

## 3: XBee Testing & Configuration

Communications between the PC and XBee can be an integral part of your system whether for XBee configuration, monitoring and control of a device, or simply for testing and feedback. Using Digi's X-CTU software and Parallax's XBee USB Adapter board, interfacing with the XBee is simple for configuration changes, firmware downloads, testing signal strength, and communications to your remote devices. In this chapter many configuration settings of the XBee will be explored, but not all. Please refer to Digi International's XBee manual for more information.



### We are using Series 1 Only!

The term "XBee" in the remainder of the tutorial will refer to any of the Series 1 XBee or XBee-Pro, unless otherwise noted.

### XBee Testing

In this section we will explore Digi International's X-CTU software for communications, signal strength monitoring and configuration of the XBee. The XBee connected to the PC using the XBee USB adapter and a remote XBee with power and a loopback will be used. For the loopback, DOUT is connected to DIN which will cause any received data to be transmitted back the PC as shown in Figure 3-1.

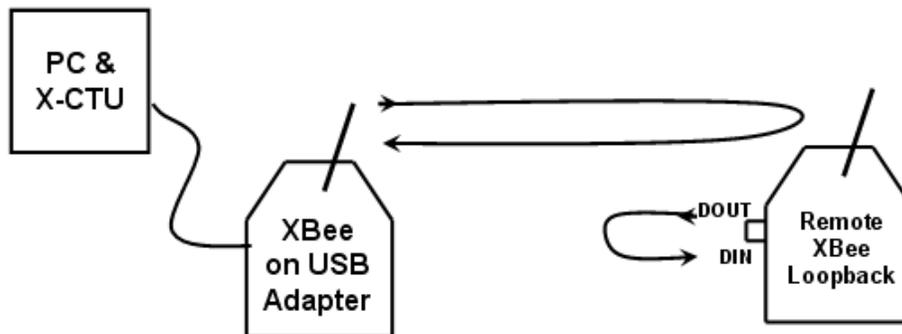


Figure 3-1: XBee Loopback Testing

### Hardware Required

- 2 or more XBee modules
- XBee USB Adapter (#32400)
- One or more XBee adapter boards appropriate for your controller (BASIC Stamp, Propeller chip, etc)
- 220  $\Omega$  resistor and LED (optional)

### Other USB Options:

- **Using Propeller chip for PC Communications:** The Propeller may be used for USB communication to the XBee using **Serial\_Pass\_Through.spin** in Chapter 4. This allows all functions except firmware downloads.
- **Using the Prop Plug for PC Communications:** The Prop Plug (#32201) may also be used for USB communications to the XBee by connecting DOUT to Rx, DIN to Tx, and Vss to VSS with the adapter board powered normally. This allows all functions except firmware downloads.

## 3: XBee Testing & Configuration

---

### Software Required

- X-CTU software available from [www.digi.com](http://www.digi.com) or through Parallax's XBee page, [www.parallax.com/go/XBee](http://www.parallax.com/go/XBee). After installing, it will ask you if you wish to download the latest versions. This is very time consuming and may be skipped initially in most cases.
- FTDI drivers are available from Parallax: [www.parallax.com/go/usbdrivers](http://www.parallax.com/go/usbdrivers).

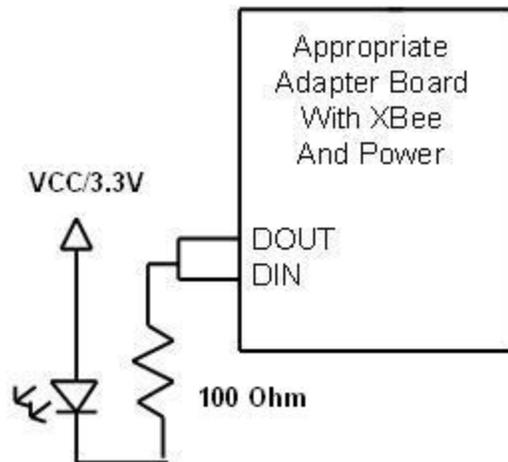


#### FTDI Drivers

If you have installed the BASIC Stamp Editor or the Propeller Tool software, you probably have the drivers installed already.

### Hardware Connections:

- ✓ With an XBee module in place, connect the XBee USB Adapter Board to the PC using an appropriate USB cable.
- ✓ With an appropriate board with an XBee for your BASIC Stamp or Propeller, connect as shown in Figure 3-2:
  - Connect VDD or VCC (as appropriate) and VSS per Chapter 2.
  - Use a jumper wire to connect DOUT to DIN on your breadboard connection.
  - If your adapter board does not have Tx/Rx LEDs, you may optionally connect an LED from DOUT/DIN to the VCC/3.3 V supply power.
  - If your adapter board does not have RSSI or ASSOC LEDs, these may be added per Chapter 2.



**Figure 3-2: Remote XBee with Loopback and optional LED**

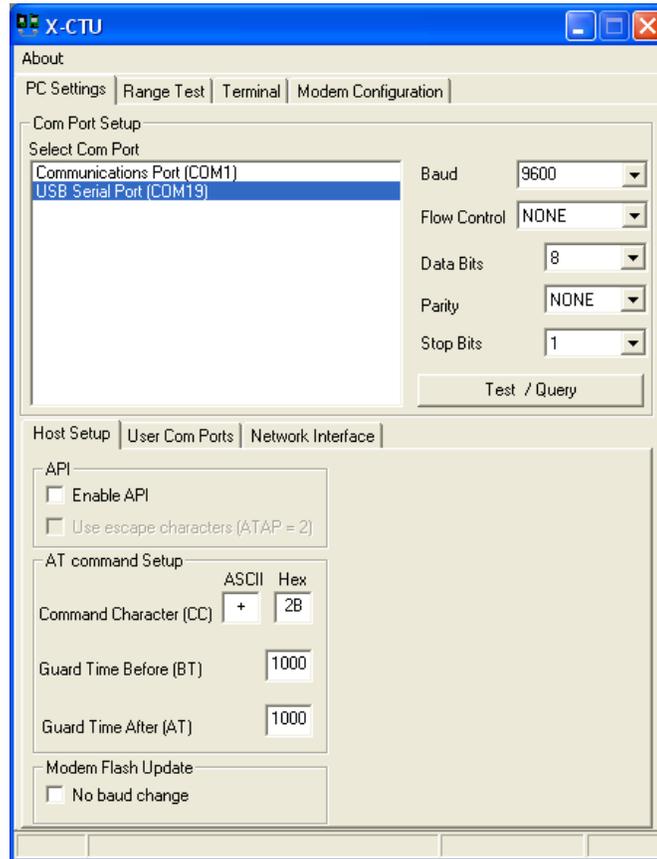
After powering up, both units should have POWER and ON/SLEEP LEDs illuminated, and ASSOC LEDs blinking.

### Testing Communications

In this section we will test basic communications with the X-CTU software.

- ✓ Ensure the remote XBee with loopback is powered.
- ✓ Open the X-CTU software.

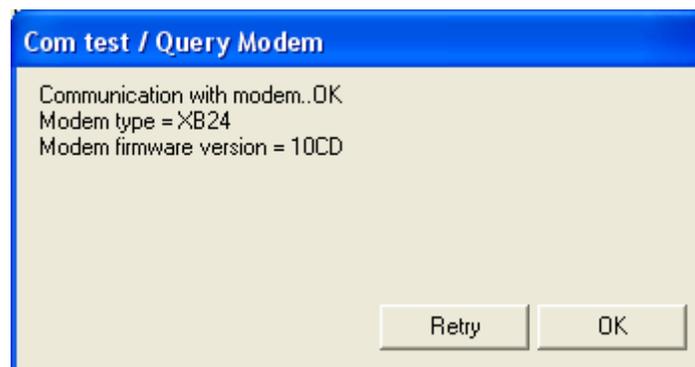
- ✓ If your XBee USB Adapter is connected you should have one or more COM ports listed as shown in Figure 3-3.



**Figure 3-3: X-CTU COM Port Selection**

- ✓ Select one and click the **Test / Query** button. The proper selection will show a response similar to Figure 3-4. The message box shows the type of XBee and the firmware version.

**Troubleshooting**  
If your Test / Query does not find an XBee or other tests in this section fail, please see XBee Troubleshooting in Appendix B .



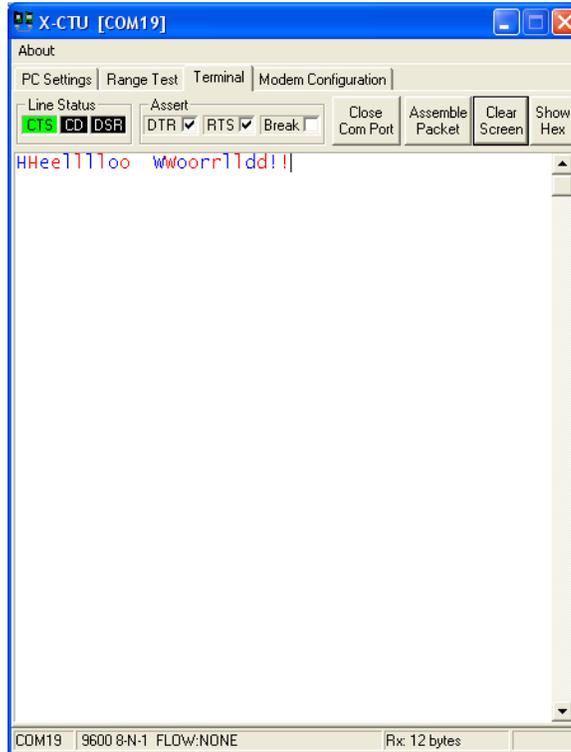
**Figure 3-4: X-CTU Test/Query Response**

### 3: XBee Testing & Configuration

---

- ✓ Click the **Terminal** tab on the software. This tab allows you to send and receive data using a terminal interface. We will also use this for XBee configuration changes shortly.
- ✓ In the window area, type **Hello World!**

If all went well, you should see two sets of each character—a blue followed by a red as shown in Figure 3-5. The blue character is what you sent, the red is what was received.

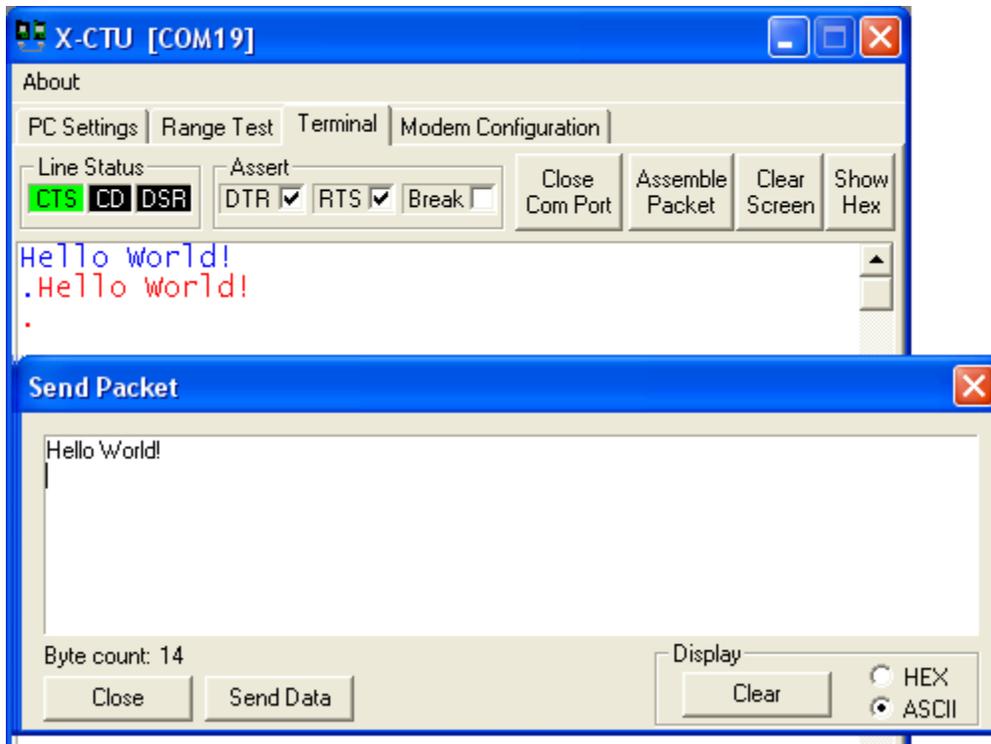


**Figure 3-5: X-CTU Terminal Test**

As each character was typed, it was transmitted, looped-back, and returned for reception and display.

- ✓ Type a few more characters and monitor the LEDs on your boards. On both boards you should see the RSSI LED illuminate for several seconds with each character received. If there are Tx/Rx LEDs, these should light on the remote board, blinking rapidly as data is sent and received.
- ✓ Click the **Clear Screen** button. In the terminal window, click the **Assemble Packet** button. This opens a window where you can type text to be sent as a single packet.
- ✓ Type **Hello World!** and press the **Enter** key in the text box then click **Send Data**.

In the terminal window you should see your text twice as shown in Figure 3-6. The entire text is sent quickly enough to send the entire text in a single packet transmitted to the remote XBee and returned.



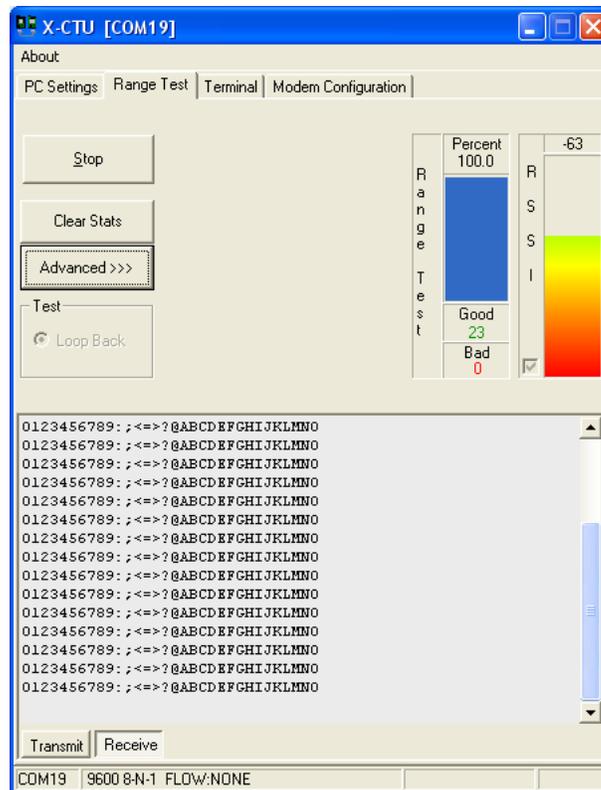
**Figure 3-6: Assembling and Sending a Text Packet**

While it may be interesting at this time to see what would happen if there were two remote XBee modules with loopback (Do I get 2 responses back?), it may not be a good idea just yet. One remote XBee transmitting will cause a repeat on the other XBee, and back again; the one transmission will be repeated over and over between XBee modules essentially causing a broadcast storm. Because all the XBee modules currently have the same address by default, they will be sending to each other as well as the base and repeating what they get. A better time to test this is when we experiment with addressing.

### Range Testing

- ✓ Close the **Send Packet** window and click the **Range Test** tab on the software.
- ✓ Under the vertical **RSSI** lettering, check the checkbox to monitor signal strength.
- ✓ Click **Start**. The software will repeatedly send packets and display the signal strength and percentage of correctly returned packets as well as the packet itself as shown in Figure 3-7. Click **Stop** once you've seen enough.

### 3: XBee Testing & Configuration



**Figure 3-7: Range Testing and RSSI Monitoring**

RSSI is the Received Signal Strength Indicator value. This value will be between -40 dBm (the highest recorded value) and -100 dBm (the minimum sensitivity of the XBee).

**Packet Errors**

In theory you should never see a malformed (bad) packet returned. All data is sent/received with error checking. If the packet between XBee modules contained errors, there is a high probability the error was detected and the packet discarded and resent up to 3 times until properly acknowledged. A more likely source of errors is the serial link between the XBee and the PC (or controllers when used).

This is a good time to test environmental conditions on your transmission link.

- ✓ Test interference of the signal by placing your hands over it, blocking line-of-sight with your body, placing a metal box over an XBee, or moving the remote XBee to a further location or a different room or floor in the building.
- ✓ Click **Stop** when done testing.

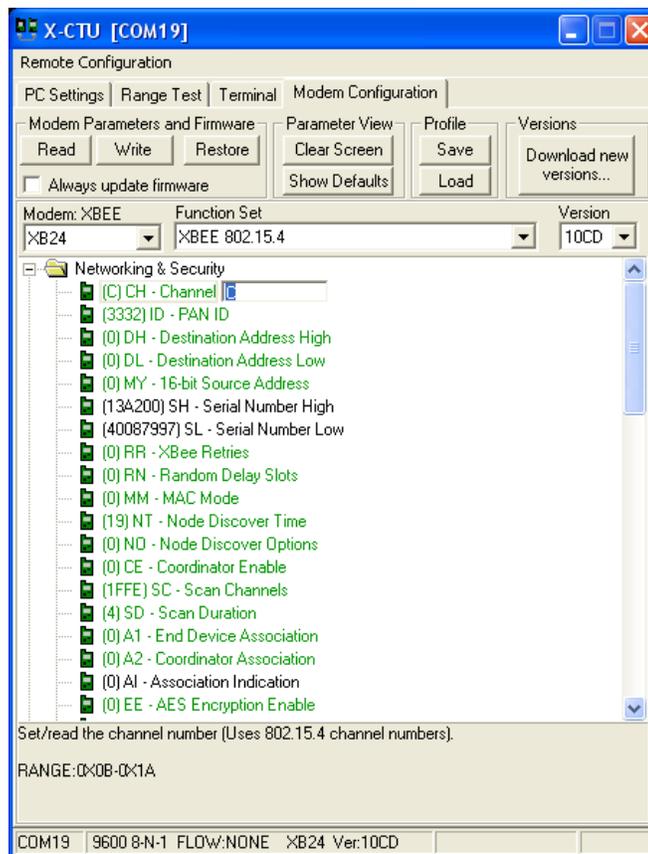
### XBee Configuration

Using X-CTU, the configuration of the XBee may be modified and the firmware upgraded. Through the software, the XBee configuration may be read, changed and written for updates. The XBee may also be configured through any terminal interface or from the controller by sending the correct character sequences.

### Using the Modem Configuration Tab

The Modem Configuration tab may be used to read, modify, and write changes the XBee's configuration. It may also be used to update the firmware on the module.

- ✓ With an XBee connected, open X-CTU and select the correct COM port.
- ✓ Click the **Modem Configuration** tab.
- ✓ Click the **Read** button. If all goes well, you should see the window populated by the current settings as shown in Figure 3-8.



**Figure 3-8: X-CTU XBee Modem Configuration Tab**



#### Troubleshooting

Should you have communications problems, please see the Troubleshooting Appendix.

- ✓ Click the **Restore** button. Once complete as indicated in bottom text area, click **Read** again. This ensures the XBee has the default configuration in the event it had been modified by you or someone else.

Notice the drop-down boxes near the top. They list the type of modem found, the firmware type on the module and the firmware version.

## 3: XBee Testing & Configuration

---

### **Modifying Settings**

To modify settings, click the setting, such as **(C) CH – Channel** as shown in Figure 3-8. The current setting is C. Note at the bottom of the window a brief description and valid range is provided. The channel (CH) may have a range of C to 1A on the XBee (non-Pro). On XBee-Pro it is C to 17. All parameters are in hexadecimal.

- ✓ Click the **CH – Channel** setting
- ✓ Modify to **D**
- ✓ **Write** then **Read** and verify the change (current setting is shown in parenthesis before the command name)
- ✓ **Restore** and **Read** and verify it is back to C.

	<b>Configuration Settings</b> Changes to configurations are stored in non-volatile memory meaning that are permanent until changed again or restored.
---	--

### **Updating Firmware**

Different firmware versions have different features, though most are simply new revisions with some fixes. In order to use analog/digital features, version 10A1 or higher is needed. To perform remote configuration features, version 10CD or higher is needed. If you feel you don't have newer versions or need to have the latest, use **Download New Versions** from the Web.

	<b>Updating Warning!</b> Updating on occasions does not go well. Multiple retries may be required. A failure to update properly could cause inoperability of your XBee. While most times updates go smoothly, there always exists the possibility of a corrupted update. Please see Appendix B: Troubleshooting if problems arise.
---	---

To change the firmware on your XBee:

- ✓ Perform a **Restore** and **Read** the XBee.
- ✓ Select the Firmware version.
- ✓ Check the **Always update firmware** checkbox.
- ✓ Click **Write**. The new firmware and current settings should be downloaded to the XBee module.
- ✓ Uncheck the **Always update firmware** checkbox.

	<b>Updating firmware requires more communications than simply Tx and Rx.</b> DTR and CTS are used as well. While configurations may be read and changed with a simple Tx/Rx interface, a full interface is needed for firmware changes, such as with the XBee USB Adapter.
---	---

### Using the Terminal Interface (AT Command Mode)

The XBee may also be configured using AT commands via the X-CTU terminal interface or any terminal program. This concept is important in that we will duplicate this when configuring the XBee from our controller. The first step is to place the XBee into a Command Mode. We don't want this to happen accidentally, so a specific sequence is required to enter command mode:

- A 2 to 3 second wait since data was sent
- A string of +++ (with no Enter key pressed)
- Another 2 second wait
- The XBee will reply with OK when ready. The timeout value on Command mode is fairly short, so too long a delay will cause it to exit the mode requiring you to re-enter the mode if needed.
- The **ATCN** command can be used to exit Command mode.
- Multiple commands may be entered on one line.

Let's test it (example shown in Figure 3-9).

- ✓ Go to the **Terminal** tab.
- ✓ Type **Hello** to ensure you are communicating with the remote loopback XBee.
- ✓ Wait 3 seconds.
- ✓ Type +++ (With no Enter or Carriage Return).
- ✓ Wait 2 more seconds, and you should see OK.
- ✓ Enter **ATCH & Enter**.
  - You should see C returned – you requested the current value for channel.
- ✓ Enter **ATCH D & Enter**.
  - You should see OK.
- ✓ Enter **ATCN & Enter** to exit the Command Mode.
- ✓ Type **Hello** again.
  - You should NOT see it returned from the remote XBee. The base XBee is on Channel D and the remote XBee is on Channel C so they will not communicate.
- ✓ Repeat the above changing the base XBee's channel back to C (**ATCH C**) and test communications with the remote XBee.
- ✓ If you timeout (suddenly your characters are echoed back when trying to enter command), use **ATCT 500** to increase the timeout value.
- ✓ When complete, use **ATRE** to restore your XBee to default values.
- ✓ Multiple command may be entered at once:  
**ATCH C,CT 500,CN & Enter**

## 3: XBee Testing & Configuration



Figure 3-9: Using AT Command Mode in Terminal



**Note: Changes made to configuration settings using AT Commands are temporary.**

The last saved setting will be used if the XBee is reset or powered down & up. To store your settings permanently, use the AT Command **ATWR** before exiting command mode. **ATRE** may be used to restore to default, but be sure to use **ATWR** after to save to memory.

### Overview of Common Configuration Settings

#### Addressing – MY & DL

The address of the XBee defines its place on the network. It can be used to specify data to a particular node (destination) and the sender's address (source) in some cases. By default, all XBee's have an address of 0 (the MY address) and a destination low address of 0 (DL). In our loopback testing, the XBee modules used the default configuration and both were sending data from address 0 to address 0 being able to communicate with each other. While this works fine for 2-node networks, in larger networks you may want to send data to a particular node instead of several using the same address. To do this, node address (MY) and destination Low (DL) address need to be set. A broadcast address (DL of FFFF) may be used to send data to all nodes on the same network.

With your loopback configuration up and communicating, let's test some addressing.

- ✓ Using the **Terminal** Tab, type something and ensure it is echoed back to the base XBee.

## 3: XBee Testing & Configuration

- ✓ Notice that on both the base and remote XBee, the RSSI LED lights and the TX/RX LEDs will blink (if available).
- ✓ Using the Modem Configuration or terminal AT commands, change DL to a value of 1 (setting the destination of the data to address 1: **ATDL 1**).
- ✓ Type a string such as **Hello**. The data should NOT be echoed back.
- ✓ Observe the LEDs on the remote XBee and loopback XBee.

The RSSI and Rx LEDs should not light. Your data is intended for address 1. Since the remote XBee is on address 0 it will see the data but ignore it. It will NOT pass it through your DOUT pin lighting the Rx LED if connected and returning the data.

- ✓ Change DL back to 0.
- ✓ Change MY to 1 to set the address of your unit to address 1.
- ✓ Send some data.

You should see the RSSI LED and Rx LED as well as the Tx LED light on the remote XBee. You should NOT see the RSSI or Rx LED light on the base XBee. The remote XBee accepted the data addressed to 0, and looped it back to send it to its DL address of 0. Since the base XBee is on address 1, it will not accept the data since the address did not match.

- ✓ Change MY back to 0.
- ✓ Change DL to FFFF, the broadcast address.
- ✓ Send some data – it should once again be looped back and displayed. The DL address of FFFF informs all remote XBee modules to accept the data regardless of their address (broadcast to all).
- ✓ Return the DL address to 0.

Remote XBee modules may be manually configured for different addresses. If you have multiple remote XBee modules, you can set them up for loopback and the addressing scheme shown in Figure 3-10.

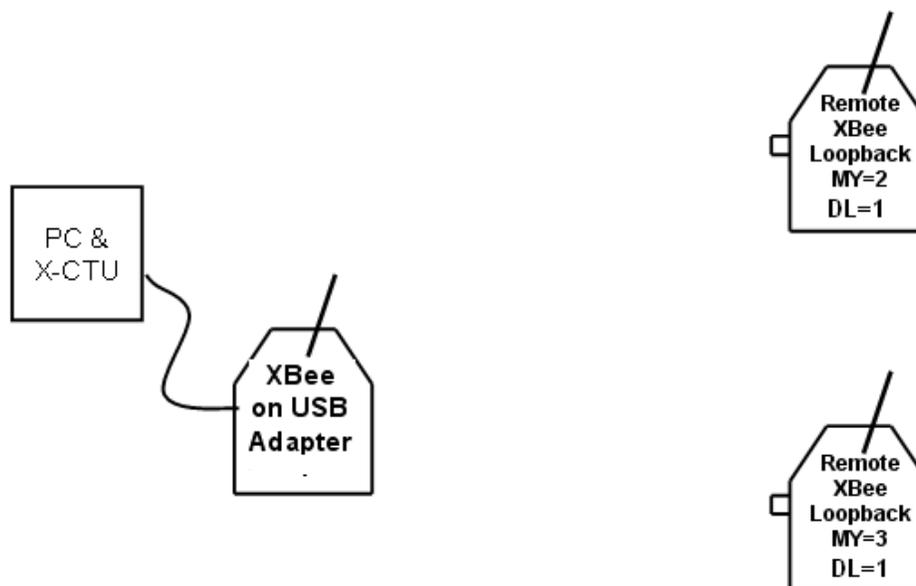


Figure 3-10: Multi-Node Addressing Test

## 3: XBee Testing & Configuration

---

- ✓ On X-CTU, return to the **PC Settings** tab.
- ✓ Disconnect the USB cable.
- ✓ Power-down and remove the remote XBee from its board and place in the XBee USB Adapter.
- ✓ Reconnect USB.
- ✓ Read and modify MY and DL address as shown in Figure 3-10 using **Modem Configuration** to ensure it is not temporary.
- ✓ Replace it in remote adapter with loopback.
- ✓ Repeat for other remote XBee modules.
- ✓ Using the base XBee, modify the DL and MY addresses to test communications to one and to both (broadcast) remote units. The base will need a MY of 1 to accept data returned. It will need a DL of 2 or 3 to get data to the remotes. Use a DL of FFFF to send and receive from BOTH remotes.



**Try it!**

Configure the addresses of all units to send data from base to 1st remote, from 1st remote to 2nd remote, and from 2nd remote back to base.

Hint: You can modify the RP setting to reduce the amount of time the RSSI light is on to better see each character being received in the even you don't have Tx/Rx LEDs connected. Use ATRP 2 for a 200 ms blink of RSSI light.

- ✓ Use **Restore** to return all XBee modules to default configurations when testing is complete.

### **Node Identifier – NI**

The node identifier allows us to assign a noun name to the XBee node to be seen when a node discovery is performed: **ATNI Node 1**

### **Node Discovery – ND**

One other interesting test is the Node Discovery (**ND**) command. Using AT Command Mode in the terminal window, enter **ATND** and wait. Other nodes in your network will be listed showing:

- Their MY address
- Their serial numbers
- Their noun names if set (NI setting).

The RSSI level (signal strength) of the received information from that node (further on the RSSI level is discussed).

### **Network Separation – ID & CH**

In addition to node addresses, sometimes we may want different group of XBee modules in the same area to be isolated from one another for communications—different networks in the same physical area. This may be done by using either unique Personal Area Network (PAN) IDs (ID setting), or by placing them on different channel frequencies (CH).

Using unique PAN IDs, even though two XBee modules may have the same address, only the XBee with a matching PAN ID will accept and use the data. This allows multiple networks to operate in the same area where XBee modules may have the same addresses.

- ✓ You may test the lack of communication by changing the ID setting of your base XBee and test the loopback.

Note that this does not reduce the number of nodes fighting for the use of the frequency medium, which is called *contention*. Even though IDs are different, the XBee modules are still on the same frequency channel (CH) and only one XBee can transmit on that channel at a time if in RF range of each other.

XBee networks may also be separated by putting different networks on different frequency channels (CH). This has the benefit of isolating the networks and having them operate on different frequencies, reducing contention. Beyond separation, there may be “better” frequencies to operate at than others in an area. With the plethora of devices in the 2.4 GHz spectrum, some channels in the spectrum may be noisier than others. Using the Energy Detect command, you may monitor the signal strengths of the channels using the **ATED** command.

- ✓ Use the terminal window and enter AT Command Mode.
- ✓ Enter **ATED** to view energy on various channels, from lowest channel (C) to highest (1A, or 17 if Pro).

Figure 3-11 shows the result of 2 tests. The first was with a nearby microwave oven off, the second with it on. The higher the magnitude it shows, the less noise on that channel. A value such as 4E would convert to a decimal value of 78, so this channel would have a noise level of -78 dBm. We want as little noise in the background as possible. A higher magnitude means a lower noise level. Having the microwave on raised the noise on nearly all channels (there is also WiFi in the area on 2.4 GHz which may be affecting channel readings as well).



The screenshot shows a terminal window titled "X-CTU [COM19]". The window has tabs for "About", "PC Settings", "Range Test", "Terminal", and "Modem Configuration". The "Terminal" tab is active. The terminal displays the following text:

```
ATED
5E
60
38
60
39
4D
60
55
56
60
59
56
+++OK
ated
4E
38
46
56
46
4E
51
4E
46
25
40
4E
```

The status bar at the bottom of the terminal window shows "COM19", "9600 8-N-1 FLOW:NONE", and "Rx: 105 bytes".

Figure 3-11: Energy Detection Tests

## 3: XBee Testing & Configuration

---

### Receiver Signal Strength – DB

The XBee can report the receiver signal strength (RSSI) in a number of ways. One way is to poll the unit using the ATDB command. The XBee will report RSSI level as a hexadecimal value indicating the magnitude. This value, converted to decimal and made negative, is the signal strength from -40 to -100 dBm. The more negative the signal strength (in dBm), the weaker the signal. Therefore, -50dBm is better than -60 dBm).

- ✓ Use the **Terminal** tab to send data to be echoed back.
- ✓ Enter Command Mode and enter **ATDB** for RSSI level.

### Packetization Timeout – RO

As the XBee accepts serial data, it only waits so long before packetizing and transmitting the data. The packetization timeout sets how long it will wait before transmitting the incoming serial data. If this value is too short, it may cause problems in that data we transmit becomes fragmented. In many cases the receiving XBee will reconstruct the data and buffer it for our use, but it does lead to more transmissions than needed and under certain conditions (such as using API mode on the receiving XBee) may cause issues.

In a program we may have code that sends some data, does some calculations, and sends more data. We wanted the data to be sent as one transmission, but while doing a calculation, the first chunk of data is transmitted. By increasing the value of RO we can help ensure our data stays together and is sent in one transmission. The value is RO is the number of character times we want the XBee to wait before sending the collected data. At 9600 bps, each character requires approximately 1 ms (0.1 ms per bit for 10 bits with the start/stop bits). With a default RO value of 3, the XBee will wait approximately 3 ms (3 character times) before packetizing and sending data in the buffer. This value can be increased to FF (255 decimal) for a timeout value of over 255 ms at 9600 baud.

- ✓ Test the loopback by typing **Hello world** quickly. Notice that each character is transmitted and echoed.
- ✓ Change the value of RO to FF using the Command Mode or Modem Configuration (**ATRO FF**).
- ✓ Type **Hello world** once again. If you're fairly quick you can type the entire phrase before it is packetized, transmitted and echoed.

### Configuring I/O – D0 to D8, P0, P1, M0, M1

The I/O on the XBee can be used for a number of uses – Digital and analog input (requires API packet parsing covered later), digital output, and communications handshaking (RTS, CTS, etc). Additionally, these and other pins can be used for special functions such as association indication, RSSI indication, Sleep mode input and PWM output. Additionally, inputs on one XBee can be controlling outputs on another using line-passing.

Due to the number of options available, the **Modem Configuration** tab is useful for seeing the options as well as configuring from there. We'll test a few out. As shown in Figure 3-12, Pin D5 can be used as the association indicator, ADC or digital input, and digital output low and high. Configuration of the pin can be done here or in command mode.

- ✓ For example, if you are tired of your association indicator on the XBee board blinking and wish to turn it off, set D5 to 4 (**ATD5 4**).

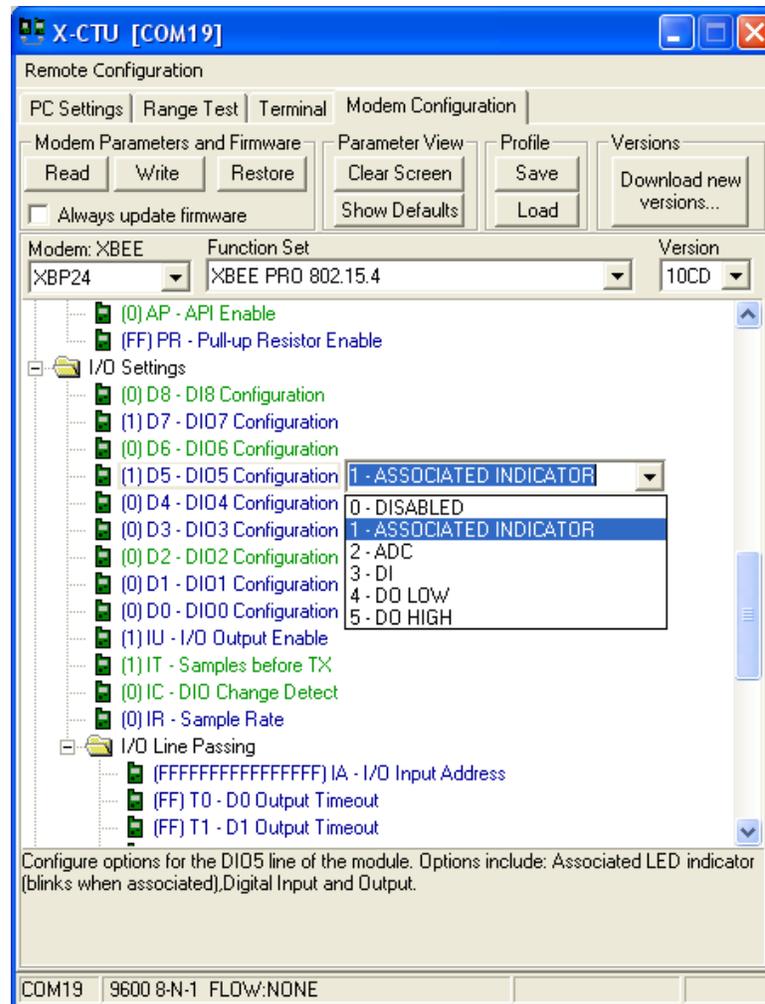


Figure 3-12: I/O Configuration Options

Two I/O can be used to generate Pulse Width Modulated output (PWM), such as the one driving the RSSI LED on your board. The function configuration is in the I/O Line Passing section. To set the PWM value, use Command Mode. To set the RSSI indicator for a certain output level to control LED brightness:

- ✓ In Command Mode, set PWM0 to PWM Output – **ATP0 2**.
- ✓ Set the PWM 10-bit value, in the range of 0 to 3FF (0 to 1023 decimal): **ATM0 5**.
- ✓ Test other PWM values and observe brightness. Your eye is filtering the LED blinking on and off quickly for the dimming effect. The output is still digital, but it may be filtered for analog voltage output.



**Note:** The XBee PWM is not the same type of PWM used for servos. This is a PWM-controlled voltage level as opposed to a PWM signal.

## Common Settings Summary

Table 1 is a list of common configuration settings used in this tutorial. For a full list, please see the XBee manual or the XBee Command Summary in Appendix C. Please see the referenced documentation or XBee X-CTU software for valid ranges and other information on use.

### 3: XBee Testing & Configuration

**Table 3-1: Common Configuration Settings**

Group	Command	Meaning & Use
<b>Networking &amp; Security</b>	CH	Channel: Sets the operating frequency channel within the 2.4 GHz band. This may be modified to find to a clearer channel or to separate XBee networks. Default value is C.
	ID	PAN ID: Essentially, the network ID. Different groups of XBee networks can be separated by setting up different PAN's (Personal Area Network). Default value is 3332.
	DL	Destination Low Address: The destination address of where the transmitted packet is intended for. We will use this often to define which node receives data. A hexadecimal value of FFFF performs a broadcast and sends data to all nodes on the PAN. The default value is 0.
	MY	Source Address: Sets the address of the node itself. This will be used often in all our configurations. The default value is 0.
	NI	Node Identifier: Sets a node's noun name.
<b>Sleep Modes</b>	SM	Sleep Mode. Allows the sleep mode to be selected for low power consumption (<10 $\mu$ A). It is discussed in Chapter 6.
<b>Serial Interfacing</b>	BD	Interface Data Rate: Sets baud rate of the serial data into and out of the XBee.
	AP	API Enable: Switches the XBee from transparent mode (AT) to a framed data version where the data must be manually framed with other information, such as address and checksum. This is a very powerful mode and will be explored in Chapter 6.
	RO	Packetization Timeout: In building a packet to be transmitted, the XBee waits a set length of time for another character.
<b>I/O Settings</b>	D0 – D8	Sets the function of the I/O pins on the XBee, such as digital output, input, ADC, RTS, CTS and other.
	P0, P1	Configures the function of PWM0 and PWM1
	RP	Sets the time length for the RSSI output.
	M0, M1	Sets the PWM value of the PWM outputs
	IR	Sample Rate: The XBee can be configured to automatically send data from digital I/O or ADC's. It required the receiving node to be in API mode and the data parsed for the I/O values. See Chapter 6.
<b>Diagnostics</b>	DB	Received Signal Strength: The XBee can be polled to send back the RSSI level of the last packet received.
	EC	CCA Failures: The protocol performs clear channel assessment (CCA) – that is, it listens to the RF levels before it transmits. If it cannot get an opening, the packet will fail and the CCA counter will be incremented.
	EA	ACK Failures: If a packet is transmitted but receives no acknowledgement that data reached the destination, EA is incremented. The XBee performs 2 retries before failure. Additional retries can be added by using RR setting.
<b>AT Command Options</b>	CT	AT Command Timeout: Once in command mode, sets how long of a delay before returning to normal operation.
	GT	Guard Time: When switching into AT command mode, defines how long the guard times should be (absence of data before the command line) so that accidental mode change is not performed.
<b>Configurations</b>	WR	Write the configuration to non-volatile memory
	RE	Restore to default configuration in volatile memory

### **Summary**

The loopback test is an excellent means of testing communications and various configurations. The XBee has a multitude of configuration settings to deal with addressing, interfacing and input and output control. Using X-CTU interface allows range testing with loopback, modem configuration and a terminal window for communications, testing and configuration. In the next chapter we will explore code for the controllers in sending and receiving data and configuring the XBee.