

Getting Started Guide



XBee-PRO[®] ZB Programmable Development Kit

This guide provides instructions for setting up, developing, and testing programs on the available host processor of the XBee-PRO ZB programmable module. The guide includes information on how to:

- Download and install software tools
- Connect the hardware
- Compile and upload the bootloader
- Run an application

Software Setup

Step 1: Download and Install X-CTU Software

X-CTU is a stand-alone tool for configuring XBee modules. This software can be downloaded from our ftp site at ftp://ftp1.digi.com/support/documentation/XBP_ZB_Programmable_DK/.

1. Click on the X-CTU Setup file. X-CTU version 5250 or newer is needed for uploading applications.
2. Click **Save** or **Save File**. In Internet Explorer, you will be prompted to specify a location for the file. Most other browsers have a designated area for downloads. Make note of where the file is being saved for future use.
3. Browse to where the file is stored.
4. Double-click on the installer file and follow the X-CTU Setup Wizard.
5. When asked if you would like to check Digi's website for firmware updates, click **Yes**.
6. After the firmware updates are complete, click **Close**.
7. Start X-CTU by double-clicking on the X-CTU icon placed on your desktop or by selecting:
Start > Programs > Digi International > X-CTU

The X-CTU software is now available for use.

Step 2: Download and Install USB Drivers (Hardware USB Bus & Virtual COM Port Drivers)

The USB interface board is a "plug-and-play" device that should be detected by the PC automatically. To interface between the modem and a PC, two drivers must be installed: a USB

driver and a virtual COM port driver that makes the USB port look and perform like a physical COM port. After the modem is detected, a wizard for installing the USB driver is launched.

To install the two USB drivers:

1. Go to ftp://ftp1.digi.com/support/documentation/XBP_ZB_Programmable_DK/ and download the driver that matches your operating system. Save the file on your Desktop or other designated area. Extract the files to another folder for future use.
2. Plug in the Digi XStick [Fig 1]. The Found New Hardware Wizard dialog box will load.

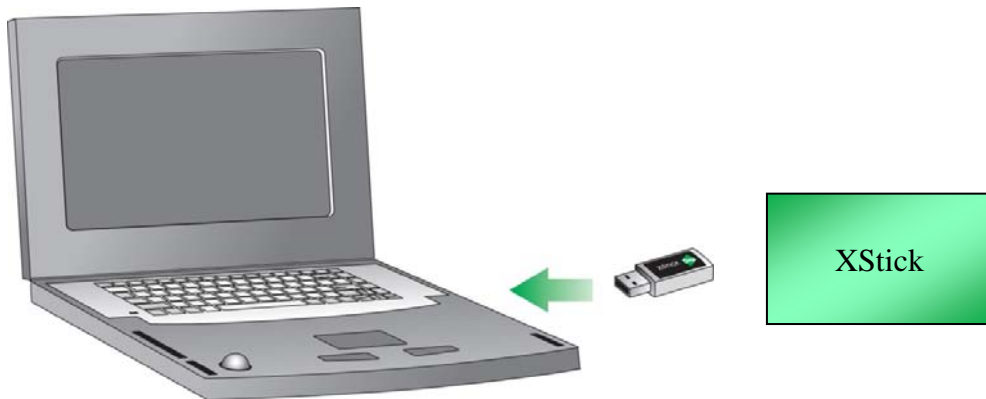


Figure 1: XStick setup

3. Select **Install from a specific list or location (Advanced)**; then click **Next**.
4. Select **Include this location in the search**. Browse to where you saved the **WINDOWS drivers** folder.
5. Click **Next**. If a hardware installation Windows alert box pops up click **Continue Anyway**.
6. Click **Finish**.

You may be prompted to install another driver, the virtual COM port driver. Repeat steps 5 and 6 to install this driver.

Step 3: Download and Install Compiler

The XBee-PRO ZB programmable module has a microprocessor that can run custom applications. The microprocessor is a Freescale® MC9S08QE32CFT. To develop applications on this processor you must have the software tool CodeWarrior® from Freescale.

1. Go to Digi's ftp site:
ftp://ftp1.digi.com/support/documentation/XBP_ZB_Programmable_DK/
2. Right-click on **CW_MCU_V6_3_SE.exe** and choose **Save Target As...** to save to your PC for installation.

Note: This is a large file and may take 30 minutes or more to download depending on your connection speed.

3. Double-click on the .exe file and click **Run**.

4. Follow the CodeWarrior Setup Wizard.
5. Upon completion of the installation process, open CodeWarrior by selecting **Start > Programs > Freescale CodeWarrior > CodeWarrior Development Studio > CodeWarrior IDE**.
6. Click **Start using CodeWarrior**.

Step 4: Download the Bootloader/Sample Application

From ftp://ftp1.digi.com/support/documentation/XBP_ZB_Programmable_DK/ download **GettingStarted.zip** to the local folder where you will be storing your apps. Unzip the folder to the same location.

USB Multilink Debugger

A debugger is required to access the XBee-PRO ZB programmable module's microprocessor. This hardware interfaces between a PC, via USB, and the 6-pin connector on the Digi XBee USB interface board, XBIB-U-DEV, that hosts the XBee module. An XBIB-R-DEV XBee RS-232 interface board can also be used in place of the USB version. With the debugger you can directly control the target's execution, read/write registers and memory values, debug code on the processor, and program internal or external FLASH memory devices.

Because the software support for this debugger is included with the current version of CodeWarrior, it is *strongly* recommended that you install and run CodeWarrior before connecting the debugger to the PC.

Note: If you do connect the debugger before installing CodeWarrior, you will need to uninstall the debugger's drivers and reinstall them after running CodeWarrior.

Connect the Hardware

1. Plug the XBee-PRO ZB programmable module into the headers on the XBee USB interface board as shown in Figure 2. The antenna will be at the top of the interface board when held so that the USB connector is at the bottom of the board. Do not power up the unit by plugging in the USB cable to the PC at this stage.

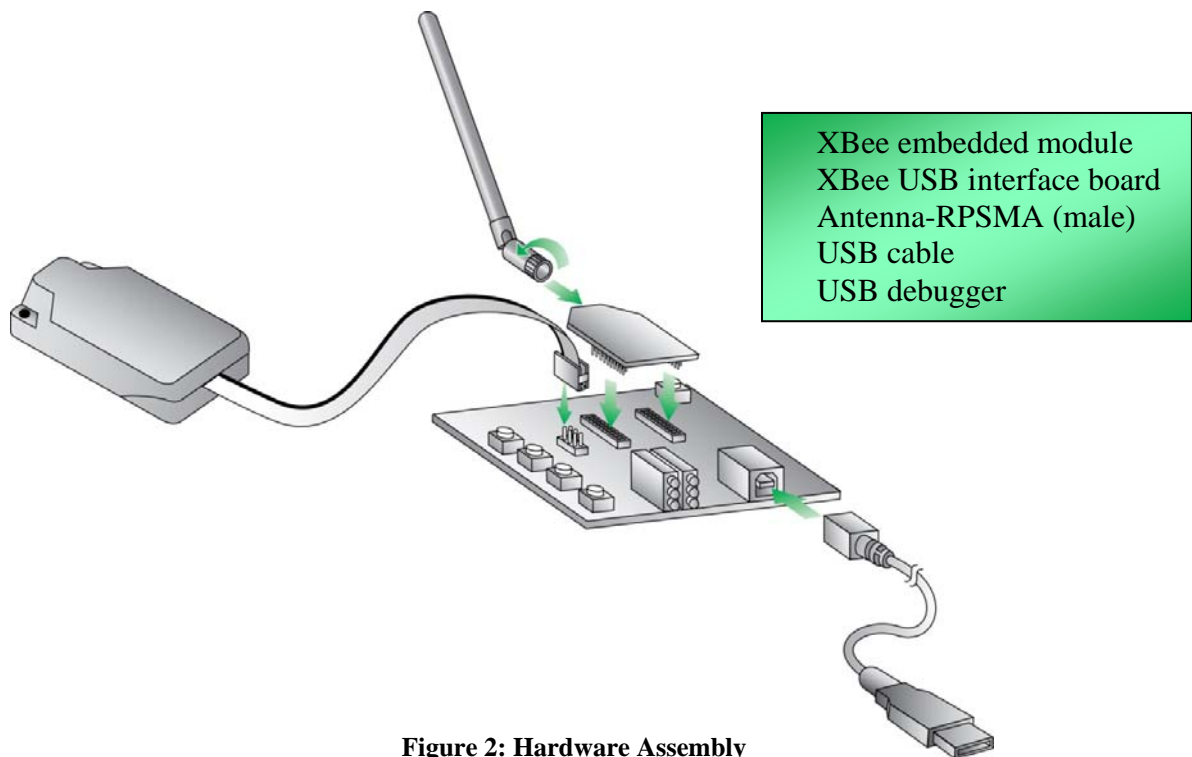


Figure 2: Hardware Assembly

2. The debugger plugs into two locations: the PC's USB port and the 6-pin header on the development board.
 - a. Verify that the power is off and the debugger and the XBee interface board are not connected to the PC.
 - b. Connect the debugger to the interface board by its ribbon cable. Make sure that the ribbon cable is plugged into the header with the proper orientation. Pin one is denoted by the red stripe running down the ribbon cable. Pin one is located at the top of the connector on the interface board (when held so that the USB connector is at the bottom of the board).

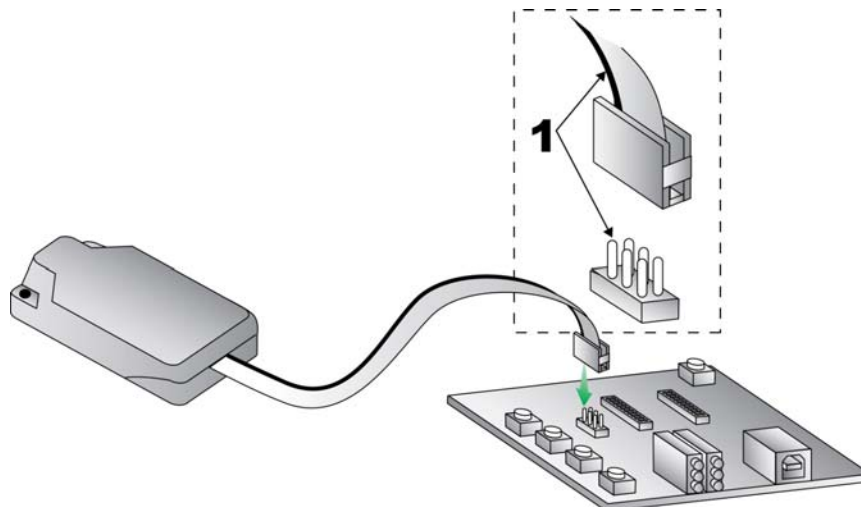


Figure 3: Debugger as Connected to an XBee USB Interface Board

- c. Connect the debugger to the PC via a USB cable. Be sure that you have downloaded and executed CodeWarrior before completing this step.
- d. A new hardware wizard box will pop up. Select **Install the software automatically**. The wizard will search for and extract the drivers from the CodeWarrior files installed earlier. The Blue LED on the debugger will come on.

- e. Apply power to the interface board by connecting the USB cable to the PC. You will need to install the USB driver for the interface board/module assembly at this point. Follow the same steps as installing the Digi XStick, as noted earlier in the guide. (Driver files are the same for the XStick and the XBee USB Interface Board.) The Yellow LED on the debugger should become lit.
- f. When you are finished working with the debugger, before disconnecting it from the interface board, turn the power off by unplugging the USB cable attached directly to the debugger.

Compile and Load: Debuggable Bootloader

XBee-PRO ZB programmable modules come from the factory with a locked bootloader, meaning that the bootloader is not debuggable. The first task to being able to create, edit, and run applications is to compile the debuggable bootloader and upload it to the XBee. The bootloader is named **MC9S08_BL32_Debug.mcp** and can be found in the sample application files you downloaded. It is located by following the path: `\Bootloader\XbeeProZBProgrammable\`. Double-click that file to launch the CodeWarrior IDE. [Fig 4]

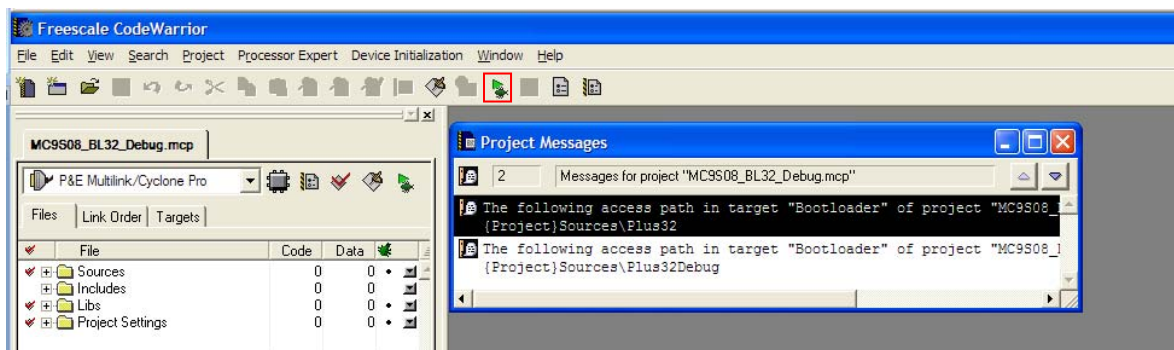


Figure 4: Open the Bootloader



In this case the project is already completed so all you need to do is press the **Debug** button (the green arrow on the CodeWarrior tool bar.) You can also click **Project > Debug** from the menu bar. A PEMICRO Connection Manager window will pop up.

If the port is listed as Undetected [Fig 5], try disconnecting and reconnecting the debugger's USB cable and click the **Refresh List** button.



Figure 5: Debugger Connection Error

After a successful compilation, the Port section will be listed as Autodetected. Check the **Use custom trim reference frequency** checkbox. The custom trim frequency should fill in at 31250.00 Hz. Press **Connect** to upload your compiled bootloader. [Fig 6]

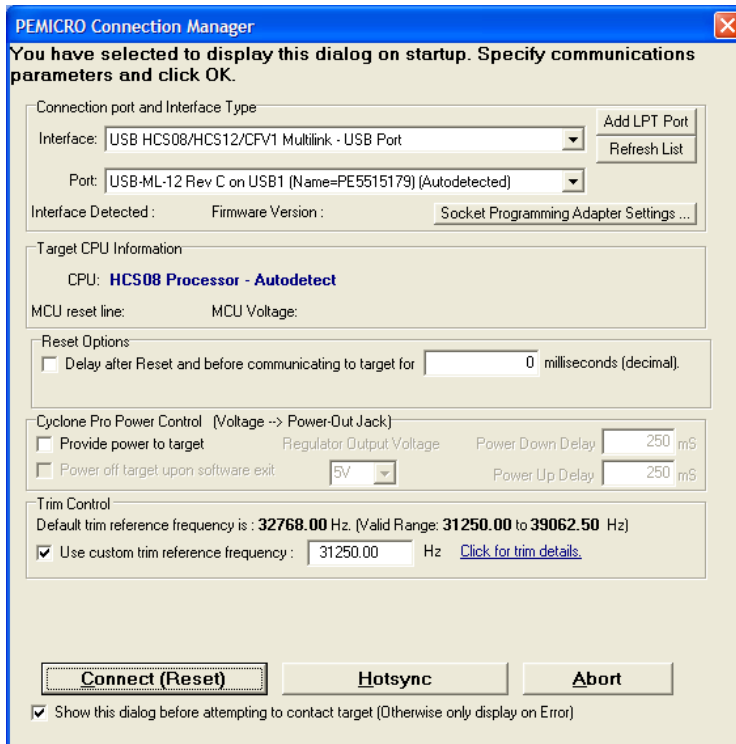


Figure 6: Load the Bootloader

A Programmer window will open and run through the uploading sequence. Once the bootloader is successfully loaded, that window will automatically close. Close the Real-Time Simulator/Debugger window.

Note: It is possible to skip this step in production and use one of these three ways to upload your application (binary file) to your programmable module:

- Over-the-air using the XStick to do a remote XModem upload
- Upload directly through a USB or 9-pin RS-232 serial port
- Use the debugger if you intend to overwrite all the flash memory

The first two methods do not erase the existing bootloader, but when using the debugger method all the flash memory is overwritten.

Compile and Load: Sample Application

Use CodeWarrior to build a binary file and load an application onto the XBee-PRO ZB programmable module.

1. From CodeWarrior's menu click **File** and then **Open...**
2. Browse to the same folder that you saved your applications to previously. The application code is found in the path: AppTransparent\XbeeProZBProgrammable\ and the file name is **App32Transparent.mcp**.
3. Click **Open**.

- Next click the **Make** button or click **Project > Make** from the menu bar [Fig 7] to compile your application.

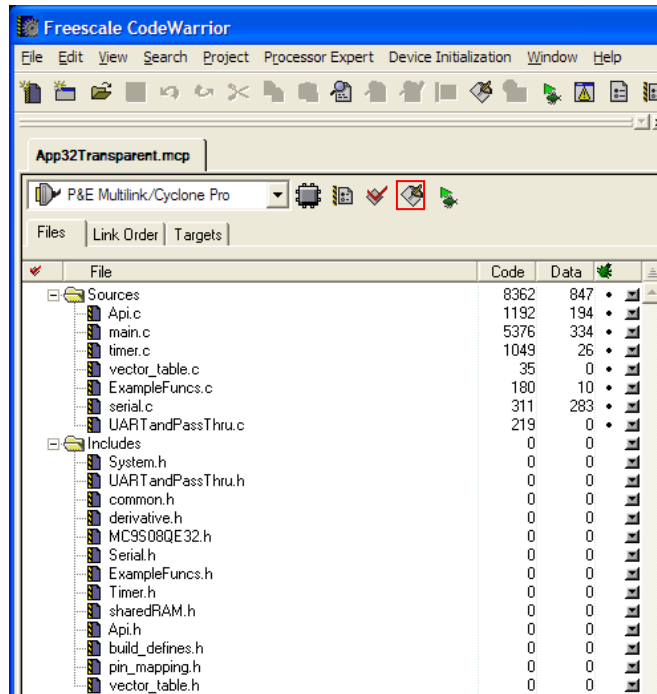


Figure 7: Open the Application File

- In the same path (\AppTransparent \XbeeProZBProgrammable\) that you found the **App32Transparent.mcp** file you will find a **Bin** folder. This is where your compiled bin file will be found. In this case it is named **App32T.abs.bin**.

Wired Uploads

You can upload an application via serial port or over the air using any XModem program. This example explains how to upload an app over the serial port using the XModem function built into X-CTU.

You can upload using the bootloader via the USB com port by:

- Make sure that the XBee-PRO ZB programmable module is still connected to your PC via the USB cable.
- Open an X-CTU session with 9600 baud, no parity, and 8 data bits with one stop bit.
- Select the appropriate COM port and then click on the **Terminal** tab.
- Click on the **XModem** menu item. A separate window will pop up.
- Click **Open File** and browse to the desired flat binary output (bin) file.
- Click **Open** and the path to the selected file will populate the top box.
- Click the main X-CTU window and hit **Enter** or **Space** to display the bootloader menu.
- Hit the **F** key to initiate a wired firmware update. A series of 'C' characters will be displayed within the terminal window.

- In the XModem window, click **Send**. The 'C' characters are a visual indication of the timeout during which the app file can be sent. If you miss that timeout, you can type in an **F** to start the process again. The file will be sent to the module. [Fig 8] Upon a successful update, the bootloader will run the newly loaded application.

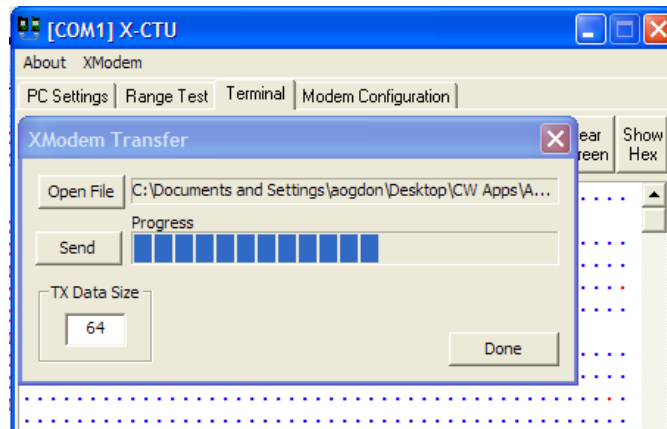


Figure 8: Upload the .bin File

Click **Done** to close the XModem Transfer window. For this particular sample application, a menu and description will be displayed.

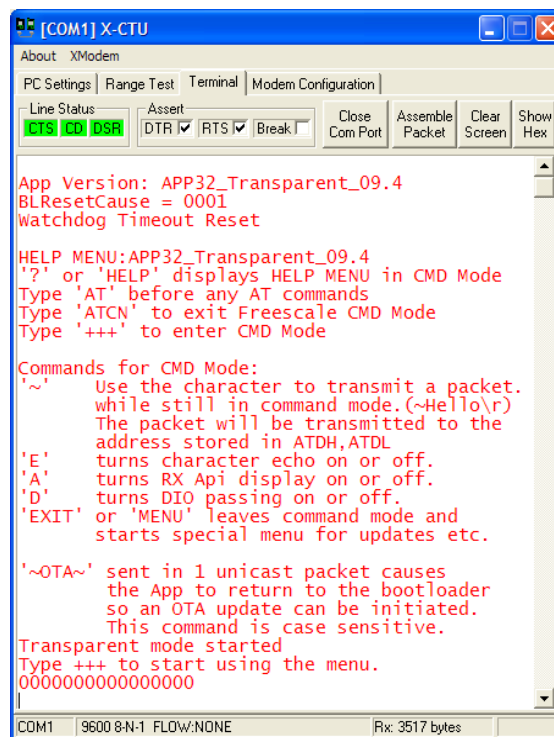


Figure 9: Sample Application Menu

You can now transfer data back and forth between this programmable XBee module and the XStick.

- Open a second X-CTU window and select the XStick's COM port.
- Go to the **Terminal** tab.
- Click the **Show Hex** button.
- In the Terminal tab of the programmable XBee, click on the **Assemble Packet** button.

5. Type your desired string into the window and click **Send Data**.

The resulting API packet is shown as received by the XStick in both ASCII and Hex as an API Receive Packet. [Fig 10]

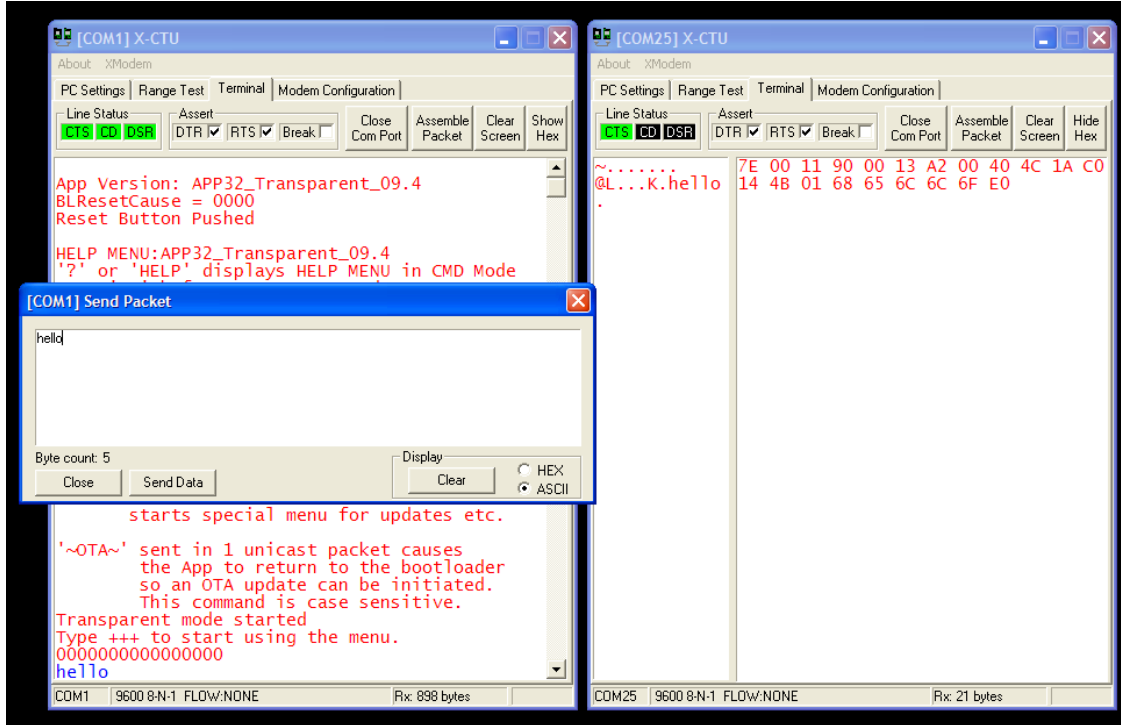


Figure 10: Transparent Data Transfer

Type +++ to enter command mode. From here you can query and change parameters using AT commands. [Fig 11] The list of commands noted in the menu can also be used.

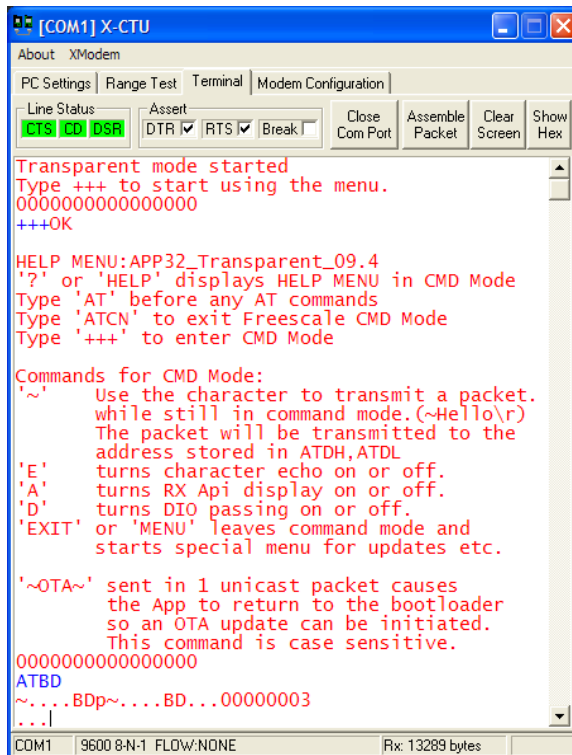


Figure 11: Command Mode

The sample application also has some built-in functionality for sampling ADC and Digital I/O lines as well as sending this data over the air. Try out these features and find out more by going to the CodeWarrior window, and clicking on main.c. The header provides more detailed information about the features of this sample application. This Freescale processor and development environment support C and Assembly programming languages.

Congratulations! You have compiled and installed a debuggable bootloader as well as your first application on the XBee-PRO ZB programmable module.

Further Information

Next Steps

- If you want to write your own bootloader or edit the sample one created by Digi, you can follow the loading steps in the **Compile and Load: Debuggable Bootloader** section to load the new code.
- When you write your own application, you can follow the **Compile and Load: Sample Application** section.

Note: If you accidentally overwrite bootloader memory while developing your own application you can repeat the bootloader process to restore the bootloader.

Supporting Documentation

The following information may be useful for developing your own applications:

[XBee ZB Product Manual \(90000976\)](#): Includes schematics, memory map diagrams, and API frame definitions.

[Freescale Product Manual \(MC9S08QE32RM\)](#): Includes details on memory registers and more.

[Quick Start Guide for CodeWarrior](#): Contains detailed information on creating new applications.

[XBee-PRO ZB Programmable Forum](#): Message boards for questions, comments, and other help.

Contact Digi International

Toll-free phone U.S.A. & Canada: (866) 765-9885
Worldwide: (801) 765-9885
Live chat: www.digi.com
Online support: <http://www.digi.com/support/eservice/login.jsp>
Email: rf-experts@digi.com

PN (1P): 90001192_A