

Setting up the Xbee Radios for the RSeries-Open Control

The setup for the Xbee radio can seem a little difficult if you are not familiar with how the radios are configured. While working on the controller and receiver set, I kept having issues with the controller board talking to the radios. Paul had posted a profile that he and Michael had used with their Xbee's that I tried to just slap on mine without really looking at it. Unfortunately, our radios are of slightly different versions and the profile caused my Xbee to brick and become unresponsive. That being said, the profiles had provided the information required to setup the Xbee network.

Setting up the Xbee's can be done in 5 fairly easy steps.

1. Install X-CTU or Moltosenso Iron (Cross-Platform)
 - a. X-CTU is available from Digi here:
(<http://www.digi.com/support/productdetail?pid=3352>)
 - b. Moltosenso Iron is available here:
(<http://www.moltosenso.com/client/fe/browser.php?pc=/client/fe/download.php>)
2. Install FTDI drivers (if required)
 - a. Located on the same page as the X-CTU software under drivers.
3. Query the Xbee for the Modem Type, Firmware, and Serial Number
 - a. Most of this information is on a label at the bottom of the Xbee, but this will save it in the X-CTU application for later use.
4. Flash the Controller Board Xbee as Coordinator AP (settings below)
5. Flash the Receiver Board Xbee as Router AP (settings below)

I'm not going to write anything regarding how to install the X-CTU or the Moltosenso Iron products used to configure your Xbee's. Needless to say, install the software as is recommended for your type of computer.

When starting out, take both of the Xbee's and turn them pin side up. We need to get some information from the stickers on the bottom.



The first line starting with XBP is the model number of your XBee (Good to have).

The second line is not needed.

The Third Line contains the Source High Address (Needed)

The Fourth Line contains the Source Low Address (Needed)

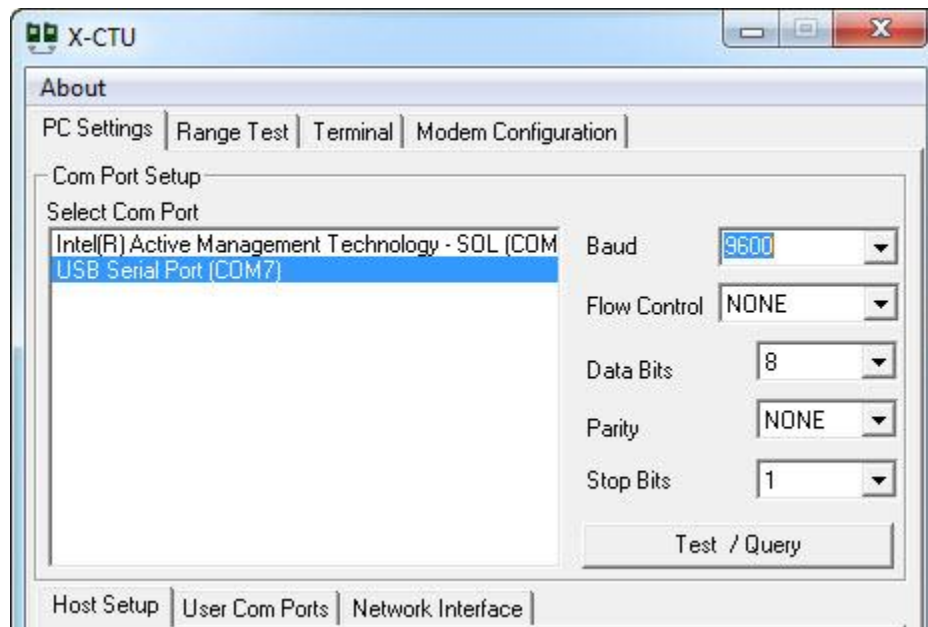
We really care about the third and fourth lines. Drop these values into a text file. We need to update both the Controller and Receiver Arduino scripts with this information.

I also labeled my Xbee's with a Sharpie so I would now know which was TX (transmitter) and RX (receiver) in case I removed them both from the boards for any reason. Say like I was doing documentation!

At this point, take the Xbee you intend to attach to your R-Series Controller Board and carefully insert it into the Xbee USB adapter. I'm using one from Adafruit seen to the right.



Start the X-CTU application, by default it puts an icon on the Windows desktop. It will identify ports on your system that it can use.

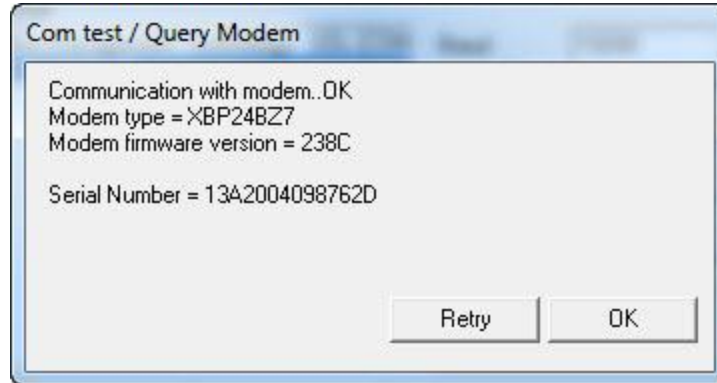


A fresh from the factory (Digi) Xbee will use the following settings:

- Baud: 9600
- Flow Control: NONE
- Data Bits: 8

- Parity: NONE
- Stop Bits: 1

Highlight the appropriate COM port (mine is 7 in the picture), and select the 'Test / Query' button.



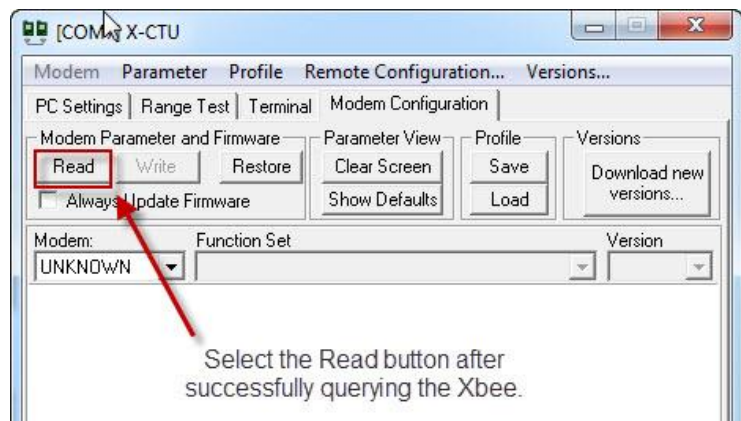
If you've done everything correctly you should receive a response similar to the one above. The modem type should be similar to the one you wrote down from the sticker. The modem type returned here is good to write down EXACTLY as it is. If you have a 'bricked' Xbee you will need this info to recover it. You can see that the serial number is a combination of the source high and source low addresses we got from the sticker.

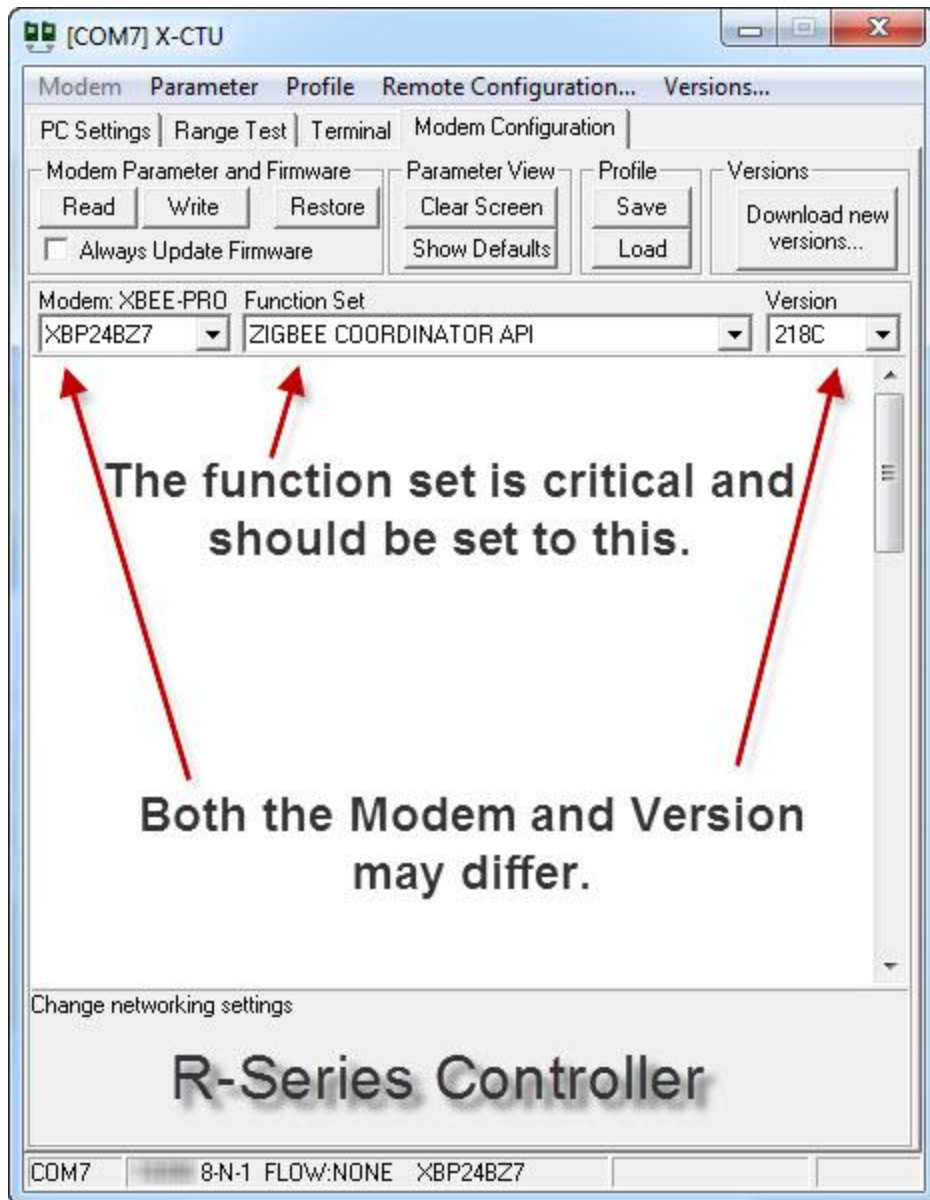
Now Select the Modem Configuration tab and then Select the 'Read' button.

This will read the installed firmware and configuration settings from the Xbee.

The default 'Function Set' installed on Xbee's from Digi or DigiKey is some version of the AT command set. We need to update the Function Set to an 'AP' version set. We also need to set this Xbee as a 'Coordinator'. Note: there should only be one 'Coordinator' in your Xbee network. I don't know if it makes a difference whether it is on the Rx side or the Tx side. I just chose Tx for this case.

Once you've 'READ' your Xbee, the Modem, Function Set, and Version should be populated. If you have an older version of Xbee, you might want to consider Selecting the Download new Versions button and getting the latest firmware for your modem. It may take awhile for it to download but it can make your Xbee more stable.

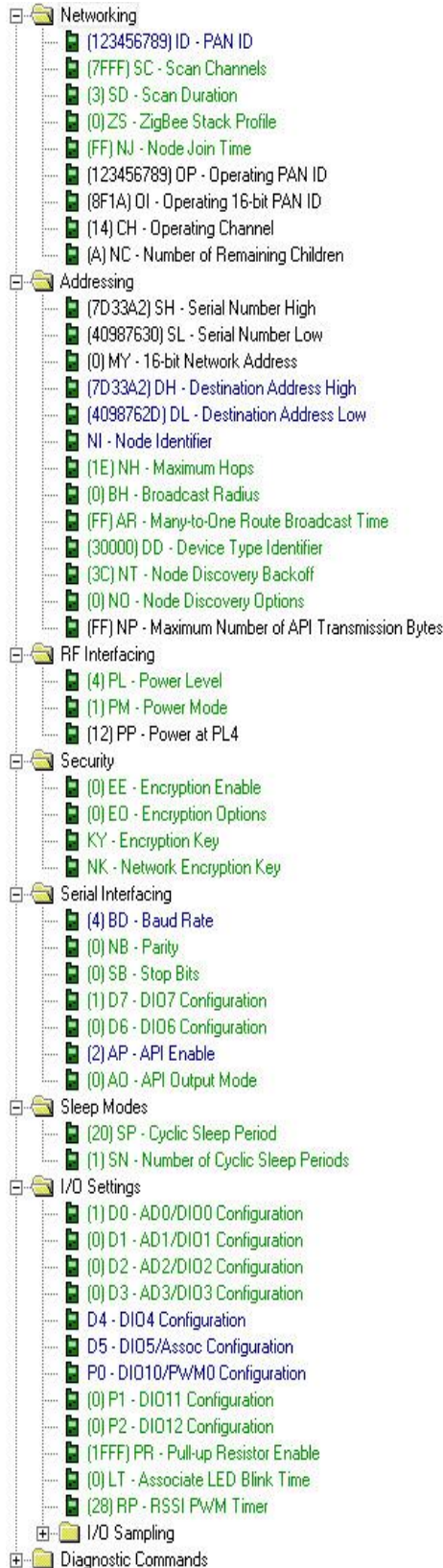




If you do decide to get the newest versions, you will want to select 'Always Update Firmware', change the 'Function Set' to 'ZIGBEE COORDINATOR API' and press the 'Write' button. Don't bother with trying to apply any additional settings as they will be wiped out by the new version.

It may take up to a minute for the Xbee's firmware to be fully updated. After the firmware finishes updating and the Xbee has reset. Press the 'Read' button again to get the new settings that are available with the firmware release. The software doesn't automatically update this so make doubly sure you do. You can brick your Xbee if you try to push the AT command set settings with AP command set installed. (Yes, I did this...)

R-Series Controller Settings



Okay, at this point you should have your Xbee running the ZIGBEE COORDINATOR API function set.

In the middle pane of X-CTU we need to start changing the settings. **UNCHECK the 'Always Update Firmware' checkbox at this time.**

The items that are in BLUE to the left are the ones that are different than the default for my firmware. They may change depending on your firmware. The table below is what you definitely need for this to work.

| Field | Description |
|-----------|---|
| ID | This is the PAN (personal area network) ID. It should be unique. So just insert something that you can remember. For this doc I used 123456789. You should make yours unique. |
| DH | Destination Address High; insert the third line value from the sticker of the Receiver here. Mine is 0013a200, the software converts this to 7D33A2. This probably isn't required BUT, by setting the DH and DL, you will ensure only your Xbee's talk to each other. |
| DL | Destination Address Low; insert the fourth line from the sticker of the Receiver here. |
| NI | Recommended; You can set a name for this Xbee, like COORDINATOR. |
| PL | Make sure this is set to 4-Highest |
| PM | Set to '1 - BOOST MODE ENABLED' |
| BD | Set to '4 - 19200' *This is why the Arduino would not talk to the radio |
| AP | Set to '2' API Enabled with Escaping |

The fields in the table above are what I would recommend to do at a minimum for the Coordinator Xbee. Once these are changed to the correct settings press the 'WRITE' button.

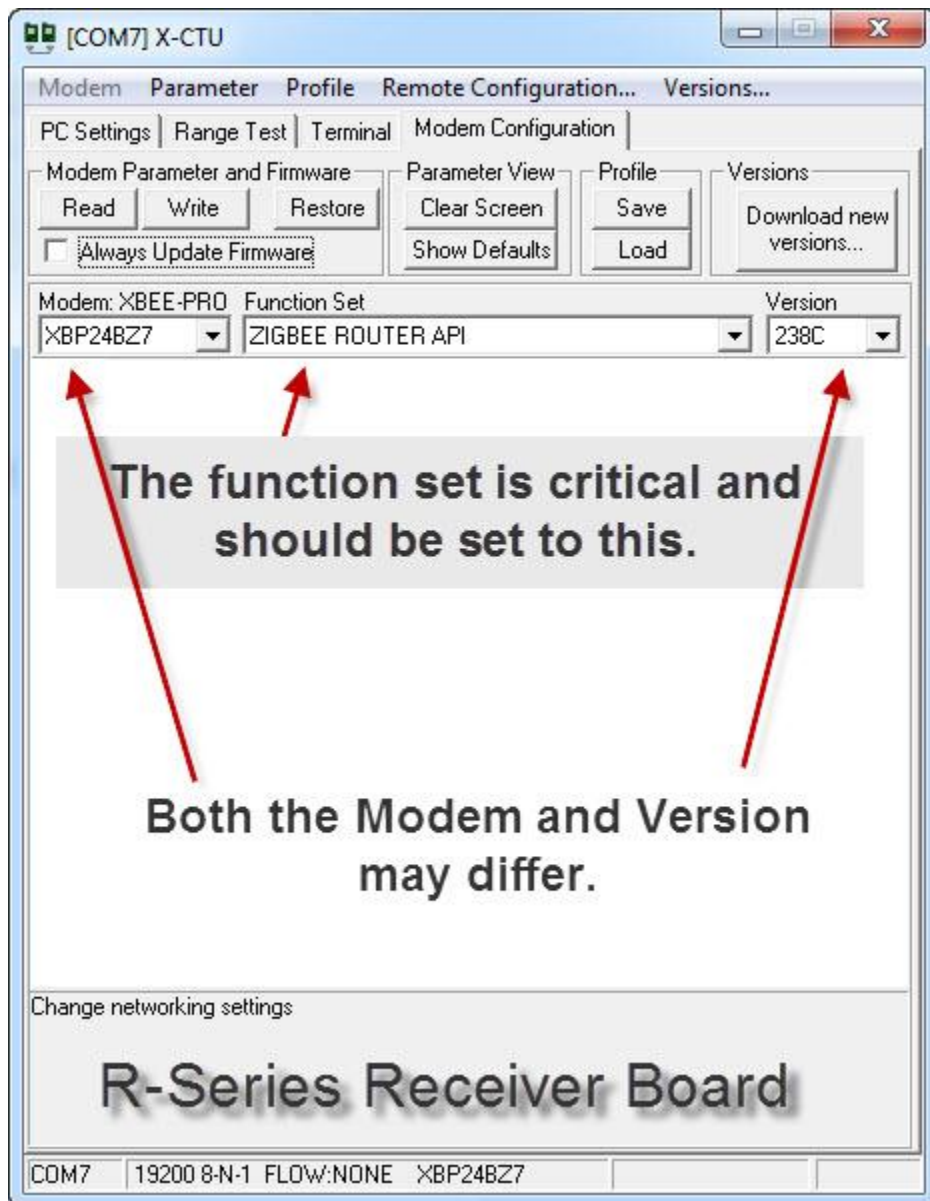
If you want to re-read this board, you will have to change the Baud rate to 19200 on the X-CTU software settings. I recommend doing this to verify that

the settings have actually been applied properly. Once you've satisfactorily verified the settings remove this Xbee and insert it into the R-Series Controller board.

Insert the Xbee destined for the Receiver Board. Go through the same steps as before, make sure you change the BAUD rate back to 9600. Query the Xbee, and download the latest firmware. It probably won't find anything new if you purchased these at the same time from the same distributor so this step is really optional.

Press the 'Read' button.

Now apply the following:

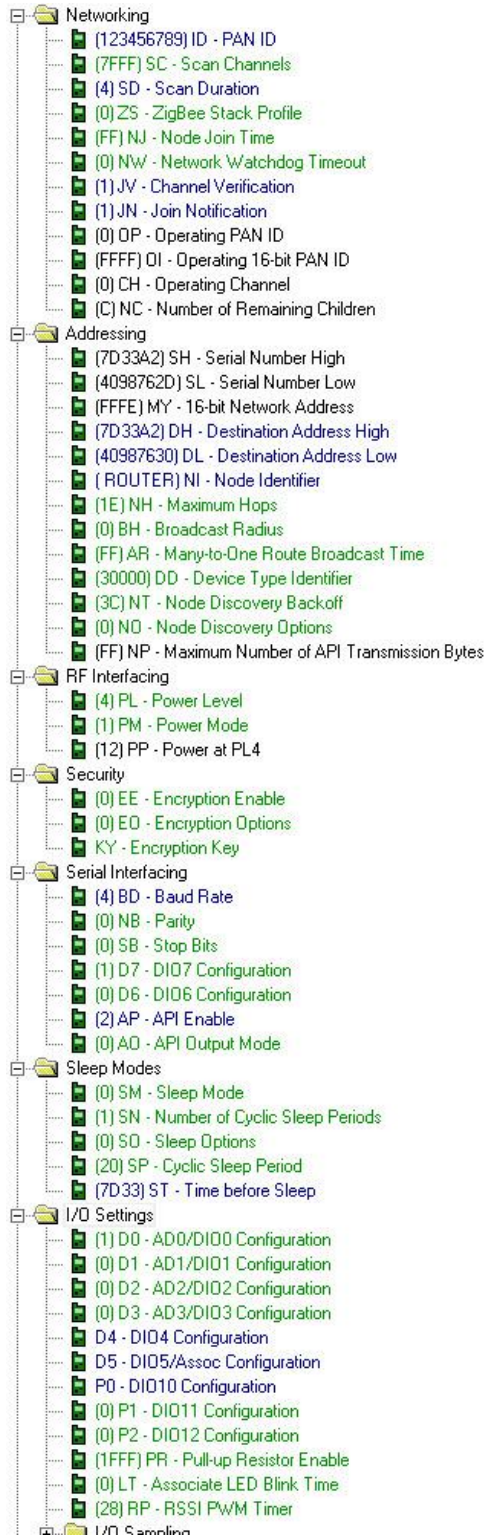


Change the Function Set to 'ZIGBEE ROUTER API', then check the 'Always Update Firmware'. Once the firmware had been updated, Uncheck the 'Always Update Firmware' before proceeding. Remember to 'Read' the new settings available with the firmware before proceeding.

Okay, at this point you should have your Xbee running the ZIGBEE ROUTER API function set.

In the middle pane of X-CTU we need to start changing the settings. **UNCHECK the 'Always Update Firmware' checkbox at this time.**

The items that are in BLUE to the left are the ones that are different than the default for my firmware. They will need to be changed on yours as well.



R-Series Receiver Settings

| Field | Description |
|-----------|---|
| ID | This is the PAN (personal area network) ID. It should be unique. So just insert something that you can remember. For this doc I used 123456789. You should make yours unique. |
| SD | Set to 4; default works, but 4 gives you some extra time. |
| DH | Destination Address High; insert the third line value from the sticker of the Receiver here. Mine is 0013a200, the software converts this to 7D33A2. This probably isn't required BUT, by setting the DH and DL, you will ensure only your Xbee's talk to each other. |
| DL | Destination Address Low; insert the fourth line from the sticker of the Receiver here. Mine is 40987630. *These are the numbers from the controller. |
| NI | Recommended; You can set a name for this Xbee, like ROUTER. |
| PL | Make sure this is set to 4-Highest |
| PM | Set to '1 - BOOST MODE ENABLED' |
| BD | Set to '4 - 19200' *This is why the Arduino would not talk to the radio |
| AP | Set to '2' API Enabled with Escaping |

The fields in the table above are what I would

recommend to do at a minimum for the Router Xbee. Once these are changed to the correct settings press the 'WRITE' button. Change your Baud Rate to 19200 in the X-CTU software and 'Read' the firmware to verify you have the appropriate settings. Some settings show up in the firmware but aren't really available, meaning they won't 'Stick' when you apply. I've tried to make sure that they were not included in the table above.

Carefully remove the Xbee from the programming socket and insert it into your R-Series Receiver board.

Changing the Arduino Sketch

In the Controller Sketch: (I'm using Controller v043):

- Use the search function for 'Address'

There should be a line that looks like this:

```
XBeeAddress64 addr64 = XBeeAddress64(0x0013a200, 0x4098762d); // Destination (Receiver) address
```

Change the highlighted addresses to the information we got off the sticker on the Receiver Xbee.

Save your work! Upload to your Controller Arduino.

In the Receiver Sketch (I'm using Receiver_v027):

- Use the search function for 'address'

There should be a line that looks like this:

```
XBeeAddress64 addr64 = XBeeAddress64(0x0013a200, 0x405e0b99); // Destination (Controller) address
```

Change the highlighted addresses to the information we got off the sticker on the Controller Xbee.

Save your work! Upload to your Receiver Arduino.

Now give your Arduino's some power and do the snoopy dance as you watch them all connect!