|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | Gadgeteer Community |  |  |
|  | XBeeClient | PM:  Dev/test: | TBD  TBD, Paul Mineau |
|  | Getting Started Guide |  |  |

# Overview

Wondering what XBees to purchase? Section 2.1 will list the required XBees and dongle, and 2.2 will list the required software.

Configuring your XBees can be frustrating; section 3.1 will show you how to configure your XBees, and section 2.2 shows how to install the XBeeClient software and source code.

Get started sending and receiving messages in minutes (3.1), as well as using the interactive PC console application for issuing commands (3.2), running tests (3.3), and saving diagnostic information (3.4). 3.5 shows building a new Gadgeteer project.

Section 4 will show advanced operations like remote configuration.

Section 5 gives resources and instructions on how to learn more.

# Prerequisites

## Shopping List

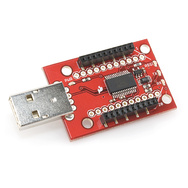
* The book *Building Wireless Sensor Networks* by Robert Faludi, O’Reilly Press (optional but very highly recommended)



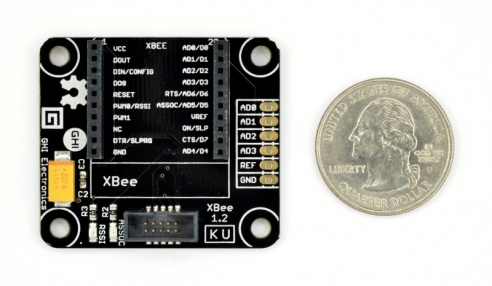
* At least 2 XBee Series 2 devices. XB24-Z7WIT-004 is the model tested with XBeeClient and recommended in *Building Wireless Sensor Networks.* 
  + <http://search.digikey.com/scripts/DkSearch/dksus.dll?vendor=0&keywords=XB24-Z7WIT-004>
  + <http://www.sparkfun.com/products/10414>



* One XBee Dongle necessary for connecting XBees to your PC for configuration as well as for running programs on your computer that communicate with XBees on your Gadgeteer creations.
  + <http://www.sparkfun.com/products/9819>



* At least 1 XBee Adapter Module from GHI Electronics for adding XBee devices to your Gadgeteer creations.
  + <http://www.ghielectronics.com/catalog/product/314>



* A Gadgeteer Mainboard. XBeeClient has been tested with the Spider mainboard from GHI Electronics, but should work with other mainboards such as the Hydra. It will not work yet with the Cerberus mainboard from GHI Electronics because XBeeClient is targeted to the .Net Micro Framework version 4.1

## Required Software

* Install all the software listed on GHI’s support page here <http://www.tinyclr.com/support>
* XCTU from Digi
  + <http://www.digi.com/support/productdetail?pid=3352&osvid=57&type=utilities>
    - Download the XCTU 32-bit

## Configuring XBees

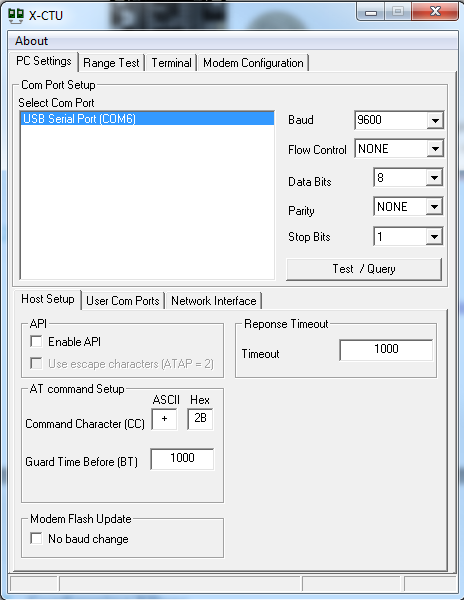
Configuring your XBees is the most difficult part. You may get stuck. This guide will do its best but may be incomplete, or you may run into issues. Use the forum on XBeeClient.Codeplex.com and use the Issue Tracker. If you have an issue that isn’t covered here, then its an issue with this document! Or its an issue with the XBeeClient test code, as it should have told you exactly what configuration issue you have.

It may help to read some blog posts. Here is a list of useful posts.

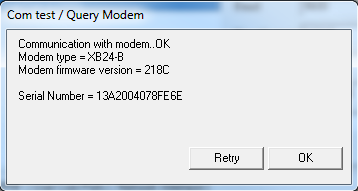
<http://mikedodaro.net/2012/04/27/xbee-radios-in-net-gadgeteer-devices/>

### Using XCTU

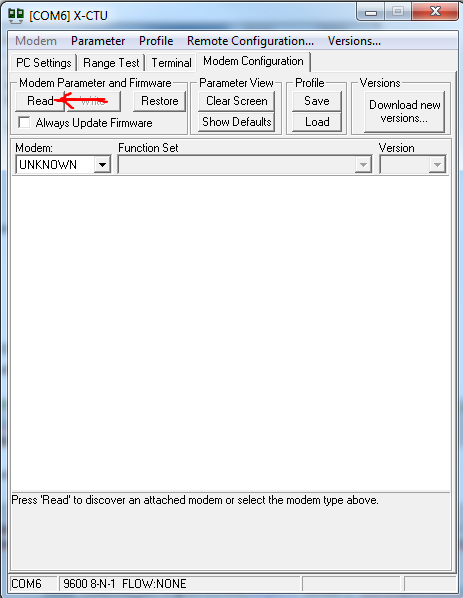
1. Plug your XBee module into the XBee dongle, and plug the dongle into your computer. Launch XCTU.



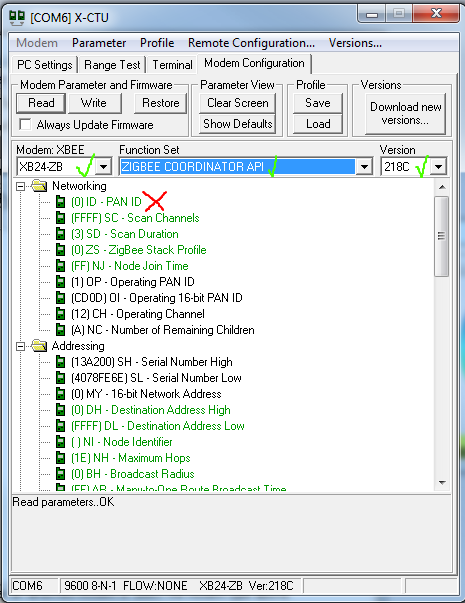
1. Click “Test/Query” to make sure XCTU can communicate with your PC.



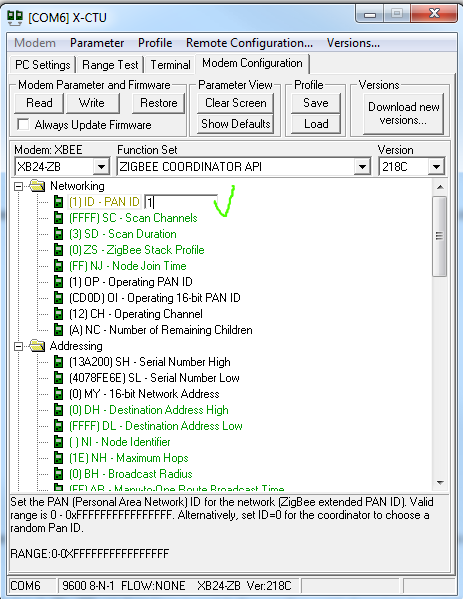
1. Go to “Modem Configuration” tab. Click on “Read”.



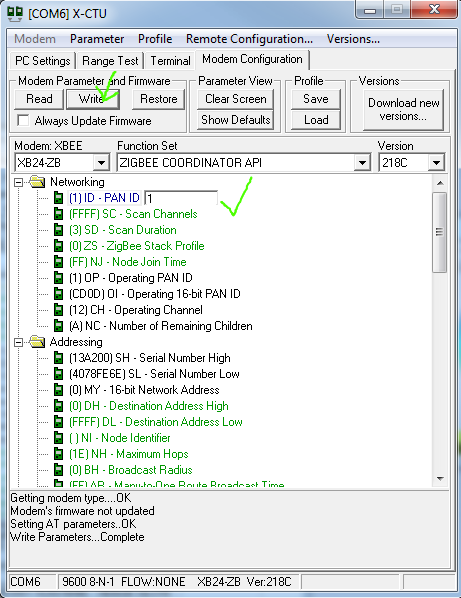
1. Update the Firmware. NOTE: This getting started guide is very incomplete. This section should describe the end goal and how to get there. The Goal is that you have 2 XBees, one Coordinator, and one Router. The coordinator should be firmware version 218C, and the Router should be 238C. The “Download new versions” button did not work for me, I had to hack my way around Digi’s site to find the firmware files, then download them to the XCTU folder C:\Program Files (x86)\Digi\XCTU\update\xbee\_zb, and I can’t recall the exact steps but do recall it was a pain that took hours. This document, once complete, will make this simple and fast. We may want to include the firmware with the XBeeClient project.
2. Once you have updated the firmware, you will see this.



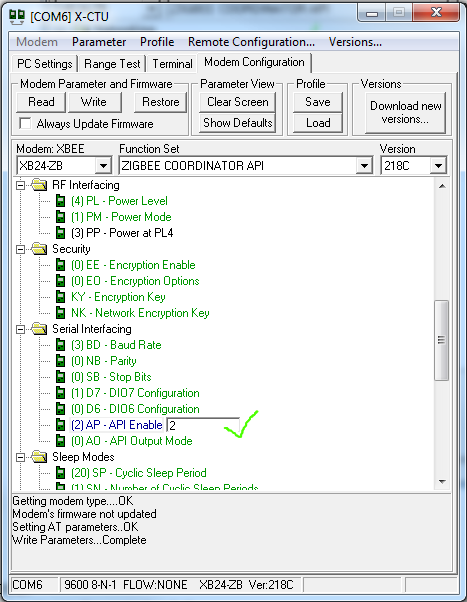
It is correctly configured as an XB24-ZB, with ZigBee Coordinator API, and version 218C, but the PAN ID is 0, and this is incorrect. Here’s why: the PAN ID stands for Personal Area Network. This is how your XBee network is separated from your neighbors. If the PAN ID is 0, and the device is a Coordinator, it will choose a random PAN ID. If it is a router, it will respect the PAN ID of 0. We want all XBees to have the same PAN ID, so choose one. I choose a PAN ID of ‘1’.



1. Click “Write” and notice that the PAN ID value of ‘1’ is now in blue. This means this is the value written to the XBee.



1. Enable API Mode with Escaping (2)



1. Plug your other XBee in, and configure it with firmware version 238C, ZigBee Router API, and make sure you set the PAN ID to 1, and enable API Mode with Escaping (2)

## Getting Started with Gadgeteer

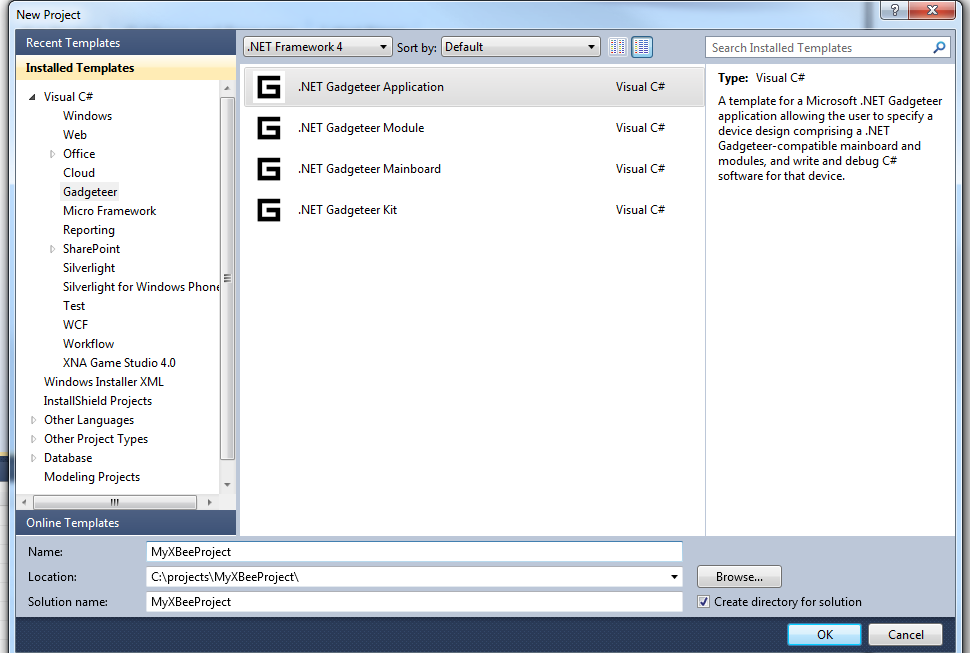
* Complete the tutorial by GHI “Getting Started with the FEZ Spider Kit for Microsoft .NET Gadgeteer” <http://www.ghielectronics.com/downloads/Gadgeteer/Mainboard/Spider_GettingStarted/>
  + This is optional, but highly recommended. Not only will it get you up to speed on writing software for your Spider, it’s also a high quality getting started guide. XBeeClient project contributors should look to that guide for inspiration and an example of what a great getting started guide can be.

## Installing XBeeClient

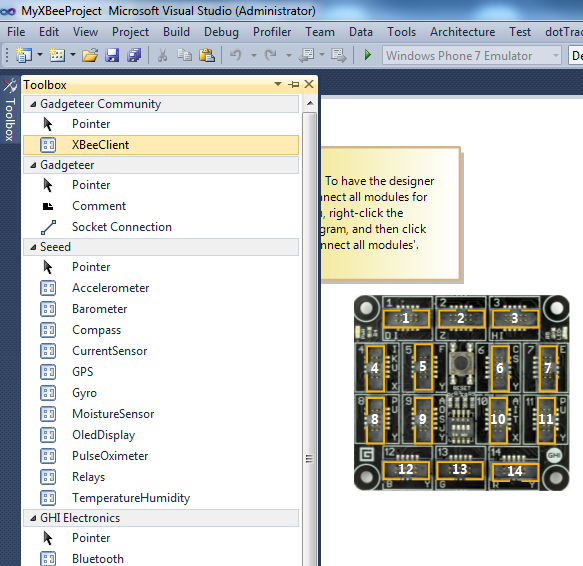
You should download the source code for XBeeClient. You can also download the XBeeClient release to use the tools and module without needing the source.

### Easy Method (no source code, just binaries)

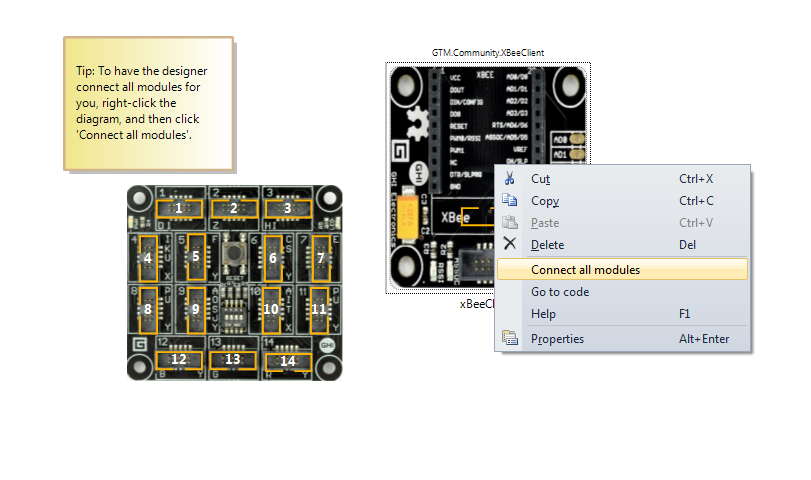
* Download the latest release
* Run the XBeeClient.msi to install
* Create a new Gadgeteer Application



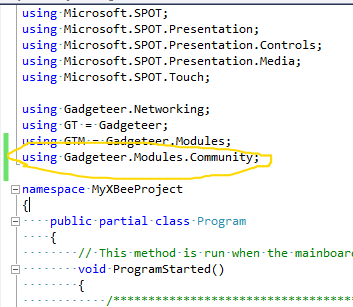
* drag and drop a Spider mainboard onto the canvas, as the default is Hydra. (If you have a Spider rather than a Hydra)
* Drag and drop an XBeeClient onto the canvas



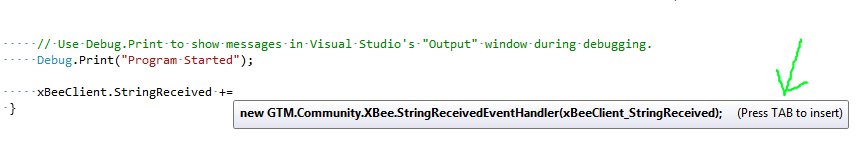
* Right click the XBeeClient and choose “Connect all Modules”. It may connect to socket 11, the valid socket types are ‘K’ and ‘U’, on the Spider these are sockets 11, 8, and 4.



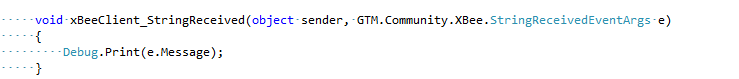
* Go to program.cs and add the namespace Gadgeteer.Modules.Community at the top (TODO: you shouldn’t have to add this, not sure why this is happening now, GTM.Community isn’t working)



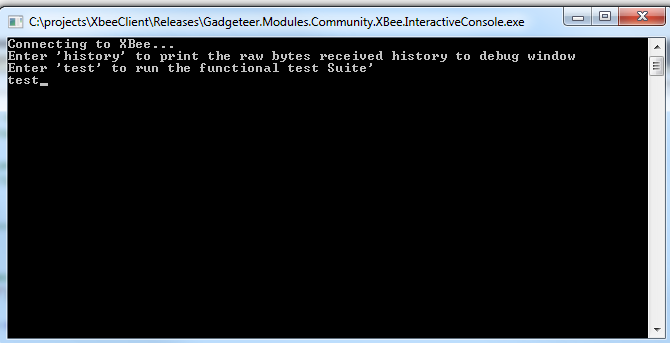
* after the Debug.Print statement, type ‘xBeeClient.StringReceived +=’ and notice the message (TODO: explain how to use intellisense and the tabs for autocomplete). Press TAB (twice) and it will add an event handler for you. The StringReceived event will be fired every time a full string is received by the XBeeClient. Large strings are divided into smaller packets, and when all the packets are received, this event will be called. Note that XBees don’t have a built in way of describing data types like ‘String’, nor is there build in methods of sending messages that take more than one packet, that is added by the XBeeClient library to make it super easy to use.



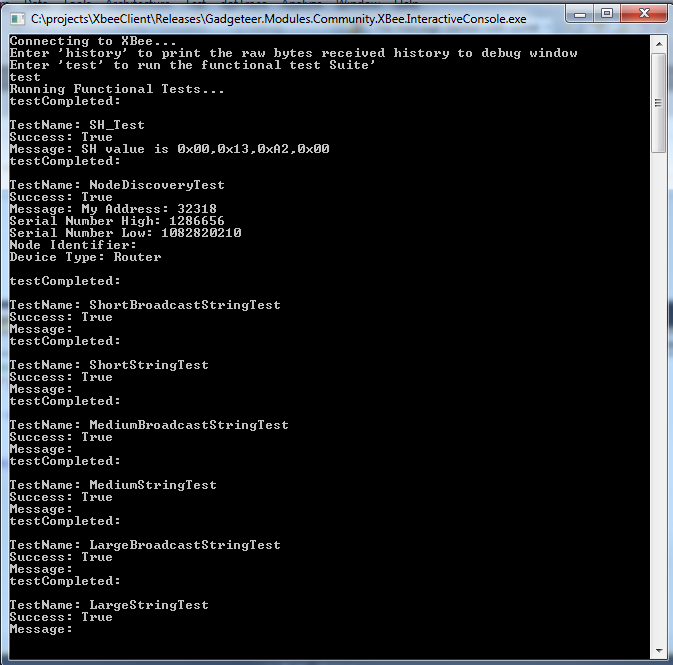
* Replace the contents of the xBeeClient\_StringReceived event handler with the following



* Hit F5 to run. This will deploy the program to the Spider, and in the output window, you should see “Program Started”.
* Now launch the Interactive Console that is included with the XBeeClient release files. Type ‘test’ and the functional tests will run. This will discover XBees and send them messages.



* If all goes well, you will see a number of tests run that pass. If the XBeeClient project lives up to its promises, when failure occurs it will be extremely easy to identify and fix. Here is the output of a successful run.



### Developing using source

When you are using the source to develop, please note that for PC projects you must pass the port number to the constructor, and for Gadgeteer projects you must pass the socket number.

//Pass in the Gadgeteer port number or on PC, the COM Port, e.g. 11

XBeeClient XBeeClient = new XBeeClient(11);

XBeeClient.UploadStringAsync(“Hello World”);

For your Gadgeteer project since you are not drag and dropping the XBee client from the toolbar, you have to add code to instantiate an XBeeClient. When you drag and drop, this code will be autogenerated behind the scenes.

// In your main method

xbeeClient = new XBeeClient(6); //the Gadgeteer port number or on PC, the COM Port

xbeeClient.StringReceived += new StringReceivedEventHandler(xbeeClient\_StringReceived);

…

void xbeeClient\_StringReceived(object sender, StringReceivedEventArgs e)

{

Debug.Print("String Received: " + e.Message);

}

# Advanced

TODO: describe how to send AT commands, local and remote.

Please see the XBeeClient Technical Design Specification document for detailed information on all the API methods and events.

# Learn More

## Blogs

<http://mikedodaro.net/> provides a wealth of information on Gadgeteer and XBee.