Setting up the Xbee

EE 299, University of Washington

Spring 2014

By: Ryan McDaniel and Alyanna Castillo

Rev. 3
Introduction

This document will outline how to set up wireless communication between two Arduino Unos using the Xbee wireless card and the wireless shield. The main points are to put your Arduinos on a separate network from other Arduinos, and to communicate to the correct addresses on those networks. If this guide is not followed, it is very likely that your Arduinos will receive junk information from other groups during the lab, causing your project to break.

Process

What You Will Need

1. Two Arduino Unos
2. Two Xbee wireless cards (datasheet)
3. Two Arduino wireless shields
4. Two USB cables
5. The Arduino development environment software
6. Serial software for a PC, the lab computers have Hyperterminal, we recommend CoolTerm.

Warning

Disconnect all power from the Arduino before adding or removing the wireless shield. It is very easy to damage the communications systems of the Arduino if the shield is added or removed while power is on.

Steps

1. Insert a Wireless shield onto each Arduino that you plan to use with wireless communication (See Warning).
2. There is a small switch on the wireless shield that can switch between two options, USB and micro. Make sure that it is switched to USB.
3. If the Xbee card is not connected to the shield, place it onto the shield so that the slanted ends point away from the center of the Arduino.
4. Connect each Arduino that you plan to use with wireless communication to a computer so that they are on separate COM ports.
5. Open two separate Arduino IDE windows, and program the “Blink” example program onto each connected Arduino.
6. Open up Hyperterminal (a) or CoolTerm (b) and configure them to connect to the correct COM port (match what the Arduino IDE says), at 9600 baud, 8 data bits, no parity, and 1 stop bit. Turn on the “Local Echo" feature of the program as well.

a) Using Hyperterminal:

i) Go to the Start Menu and type “Hyperterm” into the search bar. Open up the program called Hyperterminal Private Edition.

![Connection Description](image1)

Figure 1 - Creating a connection in Hyperterminal.

ii) Click through any windows that pop up until you see the window in Figure 1. Give your connection any name and click OK to continue.

![Connect To](image2)

Figure 2 - Configuring the port for the connection.

iii) After naming the connection, Hyperterminal must be told what COM port to connect to. Select the correct COM port for the first Arduino you wish to program from the “Connect Using:” dropdown menu, as seen in Figure 2. You should choose the same port you used in the Arduino IDE to program the “Blink” example code onto your Arduino. Click OK to continue.
iv) Next, you must configure the connection. Figure 3 shows the correct settings for communicating with the Xbee wireless chip. Click Ok to continue to the next step.

Figure 3 - Configuring the connection.
v) After configuring the connection, you will be brought to a large white terminal window. Go to File->Properties, move to the “Settings” tab, and click on the “ASCII Setup” button on the lower right of the window. In the “ASCII Setup” window, check the “Echo typed characters locally” box. This makes it so that you can actually see what you type into the terminal window. This process is shown in Figure 4.

vi) Exit all of the configuration windows and return to the big white window. Lining the top of the window is the menu bar shown in Figure 5. The phone is “answered” in Figure 5, meaning that it is listening for information on the COM port. The phone can be clicked on to toggle between “answered” (listening for data) or “hung up” (not listening for data). Make sure that you have “answered” the phone before continuing.
b) Using CoolTerm

After installing and running CoolTerm, you will be greeted by the window shown in Figure 6. The first thing to do is click on the “Options” button, indicated by the cog and wrench icon.

In this menu, click the “Re-Scan Serial Ports” button to ensure that the Arduinos are recognized by CoolTerm. The menu is shown in Figure 7.
iii) Select the correct COM port for the first Arduino you wish to program from the “Port:” dropdown menu. You should choose the same port you used in the Arduino IDE to program the “Blink” example code onto your Arduino.

![CoolTerm's Terminal menu](image)

Figure 8 - CoolTerm's Terminal menu. Check the "Local Echo" box.

iv) Next, open the “Terminal” menu by clicking on the “Terminal” text in the left menu bar. Figure 8 shows the result. In this menu, check the “Local Echo” box. This makes it so that you can actually see what you type into the terminal window. Click OK to return to the main menu from Figure 6.

v) Click the “Connect” button on the main menu to establish a connection with the Arduino and Xbee.

7. After setting up either Hyperterminal or CoolTerm, you will be sitting at a white terminal window. Anything you type while this window is active will be sent instantly (no “enter” key required) to the Xbee through the Arduino.
<table>
<thead>
<tr>
<th>Command</th>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| +++     | None      | Enter into configuration mode for the Xbee  
|         |           | Each + must be entered at least 1 second apart |
| ATID    | Hex address  
|         | Ranges from 0000-FFFE | Sets the network that your Xbee is on.  
|         |           | Any Xbees you wish to communicate to must have the same network. |
| ATMY    | 32-bit Hex address. | Sets the address of the current Xbee on the network. |
| ATDH    | 16-bit Hex address. | Sets the 16 highest bits of the destination Xbee on the network. |
| ATDL    | 16-bit Hex address. | Sets the 16 lowest bits of the destination Xbee on the network. |
| ATWR    | None      |Writes any configuration changes to the Xbee. You must do this when you are finished for your changes to take effect. |

Table 1 - List of Xbee configuration settings

8. With your Terminal window active, type 3 +’s within one second with 1 second pauses before and after attempts. After all 3 are entered, the input will be confirmed with an OK. If you don’t get an OK, try entering the +’s with different timings. This will put the Xbee into configuration mode; however, it will only stay idle in configuration mode for 10 seconds. Make sure you enter any command to “reset” the 10 second timer or you will have to enter your +’s again.

9. Table 1 shows the various commands that can be entered while in configuration mode. The goal is to set your network ID with ATID, the current Xbee’s address on that network with ATMY, set ATDH to 0, and ATDL to the address of the Xbee you want to communicate with on the network.

Remember, if you take too long to enter commands, you will be kicked out of configuration mode and will have to enter +++ again.

a) First, set ATID to be any number from 0000 to FFFE in hex by typing ATID followed by the number and pressing enter. It will be confirmed with an OK. You can check the change by typing only ATID and pressing enter. The response will be the ID that is currently set. Take note of the ID that you set so that you can set it on the second Xbee later.

b) Next, set ATMY to be something between 00 and FF by using the same process as ATID. It can be checked with the same method. Take note of the address that you set so that you can set it as the ATMY on the second Xbee later.

c) Set ATDH to 0

d) Set ATDL to a second address similar to what you did for ATMY. It MUST be different from ATMY. This is what should be set as the ATMY for the second Xbee, so take note of it and enter it later.

e) Finally, type ATWR and press enter to write your changes to the Xbee so they will not be reset when you lose power. Double check by reestablishing the serial connection, entering the +++ command, and then typing ATID, ATMY, ATDH, and ATDL, to make sure they are what you set.
<table>
<thead>
<tr>
<th>Command</th>
<th>Value on Xbee 1</th>
<th>Value on Xbee 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATID</td>
<td>&lt;Same on both Xbees&gt;</td>
<td>&lt;Same on both Xbees&gt;</td>
</tr>
<tr>
<td>ATMY</td>
<td>&lt;Value 1&gt;</td>
<td>&lt;Value 2&gt;</td>
</tr>
<tr>
<td>ATDH</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ATDL</td>
<td>&lt;Value 2&gt;</td>
<td>&lt;Value 1&gt;</td>
</tr>
</tbody>
</table>

Table 2 - Relationship between settings on Xbee 1 and Xbee 2.

10. Repeat steps 6 – 9 for the second Arduino and Xbee, but when you reach step 9, set the configuration values to match what is outlined in Table 2.
11. Don’t forget to type ATWR and press enter to save your settings.
12. To communicate between two Arduinos, program your Arduino with the program you wish to run. After programming, switch the small switch mentioned in step 2 to Micro. Make sure that the Arduino on the receiving end also has its switch set to Micro. **Anything that would normally be transmitted to the Serial Monitor will now be transmitted wirelessly to the second Arduino.**
13. If communication is still not working, you might want to check some additional configuration settings, which can be found in the Xbee datasheet. Try checking the channel with ATCH, baud rate with ATBD, or even resetting the Xbee to default settings with ATRE followed by ATWR.

**Conclusion**

By following these instructions, you should be able to set up wireless communication between two Xbees and two Arduinos. The steps include setting up the hardware correctly first, configuring the Xbee wireless controllers with a serial terminal like CoolTerm or Hyperterminal, and finally flipping the switches on the shields to Micro and communicating. This tutorial is based on a similar one provided by the DXARTs department at this [link](#), and another tutorial at this [link](#). If you have any questions, please ask your TAs.