



*XBee[®] ZigBee SMT RF Module
Development Kit
Getting Started Guide*

©2013 Digi International Inc.
All rights reserved.

Digi, Digi International, the Digi logo, the Digi web site, a Digi International Company, XBee, Digi XBee, Etherios, the Etherios logo, the Etherios website, and Device Cloud by Etherios are trademarks or registered trademarks of Digi International, Inc. in the United States and other countries worldwide. All other trademarks are the property of their respective owners.

All other trademarks mentioned in this document are the property of their respective owners. Information in this document is subject to change without notice and does not represent a commitment on the part of Digi International.

Digi provides this document “as is,” without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of fitness or merchantability for a particular purpose. Digi may make improvements and/or changes in this manual or in the product(s) and/or the program(s) described in this manual at any time.

This document could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes may be incorporated in new editions of the publication.

Table of Contents

| | |
|--|-----------|
| Using this Guide | 5 |
| Conventions used in this Guide..... | 5 |
| Contact Information..... | 5 |
| Introduction | 6 |
| Goals of the Kit..... | 6 |
| Requirements of the Kit..... | 6 |
| System Requirements..... | 6 |
| Additional Documentation..... | 6 |
| Part 1: Set up your XBee ZB SMT RF Module Development Kit | 7 |
| Identify Kit Components..... | 7 |
| Assemble your Development Kit..... | 8 |
| Part 2: Download and Install X-CTU Software | 9 |
| Installing USB Drivers..... | 9 |
| Installing X-CTU Software..... | 9 |
| Part 3: Test Communications Link and Establish a Mesh Network | 10 |
| Configure the Modules..... | 10 |
| Perform a Range Test..... | 17 |
| Establish a Mesh Network..... | 21 |
| Part 4: Over-The-Air Configuration | 22 |
| Configure Remote Modules..... | 22 |
| Part 5: XBee ZB SMT Design Considerations | 24 |
| Software Design Considerations..... | 24 |
| Hardware Design Considerations..... | 25 |
| Part 6: Managing the Network | 26 |
| Digi Gateways..... | 26 |
| To the Cloud! Make the most of your data..... | 27 |

| | |
|---|----|
| To the Cloud! Take control of your device network..... | 27 |
| Appendix A: Troubleshooting | 28 |
| Resetting the XBee ZB SMT RF Module..... | 28 |
| Why are the Modules no longer communicating with one another? | 28 |
| Appendix B: ZigBee Functionality | 29 |
| ZigBee Nodes | 29 |

Using this Guide

Conventions used in this Guide



This icon indicates a step that has the potential to be troublesome. Further information regarding items marked with this symbol can be found in “Appendix A: Troubleshooting” on page 28.



This icon indicates a hint, or concept that is learned.



This icon indicates that a goal of the kit has been completed.



This icon indicates a warning of the potential for confusion or danger.

Contact Information

For more information about your Digi products, or for customer service and technical support, contact Digi International.

| To Contact Digi International | Use |
|---|---|
| Mail | Digi International World Headquarters 11001 Bren Road East Minnetonka, MN 55343 |
| Phone 8:00 AM - 5:00 PM (U.S. Mountain Time) | 1-866-765-9885 toll-free USA and Canada 1-801-765-9885 Worldwide |
| Online Support | http://www.digi.com/support/eservice/login.jsp |
| Email | rf-experts@digi.com |

Introduction

Thank you for purchasing an XBee® ZigBee SMT RF Module Development Kit. This kit is designed to make it easy to set up an XBee mesh network using the ZigBee protocol, send data from one XBee to another, and adjust the XBee settings. Before you start working with the kit, let's cover some basics.

Goals of the Kit

As you go through the steps in this kit, you will:



1. Set up your XBee ZB SMT RF Module Development Kit.
2. Download and install X-CTU configuration software.
3. Test communications link and establish a mesh network.
4. Set up over-the-air configuration.
5. Explore SMT design configurations.
6. Manage the network.

Requirements of the Kit

System Requirements

To install the software mentioned in this guide, you will need a PC running Microsoft Windows 2000, XP, Vista or Windows 7.

Additional Documentation

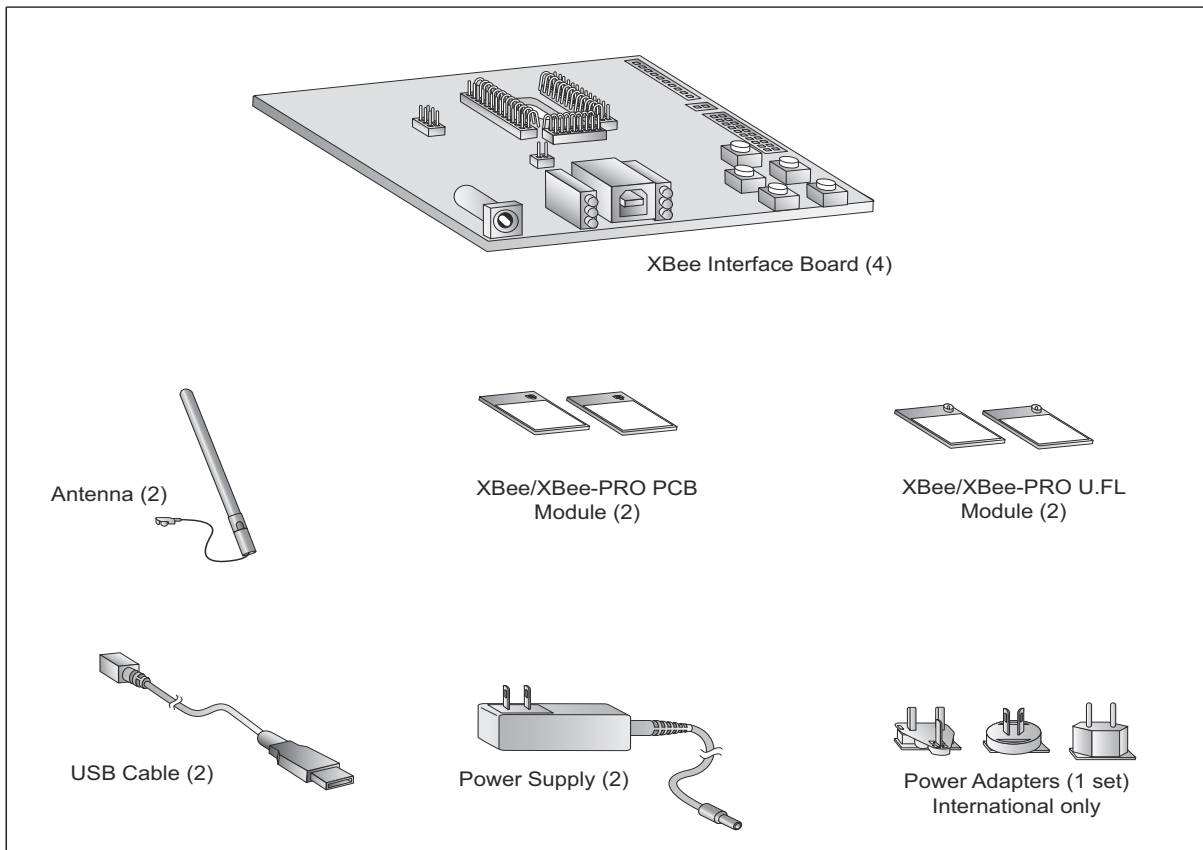
For more information about the software, API operations, AT command modes, or the form factor please refer to the [XBee ZB SMT RF Module Product Manual](#) available on the Digi Support website.

For more information on configuring and using the X-CTU utility, please refer to the [X-CTU Configuration & Test Utility Software User's Guide](#).

Part 1: Set up your XBee ZB SMT RF Module Development Kit

Identify Kit Components

Carefully unpack and verify the contents of your kit. Your kit should include the following:

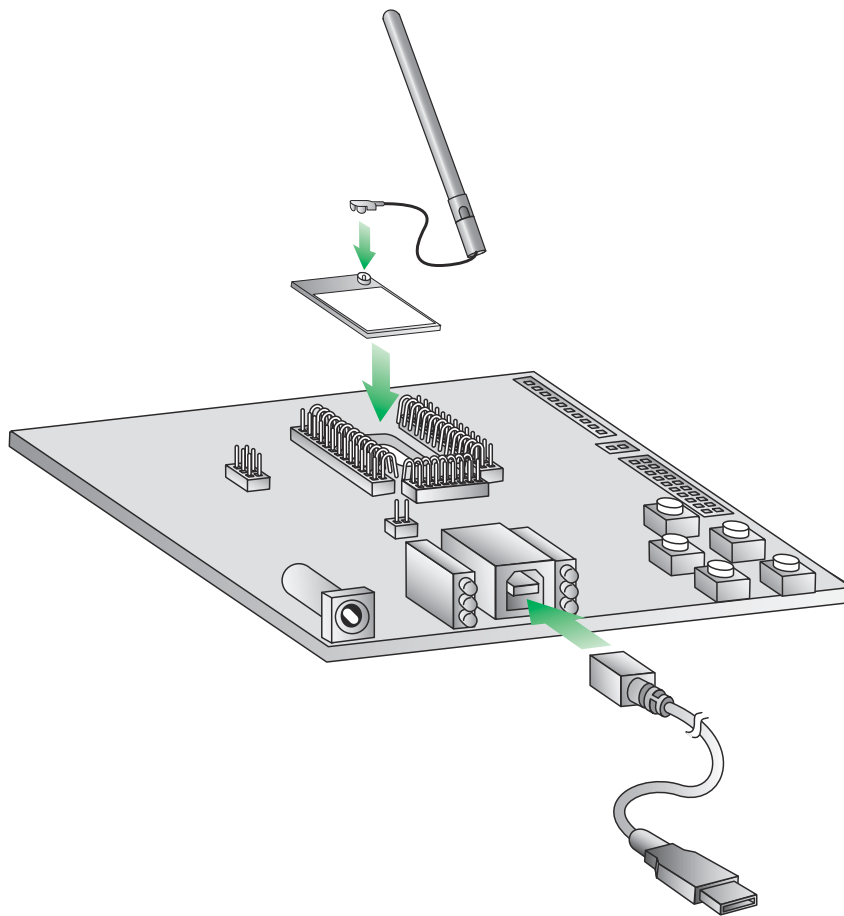


Assemble your Development Kit

To assemble your Development Kit, perform the following steps:

1. Install the modules on the XBee Interface Boards (XBIB) by lining the pins up with the headers and pressing the modules into place.
2. Attach the dipole antennas to the modules.
3. Connect the first XBIB to your computer using a USB cable. This first device will be designated as your coordinator.
4. Connect the remaining modules and interface boards, and set them aside for now.

You are now ready to run the X-CTU software.



You have just completed Goal #1 - setting up your XBee ZB SMT RF Module Development Kit.

Part 2: Download and Install X-CTU Software

For proper kit configuration and operation, X-CTU software (version 5.15 or later) must be downloaded and installed. A copy of X-CTU software and USB drivers will need to be installed on each computer used in conjunction with this guide.

Installing USB Drivers

The XBee USB interface board is a "plug-and-play" device that should be detected by the PC automatically. If you are using Windows 7 or Vista, the USB drivers should automatically install and a notification will appear in the lower right portion of your screen indicating success or failure.

If the USB drivers fail to install, please follow the USB driver installation instructions found here: <http://www.digi.com/support/kbase/kbaseresultdetl.jsp?id=3214>.

If you are using Windows 2000 or XP, download and install the driver as per the following directions.

To install the USB driver:

1. Download the driver setup file at:
http://ftp1.digi.com/support/driver/FTDI_Windows_Driver_Setup.exe.
2. Double-click on the setup file. A window will pop up during installation and automatically close when the process is complete.

Installing X-CTU Software

1. Download X-CTU at www.digi.com/xctu.
2. Browse to the folder to which you saved the above install file.
3. Double-click on the installer file and follow the X-CTU Setup Wizard.
4. When asked if you would like to check Digi's web site for firmware updates, click **Yes**. Updates may take a few minutes, please be patient.
5. After the firmware updates are complete, click **Close**.
6. Start X-CTU by double-clicking on the X-CTU icon on your desktop, or by selecting **Start > Programs > Digi > X-CTU**.

The X-CTU software is now ready to be used.



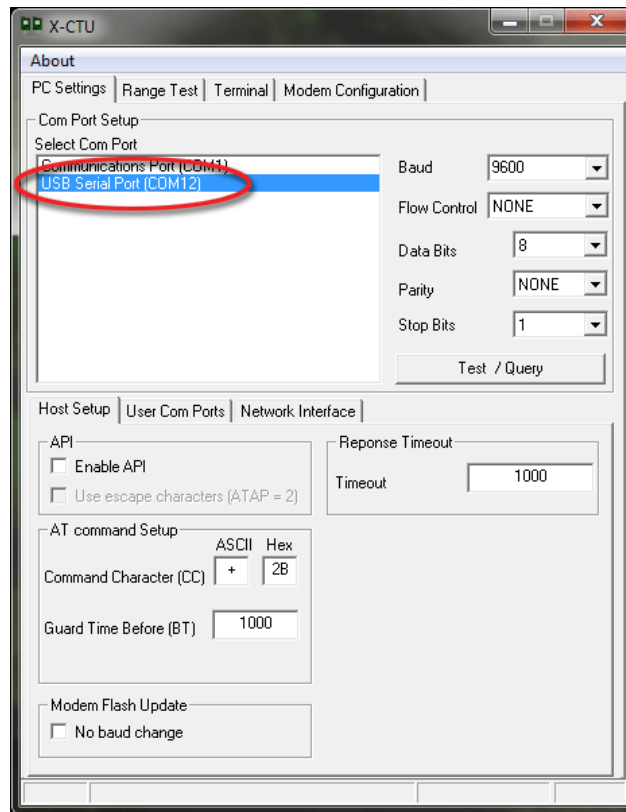
You have just completed Goal #2 - downloading and installing the X-CTU configuration software.

Part 3: Test Communications Link and Establish a Mesh Network

Configure the Modules

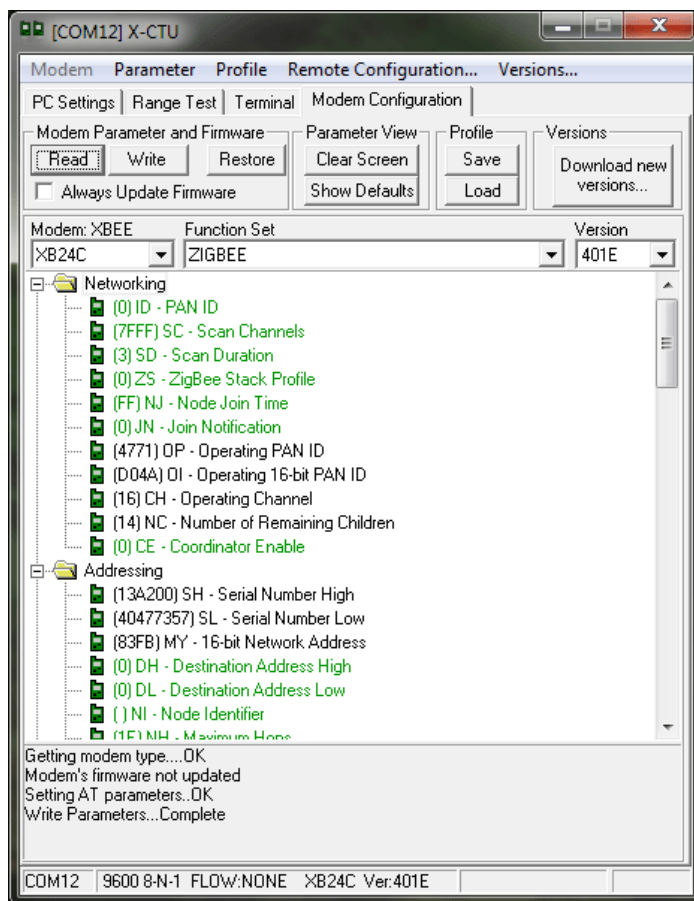
Each network requires one and only one coordinator. For more information on the function of the coordinator, see the [Product Manual for the XBee ZB SMT RF Module](#). The first module will be configured as the coordinator. The remaining modules will be configured as routers.

1. Connect an XBee Interface Board (with module installed) to your computer.
2. Double-click the X-CTU shortcut on your desktop.
3. Under the PC Settings tab, select the serial COM port associated with the interface boards you have just attached to your computer.



4. Verify that the baud rate and data settings match the internal settings of the devices:
 - Baud Rate: 9600
 - Flow Control: NONE
 - Data Bits: 8
 - Parity: NONE
 - Stop Bits: 1

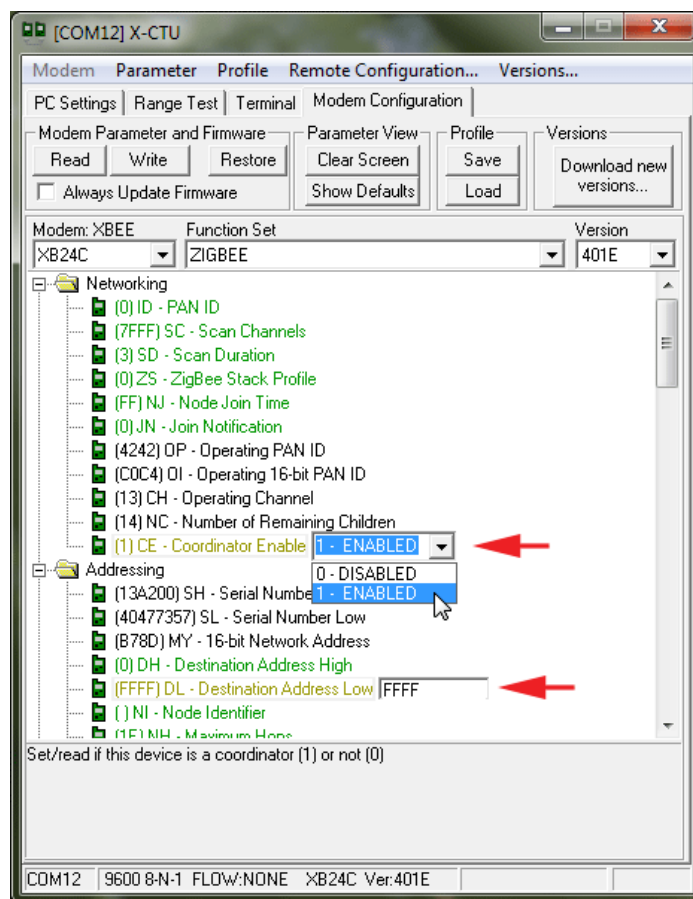
5. Click the **Test/Query** button to verify communication with the module. A pop-up will be displayed showing status and some basic information about the attached module.
6. In the X-CTU **Modem Configuration** tab, click **Read**.



X-CTU displays the available settings for the module.

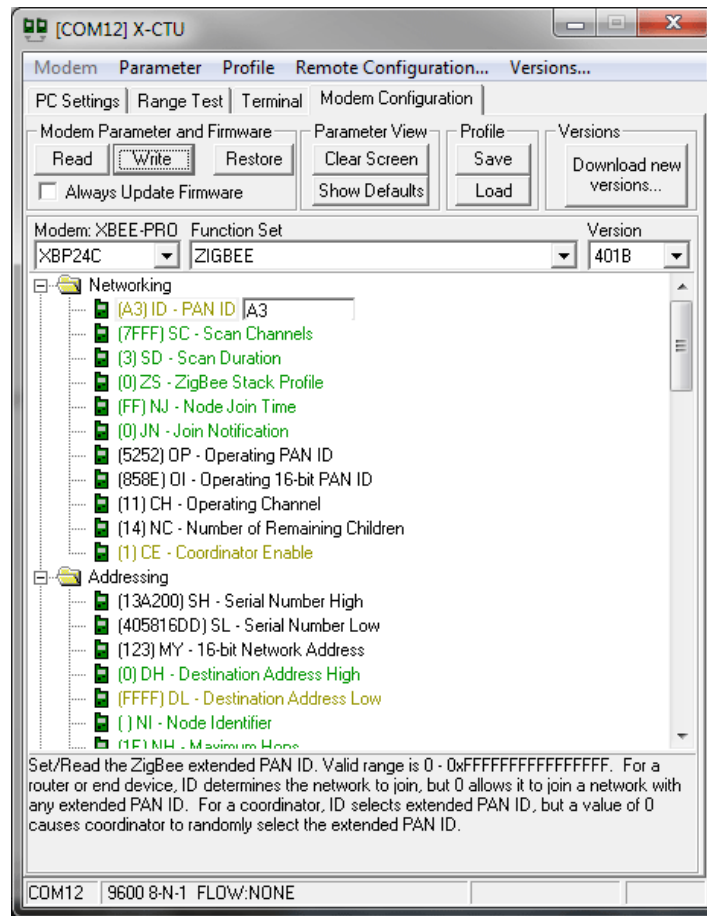
7. The following procedure will set one device as the coordinator. Perform this procedure on the first module that was connected to your computer during kit assembly.
 - a. Click on **CE - Coordinator Enable** and select **1-Enabled**.
 - b. Click on **DL - Destination Address Low** and type in FFFF.
 - c. Then click the **Write** button.

Note: By assigning the DL as FFFF, this assigns the coordinator to broadcast transmission. Unicast transmission will be covered later in this guide.



Use the parameter window shown above to designate one device as the coordinator.

8. (Optional) If you are working in an area where other ZigBee networks are running, it is possible for router modules to join an unintended network. To avoid this situation, you can set a single PAN ID for all of modules you wish to be on the same network. To set the PAN ID:
 - a. Click on **ID - PAN ID** and enter a hex number (for example, A3 in the figure below).
 - b. Then click the **Write** button.

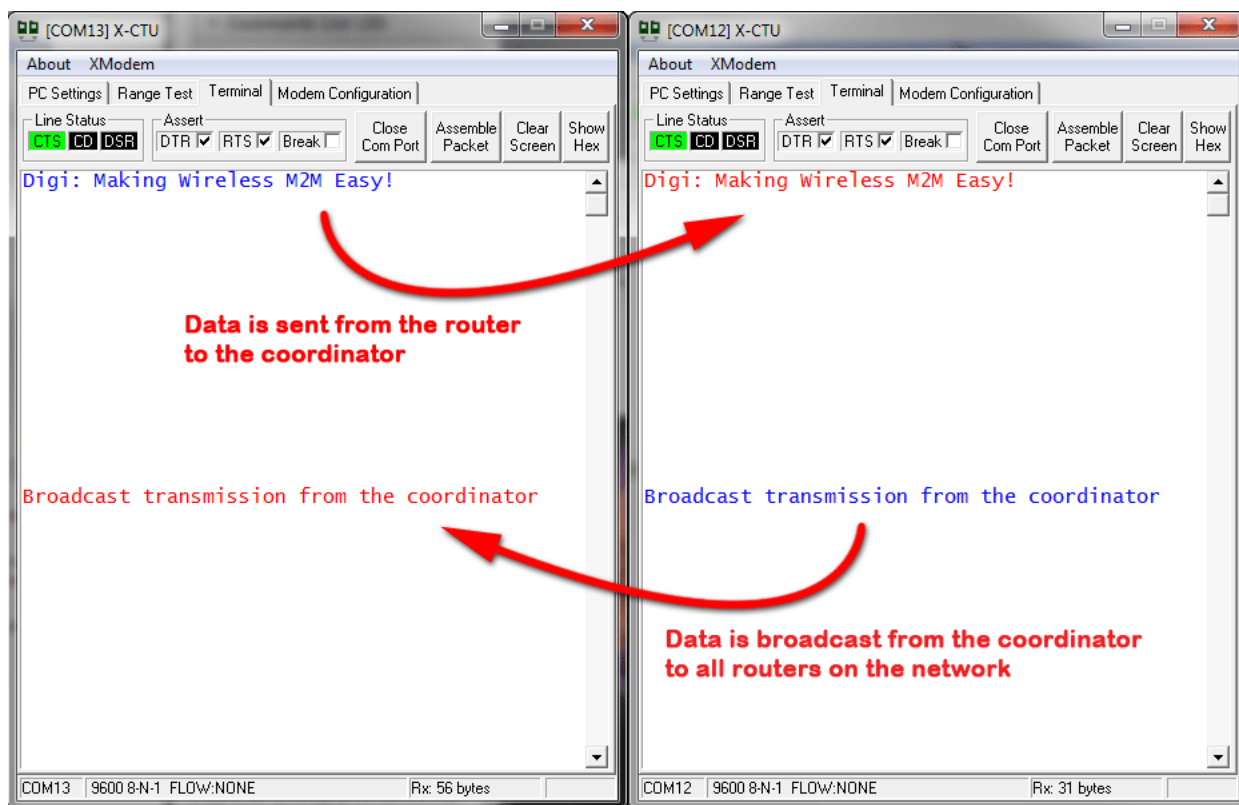


Note: You can set multiple parameters at once and then click the **Write** button to save them in the module.

9. Configure the remaining 3 modules following Steps 1-8. Ensure that CE is set to 0 on these modules. These will be designated as router (non-coordinator) modules.
 - a. Connect router modules in the network to the computer using the USB cable (two are included in the kit). Once the network is established, the red LED near the USB connector on the USB interface board will flash. Coordinators will flash at a rate of about once per second, and routers will flash at a rate of about twice per second.
 - b. Open an X-CTU window for each module connected to the computer, select the appropriate COM port, and go to the **Terminal** tab. USB interface boards can be connected one by one to the network and computer so you can identify which module goes with which COM port.

Note: X-CTU only looks for available COM ports on start-up. If you plug in a USB interface board after starting X-CTU, it will not be detected unless you close and re-start X-CTU.

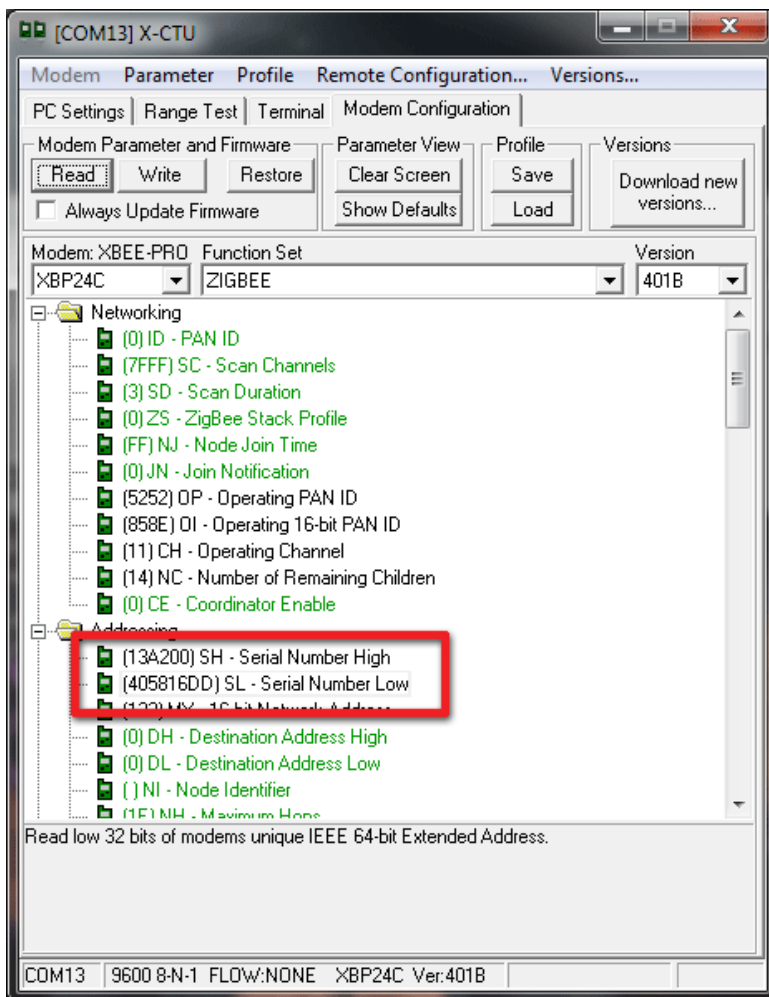
10. In this configuration, the routers are set up to communicate with the coordinator. Using the **Terminal** tab as shown below, any text entered in the window of a router module's terminal will show up in the coordinator's window as a repeat of the character. Text transmitted is shown in blue while text received is shown in red.



Note: In this configuration, the coordinator is broadcasting its data to the routers (the destination address of 0xFFFF is the broadcast address). The routers are unicasting their data to the coordinator (the destination address of 0x00 is the coordinator). You might notice that if data is rapidly entered into the terminal window of the coordinator, there will be a long delay before it reaches the routers. This is because broadcast transmissions in a ZigBee network require a large amount of network overhead and are not practical in most situations.

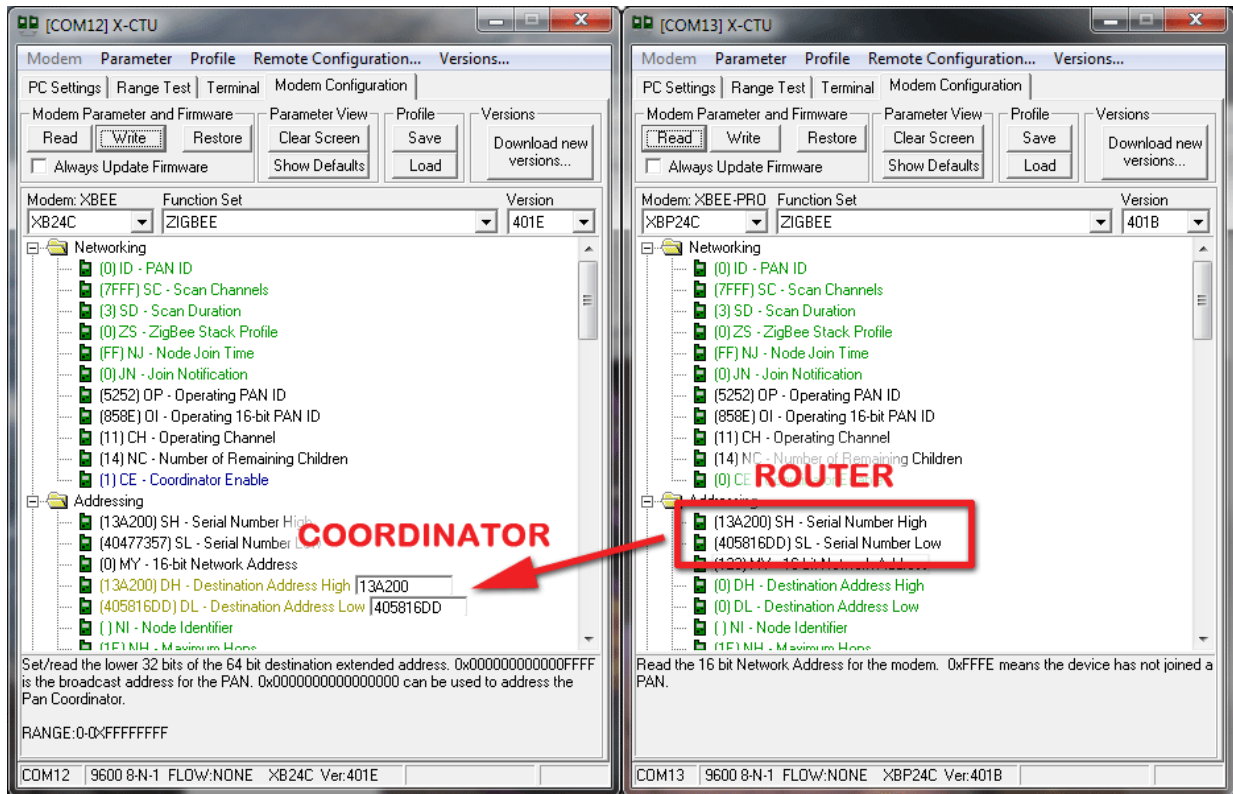
In the next step, we will change the addressing of the coordinator to Unicast to a specific router.

11. In order to address to a specific router, we need to determine what the serial number of the router is:
 - a. Open the Modem Configuration tab of the router that is connected to your PC and click **Read**.
 - b. Make a note of the SH (Serial Number High) and SL (Serial Number Low).



Note: The serial number is also printed on the label located on the RF shield of the XBee module.

12. We now need to assign this serial number as the destination address on the coordinator:
 - a. On the coordinator, open the **Modem Configuration** tab, and click on the **DH - Destination Address High** parameter and enter the **SH - Serial Number High** of the router.
 - b. Click on the **DL - Destination Address Low** and enter the **SL - Serial Number Low** of the router.
 - c. Then click the **Write** button.



You can now open the terminal tab for both modules as you did before to verify that the addressing is assigned correctly.

This router will be used in the next section to perform a range test.

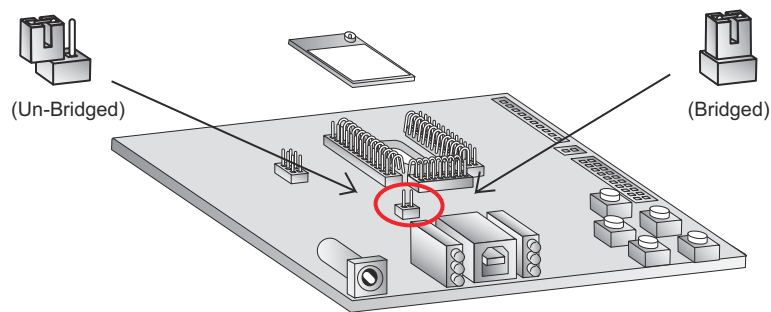
Perform a Range Test

1. While the coordinator module remains plugged into the computer, apply power to the router we addressed earlier using a power supply in a fixed location. This module will be designated as your remote module.

Note: You will need to have a jumper at P8 on the loopback header on the XBIB for the remote module.

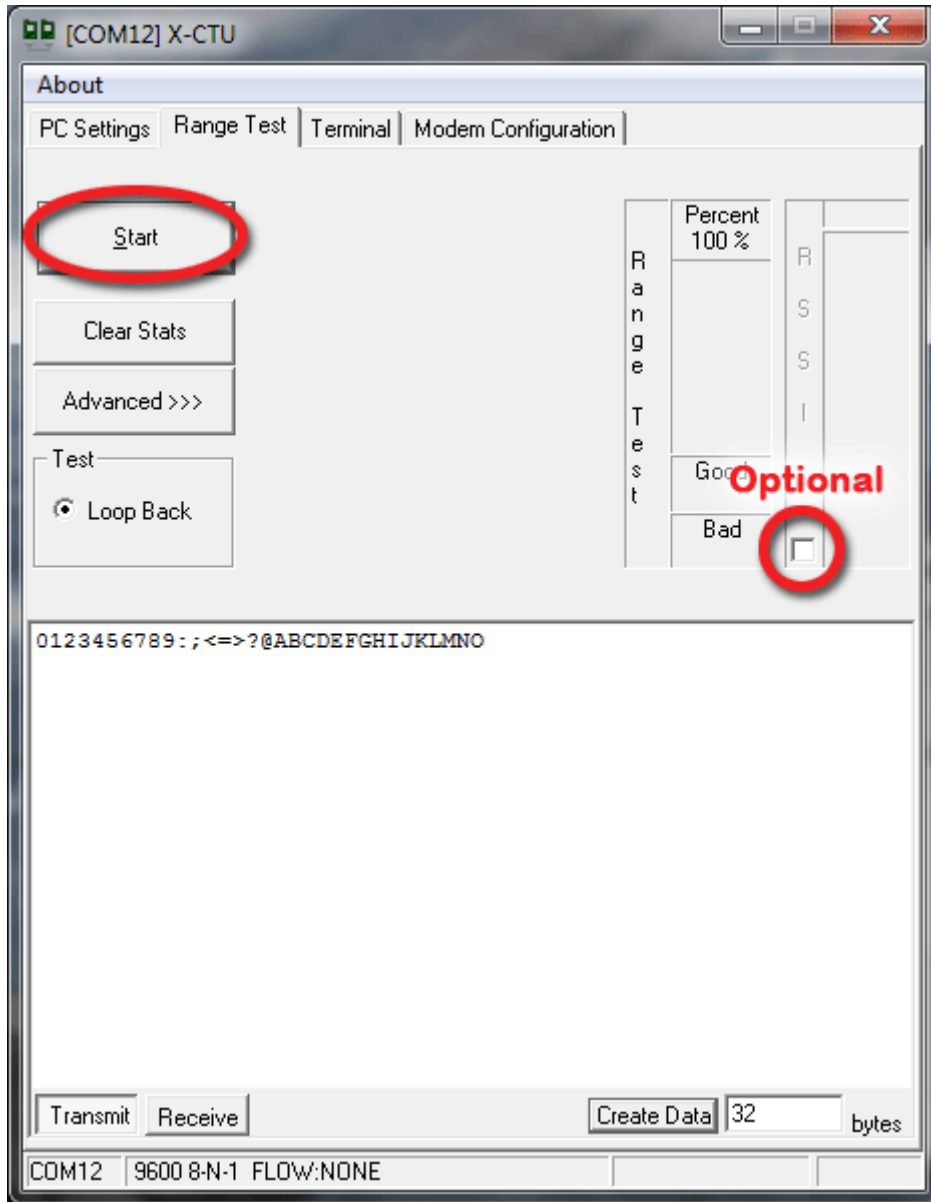


Ensure that the P8 jumper is not bridging the two pins together on the base module. This could cause X-CTU to stall if the jumper is populated and requests are sent to the module.



2. Select the **Range Test** tab.
3. (Optional) Check the **“RSSI” check box** to display the Received Signal Strength Indicator.

4. Click **Start** to begin the range test.



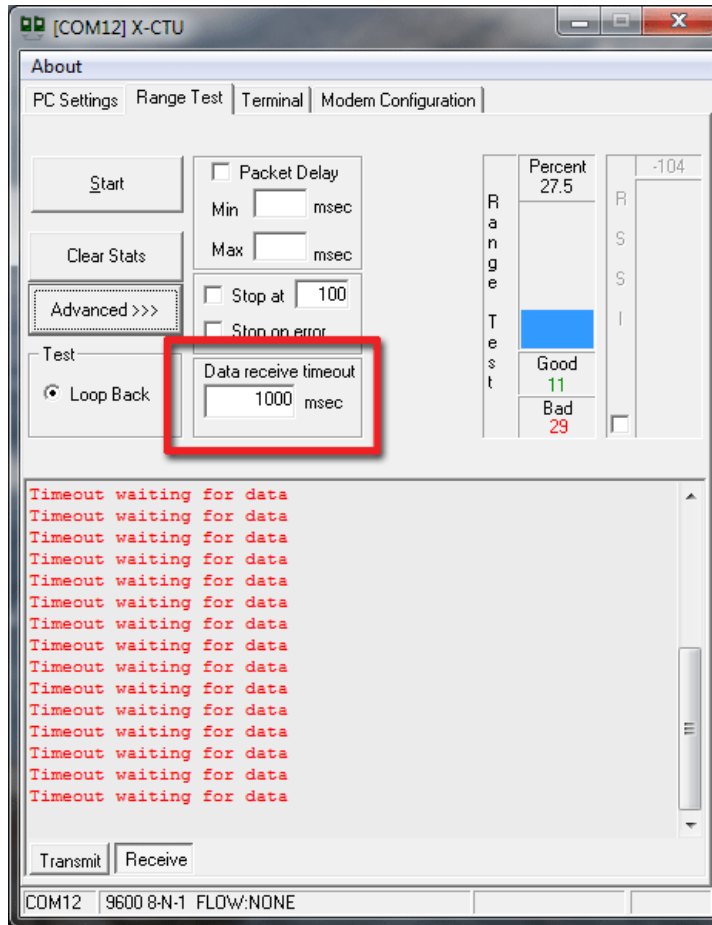
5. Monitor the link quality by reading the Percent section on the Range Test tab. This section displays the running percentage of good packets sent to the receiving module and looped back to the base.



As your distance increases beyond the maximum range of the modules, you will start seeing greater packet loss.

6. Click **Stop** to end the range test.

The **Advanced** tab provides you with more options. If you are running the range test with several mesh modules acting as intermediaries, increase the Data receive timeout value to compensate for additional latency.



You have just completed Goal #3 - performing a range test.

Establish a Mesh Network

You have already established a point-to-multipoint network. Now it is time to extend this network by creating a self-healing mesh network.

1. Restart the range test.
2. Move the remote module further away from the base module until the signal is lost.
3. Assemble another module and interface board.
4. Connect with a power supply and place this device halfway between the base and remote modules.

The intermediate module bridges the gap between the remote and the base and reestablishes communication. The network "self-healed" by redirecting communications as soon as a pathway became available.



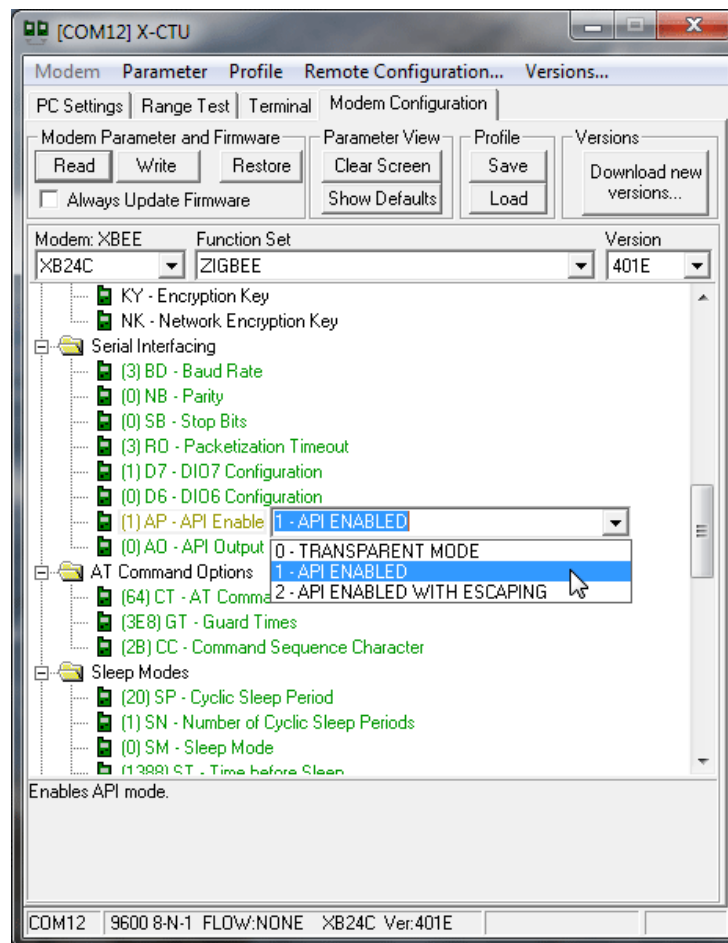
You have just completed Goal #4 - establishing a mesh network.

Part 4: Over-The-Air Configuration

Configure Remote Modules

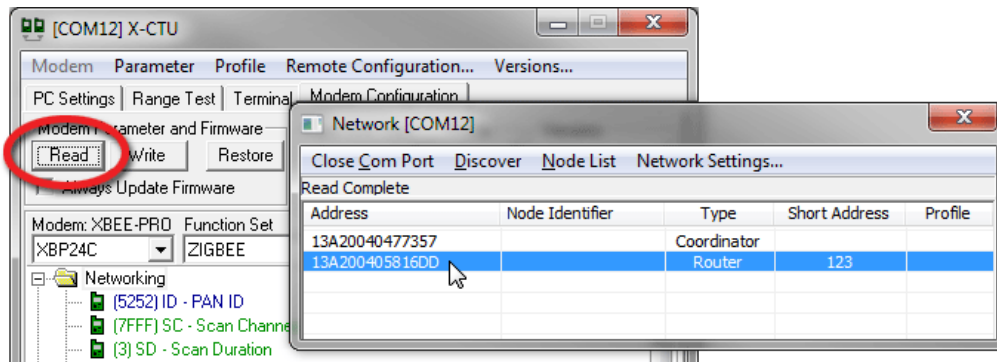
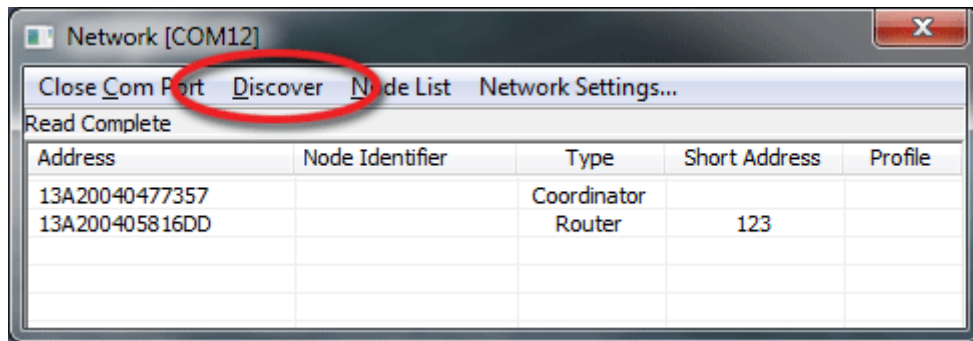
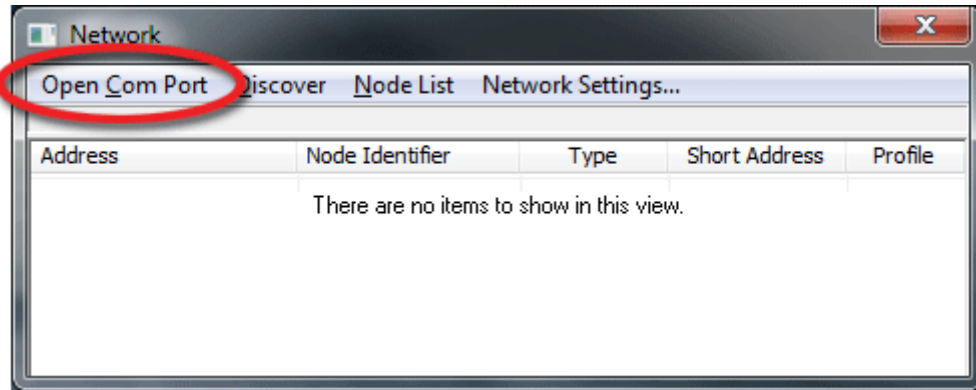
You can configure the XBee modules over-the-air using X-CTU. To use this feature, the coordinator must be configured for API mode. For more information about API mode, please reference the [Product Manual for the XBee ZB SMT RF Module](#).

1. On the base module, go to the **Modem Configuration** tab in X-CTU.
2. Click **Read** and the parameter types will appear in the window.



3. Set AP = 1 on your base module by clicking on the **AP** command in the **Serial Interfacing** folder in the **Modem Configuration** window.
4. Click the **Write** button.
5. Next, click **Remote Configuration** in the middle-top of the **Modem Configuration** tab. The **Network** window will appear.

6. Click **Open Com Port** and **Discover**; a list of network nodes will appear on the screen.



By selecting a particular node from the list, you can interact with it as if it were connected to the PC directly. Now you can **Read**, **Write**, and **Restore** parameters on the main X-CTU window, and those changes will occur over the air on the remote module selected in the **Network** window. Close the **Network** window when you have finished with remote configuration.



You have just completed Goal #5 - over-the-air configuration.

Part 5: XBee ZB SMT Design Considerations

Software Design Considerations

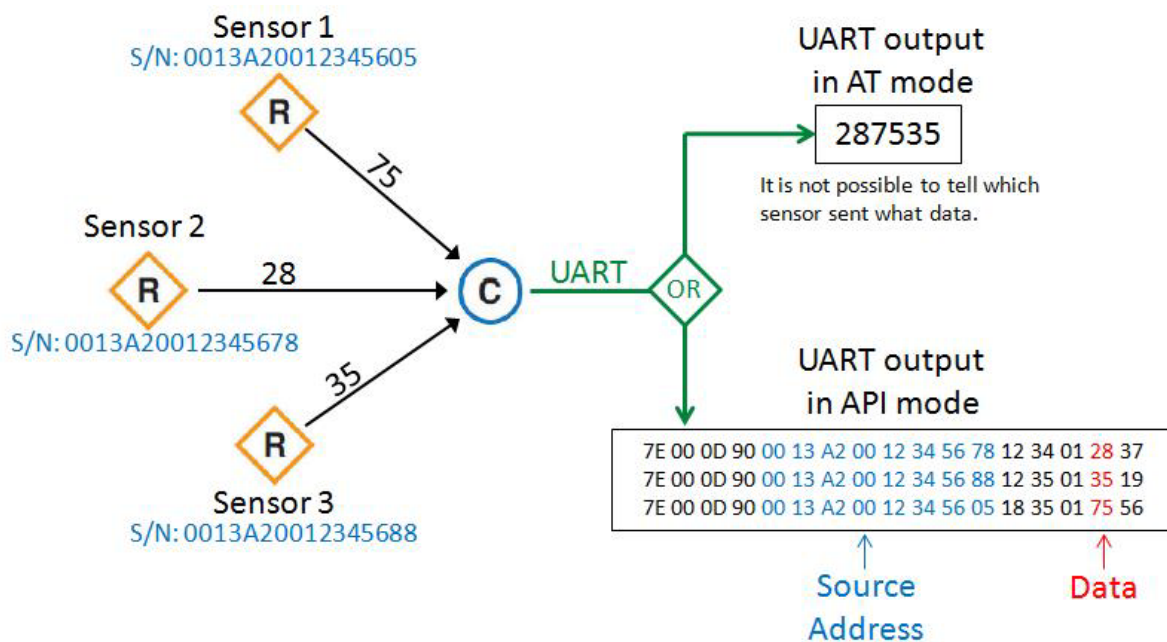
So far, we have been transmitting data between the modules in AT (transparent) mode. AT mode on the XBee is beneficial because it is very simple to send and receive data. In this configuration, it acts as though you had connected the two modules with a wire: whatever data goes into one module comes out of the other transparently.

The other operating mode is API. This is a much more powerful way of transmitting data and remotely controlling the module. If you performed the “over-the-air configuration” earlier in this guide, you have already performed some operations using API (Remote AT commands through X-CTU).

Here is an example of API mode in a real-world application:

A user has installed three sensors to monitor his factory equipment. He would like to wirelessly gather the sensor data to a base module and be able to remotely control a relay located at each sensor. If he were to use AT (transparent) mode, there would be no way to determine which sensor sent what data. The data from the three sensors would come out of the base module’s UART in a jumbled mess.

By simply changing the base module to API mode, the data that is sent by the remote modules will come out of the UART in an API frame. Included in this frame is the source address, as well as the sensor data, so it is very easy to keep track of each sensor. This process works similarly for sending data back to a specific module. The user can also send a Remote AT command to toggle a pin on one of the sensor modules, allowing him to remotely control the relay.



API mode is covered in detail in Section 9 (API Operation) of the manual. We also have a useful tool available to aid in prototyping and testing API frames called [Digi API Frame Maker](#).

Hardware Design Considerations

It is highly recommended to review the Design Notes section of the XBee ZB SMT manual. This will provide you with board layout considerations, powersupply design, and recommended pin connections.

There are some additional recommendations for implementing the XBee into your hardware design.

The XBee ZB SMT module is designed to be soldered in place. This creates some unique challenges for troubleshooting and reprogramming the XBee. The primary means of updating and configuring the module will be via over-the-air configuration, however there are some factors that will cause the module to not be accessible in this way (bad firmware flash, unknown PAN ID, etc.) Reworking the module after it has been installed will void the warranty, so it is recommended to have a physical programming header installed on your board. This header will allow you to access the necessary pins of the XBee to configure the module or perform a recovery if needed. Note that two devices cannot share the same UART connection, so if a programming header is installed there should be some way to allow the header exclusive access to the XBee, either with a physical jumper or software solution. The recommended pin connections are listed in the manual.

The XBee ZB SMT module is 3.3v only; it is not 5v tolerant on any pin. Applying a voltage higher than 3.3v will damage the module and void the warranty.

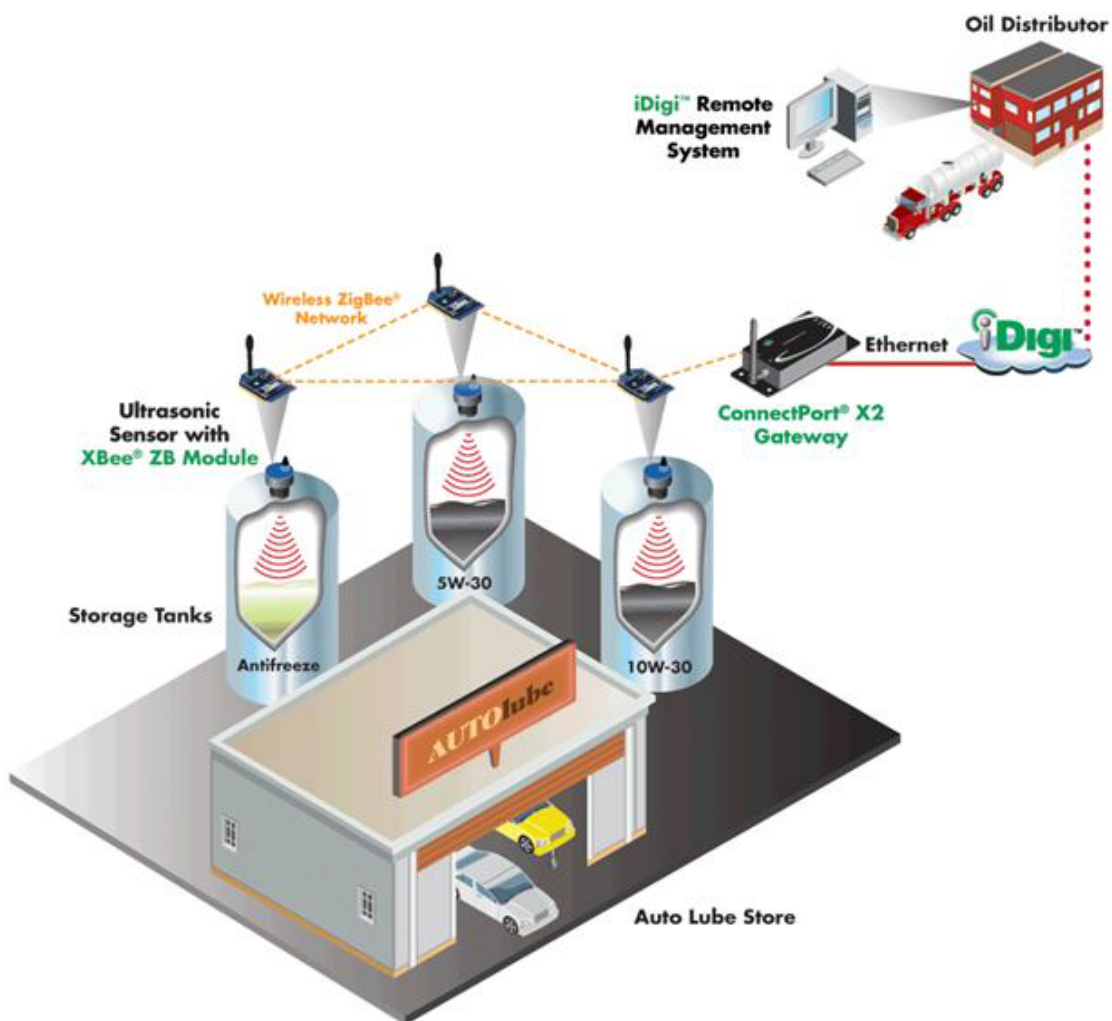
You may also want to consider implementing a dual-footprint design. This footprint gives you the flexibility to implement either an SMT or through-hole XBee. Details on this dual-footprint are in Appendix B of the manual.

There is a [schematic](#) available for our X-BIB-U interface board; this can be used as a reference for how to design a proper interface to the XBee.

Part 6: Managing the Network

Digi Gateways

Now that you have an RF network running, it is important to know how to manage that network. X-CTU software and the various RF module features provide great tools for on-site XBee configuration, firmware upgrades, and network diagnostics. However, in many cases it is necessary to be able to perform these functions from a remote location via the Internet. Digi has solved this problem by creating RF-to-Internet gateways to allow remote access, monitoring and control of your RF network over the Internet.



In the diagram above, a Digi ConnectPort X2 gateway is used to connect an RF network (plus other peripherals) to the Internet via either the Ethernet port or a cellular WAN connection. Note that gateways are programmable and can also be used to run local applications, which can monitor/control the RF network, with or without Internet connectivity.

See <http://www.digi.com/products/wireless-routers-gateways/gateways/> for an overview of Digi's suite of gateway products. A simple ConnectPort X2 ZigBee to Ethernet gateway costs around \$100 and can quickly internet-enable your ZigBee network.

To the Cloud! Make the most of your data.

So you've created an RF network and connected it to the internet with a Digi gateway. Now what? The next step is to decide what you want to do with all of that valuable data. Ask yourself some fundamental questions. Does an application need to be developed, or is there an existing application that will be utilized? How should the data be presented? Who will be using this data? Is the audience internal, external or both? Whatever the answers to these questions, Device Cloud by Etherios™ can help you realize your vision and meet your needs for data capture.

Device Cloud is a public cloud platform-as-a-service that allows any application, anywhere, to connect with anything, anywhere. Device Cloud takes care of the infrastructure, scalability and security, letting you focus on what you do best - developing awesome products and applications. Get up and running with Device Cloud for free and connect your device to explore what Device Cloud can do for you. Simply go to www.etherios.com/devicecloud and click on Get Started Now to set up a free demo account.

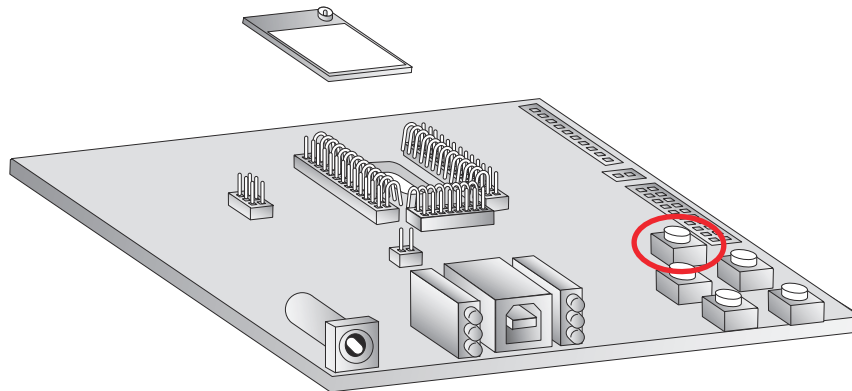
To the Cloud! Take control of your device network.

Picture this. You have a network of thousands of devices to manage, many of them remote. You need to update firmware, edit configurations, download software onto the various device types, dealing with IP addresses, firewalls, security and server reliability. Sounds like a mammoth task. The Device Manager application, hosted on Device Cloud, meets this challenge head on, allowing you to connect, control and monitor the vast array of devices in your network. Perfect for large-scale deployments, you can group devices and schedule operations, making device network management easy. Take a tour of Device Manager at <http://www.etherios.com/devicecloud/devicemanager> and get started for free.

Appendix A: Troubleshooting

Resetting the XBee ZB SMT RF Module

Each XBee USB Development Board has a reset button (located as shown below):



Pressing this button power cycles the module, but will not clear any changes written to the module. This is useful if you are having issues accessing the COM port. This will also reset any parameters that were changed but not written into memory. If your module is sleeping, pressing the reset button will wake it temporarily.

Note: The remaining buttons are connected to various I/O lines and are not used in this kit. See the [product manual](#) for more details relating to this functionality.

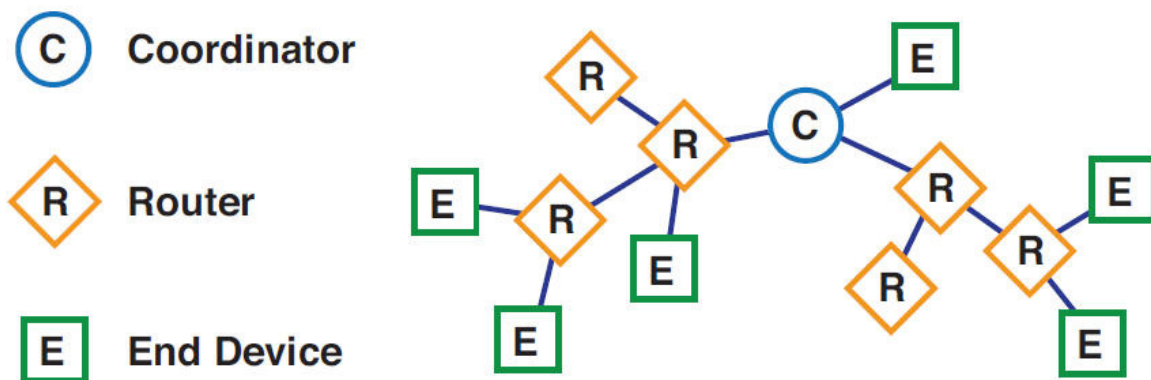
Why are the Modules no longer communicating with one another?

Network settings that can cause loss of communication include Baud Rate (BD), Parity (NB), and Encryption Enable (EE) among others. Check to see if these parameters are set appropriately. If you are unsure if your settings are affecting your communication, you might want to try setting your modules back to their default settings. To do so, go to the Modem Configuration tab in X-CTU and click Restore.

Appendix B: ZigBee Functionality

ZigBee Nodes

The ZigBee Protocol defines three types of nodes: Coordinators, Routers, and End Devices, with a requirement of one Coordinator per network. While all nodes can send and receive data, there are differences in the specific roles they play.



Coordinators are the most capable of the three node types. There is exactly one coordinator in each network and it is the device that establishes the network originally. It is able to store information about the network, including security keys.

Routers act as intermediate nodes, relaying data from other devices.

End Devices can be low-power / battery-powered devices. They have sufficient functionality to talk to their parents (either the coordinator or a router) and cannot relay data from other devices. This reduced functionality allows for the potential to reduce cost.

ZigBee offers these advantages:

- Open standard with interoperability between vendors
- Option for lower cost, reduced function end nodes