out of reset, LED4 is set to indicate that the first channel (2.405 GHz) is being used. To change channels, press SW2 on board one. The buzzer chirps and LED3 is set to indicate that the second channel (2.430 GHz) is now being used on board one. Press SW2 on board two which will set LED3 and thus match channels with board one.

Both boards are required to be on the same channel to work correctly. If communication problems exist, it is possible that one of the boards is on a different channel. Reset both boards to ensure that both are on the first channel (2.405 GHz), or push button SW2 until both boards LEDs match.

NOTE

There is no reason to change channels unless the system encounters an interference issue or users want to simultaneously run multiple demonstrations.

3.4 Setting Up the UART/USB Virtual Com Port

Figure 18 shows the default SMAC RS-232 settings.

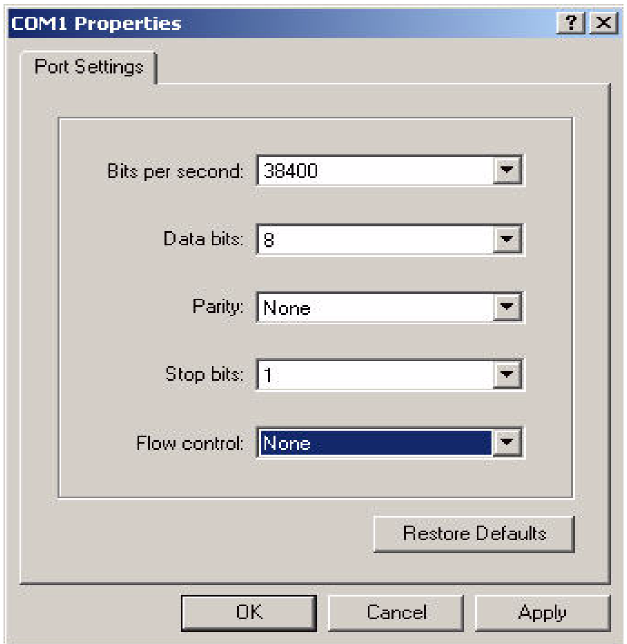

Figure 18. Default SMAC RS-232 Setting

1. Use a PC terminal program and set the correct baud rate, data bits, parity, COM port, and flow control.
2. In the PC terminal program, set the properties in the optional settings as shown in Figure 19.

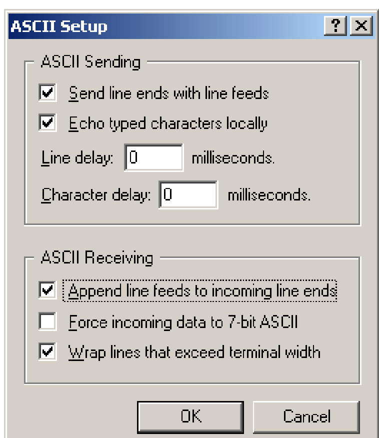

Figure 19. Additional Terminal program settings

3. Connect the PER RX radio to the PC with either a RS-232 cable or a USB cable. If using a USB cable, a corresponding USB driver is required. This driver can be found on the CD in the hardware kit or the CD image is available from the Freescale ZigBee web site at www.freescale.com/zigbee.
4. To check which COM port is being used by the USB, do the following:
   a) Open the System Properties window using Start->Settings->Control Panel->System.

b) Select the Hardware tab, and click the Device Manager button. The Device Manager window appears as shown in Figure 20.

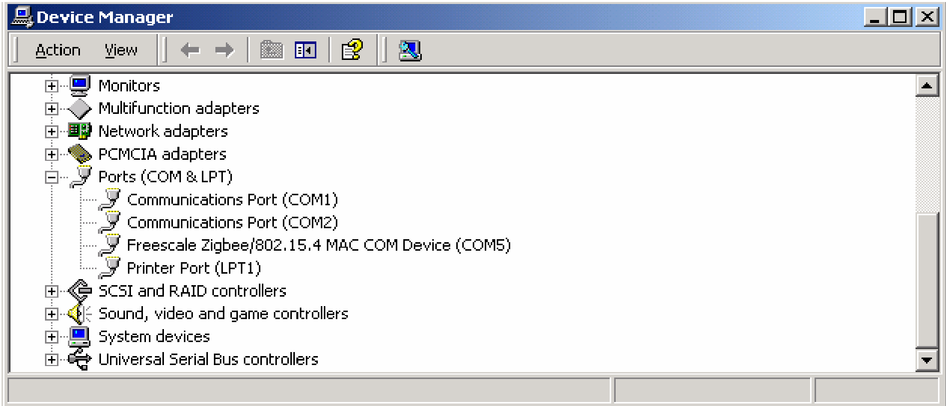

Figure 20. COM Port Determination in Device Manager

c) Scroll to the Ports label and expand the tree by clicking the "+" sign. This shows the COM ports in the system.

d) As shown in Figure 20, the COM Port chosen by the system is titled `Freescale ZigBee/802.15.4 MAC COM Device (COM5)`.

5. If the COM port chosen for `Freescale Zigbee/802.15.4 MAC COM Device` is not a port within Ports 1-10, then perform the following tasks:

   a) Double click on `Freescale Zigbee/802.15.4 MAC COM Device` in the Device Manager window. The Properties window appears.

   b) Select the Port Settings tab, and then click on the Advanced button.

6. Go to the Com Port Number drop down menu and select a COM port between 1-10 that is not in use.

3.5 Starting the PER Test

Ensure that both the PER-TX board and the PER-RX board are on the desired channel as described in Section 3.3, "Changing the Frequency of the Board". The PER-TX board and the PER-RX board are asynchronous to each other which means that the PER-TX board does not know when the PER-RX board is on. Users must place the PER-RX board into Listen Mode prior to setting the PER-TX board into Transmit Packet Mode. If this is done out of order, the reported packet error rate may be incorrect.

Press the Reset button on the PER-RX board. An Out of Reset or a Power Up message is echoed through the RS-232 interface and displayed on the PC as shown in Figure 21. To place the PER-RX board into Listen Mode, press SW1 on the PER-RX board. All four LEDs blink twice to indicate that the PER-RX board is in Listen Mode and a message to indicate that the PER-RX board is listening is echoed to the terminal as shown in Figure 22.

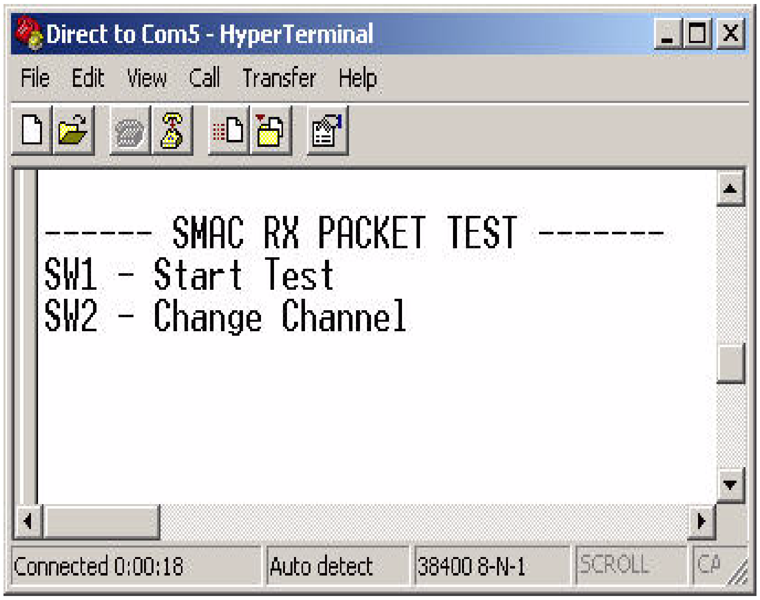

Figure 21. PER Power on Reset Message

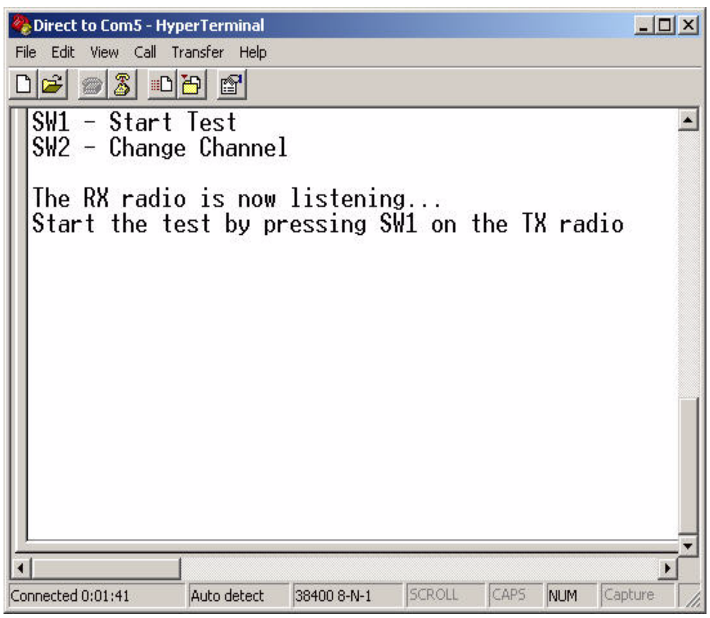

Figure 22. Output to the PC (From Receiving Board When in Listen Mode)

The PER-TX board setup follows a procedure similar to the PER-RX board setup. Place the PER-TX board on the same channel as the PER-RX board by pressing SW2. This ensures that the PER-TX board is on the same channel as the PER-RX board as described in Section 3.3, "Changing the Frequency of the Board". Press SW1 on the PER-TX board to place the application into TX mode. The PER-TX application turns on LEDs 1 through 4 in order to show that the PER-TX board is in Packet Transmit Mode and then it begins to transmit packets.

3.6 Basic PER operation

Since the PER-TX board and the PER-RX board are on the same channel and the PER-RX board started to listen before the PER-TX board sent any packets, the PER-RX board will toggle LED1 when a packet is being received. Then the PER-TX board blinks LED1 on every packet that is transmitted. If both boards are blinking, then both boards are on the same channel. If only the PER-TX board is toggling LED1, then either the PER-RX board is not in Listen Mode or it is listening to the wrong channel. If this occurs, reset both boards and try the procedure again without switching channels. If a packet is received, the radio echoes data to the PC through the UART connection as shown in Figure 23.

- N represents the number of packets received
- L is the length of the packet
- LQI is the Link Quality Indicator (measured in dBm)
- CRC indicates if the Cyclical Redundancy Check (CRC) is good
- Data is the hex formatted payload

```
N:980 L:18 LQI=-24 CRC=1 Data=000102030405060708090A0B0C0D0E0F1011
N:981 L:18 LQI=-24 CRC=1 Data=000102030405060708090A0B0C0D0E0F1011
N:982 L:18 LQI=-24 CRC=1 Data=000102030405060708090A0B0C0D0E0F1011
N:983 L:18 LQI=-24 CRC=1 Data=000102030405060708090A0B0C0D0E0F1011
N:984 L:18 LQI=-24 CRC=1 Data=000102030405060708090A0B0C0D0E0F1011
N:985 L:18 LQI=-24 CRC=1 Data=000102030405060708090A0B0C0D0E0F1011
N:986 L:18 LQI=-24 CRC=1 Data=000102030405060708090A0B0C0D0E0F1011
N:987 L:18 LQI=-24 CRC=1 Data=000102030405060708090A0B0C0D0E0F1011
N:988 L:18 LQI=-24 CRC=1 Data=000102030405060708090A0B0C0D0E0F1011
N:989 L:18 LQI=-24 CRC=1 Data=000102030405060708090A0B0C0D0E0F1011
N:990 L:18 LQI=-24 CRC=1 Data=000102030405060708090A0B0C0D0E0F1011
N:991 L:18 LQI=-24 CRC=1 Data=000102030405060708090A0B0C0D0E0F1011
N:992 L:18 LQI=-24 CRC=1 Data=000102030405060708090A0B0C0D0E0F1011
N:993 L:18 LQI=-24 CRC=1 Data=000102030405060708090A0B0C0D0E0F1011
N:994 L:18 LQI=-24 CRC=1 Data=000102030405060708090A0B0C0D0E0F1011
N:995 L:18 LQI=-24 CRC=1 Data=000102030405060708090A0B0C0D0E0F1011
N:996 L:18 LQI=-24 CRC=1 Data=000102030405060708090A0B0C0D0E0F1011
N:997 L:18 LQI=-24 CRC=1 Data=000102030405060708090A0B0C0D0E0F1011
N:998 L:18 LQI=-24 CRC=1 Data=000102030405060708090A0B0C0D0E0F1011
Good:   999/1000

Halt
_
```

Figure 23. The UART output during PER test

When the PER-TX board is done sending 1000 packets, the PER-TX board sends packets with the payload “DONEDONE”. If the PER-RX board receives the “DONEDONE” packet, the application echoes the total number of good packets received. The measured PER is echoed to the terminal in a 999/1000 format. Also, the PER value is echoed to the LEDs on the PER-RX board. For example, if the measured packet error rate was 983/1000 then at the completion of the test, the value 1001b (9d), 1000b (8d), 0011b (3d) will blink. This allows PER to be measured without the need for a PC.

3.7 Restarting the Basic Packet Error Rate Application

The first task to perform for a restart is to reset both boards and repeat the procedures as shown in Section 3.5, “Starting the PER Test”. However, this may result in resetting the RF channel back to Channel 0 (2.405 GHz). If the channel was changed from the default Channel 0 and the test needs to be repeated on the changed channel, press SW1 on the PER-RX board and then press SW1 on the PER-TX board to repeat the test. If users need to choose a different channel, then use SW2 to change the channel on both the PER-TX board and PER-RX board as described in Section 3.3, “Changing the Frequency of the Board”.

4 Wireless UART Demonstration

The Wireless UART application allows the Freescale ZigBee family of boards to communicate at typematic rates from one board to another over RS-232 cables or USB virtual com ports.

The following is a simplified example and the code as it exists cannot be used as a cable replacement. If cable replacement is the goal, then queues, buffers, and other constructs must be added to increase the reliability and efficiency of this demonstration.

Prior to loading the Wireless UART application, the application must be generated using BeeKit using the steps shown in Section 2.3, “Generating an Application in BeeKit”.

1. Connect the boards to their respective ports and select a Com Port using the Device Manager as shown in Section 3.4, “Setting Up the UART/USB Virtual Com Port”. Once the Com Ports are determined, open the Com Port with the settings as shown in Figure 18.
2. Open the `Wireless Uart.mcp` file as shown in Figure 24.

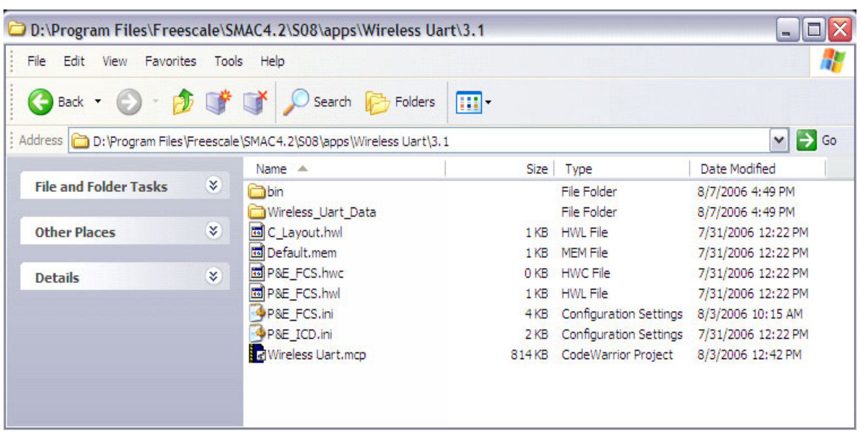

Figure 24. Wireless UART MCP File Location

3. Load the Wireless UART application into two boards as shown in Section 2.4, “Loading Applications Into a Board Using the BDM”.
4. Reset both boards. The application’s Out of Reset text echoes a welcome message as shown in Figure 25.

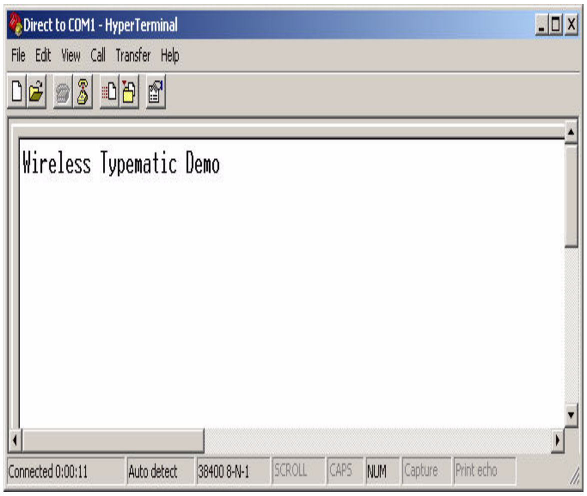

Figure 25. Wireless UART Reset UART Message

5. Once both boards are reset and proper RS-232 connectivity is established, type some characters in the HyperTerminal Session (Board 1) window (Figure 26) and the typed message will appear on the other PC's HyperTerminal Session (Board 2) window (Figure 27).

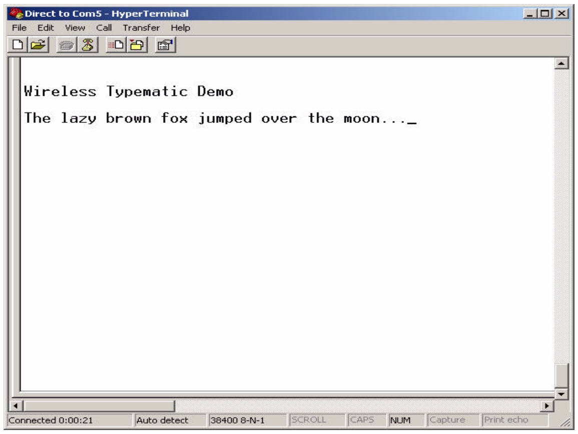

Figure 26. Wireless UART Board 1 (Output to Board 2)

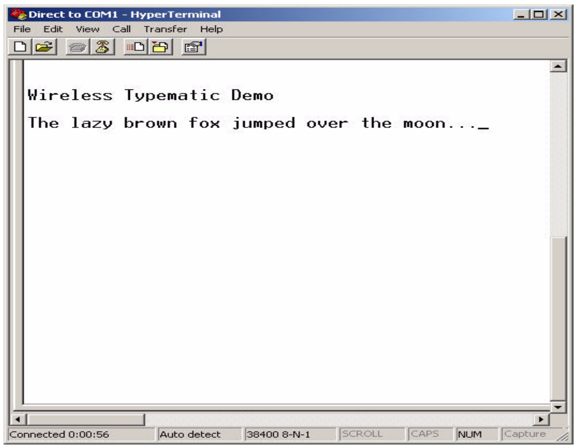

Figure 27. Wireless UART Board 2 (Input from Board 1)

This is a two-way communication protocol where the boards will retry their packets up to three times if an acknowledgement is not received. This application highlights a very basic SMAC wireless UART implementation and as already stated, because it is a basic demonstration application, it is not intended for large file transfers.

5 Accelerometer Demonstration

The Accelerometer Demonstration application shows various uses of the Freescale X, Y, and Z axes accelerometers. For a complete overview of the Accelerometer Demonstration application, see AN3232, *Accelerometer Demonstration With the 13213-SRB (Sensor Reference Board)*.

5.1 Loading the Accelerometer Application

1. Flash the Accelerometer Demonstration into two boards. One board must be either a 13192-SARD or an SRB to provide the X, Y, and Z axes accelerometer data and the other board must have an available UART and be able to connect to a PC. This example uses two SRBs.
2. If users need to run the Accelerometer on different targets, those projects must be generated by BeeKit as described in Section 2.3, "Generating an Application in BeeKit". Import the `.xml` files and generate the CodeWarrior `.mcp` project files as described in Section 2.4, "Loading Applications Into a Board Using the BDM".
3. Download the appropriate target image by following the procedure in Section 2.4, "Loading Applications Into a Board Using the BDM", with the `Accelerometer V3.mcp` project.

5.2 Board Setup

To setup both boards for operation (transmit and receive), perform the steps shown in this section. See Figure 1 for a picture of the SRB.

5.3 Board One Setup (PC_Radio Board)

1. Connect a SRB to the PC using the USB cable included in the kit.
2. Connect the USB bus to the USB hub. This provides power to the board through the USB bus. Power on the board by setting the Power Switch (S100) to the ON position.
3. Press the Reset Button once. The LEDs are off, but the PC_Radio application is running and in receive mode.

5.4 Board Two Setup (Accelerometer Board)

1. Connect 9 VDC power to DC power connector with the 9 VDC wall transformer included in the kit. Power on the board by setting the Power Switch (S100) to the ON position. Another board power option is to use the batteries which provides the advantage of true wireless operation (no cables) and allows users to easily move and manipulate the SRB.
2. Press the Reset Button once.
3. Press button SW1 once and LED3 will momentarily flash to indicate Application Two is running (Accelerometer Mode).

4. The Accelerometer application sends a ping packet to the PC_Radio every two seconds. This is indicated on the Accelerometer Board by a blinking LED2.
5. The Accelerometer application detects when the Accelerometer Board is moved and quickly toggles LED1, then sends a packet to the PC_Radio Board with an updated reading.

5.5 PC Setup

1. If the Evaluation Kit tools are already installed, then users can run the Triax software by navigating to the link using the Start menu.
2. From Windows, select Start->Programs->Freescale ZigBee->Test Tool->Triax.
   a) If the Triax application does not exist, go to Step 3 to install it.
   b) If the application is in the Start Menu, go to Step 8.
3. To run only the Triax application, insert the CD supplied with the kit into the PC CD-ROM drive.
4. If autorun is not enabled on the PC, locate the Autorun directory on the CD and run the `CDBrowser.exe` file.
5. After launching the Evaluation Kit CD Browser, select Install/Uninstall.
6. Select the SMAC packet under Additional.
7. Run the installation and select the appropriate install folder.
8. Run the `Triax.exe` file from the selected install location on the PC hard drive.
9. The Triax application is set up to run on COM ports 1-10. To check which COM port is being used by the USB, perform the following tasks:
   a) Open the System Properties window by clicking on Start->Settings->Control Panel->System.
   b) Select the Hardware tab, and click the Device Manager button. The Device Manager window appears as shown in Figure 28.
   c) Scroll to the “Ports” label and expand the tree by clicking the “+” sign to show the COM ports in the system.
   d) The COM Port selected by the system is titled `Freescale ZigBee/802.15.4 MAC COM Device (COM5)` as shown in Figure 28.

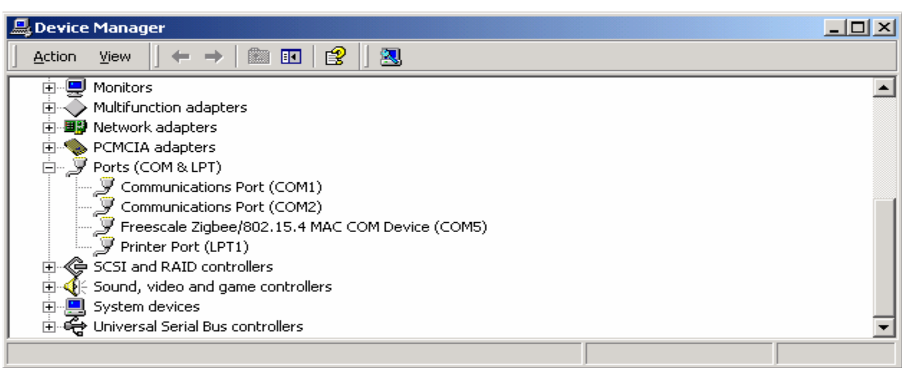

Figure 28. COM Port Determination in Device Manager

10. If the COM port chosen for `Freescale Zigbee/802.15.4 MAC COM Device` is not a port within Ports 1-10, then perform the following tasks:
    a) Double click on `Freescale Zigbee/802.15.4 MAC COM Device` in the Device Manager window and the Properties window will appear.
    b) Select the Port Settings tab and then click the Advanced button.
    c) Go to the Com Port Number drop down menu and select a COM port between 1-10 that is not in use.

5.6 Verifying Operation

1. Check for the ping packet. The Accelerometer application sends a ping packet to the PC_Radio every two (2) seconds as indicated by LED2. The Accelerometer Board quickly blinks LED2 indicating that the ping packet is sent. The PC_Radio receives the ping and toggles LED2 upon reception. Once connectivity between the PC_Radio and the Accelerometer Board is verified, go to Step 2.
2. To check if the Accelerometer Board is working, move the Accelerometer Board. LED1 blinks, which indicates that the Accelerometer Board has detected movement.
3. With the default application, data that represents movement of the Accelerometer Board is sent to the PC_Radio. When the PC_Radio receives this data, LED2 toggles.
4. Select the Raw Data applet from the Triax application. The Raw Data applet shows the X, Y, and Z axes A/D values as reported by the Accelerometer Board. When the board is laying flat, the raw values should read approximately 0g for both the X and Y axes. However, the Z axis should read approximately 1g. As shown in Figure 29, when the Accelerometer Board is moved, the values of the X, Y, and Z axes are updated on the Raw Data applet.

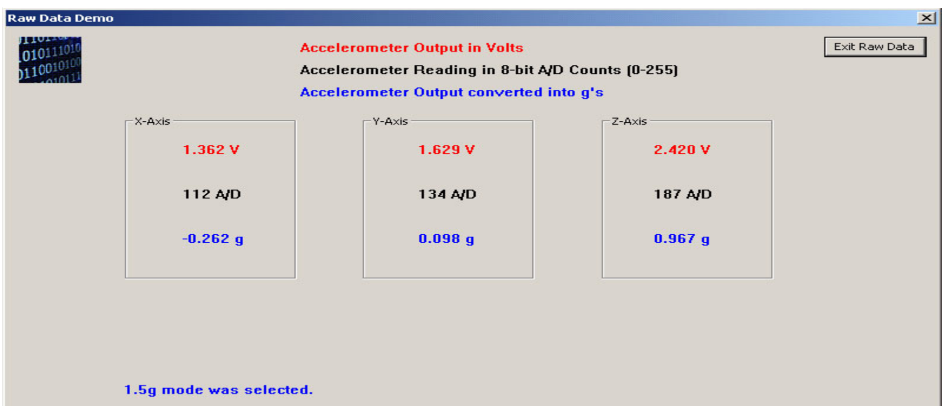

Figure 29. Raw Data Demonstration

NOTE

To improve accuracy of the accelerometer readings, refer to Section 5.7, "Setting Up the Calibration Application".

The Triax main window appears as shown in Figure 30.

Figure 30. Triax Main Window

When the accelerometer is moved, the Accelerometer Board sends data to the PC_Radio to update its data. Data is only sent when it changes within a small tolerance.

The receiver receives these packets and updates its cached value of the accelerometer data. The PC periodically sends a command string through the USB to the PC_Radio that requests the accelerometer data. This data is interpreted by the Triax software to enable several applications and demonstrations. Again, for a complete overview of the accelerometer demonstration, see AN3232, *Accelerometer Demonstration With the 13213-SRB (Sensor Reference Board).*

5.7 Setting Up the Calibration Application

In order to verify and visualize the calibration procedure, the `Triax.exe` PC GUI must be running, but it is not required.

1. Press the Reset button on the PC_Radio Board once. (Out of reset, the PC application is running.)
2. Press the Reset button on the Accelerometer Board once.
3. Place the Accelerometer Board on a flat surface. (component side up)
4. Press SW1 on the Accelerometer Board to advance the application to the Accelerometer Mode (Application Two). LED3 momentarily lights to indicate that Application Two is running. The Accelerometer Board sends out a ping packet every two (2) seconds and updates the data on the XYZ Demo PC application each time the board is moved.
5. If the data points for the X, Y, and Z axes shown on the PC GUI are not near the origin of the graph, then continue to Step 6.

6. Press SW4 on the Accelerometer Board once until all LEDs are lit. This shows that the calibration is being performed.
7. Verify that the Accelerometer Board is calibrated by viewing the X, Y, and Z axes on the PC GUI. The squares on the display should be approximately at axis point 0,0. If the calibration is still incorrect, repeat the calibration process until the data points are nominally at axis point 0,0.
8. To quit, exit the application in the GUI and press reset on both boards. Press SW1 on the Accelerometer Board once to return to main application functionality.

6 Range Demonstration Plus

The Range Demonstration Plus application consists of two binary images:

- The TX range demo plus application (TX_APP)
- The RX range demo plus (RX_APP)

  The TX_APP continuously transmits a packet at a certain rate and waits for an ACK packet from the RX_APP.

1. These applications must be generated by BeeKit as shown in Section 2.3, “Generating an Application in BeeKit”. Import the `.xml` file and generate the `.mcp` file as described in Section 2.4, “Loading Applications Into a Board Using the BDM”.
2. Download the appropriate target image by following the procedure shown in Section 2.4, “Loading Applications Into a Board Using the BDM” with the `Range Demo Plus RX.mcp` into one board and `Range Demo Plus TX.mcp` into another board.
3. The TX_APP sends a packet with a payload of “ZigBee” to the RX_APP. If the RX_APP receives the “ZigBee” packet, LED1 toggles.
4. The Link Quality Indicator (LQI) is also read and the following indications apply:
   — If the LQI is greater than -40 dBm, the RX_APP turns on LEDs 2, 3, and 4 to indicate highest LQI.
   — If the LQI returns -60 through -40 dBm, the RX_APP turns on LEDs 2 and 3.
   — If the LQI returns -80 though -60 dBm, the RX_APP turns on LED2.
   — If the LQI reports <-88 dBm, the RX_APP leaves LEDs 2, 3, and 4 off.

   Power ranges can be adjusted by modifying the SPEC_POWER, L1_POWER, L2_POWER, and L3_POWER defines in the `range_demo_plus_header.h` header.

Once the RX_APP decodes the power range for the last packet, it forms an ACK packet with the decoded power bin at the end of the ACK packet.

- For a strong signal in the L3_POWER range, the ack packet is “ACK3” which is interpreted by the TX_APP to turn on the three LEDs on the TX_APP side.
- For a signal in the L2_POWER range, the ack packet is “ACK2” which is interpreted by the TX_APP to turn on the two LEDs on the TX_APP side.
- For a signal in the L1_POWER range, the ack packet is “ACK1” which is interpreted by the TX_APP to turn on the one LEDs on the TX_APP side.

- For a signal in the SPEC_POWER range, the ack packet is "ACK0" which is interpreted by the TX_APP to turn off the three LEDs on the TX_APP side.

The TX_APP receives the ACK packet and turns on the appropriate number of LEDs and toggles LED1. In this way, users can perform rough power characterizations by approximating the power received by the RX_APP on either the RX_APP or the TX_APP board.

6.1 Range Demonstration

1. Reset both boards.
2. Out of reset, the TX_APP walks through LEDs 1 through 4 to indicate that it is a transmitter while the RX_APP blinks all four LEDs twice. This allows users to know which is the transmitter and which is the receiver.
3. Because the boards are physically located close to each other, packets should be received by both boards and LED1 should be toggling to indicate that packets are being received. LEDs 2 through 4 are on which indicates a strong signal because the boards are close to each other.
4. As the distance increases between the TX_APP and RX_APP boards, the LEDs should begin to turn off one by one. When only LED1 is blinking, the radio should be operating at approximately -88 dBm.

7 Lighting Demonstration

This section describes the Lighting Demonstration.

1. The following applications must be generated by BeeKit as described in Section 2.3, "Generating an Application in BeeKit".
   — Lighting Demo — Controller
   — Lighting Demo — Device
2. Import both projects as described in Section 2.4, "Loading Applications Into a Board Using the BDM", and open them.
3. Download the appropriate target images by following the procedure as described in Section 2.4, "Loading Applications Into a Board Using the BDM" with the `Lighting_demo_device.mcp` into one board and `Lighting_demo_controller.mcp` into another board.

7.1 Lighting Controller Selection and Operation

This section demonstrates how the board loaded with the Lighting_demo_controller application is used as the Lighting Controller.

Upon power-up or reset, any push button can be pressed and held to manually select an operating channel. On power-up or reset, all LEDs flash for one (1) second to indicate that the board is a Lighting Controller. Next, the group number (i.e. SERNUM) LED is flashed on for one (1) second to identify the group. If the channel is selected manually, the respective push button is released and the channel LED flashes for one (1) second. If channel selection is automatic, all LEDs are on while the unit searches for the clearest channel. Once this channel is detected, the respective LED flashes on for one (1) second and initialization of the board as a Lighting Controller is complete.

To use the Lighting Controller, perform the following steps.

1. Press SW1 to wake the controller out of its very low-power mode (i.e. MCU in STOP3 and MC1320x in Reset.)
2. The LEDs cycle each time SW1 is pressed to select a desired remote board. If the push button remains inactive for approximately four (4) seconds, the controller returns to the very low power state.
3. Press SW2 to select the desired remote board's light (LED).
4. Press SW3 to toggle the destination remote on/off. Remember, the remote board is selected using SW1 and the LED is selected using SW2.
5. The remote board acknowledges the toggle command and if received successfully by the controller, is indicated by flashing all LEDs on one time.
6. Instead of pressing SW3, users can press SW4 to determine the status of the selected board's light. If the status is received successfully by the Lighting Controller, the first two LEDs are turned on and the last LED indicates the status (on/off). If either the remote acknowledges or status is not received by the Lighting Controller, all the LEDs blink on/off several times. After any completed operation, the Lighting Controller automatically returns to the low-power state. The program assumes that all LEDs are connected to the same PORT. The target platform may be selected in the `pub_def.h` file. Many application specific details are located in the in `remote_controller.h` file.

7.2 Lighting Device Selection and Operation

This section demonstrates how the board loaded with the Lighting_demo_device application is used as the Lighting Device.

On power-up or reset, the LEDs cycle for one (1) second to indicate that the board is a Lighting Device. Next, the group number (i.e. SERNUM) LED flashes on for one (1) second to identify the group. Then the board pauses and waits for the operator to select a desired channel by pressing the corresponding push button. The selected channel flashes the LED on for one (1) second. The board again pauses and waits for the operator to select the desired device number by pressing the corresponding push button. The selected device number flashes the LED on for one (1) second and initialization of the board as a Lighting Device is complete.

To use the Lighting Device, any push button may be pressed to toggle the status of the corresponding LED. This is for local control. If addressed (and if the SERNUM matches), the Lighting Controller may request that any LED be toggled, or the Lighting Controller may request the status of an individual LED. The Lighting Device retransmits any data received and intended for another device within the SERNUM grouping, acting as a repeater. The Lighting Device has a unique “group” number (i.e. SERNUM) and device number (i.e. DEVICE_NUM). The program assumes that all LEDs are connected to the same PORT. The target platform may be selected in the `pub_def.h` file. Many application specific details are located in the `lighting_demo_device.h` file.

8 Test Mode Application

The Test Mode application is an easy way to test the RF performance of the transmitter for basic transmitter tests. The Test Mode application is a collection of test modes that consist of the following:

- Idle
- TX 9th Order Binary Polynomial (PRBS9)
- RX_ON
- TX_ON with modulation
- TX_ON Continuous Wave (CW) without modulation

These tests allow for basic RF characterization of the transmitter. To start the test, perform the following steps:

1. The Test Mode application must be generated by BeeKit as described in Section 2.3, "Generating an Application in BeeKit". Import the generated `.xml` file into CodeWarrior and generate the `.mcp` file as described in Section 2.4, "Loading Applications Into a Board Using the BDM".
2. Download the appropriate target image by following the procedure in Section 2.4, "Loading Applications Into a Board Using the BDM" with the `Test Mode App.mcp` into one board.

8.1 Test Mode Application Operation

Out of reset, the board is waiting for either a channel selection via SW2 as described in Section 3.3, "Changing the Frequency of the Board" or waiting for SW1 to be pressed to cycle through the various modes. The modes are as follows:

- Mode 1 — Idle
- Mode 2 — 9th Order Binary Polynomial (PRBS9)
- Mode 3 — RX_ON
- Mode 4 — TX with modulation
- Mode 5 — TX with modulation off (CW mode)

By default, the mode is displayed on the LEDs after SW1 is pressed for the first time. If SW1 is pressed and the radio is in Mode 5 the application wraps the mode back to Mode 1.

8.2 Spectrum Analyzer Captures

The measurements shown in this section are example illustrations only and were captured from a SRB using the F-antenna. These measurements are radiated measurements and not conducted measurements.

8.2.1 Test Mode 1 (Idle Mode)

The Idle Mode captures the noise floor as shown in Figure 31.

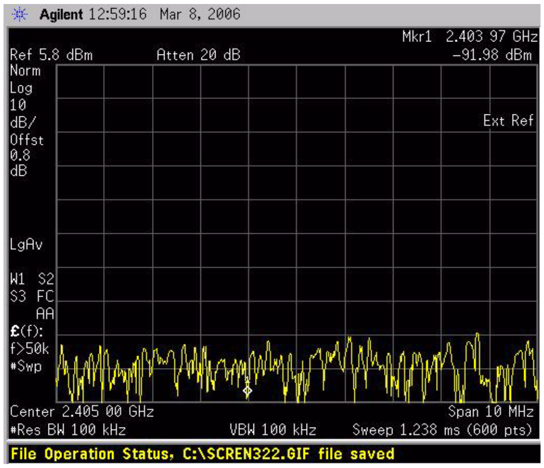

Figure 31. Spectrum Analyzer Capture (Idle Mode)

8.2.2 Test Mode 2 (PRBS9)

This mode captures the 9th Order Binary Polynomial as shown in Figure 32.

The capture shown in Figure 32 is set to an average of 16 frames. Averaging is turned off, so the spectrum analyzer trace has discontinuities and the response as shown, is less than smooth.

Figure 32. Spectrum Analyzer Capture (PRBS9 Mode)

8.2.3 Test Mode 3 (RX_ON)

Figure 33 shows the RX_ON Test Mode. Because the radio is in receive mode, their is little or no energy captured in the 2.4 GHz band.

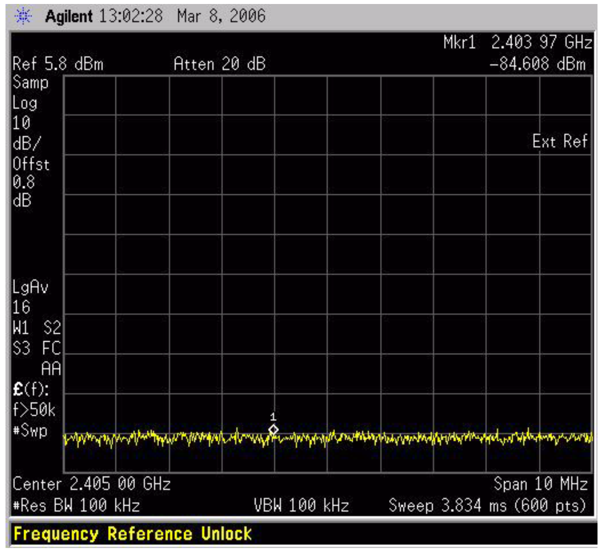

Figure 33. Spectrum Analyzer (RX_ON Mode)

8.2.4 Test Mode 4 (TX With Modulation)

Figure 34 shows transmitting with modulation.

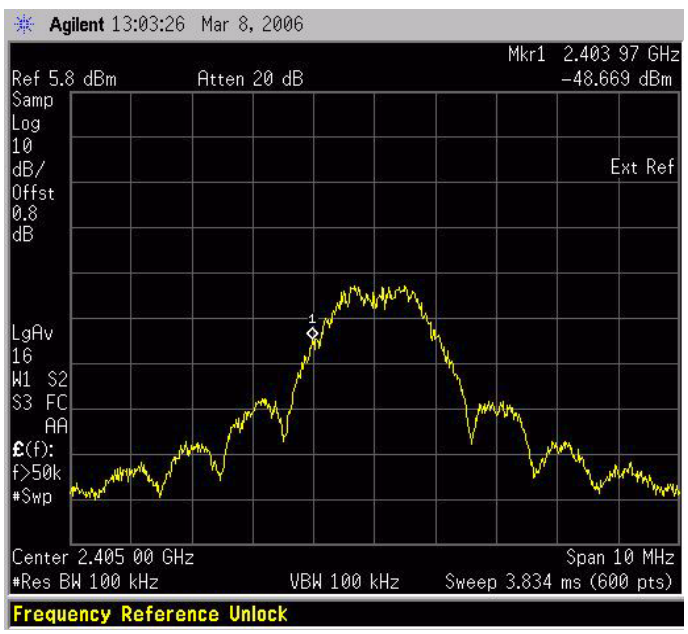

Figure 34. Spectrum Analyzer (TX with Modulation)

8.2.5 Test Mode 5 Transmit Without Modulation (CW Mode)

Figure 35 shows Continuous Wave (CW) Mode.

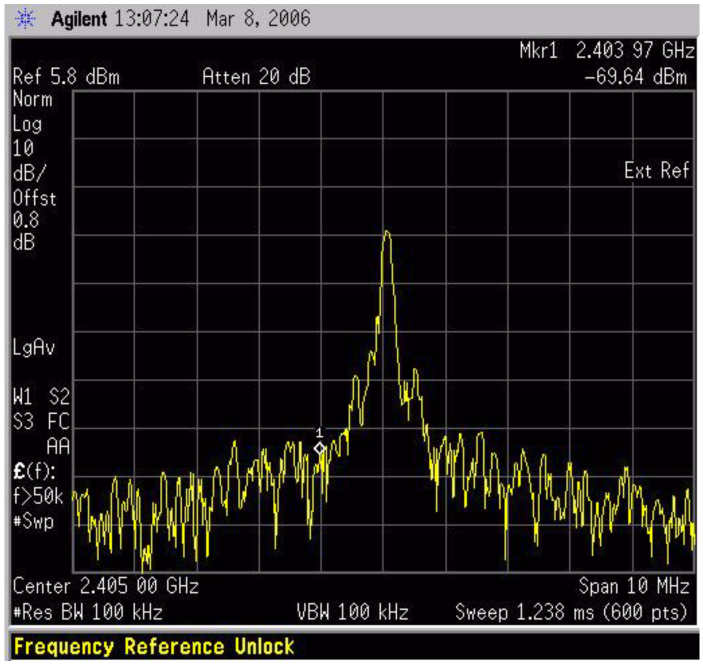

Figure 35. Spectrum Analyzer Capture (CW Mode)

9 Repeater Application

The Repeater application repeats all the messages received in a specific channel in order to extend the range of a Point-to-Point link. The Repeater application is configured through the board's UART port to determine the working channel, the output power, and statistics about packets received and repeated.

9.1 Downloading the Repeater Application

1. Generate the Repeater application using BeeKit using the procedure described in Section 2.3, "Generating an Application in BeeKit".
2. Import the `.xml` file generated by BeeKit and create the `.mcp` file as described in Section 2.4, "Loading Applications Into a Board Using the BDM".

9.1.1 Compiling the Application

To compile the Repeater application, perform the following steps:

1. From the IDE window, click on the Build button to compile the application.
2. After clicking on this button, the specific `.s19` file for the chosen target is generated in the following directory:

```
<Solution folder>\<Project folder>\bin
```

9.1.2 Downloading the Code into the Application

To download the Repeater application code, follow the steps as previously stated in Section 2.4, "Loading Applications Into a Board Using the BDM".

To download code into the Repeater application for Embedded Bootloader targets, follow the instructions as described in the *Simple Media Access Controller (SMAC) User's Guide* (SMACRM).

9.2 Repeater Functionality

Prior to evaluating the Repeater application, the following hardware and software are required.

- A serial cable that connects a PC to the board.
- A HyperTerminal program or another compatible communications software opened and configured for serial communications using the following parameters:
  - — 8 Bits
  - — No Parity
  - — 1 Stop
  - — 38400 Baud
  - — Ensure that the Append line feeds to incoming line ends option is selected.

1. Turn on the board and the following menu appears on the PC display:

```
-------------- Repeater Application V0.0 --------------
<s> Show Statistics:
<0> Reset Statistics:
<r> Repeat menu
<m> Switch Repeater mode:
<p> Change Power:
<d> Change Delay length:
<c> Switch Channel:
<z> EXECUTE
```

2. To perform the following action, press the appropriate key.

   s — Pressing this key shows the current Repeater statistics

   0 — Pressing this key resets the current Repeater statistics

   r — Pressing this key re-displays these key options

   m — Pressing this key toggles the operating mode as follows:

      Dummy Mode [immediate TX]

      Normal Mode [wait then TX]

   p — Pressing this key scrolls through the configured power output

   c — Pressing this key scrolls through the current channels

   d — Pressing this key sets the time to wait before transmit when running in Normal Mode

   z — Pressing this key starts and stops the Repeater application

3. After choosing one of these options, press the “z” key to start and stop the Repeater application as needed. All data received on a specific channel is sent and displayed through the UART interface and appears on the PC display that is running the HyperTerminal or other appropriate communications program.

   As shown in Figure 36, users can verify if the Repeater application is working by setting up a test using two boards that are physically set far enough apart so that they can not actually communicate with each other. (The Range Demo Plus application can be used for this test. See AN3231 for a description of the Range Demo Plus application.) Use a third target that is running the Repeater application, power it up, and physically place it in range of both of these boards. Both boards should start normal communication once the board running the Repeater application is set in place.

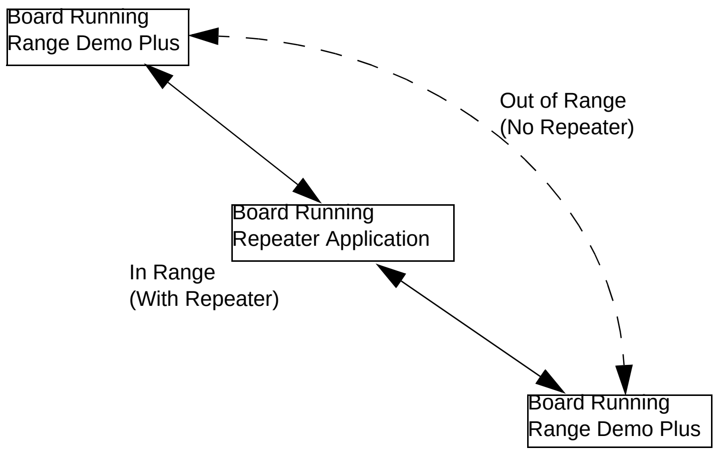

Figure 36. Repeater Application Test Setup

NOTE

Freescale recommends using only one hop for message re-transmission to avoid connection problems. Using more than one hop can lead to connection problems due to high latency.

10 Simple Protocol Test Client (SPTC)

The SPTC application allows users to test the SMAC primitives by sending special data frames through the Serial/USB port. The SPTC application requires two boards and the appropriate Python scripts to communicate with the boards or a serial port packet generator. Users can develop their own serial program to communicate with the SPTC by reviewing the protocol located in the `SPTC.xls` file.

NOTE

The Freescale Embedded Bootloader is not available in this version of the SPTC application.

10.1 Downloading the SPTC Application

1. Generate the SPTC application code using BeeKit as described in Section 2.3, “Generating an Application in BeeKit”.
2. Import the BeeKit generated `.xml` file and create a `.mcp` file as described in Section 2.4, “Loading Applications Into a Board Using the BDM”.
3. Ensure that the BDM debugger is already connected to the PC and the board and that the appropriate drivers are already installed.
4. Back in the Project window, click the Download button (Figure 37) and the ICD-Connection Manager window appears as shown in Figure 38.

Figure 37. Download Button

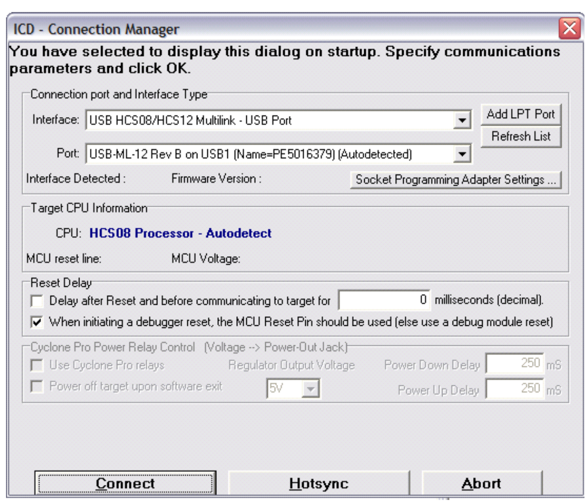

Figure 38. ICD - Connection Manager Window

5. Select the appropriate Interface from the drop down menu. This is usually the USB Port.
6. Select the appropriate Port from the drop down menu if multiple BDMs are connected to the system and click on the Connect button.

7. The Status window appears as shown in Figure 39. This window shows the download progress if the connection is working.

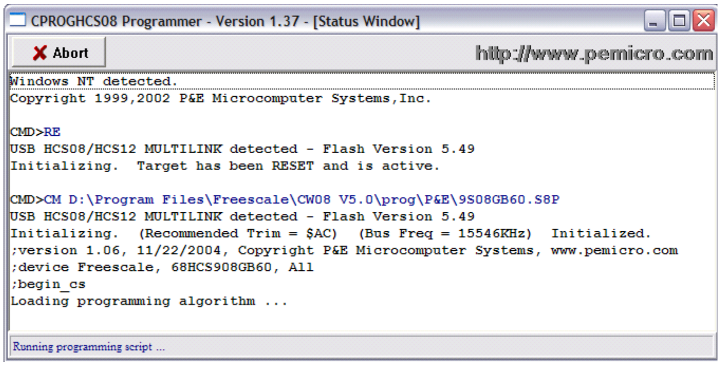

Figure 39. Status Window

8. After the code is downloaded to the target, the system is running in Debugger Mode and the Debugger application window appears as shown in Figure 40.

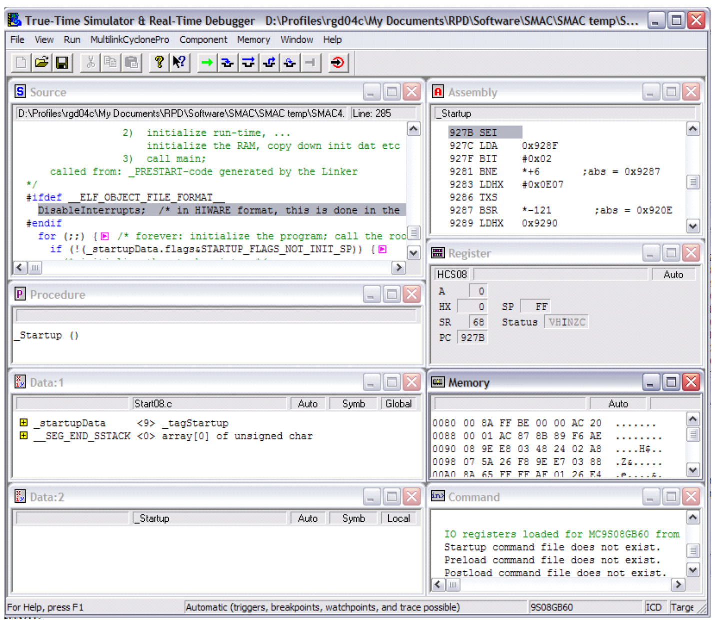

Figure 40. Debugger Application Window

9. Either disconnect the BDM cable from the target board and press its reset button or run the program on the target via the debugger session by clicking on the Run button (Figure 41).

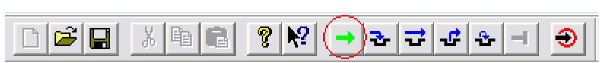

Figure 41. Debugger Run Button

10.2 Testing the SPTC with the WirelessUART

1. Download the SPTC application to a board that will be the tester.
2. Download the WirelessUART application to a board that will be the subject of the test.
3. Open a HyperTerminal session (or another compatible communications program) and configure it for serial communications using the following parameters:
   - 8 Bits
   - No Parity
   - 1 Stop
   - 38400 Baud
   - No Flow control
4. Connect the board that is running the WirelessUART application to the serial port of the PC and press that board's reset button. A message appears on the PC display as shown in Figure 42:

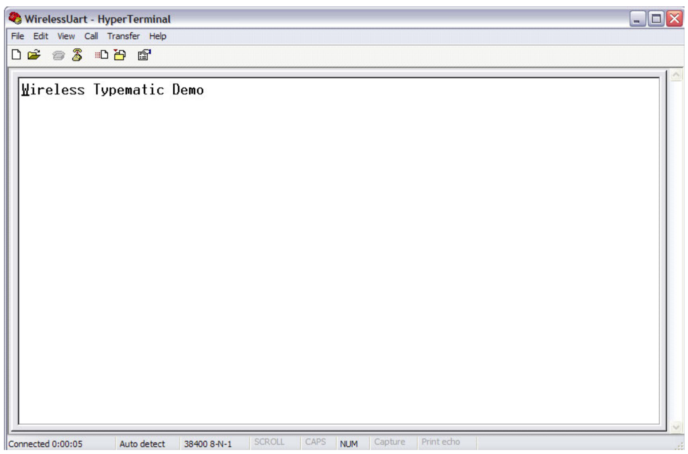

Figure 42. WirelessUART Message

5. Open the Freescale Test Tool located in the following directory:

```
Programs/Freescale ZigBee/Test Tool
```

6. As shown in Figure 43, from the Test Tool window, select Tools -> Communication Settings.

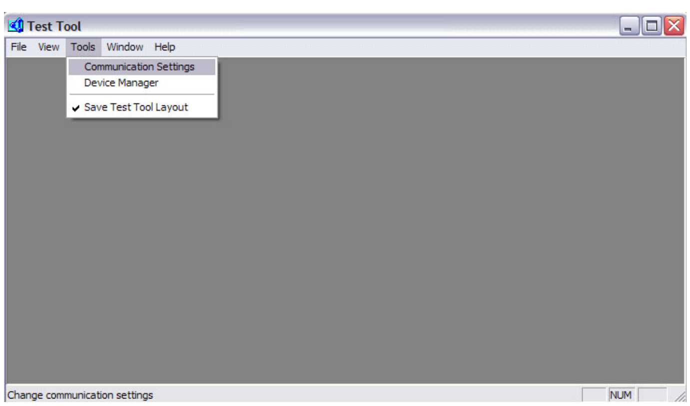

Figure 43. Communication Settings in Test Tool

7. As shown in Figure 44, for the board that is running the SPTC application, create a new serial connection. Select the appropriate Comport and set the baud rate at 19200. Click the OK button

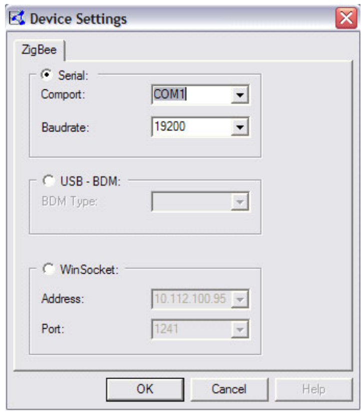

Figure 44. Test Tool Device Settings Window

8. The List of Devices window appears as shown in Figure 45. Disable all connections except the SPTC connection that was just created by highlighting the connection to be disabled, and clicking on the Disable button. The disabled connections are grayed out.

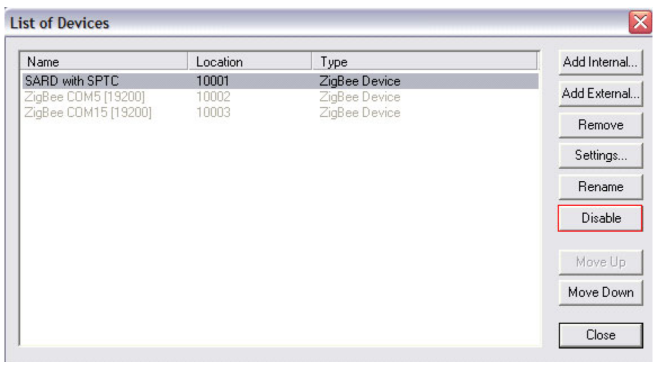

Figure 45. Enabling and Disabling Connections

9. As shown in Figure 46, use the Test Tool main menu to open the Script Server application by clicking on View -> Script Server.

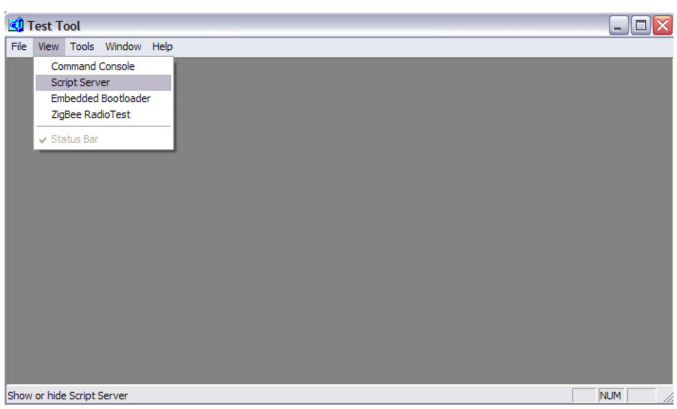

Figure 46. Launching the ScriptServer

10. The Script Server window appears as shown in Figure 47. Load the `sendingletterstotheWirelessUart.py` Python script, which is located in the following directory:

```
~\<Solution Name>\<Project name>\Sources
```

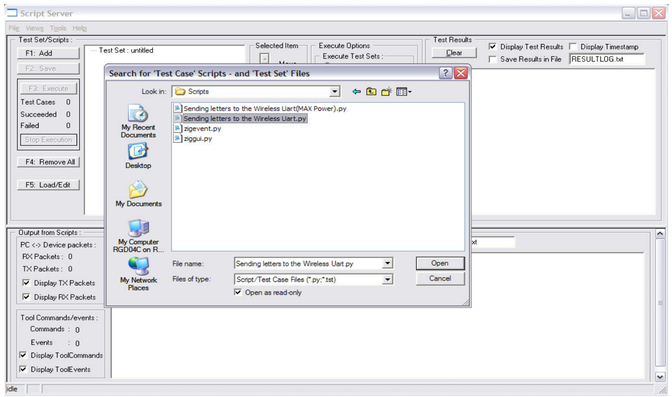

Figure 47. Loading the Script

11. As shown in Figure 48, click on the Execute button to run the script.

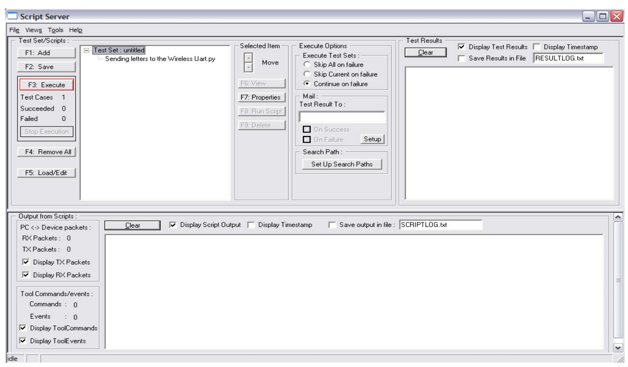

Figure 48. Running the Script

12. On the PC that is running the HyperTerminal application and is connected to the board running the WirelessUART application, the expected end result is shown in Figure 49. If this is not the end result, try running the WirelessUART application on the SPTC again.

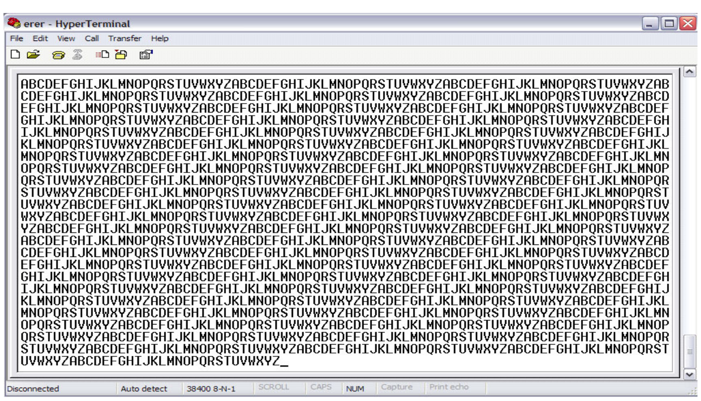

Figure 49. WirelessUART Expected Result

11 Over the Air Programmer (OTAP)

The Over the Air Programmer (OTAP) is beneficial in wireless applications when physical connections to existing installations are costly, inconvenient or impossible. The OTAP application allows users to update FLASH remotely without a physical connection.

Given an OTAP enabled application and an OTAP programmer, users can replace an existing application with a new application. For the OTAP application, users will utilize two boards; one board will be the programmer (OTAP Programmer) and the other board (OTAP-enabled board) will have its firmware updated. It is recommended that both images be an OTAP-enabled application.

The following summarizes what users will accomplish with the OTAP application:

- OTAP Programmer Board:
  - Download and run the OTAP Programmer software
  - Download an OTAP-enabled application via the UART
  - OTAP Programmer boards include the 13192-NCB and the Axiom-0308 with the MC1320x/MC1319x RF Daughter Cards.
- OTAP-enabled application Board (board having its firmware updated):
  - Download and run the OTAP-enabled Accelerometer application on the board that will have its firmware updated.
  - An OTAP enabled board can include any of the boards listed in Section 2.1, “Supported Hardware”.

In the following subsections, users will download the Wireless UART image from the OTAP Programmer board to the OTAP-enabled board that was previously running the Accelerometer application. Both Wireless UART and Accelerometer images must have the OTAP module property enabled when the project is generated in BeeKit.

NOTE

All SMAC applications support OTAP except the applications that run both the Embedded Bootloader and the OTAP Programmer. As stated, the only two target boards that can run the OTAP Programmer software are the 13213-NCB and the Axiom-0308 with the MC1320x/MC1319x RF Daughter Cards.

11.1 Downloading the OTAP Programmer Software

This section describes how to download the OTAP Programmer software to the board that will be used as the OTAP Programmer.

1. Generate the OTAP Programmer code using BeeKit as described in Section 2.3, “Generating an Application in BeeKit”.
2. Import the OTAP Programmer `.xml` file and generate the `.mcp` file according the procedure described in Section 2.4, “Loading Applications Into a Board Using the BDM”.
3. Ensure that the BDM debugger and the OTAP Programmer board are connected to the PC.
4. Compile the target in the project by selecting the Compile button (Figure 50).

Figure 50. Compile Button

5. Back in the Project window, click on the Download button (Figure 51) and the ICD-Connection Manager window appears as shown in Figure 52.

Figure 51. Download Button

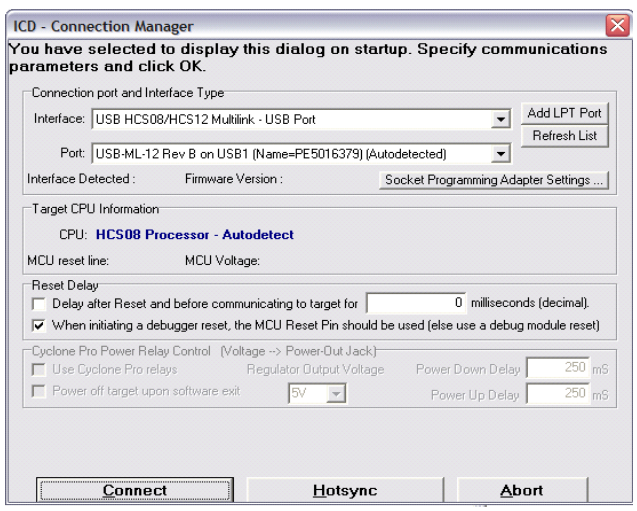

Figure 52. ICD - Connection Manager Window

6. Select the appropriate Interface from the drop down menu. This is usually the USB Port.
7. Select the appropriate Port from the drop down menu if multiple BDMs are connected to the system and click on the Connect button.
8. The Status window appears as shown in Figure 53. This window shows the download progress if the connection is working properly.

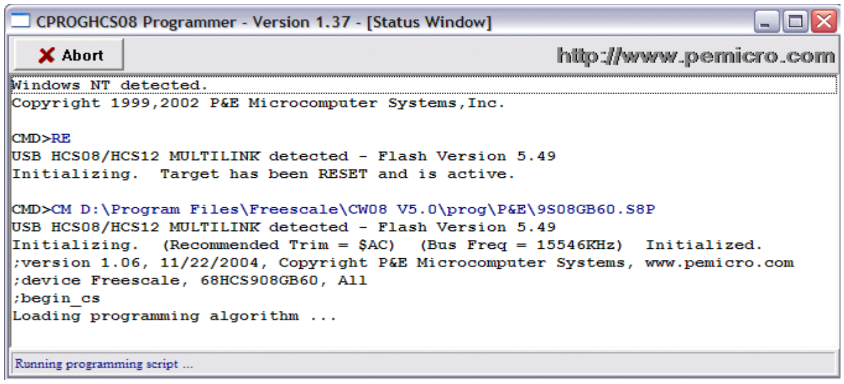

Figure 53. Status Window

9. After the code is downloaded to the target, the system is running in Debugger Mode and the Debugger application window appears as shown in Figure 54.

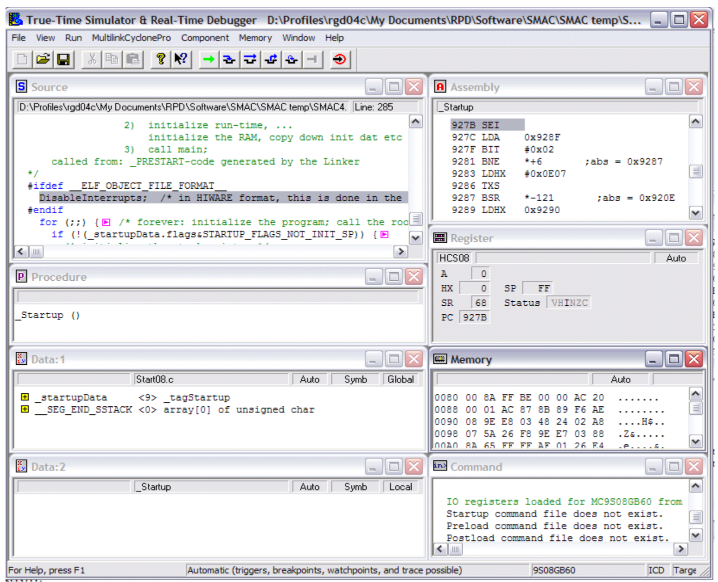

Figure 54. Debugger Application Window

10. Either disconnect the BDM cable from the OTAP Programmer board and press its reset button, or run the program on the target via the debugger session by clicking on the Run button (Figure 55).

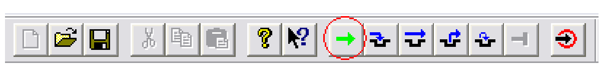

Figure 55. Debugger Run Button

11.2 Downloading an OTAP-enabled Image to the OTAP Programmer Board

This section describes how to download the Wireless UART application to the OTAP Programmer board.

1. Open the HyperTerminal or Tera Term configuration file contained in the following OTAP Programmer directory:

```
~\<Solution Folder>\<Project Folder>\Sources\OTAP.ht
```

2. As shown in Figure 56, use the following settings to manually configure HyperTerminal:
   — 8 Data Bits
   — No Parity
   — 1 Stop
   — 19200 Baud
   — Flow control hardware
   — Send line ends with line feeds
   — 1 millisecond delay between characters

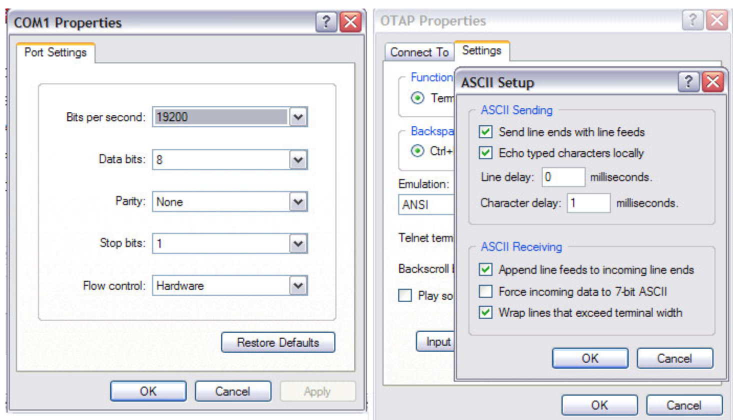

Figure 56. Manual HyperTerminal Configuration

3. Turn on the OTAP-Programmer application board.
4. Press Button 1 on the OTAP Programmer board to select the UART mode. Wait for the menu to appear on the HyperTerminal window as shown in Figure 57.

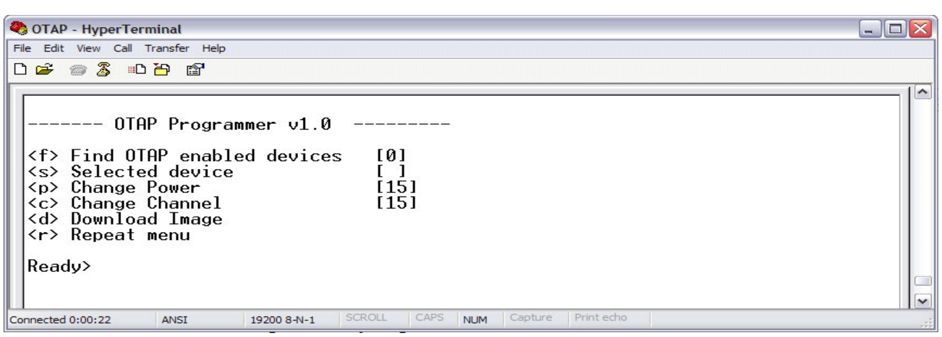

Figure 57. OTAP Menu in HyperTerminal Window

5. As shown in Figure 58, press 'd' to download the Wireless UART image to the OTAP-Programmer board.

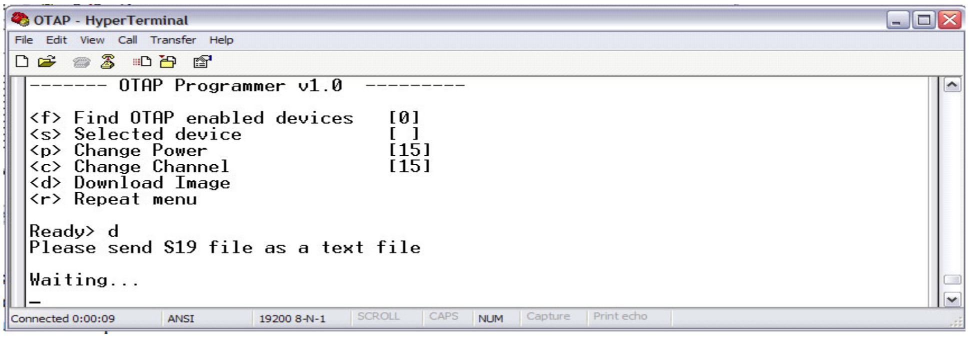

Figure 58. OTAP Download Image Window

6. Send the image as a text file by selecting Transfer -> Send Text File, as shown in Figure 59.

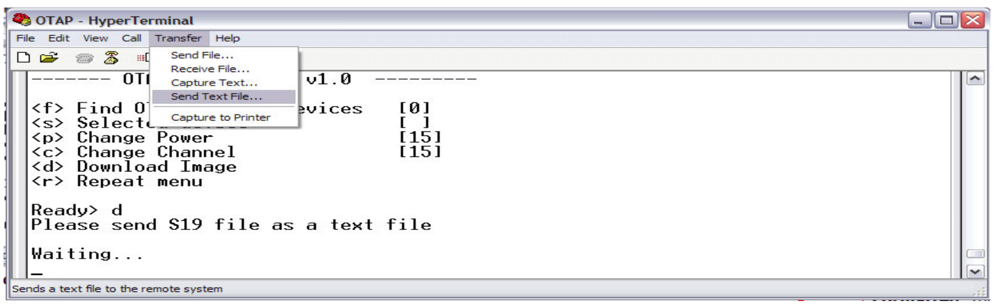

Figure 59. Sending the OTAP Image as a Text File

7. Configure the Send Text File window to look for all file types as shown in Figure 60.

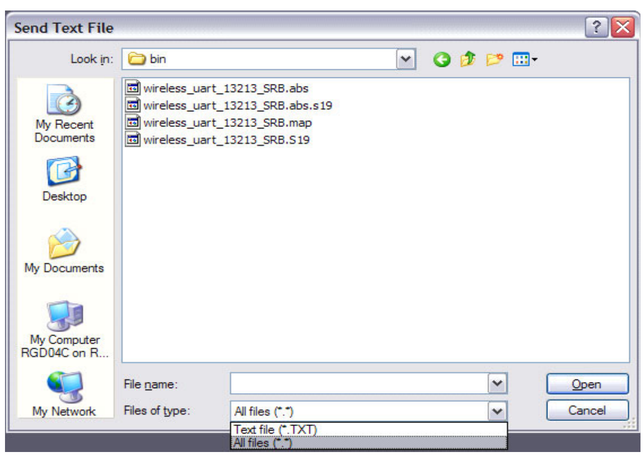

Figure 60. Setting the Send Text File Window to Look For All File Types

8. Search for the image file (in this example, the `wireless_uart_13213_SRB.S19`) inside the bin folder of the OTAP-enabled Wireless UART application. Compile the Wireless UART example as follows:

   a) Use BeeKit to generate the Wireless UART application with the SMAC/OTAP Enabled property set to True and a specific string in SMAC/OTAP ID property.

b) Import the Wireless UART `.xml` file and generate the `.mcp` file following the steps described in Section 2.4, “Loading Applications Into a Board Using the BDM”.

c) Compile the target and Build All in the SMAC project by clicking on the Compile button.

d) Compile the target to use in the WirelessUART project.

As shown in Figure 61, this example loads the `wireless_uart_13213_SRB.S19` file.

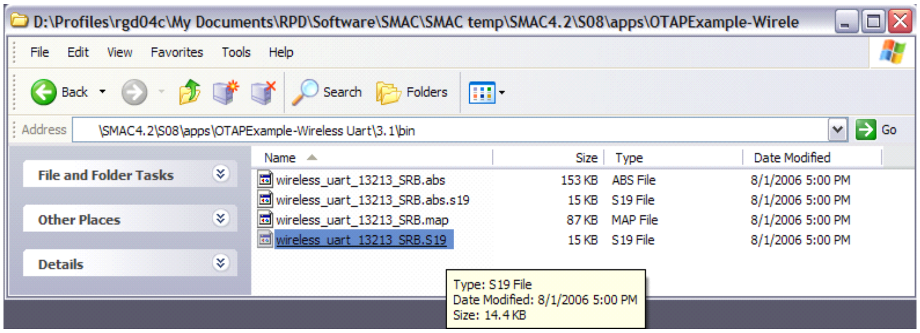

Figure 61. Selecting the S19 File

9. As shown in Figure 62, wait for the image to transfer until the following message appears:

```
Would you like to update NV-RAM? Yes(y) No(n)
```

10. When prompted to update NV-RAM, press 'n' (No) to return to the previous menu. The NV-RAM update feature is not currently available.

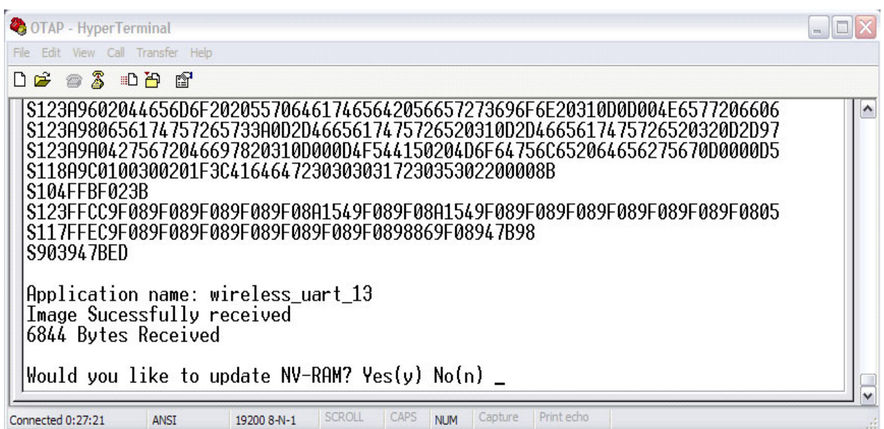

Figure 62. Image Download Complete

NOTE

The download process may take several minutes to complete. It can take as long as 3 to 5 minutes depending on the image size. This is normal operation.

11.3 Downloading the OTAP-enabled Accelerometer Application

This section describes how to download the Accelerometer Application to the board that will have its firmware updated by the OTAP Programmer.

1. Use BeeKit to generate the Accelerometer application with the following SMAC property settings; OTAP Enabled property set to True and a specific string in OTAP Default ID property field as shown in Figure 63.

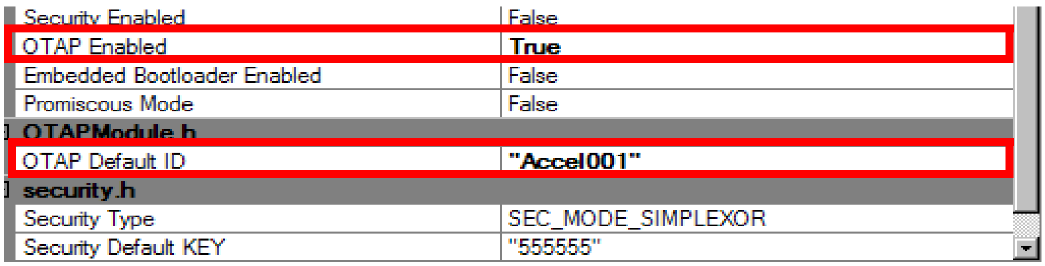

Figure 63. BeeKit SMAC-OTAP Enabled and OTAP Default ID Property Settings

2. Import the Accelerometer `.xml` file and generate the `.mcp` file following the steps described in Section 2.4, “Loading Applications Into a Board Using the BDM”.
3. Compile the target and Build All in the SMAC project by clicking on the Compile button.
4. Compile the target to use in the OTAPWirelessUart project.
5. Click on the Download button and follow the steps already described, starting with Step 5 in Section 11.1, “Downloading the OTAP Programmer Software”.

11.4 Transmitting New Image to an OTAP-enabled Board

This section describes how to send an image to an OTAP-enabled board using the LCD Interface on the OTAP Programmer board.

1. Turn on the OTAP programmer board and download the image to that board using HyperTerminal or Tera term as already described in Section 11.2, “Downloading an OTAP-enabled Image to the OTAP Programmer Board”.
2. On the board that will have its image updated and currently has the Accelerometer application loaded, enable its OTAP mode by pressing and holding Button 1 during a reset.
3. Reset the OTAP Programmer board by pressing its reset button. After reset, but before pressing any buttons, the LCD on the OTAP Programmer will display the following:

   OTAP SW v.1.0

   |UART|LCD|**|**|
4. On the OTAP Programmer board, press LCD (Button 2) to select the LCD interface. Before pressing any buttons again, the LCD on the OTAP Programmer will display the following:

   wireless_uart_13

   |FIND|***|**|CS

NOTE

The “wireless_uart_13” text shown in this example will change according to the name of the project.

5. Press FIND (Button 1) to start finding OTAP-enabled devices. Before pressing any buttons again, the LCD on the OTAP Programmer will display the following:

   FINDING… 01

   |DONE|ESC|**|**|

NOTE

The number shown in the upper right corner of the LCD display will change according to the number of devices found.

6. Press DONE (Button 1) to stop looking for devices. Before pressing any buttons again, the LCD on the OTAP Programmer will display the following:

   OTAP DEVICES 01

   |FIND|SEL|**|CS|
7. Press SEL (Button 2) to select the device that was found. Before pressing any buttons again, the LCD on the OTAP Programmer will display the following:

   SEL ACCEL001 ?

   |OK|ESC|<<|>>|*---
8. Press OK (Button 1) to select the device. Before pressing any buttons again, the LCD on the OTAP Programmer will display the following:

   ACCEL001 SEL

   |FIND|SEL|SND|CS

NOTE

The name of the device in this example is ACCEL001. This name will change according to the value assigned in BeeKit. Use the Arrows (<< >>) on the LCD display by pressing the corresponding buttons on the board to select any found devices.

9. Press SND (Button 3) to start sending the image from the OTAP Programmer to the OTAP-enabled board.
10. Wait until the countdown of the messages shows DONE. If the image was successfully transmitted from the OTAP Programmer to the OTAP-enabled board, the LCD will display the following:

    SENDING 0/174

    |DONE|ESC|**|**|
11. If the image transmission failed, the LCD will display the following:

    SENDING 174/174

    |CANC|ESC|**|**|

NOTE

The displayed number (174) will change according to the number of packets sent.

12. If the image transmission failed, go back to Step 2 in Section 11.4, "Transmitting New Image to an OTAP-enabled Board" and try again.

11.5 Verifying the Transmitted Image

Perform the test described in this section to verify that the image was successfully transmitted.

1. On the board having its image updated, enable OTAP mode by pressing and holding Button 1 while resetting the board.
2. Reset the OTAP Programmer board by pressing its reset button.
3. Press Button 2 (LCD) on the OTAP Programmer board to select the LCD interface.
4. Press Button 1 (FIND) on the OTAP Programmer board to start finding the OTAP-enabled boards.
5. Before pressing any buttons, ensure a device was found. This is indicated by the LCD displaying the following:

   OTAP DEVICES   01
6. Press Button 2 (SEL) on the OTAP Programmer board to select the OTAP-enabled board. The LCD displays the new image previously sent to the OTAP-enabled board.

In summary, the name of the boards found will change to the ones established in the previous configuration. For example, if users previously had a board named ACCEL001, they will now have a board named UART0001 using the information from the example described in Section 5.1, "Loading the Accelerometer Application".

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064, Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

***For Literature Requests Only*:**
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-521-6274 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Document Number: AN3231
Rev. 1.1
10/2006