

Nov 14, 2012

Posted by micke in R2-D2 | 0 Comments

Configuring the XBee's

So: Just because my droid is finished, doesn't mean that I've stopped working on it!



Lately, I've been working on a new droid controller, that will be a lot more stealthy, give a lot more options than my current radio, and could, potentially, give me many many more options!

I've got a design in mind, but I'll need to build a prototype of it first before showing you guys!

Anyway, After soldering the controller together, I tried to start it up to see if it would boot and run my program. The program is made in the US by a couple of guys who came up with this new droid controller, and I've been fortunate enough that they graciously shared their hard work!

When starting the control unit up, I get the following picture:



So, it boots up, and then stops here:



This is because the XBee have not been configured at all. So, the transmitter and the arduino is not talking to each other. That is the primary issue. The secondary issue is that neither does my XBee's, so transmitter and receiver are not aware of each other. Doesn't matter how much I hold them facing each other and scream "THEY ARE RIGHT THERE!!! *GAAAAAAAAH*" They will still go "nope, no idea what you're talking about. There's no transmitter/receiver".

sigh

So, configuring these is relatively easy.

–If you have a windows machine, that is! Me, I'm on a glorious Mac! Makes some things harder, but most things way WAY better!

So, without further ado, I thought I'd give a brief tutorial on how to get the configuration of the XBee's work on a mac!

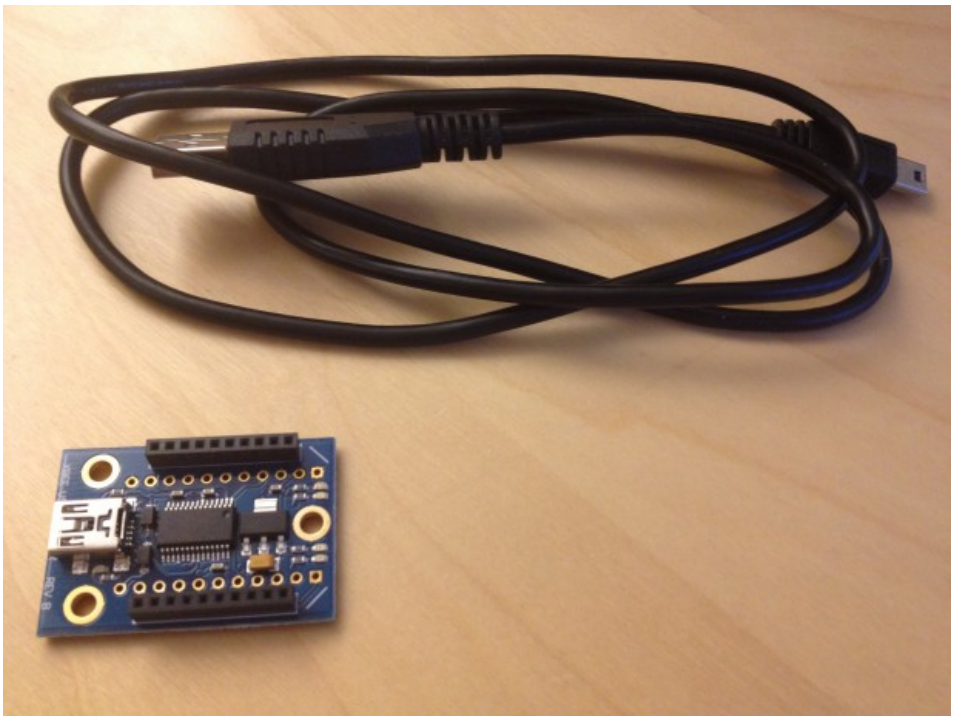


First off, I downloaded a free tool called moltosenso Network Manager Iron Best part? It's free! You can get the software here

In addition to your trusty XBee's,

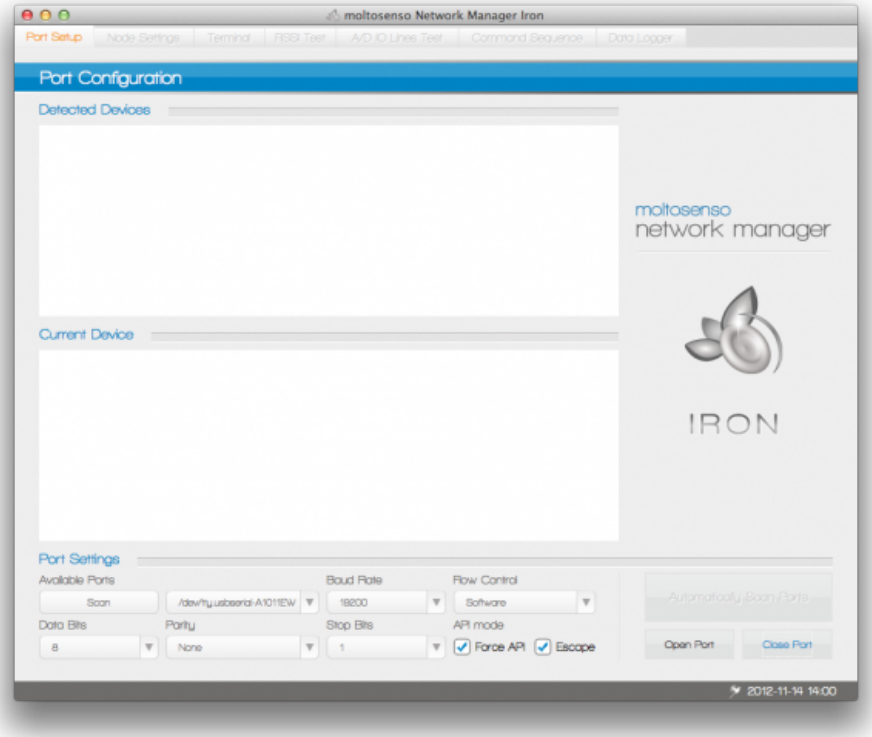


you'll also need a XBee USB Shield and a USB Cable. I mooched my XBee USB Shield of Sigge, since I was too lazy (=forgot) to buy one with me XBee's.



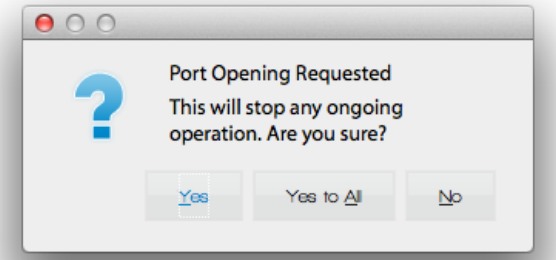
So, I started with my XBee that I marked as Transmitter. Very important to take note of the numbers on the backside, as they give which addresses the XBee are working within!

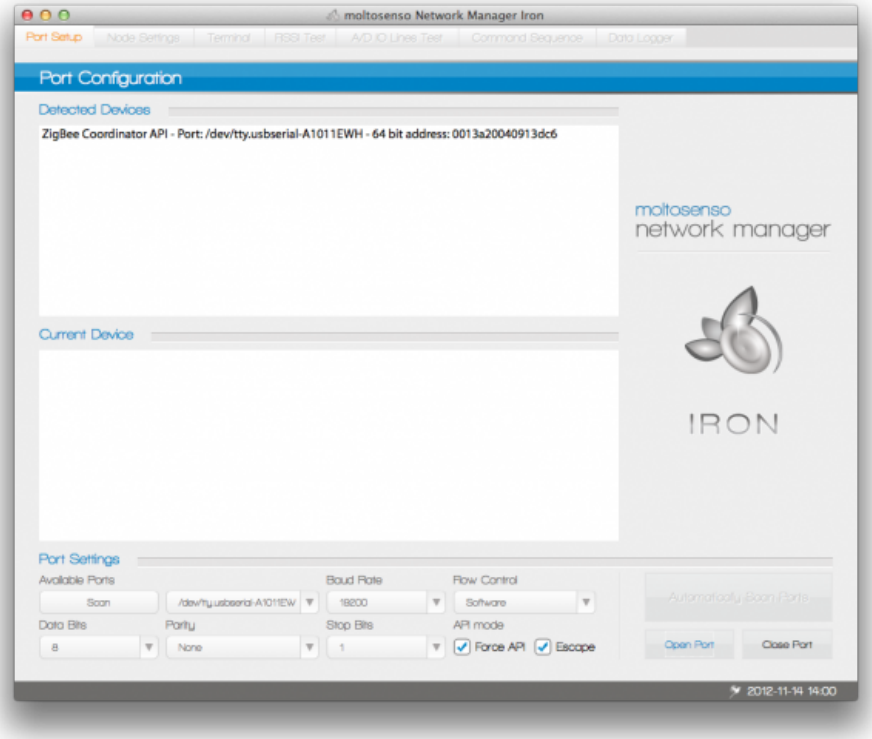
XBee goes into the USB Shield, and cable into the computer, then I fire up the software.



For a never configured XBee, make sure the Baud Rate is set to 9600, and click on “Open Port” in the lower right. Let the checkboxes “Force API” and “Escape” be unchecked for now.

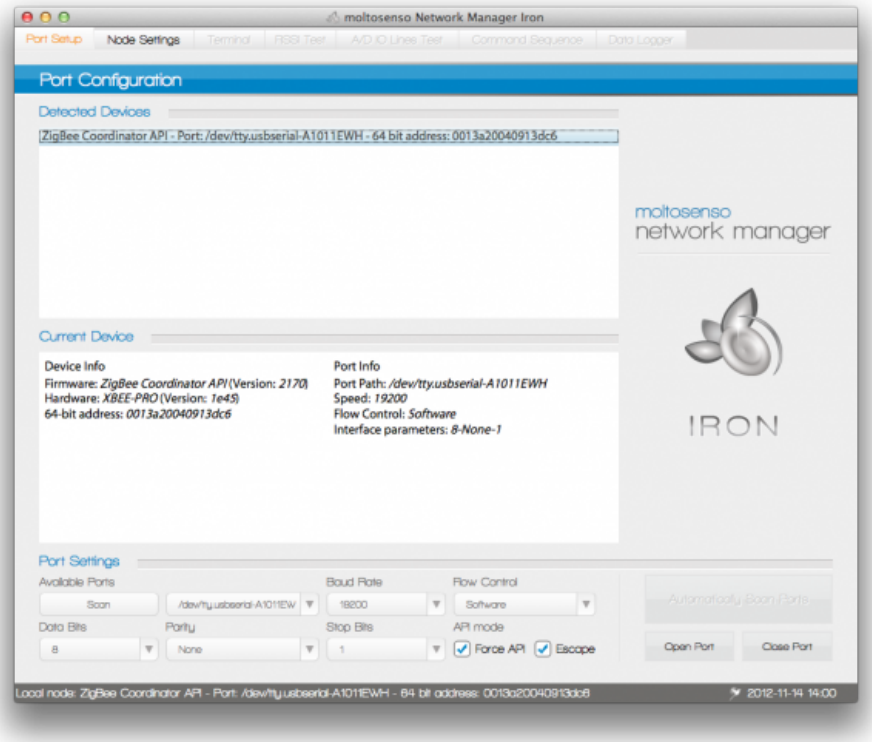
You will get this window a lot, by the way:





Look, it found my XBee!

I mark the XBee in the list, and I should get some response in the lower white area like this.

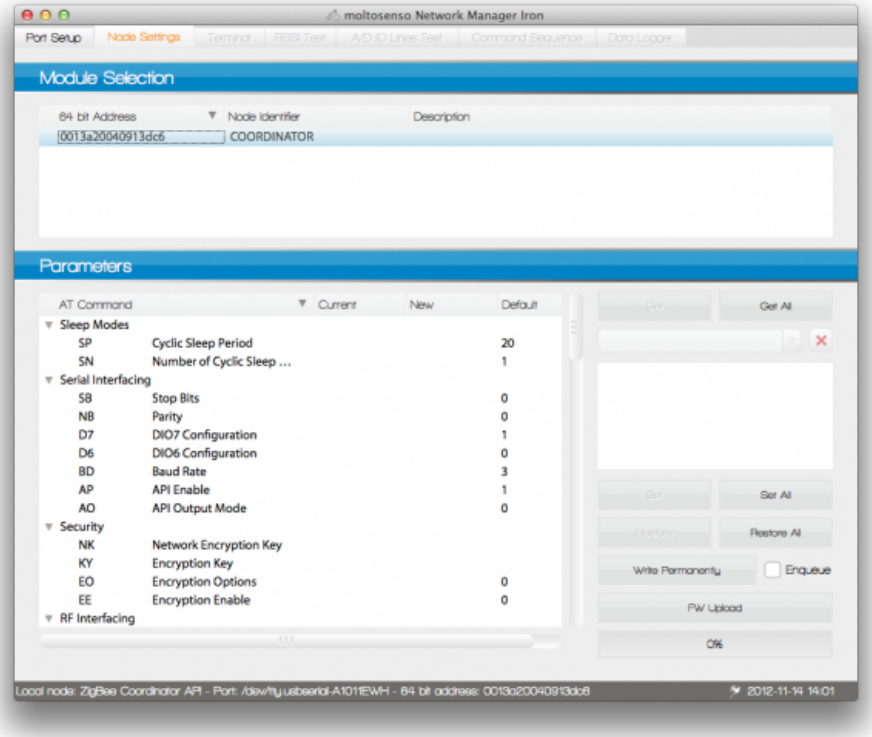


So this tells me some information about my XBee.

Please note that on my pictures, the XBee is already configured, so it will differ from the info you get. Most importantly is that you get a response from your XBee.

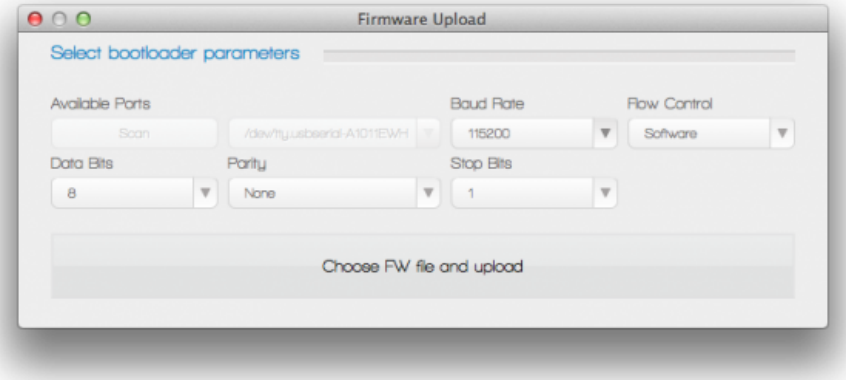
Next step: Click on “Node Settings” in the window menu.

Click on the XBee that shows up at the top, and then the button “Get All” and it should look something like this:



These are all the lovely settings you can change in the XBee config, but WAIT! Before you start changing them, you need to flash the XBee cards with the proper firmware, as the firmware is not only an update, it is ALSO programming the ROLE of what this XBee will do!

Click on the button FW Upload.



Leave the settings as is, and click on “Choose FW file and upload”.

Now this is where it got tricky! Which file? from where?

The XBee official site only lets X-CTU download the firmware from within the client!

Fortunately, moltosenso has thought about this, and have actually enclosed the firmware on disk in the installation!

When you click on “Choose FW file and upload” navigate to the “ebl_files” folder, which should be directly visible from the folder you are in.

Since the ROLE of the card is programmed in with the firmware, there are different firmware files that should be used for the transmitter and the receiver!

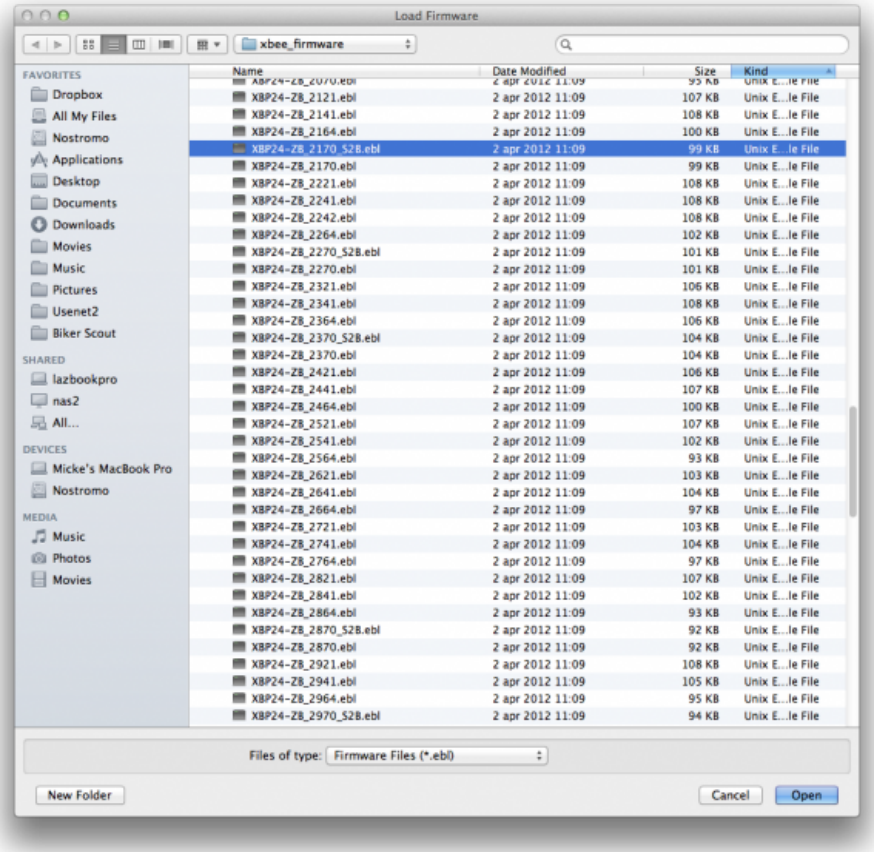
This is taken from the XBee Firmware release notes:

> XBee version numbers will have 4 significant digits. A version number is reported by issuing an ATVR command. The response returns 3 or 4 numbers. All numbers are hexadecimal and can have a range from 0-0xF. A version is reported as “ABCD”. Digits ABC are the main release number and D is the revision number from the main release. “B” is a variant designator. The following variants exist in ZB firmware:
>
> - “0″ – Coordinator, AT Command Mode (AP=0)
> - “1″ – Coordinator, API Mode (AP=1,2)
> - “2″ – Router AT Command Mode (AP=0)
> - “3″ – Router API Mode (AP=1,2)

- “8” – End Device, AT Command Mode (AP=0)
- “9” – End Device, API Mode (AP=1,2)

So, in clear text, this means that whatever the latest firmware is, the SECOND digit HAS to be a “1″ for the Coordinator XBee.

Searching in the folder, for Firmware for XBee Pro gives this one:



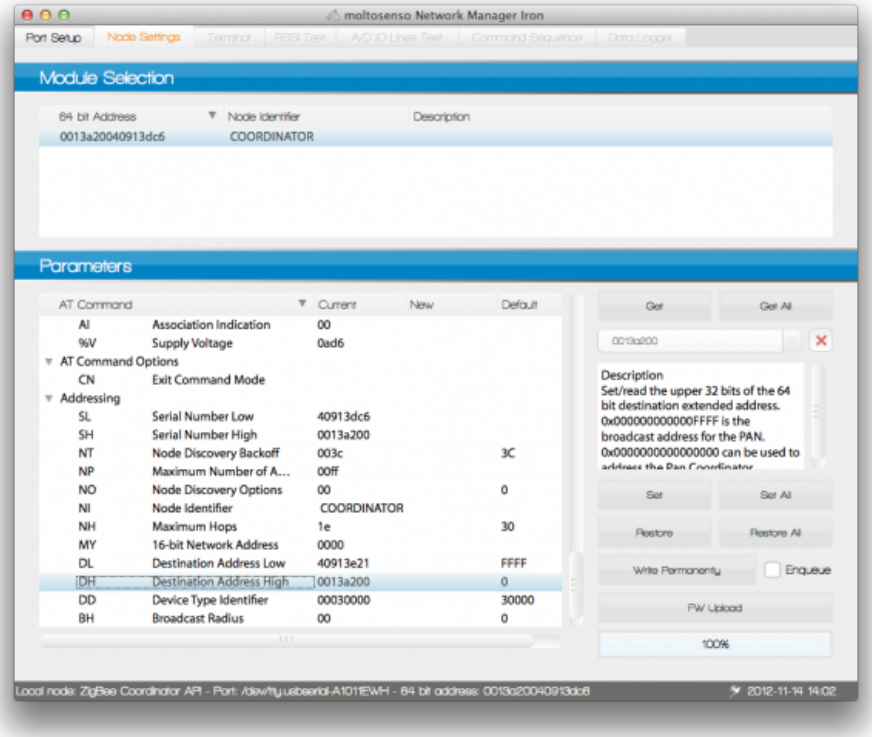
XBP24-ZB-2170_S2B.ebl – sounds great, let’s use that one!

So, the Firmware is uploaded into the XBee, and then it is time to change the information.

Select the card and Click “Get All” to get the configuration data.

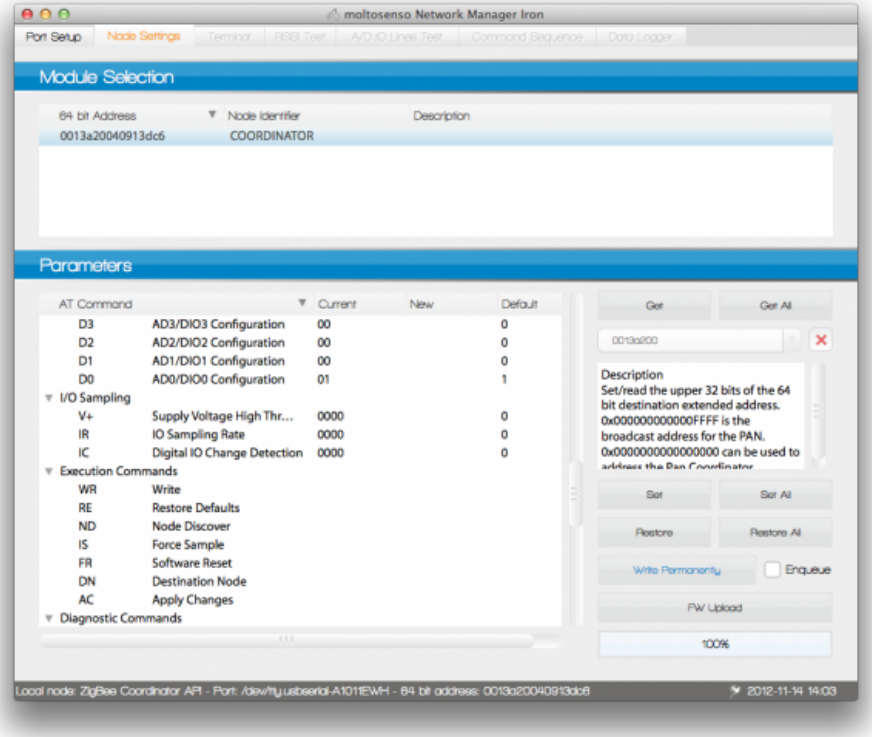
There are really a few things that should be changed on them to get this to work:

- ID – Your Personal Area Network. This is something that should be unique to you. Remember this as it should be set on both XBee’s. Don’t use mine! =)
- DH – Destination High Address, which is the THIRD Line listed on the back of your OTHER XBee. You DID write these down in advance, didn’t you? Note that it is the 3rd line of the RECEIVER XBee you should put here, since you are telling the CONTROLLER which addresses it can talk on.
- DL – Destination Low Address, which is the FOURTH Line listed on the back of your OTHER XBee. You DID write these down in advance, didn’t you? Note that it is the 4th line of the RECEIVER XBee you should put here, since you are telling the CONTROLLER which addresses it can talk on.
- NI – Recommended. Set this to “COORDINATOR”. This is just a Name for your XBee. It will show up when you connect it, and make future configuration easier!
- PL – Power Level. This should be set to 4.
- PM – Power Mode. This should be set to ’1 – BOOST MODE ENABLED’.
- BD – Baud Rate. Set this to ’4 – 19200′. This enables the Arduino to talk to the XBee! *weeeee*
- AP – This should be set to ’2′, which is API with Escaping. However, I’ve find that you can’t set this to two at this moment. We’ll get back to this in a moment. Just leave it at ’1′ for now.



Make sure you click on “Set” after each time you changed a value!

Now, when the values are changed, it is REALLY Important to write these permanently to the XBee. Click on the “Write Permanently” box. I missed this the first time I did this…



Since you have written the baud rate, you will lose connection with the card, even though the software hasn’t realized it. Click on the “Port Setup” tab and change the baud rate at the bottom to ’19200′.

Check the “Force API” and “Escape” checkboxes, then click on ‘Open Port’ again.

You are back in to the card, and it should show some different info than before. Again, select the card and go into “Node Settings” tab. Select the card and Click “Get All” to get the configuration data.

You can now go back and change the AP value:

- AP – This should be set to '2', which you can do now. Save this

Don't forget to "Write Permanently" on to the card, and you are all set!

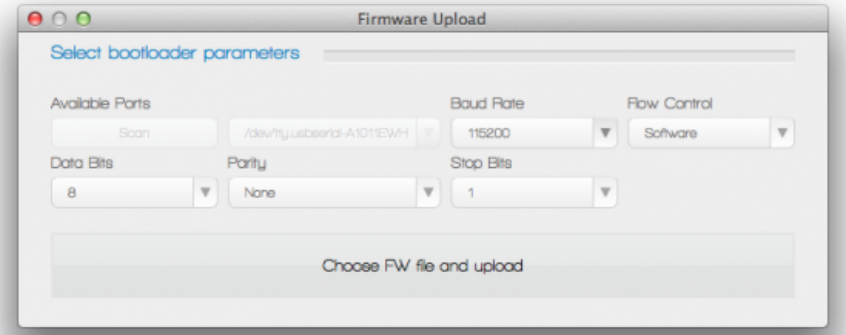
Next, click on the "Port Setup" tab and click on "Close Port". Disconnect the USB cable from the XBee USB Shield and remove the card. This is now DONE! Good Work!

Time to change the Receiver Card!

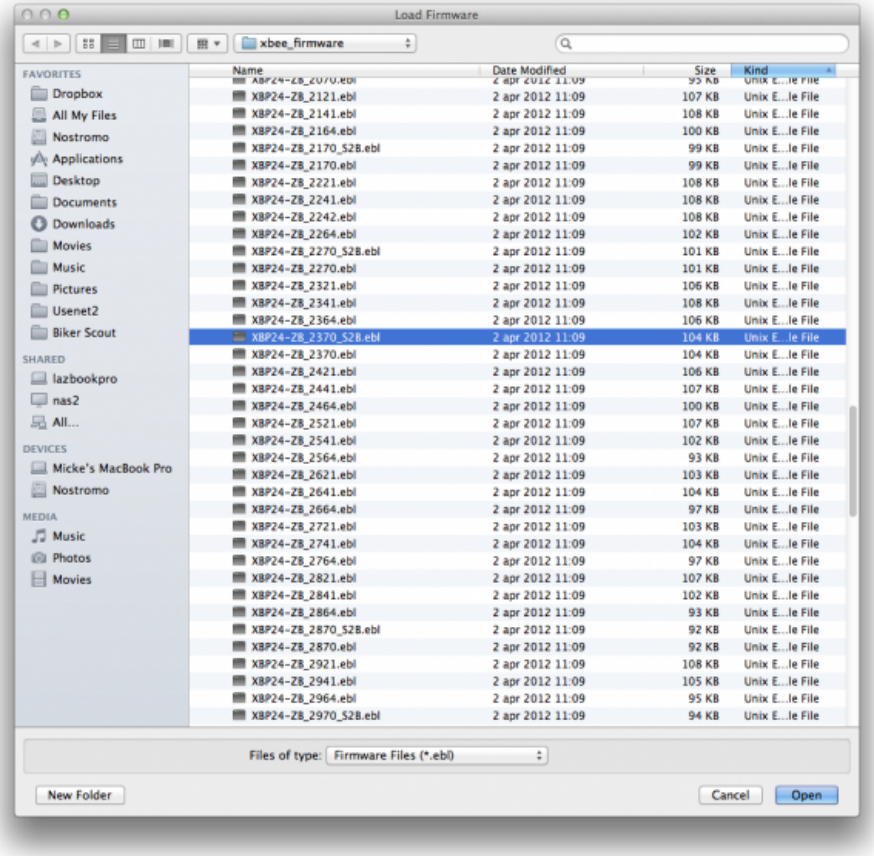
Same procedure as last time, almost.

You need to flash the firmware of the receiver as well, and by reading the little list from the Release Notes earlier, you realize that this Firmware should have a 3 as the second digit in order to be the role of 'Router Api', which is what we want!

Click on the button FW Upload.



Searching in the folder, for Firmware for XBee Pro gives this one:



XBP24-ZB-2370_S2B.ebl – Same version, but a 3 instead of a one. Just the one I want!

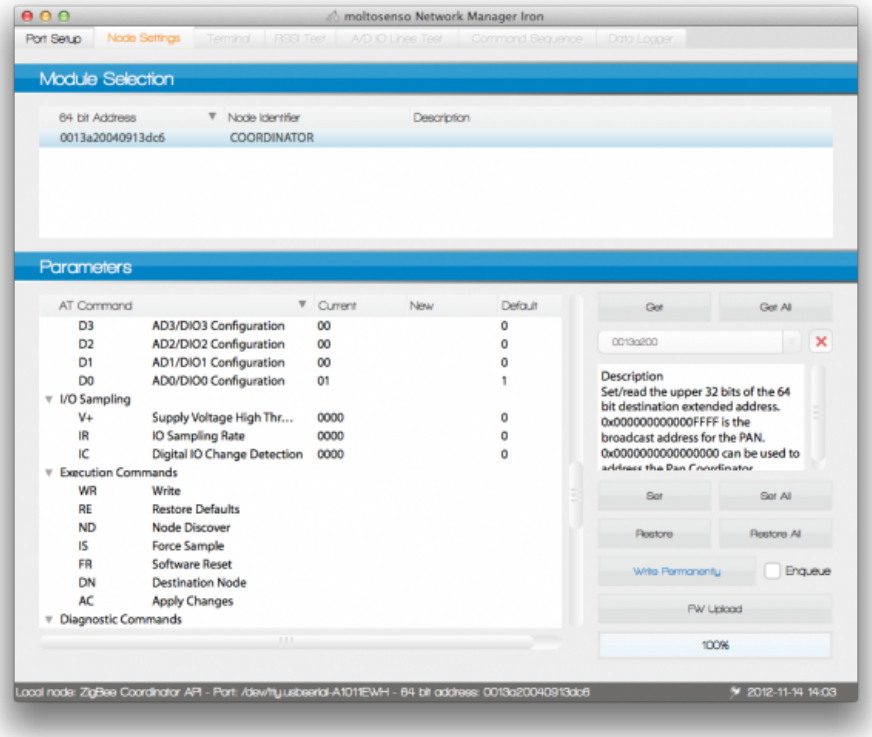
Upload this, then start changing the values as before.

Select the card and Click “Get All” to get the configuration data.

- ID – Your Personal Area Network. This should be the same as you put on your other XBee! Otherwise they can’t talk to each other!
- SD – Scan Duration. Set this to 4. 3 should be enough, but it will allow a longer scan for connections.
- DH – Destination High Address, which is the THIRD Line listed on the back of your OTHER XBee. You DID write these down in advance, didn’t you? Note that it is the 3rd line of the CONTROLLER XBee you should put here, since you are telling the RECEIVER which addresses it can talk on.
- DL – Destination Low Address, which is the FOURTH Line listed on the back of your OTHER XBee. You DID write these down in advance, didn’t you? Note that it is the 4th line of the CONTROLLER XBee you should put here, since you are telling the RECEIVER which addresses it can talk on.
- NI – Recommended. Set this to “ROUTER”. This is just a Name for your XBee. It will show up when you connect it, and make future configuration easier!
- PL – Power Level. This should be set to 4.
- PM – Power Mode. This should be set to ’1 – BOOST MODE ENABLED’.
- BD – Baud Rate. Set this to ’4 – 19200′. This enables the Arduino to talk to the XBee! *weeeee*
- AP – This should be set to ’2′, which is API with Escaping. However, I’ve find that you can’t set this to two at this moment. We’ll get back to this in a moment. Just leave it at ’1′ for now.
- JV – Channel Verification. Set this to 1. It will verify the channel to the controller board.
- JN – Join Notification. Set this to 1 as well, so your receiver board actually tells the controller board that they are paired!

Make sure you click on “Set” after each time you changed a value!

Now, when the values are changed, it is REALLY Important to write these permanently to the XBee. Click on the “Write Permanently” box.



Since you have written the baud rate, you will lose connection with the card, even though the software hasn’t realized it. Click on the “Port Setup” tab and change the baud rate at the bottom to ’19200′.

Check the “Force API” and “Escape” checkboxes, then click on ‘Open Port’ again.

You are back in to the card, and it should show some different info than before. Again, select the card and go into “Node Settings” tab. Select the card and Click “Get All” to get the configuration data.

You can now go back and change the AP value:

- AP – This should be set to ’2′, which you can do now. Save this

Don’t forget to “Write Permanently” on to the card, and you are all set!

Next, click on the “Port Setup” tab and click on “Close Port”. Disconnect the USB cable from the XBee USB Shield and remove the card. This is

now DONE! Good Work!

So, in theory, if you've followed this, both your XBee's should now be complete and configured!

Last step is to change the Arduino Sketches. You need to change both of them.

Like in the XBee's, you need to add the correct address for the receiver XBee in the Arduino controller sketch, and the correct address for the controller XBee in the Arduino receiver sketch.

Search for "Address" in the Arduino sketch.

The line should look something like this

From my RSeries_Controller sketch:

XBeeAddress64 addr64 = XBeeAddress64(0x**0013a200**, 0x**40913e21**); // Destination (Receiver) address

(Note the values are the same values I have on my XBee marked as my receiver in the picture up top. The value will be same, but start with 0x before the actual value. This needs to be there, by the way.)

From my RSeries_Controller sketch:

XBeeAddress64 addr64 = XBeeAddress64(0x**0013a200**, 0x**40913dc6**); // Destination (Controller) address

(Note the values are the same values I have on my XBee marked as my transmitter(controller) in the picture up top. The value will be same, but start with 0x before the actual value. This still needs to be there.)

Upload these into your arduinos, and you should be done! *yay*

A BIG THANK YOU to **Steele** who wrote the how-to for Windows / X-CTU. It was his guide I followed and used as basis for my own setup. I just didn't want to pollute my Mac, and I'm sure others would like to know how to get this done natively on a mac/Linux system as well.

Also a BIG THANK YOU to Ted (buhatkj) for finding the Join channel values to change!

Like 2 Tweet 1 0 2

Comments

0 comments

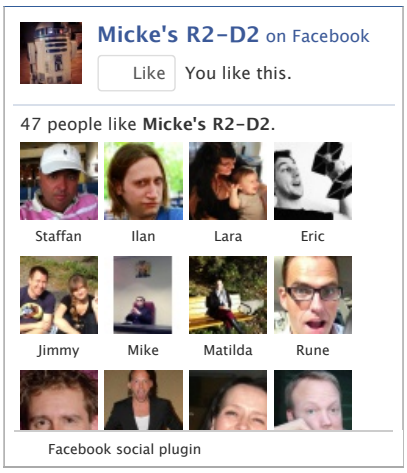

Add a comment...
Post to Facebook
Posting as Micke Askernäs (Change)
Comment
Warning: http://r2d2.askernas.com/?p=1345 is unreachable.
Facebook social plugin

Powered by Facebook Comments

Edit this page

Comments are closed.

R2-D2 on Facebook



Recent Posts

- Configuring the XBee's
- The final stretch..
- Ok, then, just a few pictures..
- R2-D2 Complete!
- I'm so close to the finish, I can smell it!

R2-2DO

R2-2DO:

========
Shoulders:

- Wet sand and final paint second horseshoe

R2-D2 Dome:

- Sand painted (primed blue) panels
- Paint painted panels in proper colour!

Body:

- mount utility arms properly

Center Foot:

- Cut out wooden thick shim for center foot, to raise droid slightly.
- Attach wooden thick shim to frame (under skirt)

Main Feet:

- paint right foot
- Touch up right battery box*

Legs:

- Insert metal tube in pivot point/mount on ankle*
- Add steel mount for magnetized shoulder hubs
- Possibly sand holder for new, alu shoulder hubs for perfect fit*
- Fix/remake Ankle Cylinders*
- Paint Ankle Cylinders*
- Paint Ankle Wedges*
- Wet sand and final paint*

(* partially done)

Future Work:

- Cut inner dome pie panels for Life Support Scanner and Periscope
- Work on hinging doors on styrene frame*
- Attach styrene pieces on inside foot shells for KHF's to screw in to better.*
- Add magnet holders to foot plates in order to stick them more firm onto the foot*

Secondary Dome:

- Fix faulty FLD.
- Cut holes for FLD, RLD
- Mount HP's
- Mount PSI's
- Mount FLD&RLD
- Mask for blue panels
- prime and paint blue panels
- Fix unique sound scheme for secondary identity!

Like 2 Tweet 1 0 2

Comments

0 comments


Add a comment...
Post to Facebook
Posting as Micke Askernäs (Change)
Comment

Warning: http://r2d2.askernas.com/?p=1345 is unreachable.

Facebook social plugin

Powered by Facebook Comments

Tags

aluminium construction dome electronics feet finish frame horseshoes legs Paintjob public appearances r2controller resin skins skirt status update styrene weathering

Search

Archives

- November 2012
- October 2012
- September 2012
- August 2012
- June 2012
- April 2012
- March 2012
- February 2012
- January 2012
- December 2011
- October 2011

Meta

- Site Admin
- Log out
- Entries RSS
- Comments RSS
- WordPress.org

5