The Life and Times of a Cell Phone

Wireless Communication Activity

University of Washington, Dept. of Electrical Engineering

Distributed Microsystems Laboratory

**Introduction**

This activity will introduce the student to the process of wireless communication using two transmitter/receivers called Xbees. The Xbees are designed to work easily with Arduinos and we will take name these Xbees, talk to them, test their range, and use them to collect sensor information remotely.

**Materials & Tools**

*This activity will require the following materials for each team of two students:*

1 USB cable (to connect the Arduino card to the computer)

1 Arduino Card

1 Arduino Shield

2 Xbees

1 USB Explorer

1 Breadboard

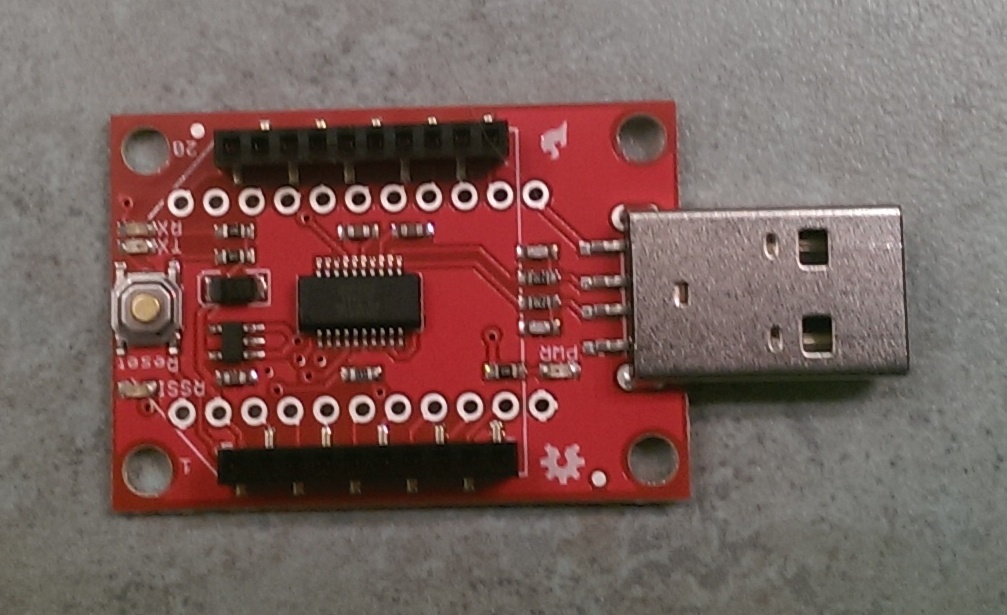
Variable Sensor and components to complete Procedure 4 (open design)

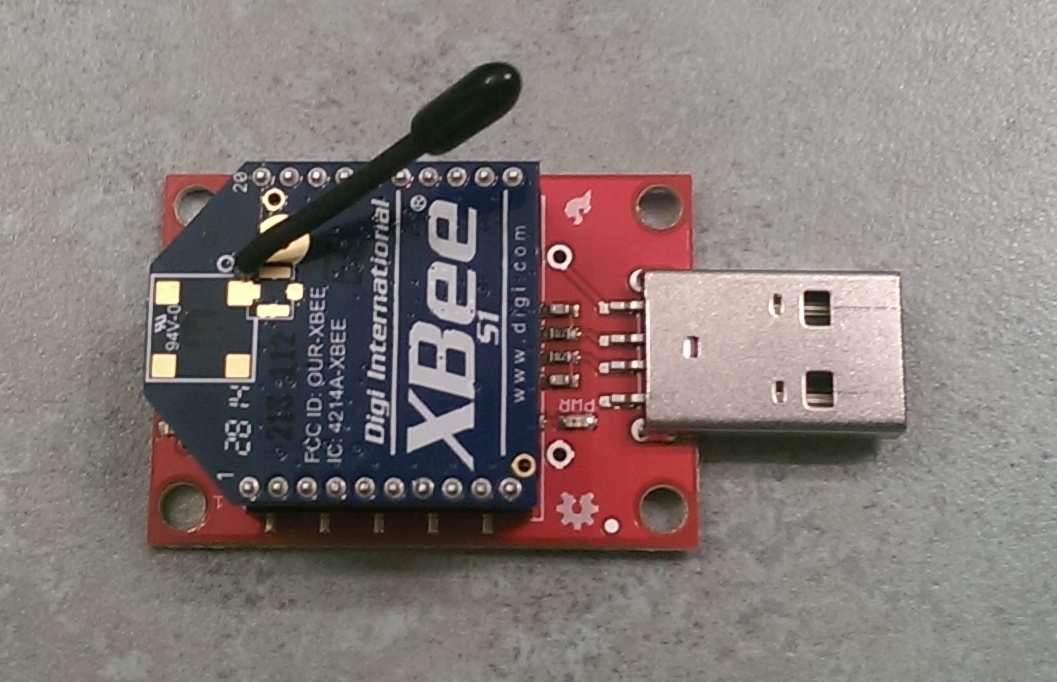
**Precautions**

Take care when inserting and removing electronic components to avoid bending pins or touching adjacent pins together. Do not touch circuit components when power is applied… always turn power off before changing connections, moving components, or otherwise altering a circuit. Don't forget to wash your hands after the activity!

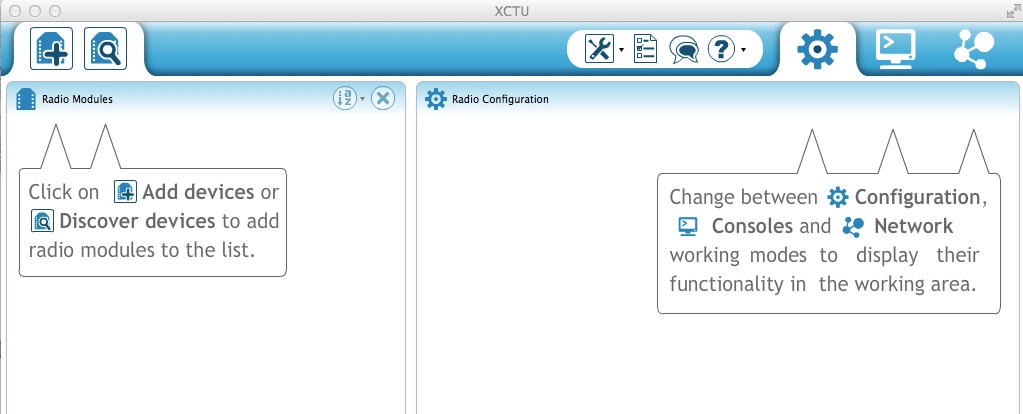
**Procedure #1 (Naming and Marrying an Xbee pair)**

In this first procedure, students will work in groups of two to program their Xbee wireless transceivers so that they can only talk (transmit and receive) to each other (and no other Xbee in the room). This will involve naming each Xbee and ensuring that each Xbee is “married” to only one other Xbee. We’ll do this with the XCTU program as follows:

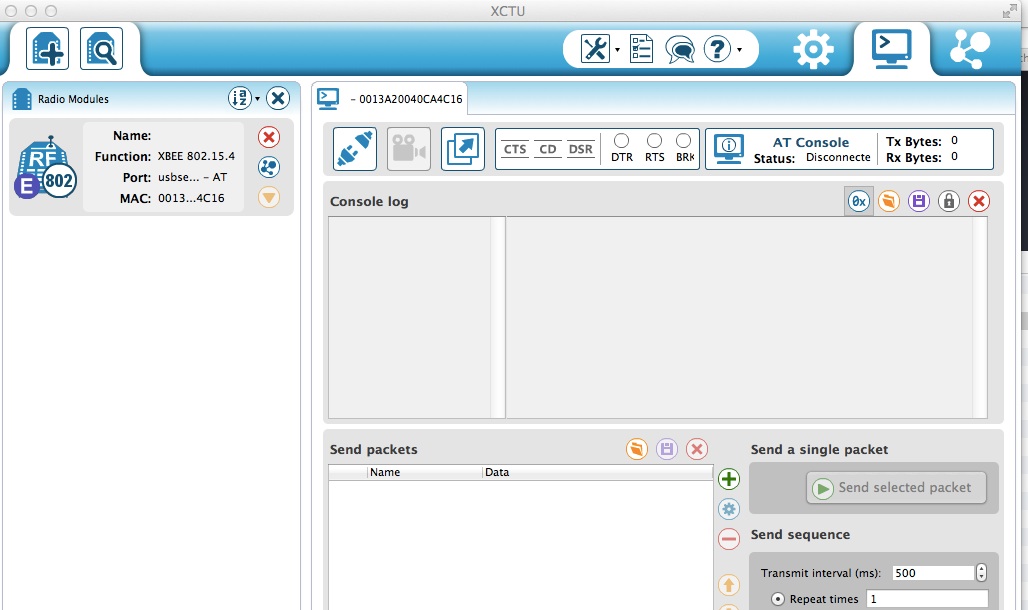
1. Locate the USB Explorer. It looks like this: 
2. Carefully plug in the AWAY Xbee to the USB explorer in the orientation shown below. The AWAY Xbee is the one that will be used as the remote transceiver (the one that will not be connected to a computer) Be careful not to bend any pins while inserting the AWAY Xbee into the USB explorer:



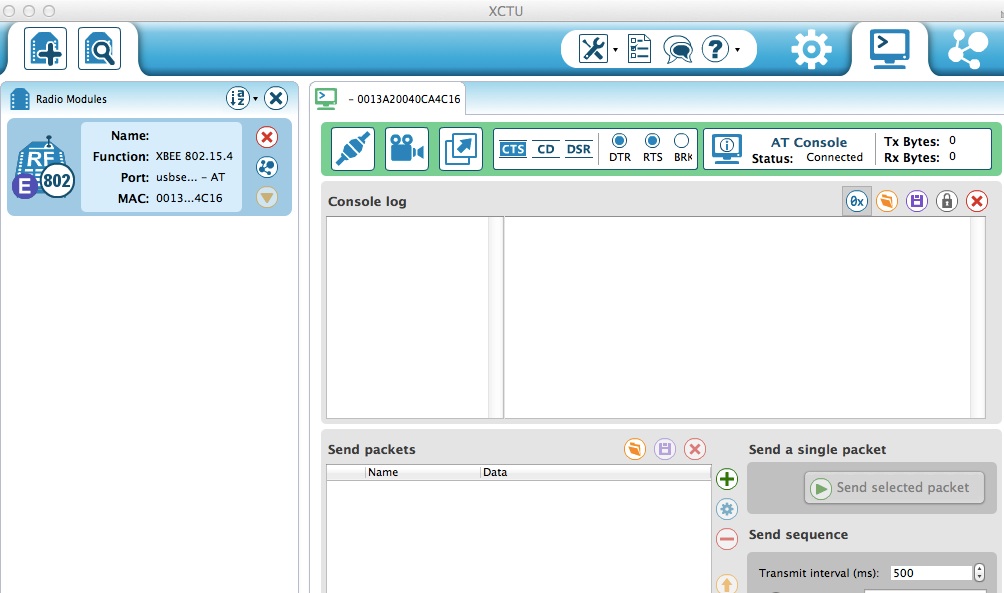
1. Plug the Xbee and USB explorer assembly into the USB port of the laptop computer.
2. Open the XCTU program. You should get a screen that looks like this:



1. Select the + icon (Add devices). Select the serial port to which the USB explorer and AWAY Xbee are connected. If you do not see the serial port to which the USB explorer is connected, click refresh ports. Once you have selected the correct serial port, click finish. You have now told the XCTU program that you will be communicating with your AWAY Xbee through the chosen serial port.
2. Select the monitor icon at the top of the XCTU screen (second from the right at the top). You should then get the following screen in XCTU:



1. Click on the Connect icon (it will say “open the serial connection with the radio module” when you mouseover it). The Connect icon looks like two mating plugs that are disconnected. When you click on it, it will transform to mating plugs that are connected, indicating that you are now connected to your AWAY Xbee and USB explorer.



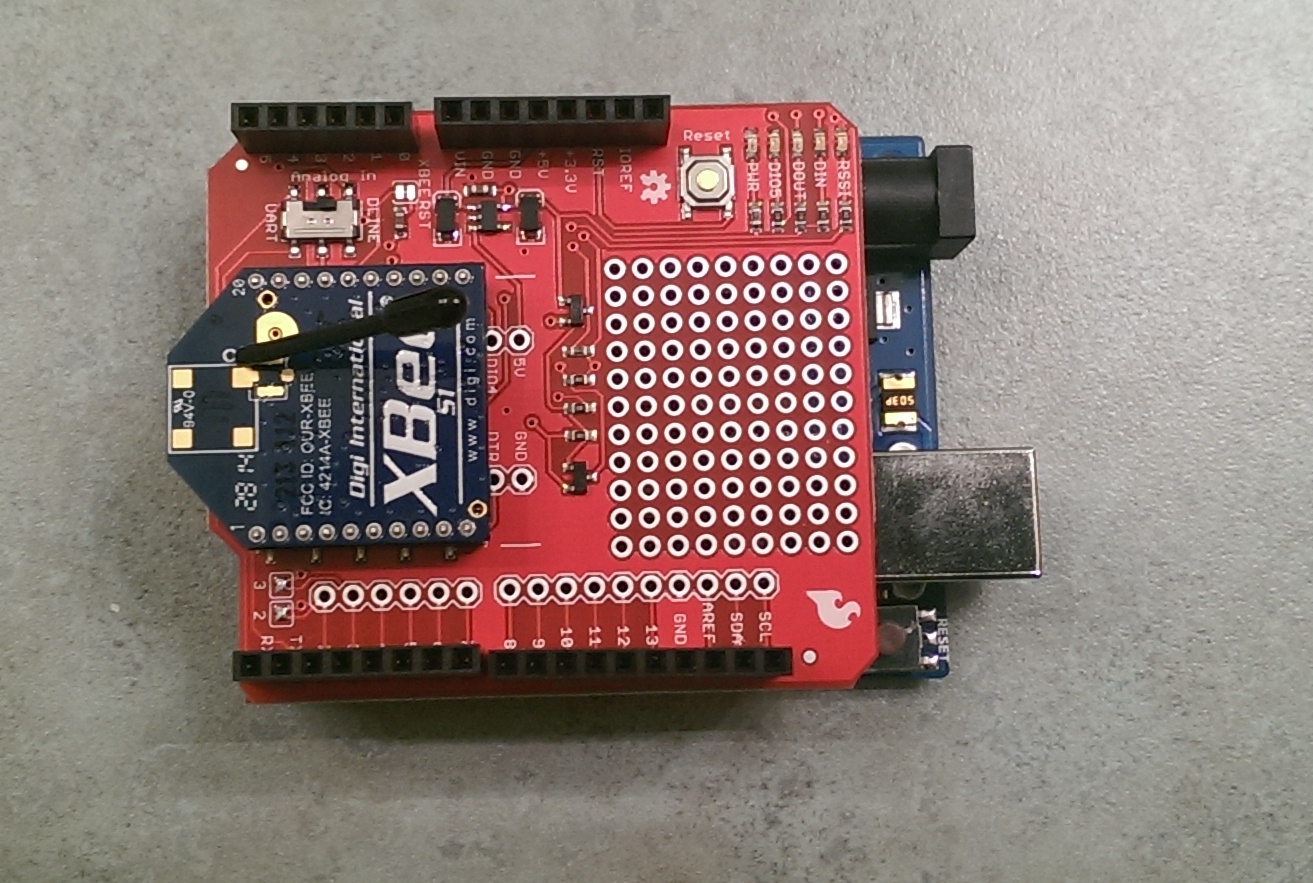
1. In the Console log screen, type +++ to enter command mode. Do not press enter. You should receive a response of OK.
2. Select a name for the AWAY Xbee (it should be four digits, all numbers). Select a name for the HOME Xbee (it should also be four digits, all numbers). Make sure both names are different from names chosen by others within 100 feet of the Xbees.
3. In the Console, type ATMYXXXX where XXXX is the name of the AWAY Xbee. Press enter after you complete typing in this command. This command sets the current Xbee address.
4. In the Console, type ATDLXXXX where XXXX is the name of the HOME Xbee Press enter after you complete typing in this command. This command sets the destination address (the HOME Xbee with which your AWAY Xbee will be communicating).
5. Now that you have named both of your Xbees and told your AWAY Xbee its own name as well as the name of the HOME Xbee to which it will be married, store the data in the Xbee. Type ATWR and press enter. You have now completed naming your AWAY Xbee.
6. Check the settings on your AWAY Xbee to ensure the baud rate is 9600. Do this by opening the settings (the icon next to the monitor icon) on the XCTU screen.
7. Scroll down to the BD Interface Data Rate. Make sure 9600 is selected. This ensures that the AWAY Xbee is communicating at a rate of 9600 baud (bits per second). Both the HOME and AWAY Xbees need to communicate at the same rate, or they will not understand each other.
8. Disconnect the AWAY Xbee by clicking on the Connect icon.
9. Remove the AWAY Xbee module from the USB port.
10. Remove the AWAY Xbee from the USB explorer.
11. Repeat this procedure ith the HOME Xbee. All procedures (steps 1-18) will be the same, EXCEPT the ATMYXXX and ATDLXXXX commands will be swapped (the name used for ATMY for the AWAY Xbee will become the ATDL name for the HOME Xbee and vice versa.
12. Close the XCTU program.

You have now finished programming your Xbees to communicate only with one another and no one else. Congratulations!

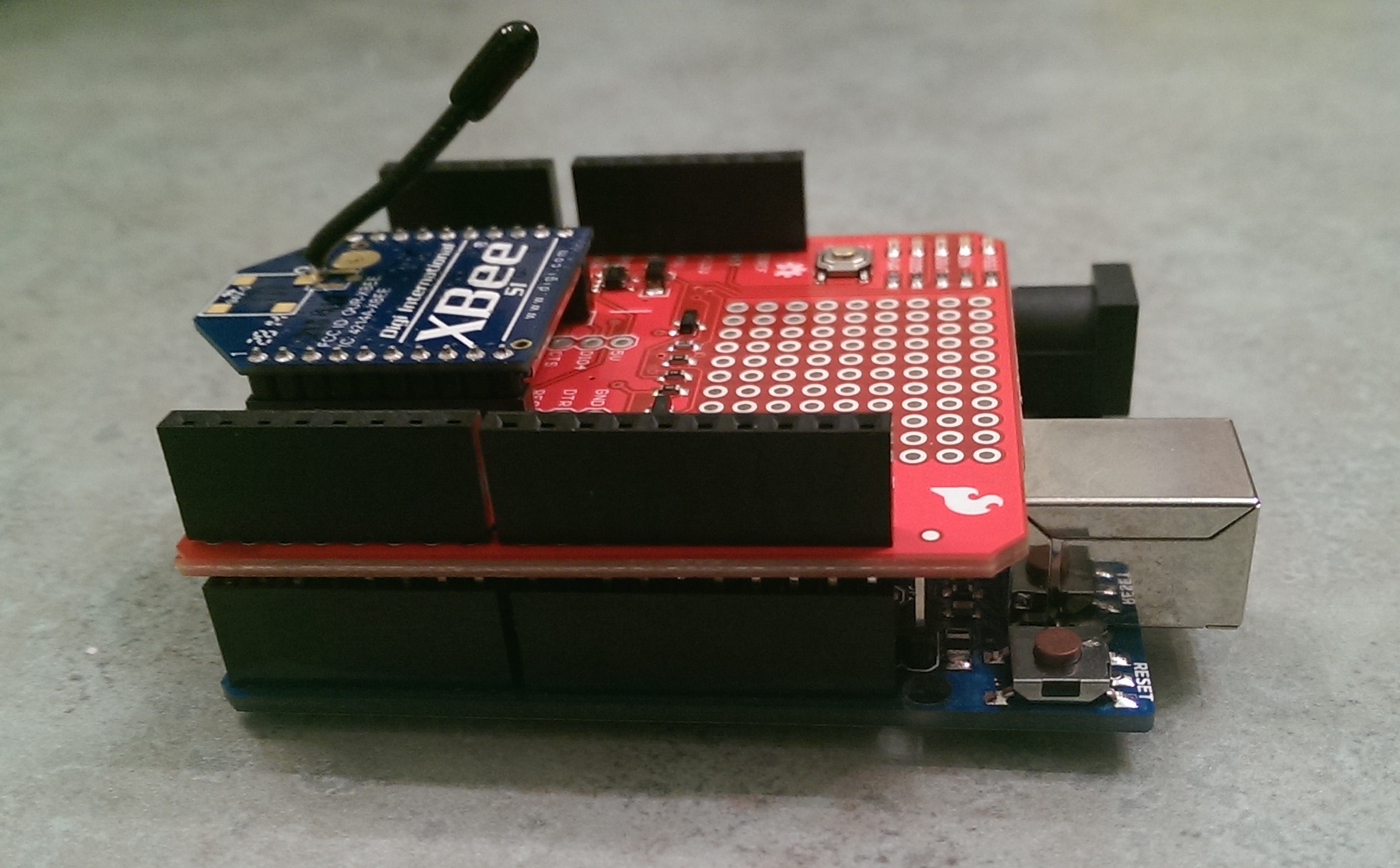
**Procedure #2 (Talking Xbees)**

In this second procedure, students will work in groups of two to test the range of the two named and married Xbees that were set up in Procedure #2.

1. Plug the Home Xbee and USB explorer into one USB port in the computer.
2. Insert the AWAY Xbee into the Xbee shield in the following orientation:



1. Insert the AWAY Xbee and Xbee shield assembly into the Arduino card as shown below:



1. Plug one end of the USB cable into the Arduino and one end into another USB port in the computer.
2. Open the Arduino program (IDE). Through the Tools drop-down menu, and serial port option, select the serial port to which the Arduino is connected.
3. Write a sketch that will allow the Arduino to allow the AWAY Xbee (now connected to the Arduino) to talk to the HOME Xbee. Comment your code (using the // to indicate comments) thoroughly so you can understand what each piece of code does. Remember to use void (setup) and the { } characters to bound the set-up parameters and the void (loop) and the { } characters to bound the continuously looping commands in this sketch.

|  |  |
| --- | --- |
| #include <SoftwareSerial.h> | // Accesses a file in the Arduino library that allows the Arduino to talk to the Xbee |
| SoftwareSerial XBee(2, 3); | Tells Arduino that the Xbee’s transmit (TX) pin is connected to pin 2 on the Arduino and the Xbee’s receive pin is connected to pin 3 on the Arduino. |
| XBee.begin(9600); | Sets the baud rate on the Xbee (this number must match the one configured in the Xbee setup in Procedure 1. |
| Serial.begin(9600); | Sets the baud rate on the serial port on the computer (this number must be identical to the Xbee baud rate) |
| if (Serial.available())  {  XBee.write(Serial.read());  } | Tells the Arduino that if serial data comes in from the serial monitor on the Arduino to send it back out through the AWAY Xbee. The AWAY Xbee will then transmit this data to the HOME Xbee that is connected to the USB explorer where it will then enter the computer.. |
| if (XBee.available())  {  Serial.write(XBee.read());  } | Tells the Arduino that if serial data comes in from the AWAY Xbee to send it out to the serial monitor. The data will then come in to the computer through the cable that connects the Arduino card to the computer. |

1. Verify the sketch you have written. Correct any errors that come up.
2. Upload the sketch to the Arduino.
3. Open XCTU and Connect to the HOME Xbee using a similar process as that described in Steps 4-7 in Procedure 1.
4. Open the serial monitor in Arduino.
5. Type anything in the blank at the top of the serial monitor and press Send. Whatever you typed should now appear in the rightmost screen of the XCTU Monitor screen (the window that lies to the right of the console log):
6. Type anything in the console log in XCTU. Whatever you typed should now appear in the Arduino serial monitor.

Congratulations! Your HOME and AWAY Xbees are officially talking to one another!

**Procedure #3 (Testing the Range of the Xbee Pair)**

In this third procedure, we will program the Arduino to read a simple voltage from the Xbee and then use that voltage to test the range of the Xbee Pair (How far apart can they be and still communicate successfully with one another?)

1. Plug the HOME Xbee and explorer assembly into the USB port of the computer.
2. Plug the AWAY Xbee and Arduino assembly into another USB port on the computer.
3. Connect the 3.3 V signal on the assembly in part 2 to the A0 signal on this same assembly.
4. Write a sketch that will allow the Arduino to send the signal at A0 to the HOME Xbee. Start out by sending the digital equivalent signal (the raw signal that Arduino will provide after reading what is connected to A0) to the HOME Xbee and then do a little bit of math to convert this digital signal to an analog one (equal to the actual voltage that the Arduino is reading from A0 which according to the connection you made in Step 3 should be approximately 3.3V). Comment your code (using the // to indicate comments) thoroughly so you can understand what each piece of code does. Remember to use void (setup) and the { } characters to bound the set-up parameters and the void (loop) and the { } characters to bound the continuously looping commands in this sketch. In addition to some of the commands used in Procedure 2, you will likely use some of the following commands:

|  |  |
| --- | --- |
| float <variablename>; | Creates a floating point number called variablename |
| int < variablename>; | Creates an integer called variable name. |
| Xbee.print (variablename); | Prints to the AWAY Xbee transmit pin the value of variablename |
| Xbee.print (“,”); | Prints to the AWAY Xbee transmit pin the text contained between the quotation marks. |
| delay(variablename); | Creates a delay or pause of length variablename in milliseconds. |

1. Verify and upload your sketch to the Arduino.
2. Open XCTU and Connect to the HOME Xbee using a similar process as that described in Steps 4-7 in Procedure 1.
3. In the monitor window, watch the incoming serial data. Is it what you expect? Is it the same as what appears in the Arduino serial monitor?
4. If the answers to Step 7 are yes, then disconnect the Arduino USB cable. Connect a 9V battery to the Arduino card using the barrel jack connector.
5. Verify that the XCTU monitor still reads the value of the 3.3V power supply on the Arduino and AWAY Xbee assembly.
6. Gradually increase the distance between the HOME and AWAY Xbees until the value read on XCTU is no longer valid.

WHAT IS THE RANGE OF YOUR XBEE PAIR?

**Procedure #4 (Open Design Time)**

In this last procedure, design a circuit that senses some parameter of interest (which you desire to know from a remote location). Breadboard the circuit and connect the desired sensor outputs to the Arduino Analog (A0, A1, etc) pins. Plug the Arduino (and AWAY Xbee) into the computer. Write a sketch to read the values of these Analog pins. Disconnect the Arduino (and AWAY Xbee) from the computer and plug in the 9V battery. Take your AWAY Xbee and Arduino Assembly somewhere to test your remote sensing capability.