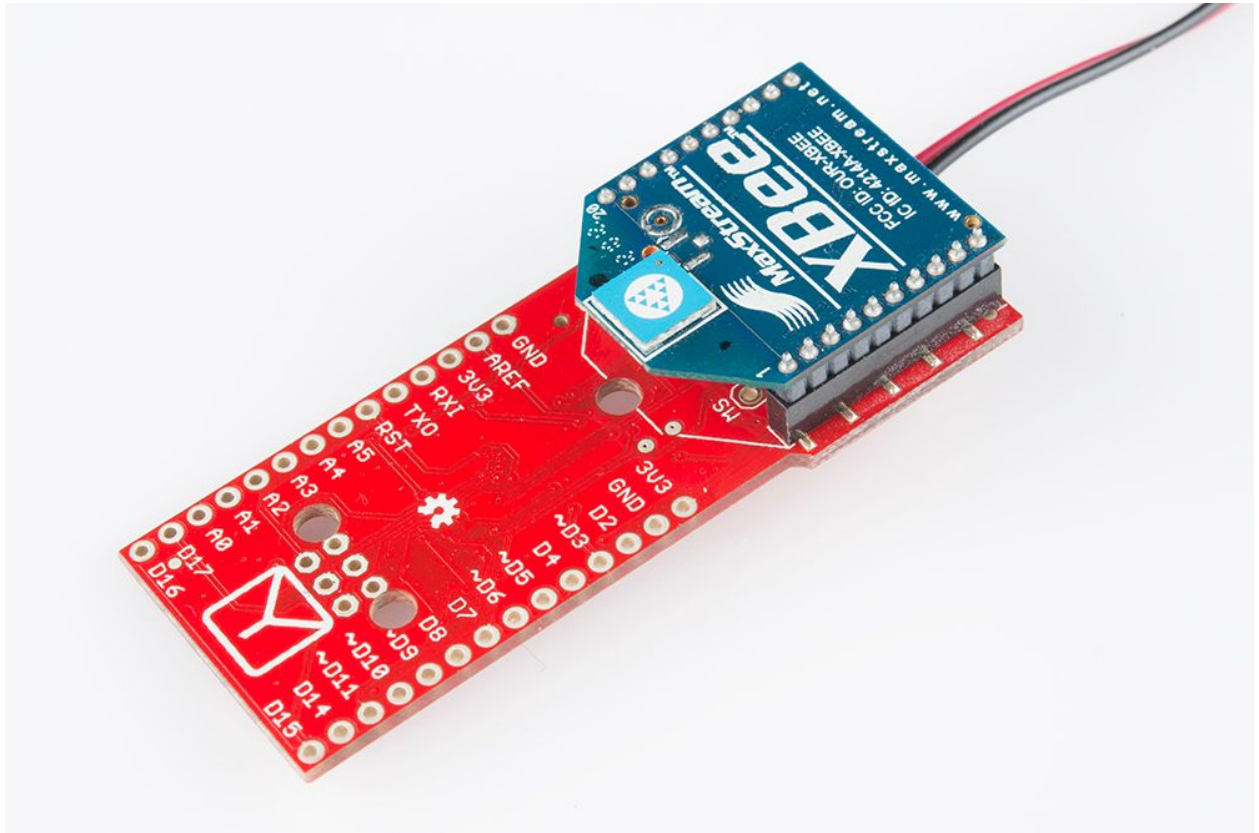


# XBee, Arduino, and Code Documentation

By Emma Teehan and Erica Hung

January 31, 2019



## I. Code Descriptions

### A. 2019 Receive Code

1. The purpose of this code is to find the heading of the transmitting beacon in degrees. Depending on the location of the receiver to the beacon, the degrees of the code will vary (refer to **Section II, B3** for specific degrees in each direction). The code will display the index heading of the signal with the strongest signal strength (RSSI) in the serial monitor.

### B. 2019 Transmitter Code

1. This code is used for delivering the direction of the compass in degrees. It is used for the HMC compass system. (Refer to **Section II, A1 and A2**)

### C. 2019 Signal Test - Receiver

1. This is useful in debugging the beacon and the receiver. When uploaded to the receiver, it will output every heading and RSSI level that it receives. (Refer to **Section II, C1 through C5**)

## II. How the Codes Run

### A. 2019 Transmit Code

1. Sends out: The degree heading of the compass
  - a) This code is received in different ways depending on if the receiver has the 2019 Receive Code or the 2019 Signal Test code
2. Receives: Nothing
3. Requires: The beacon to spin for the transfer of data to the receive code.

### B. 2019 Receive Code

1. Sends out: Nothing
2. Receives: The direction that the compass is pointing towards (in degrees)
  - a) Constantly receiving
  - b) Reverses the direction it receives by  $180^\circ$  (states direction the robot should go), this is what is displayed
  - c) Always a positive value
3. Degrees:
  - a) If receives north ( $0, 360$ ) will output  $180^\circ$
  - b) If receives South ( $180^\circ$ ) will output  $0, 360^\circ$
  - c) If receives East ( $270^\circ$ ) will output  $90^\circ$
  - d) If receives West ( $90^\circ$ ) will output  $270^\circ$
4. Example:

```
0
280
154
155
```

## C. 2019 Signal Test - Receiver

1. Sends out: Nothing
2. Receives: The signal bearing (not flipped 180 degrees) and displays the strength of the received signal. This tells the robot which direction the beacon is.
3. Degrees:
  - a) North = 0, 360°
  - b) South = 180°
  - c) East = 270°
  - d) West = 90°
4. You should see a radial pattern, which should include numbers (RSSI values) getting bigger and smaller repeatedly
  - a) As you walk away from the beacon, you should continue to see these patterns, but with weaker values
  - b) Note: The strength is the RSSI strength
    - (1) It will always turn out as negative values, so the smaller numbers will be less negative
5. Example:

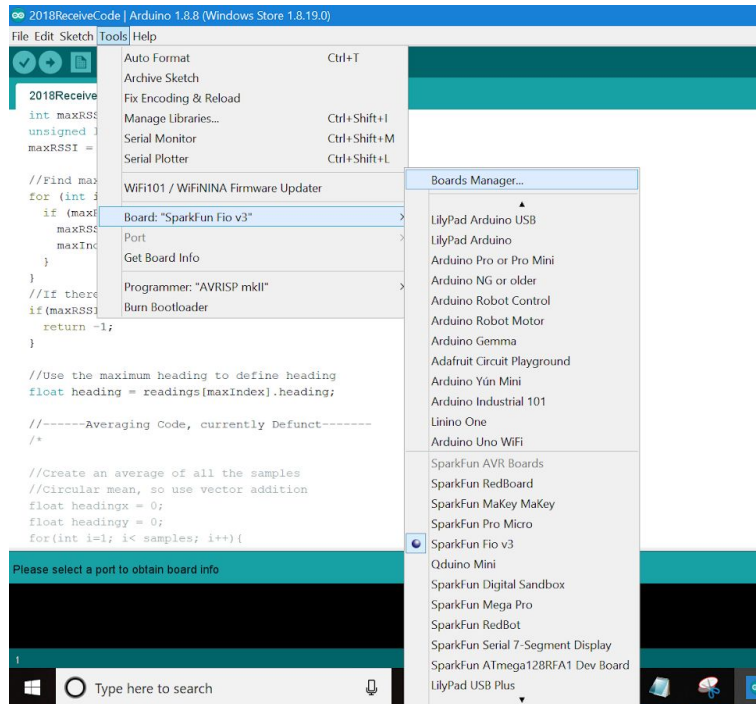
```

Recieved: 13.44 with strength -71dB
Recieved: 116.47 with strength -73dB
Recieved: 77.64 with strength -73dB
Recieved: 353.53 with strength -71dB
  
```

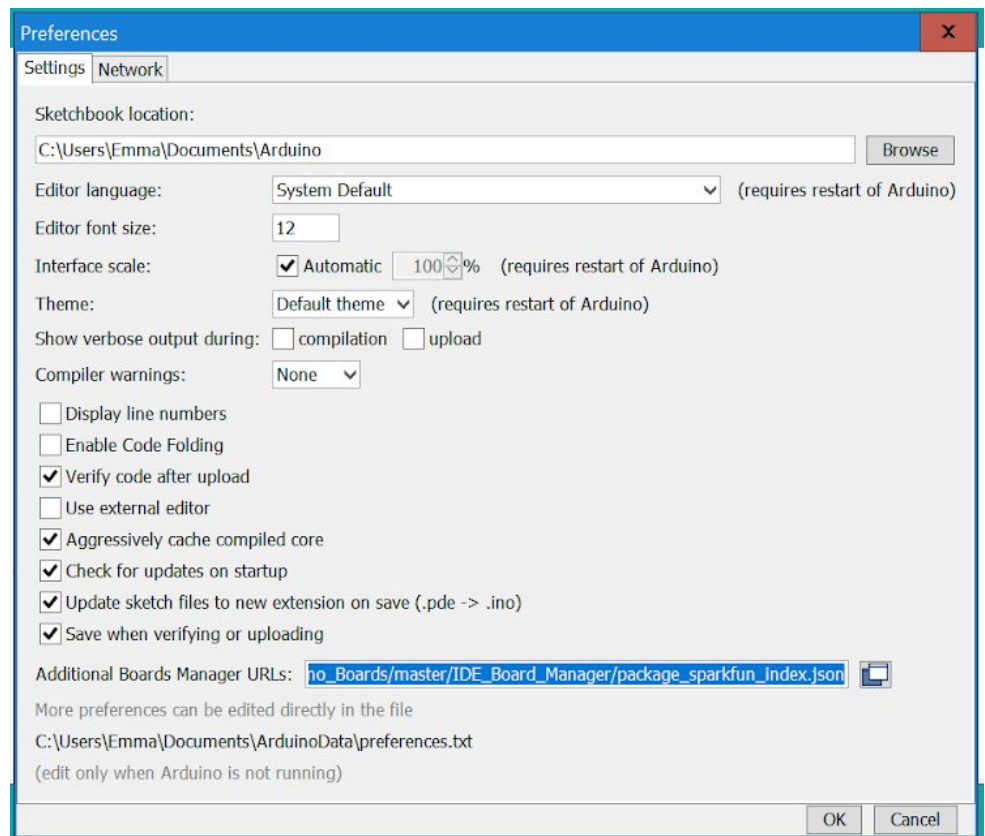
**III. Arduino**

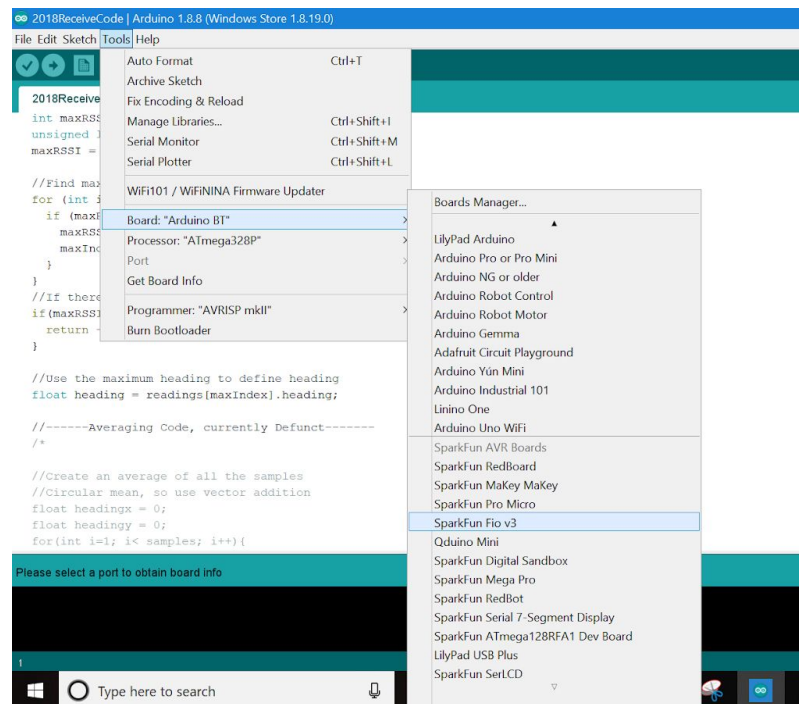
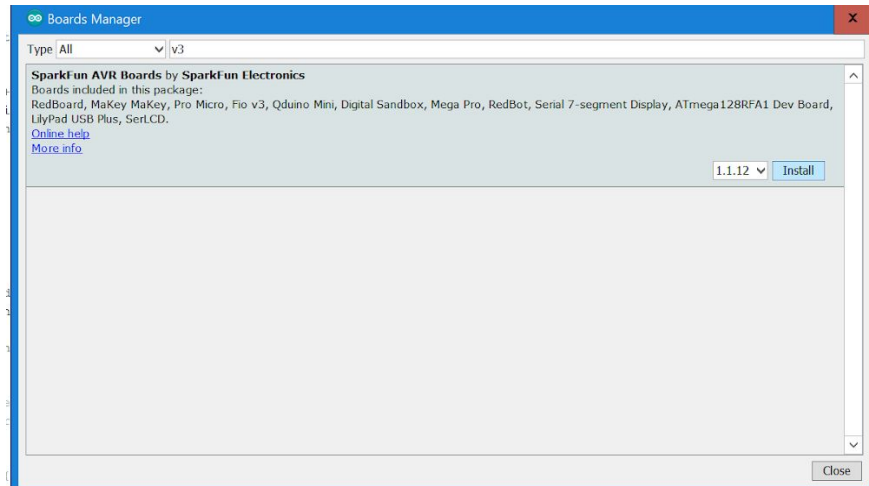
## A. Set SparkFun Fio V3 board up using this link:

1. [https://raw.githubusercontent.com/sparkfun/Arduino\\_Boards/master/IDE\\_Board\\_Manager/package\\_sparkfun\\_index.json](https://raw.githubusercontent.com/sparkfun/Arduino_Boards/master/IDE_Board_Manager/package_sparkfun_index.json)



a) This link should be copied into the “boards manager” in Arduino

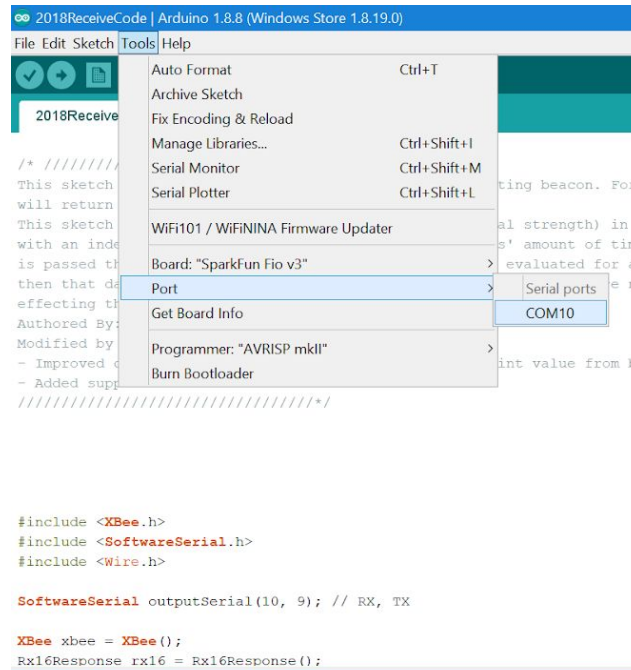




B. Download these ZIP files from github to allow arduino to understand the XBee and the arduino code:

1. <https://github.com/andrewrapp/xbee-arduino.git>
2. <https://github.com/jarzebski/Arduino-HMC5883L.git>

C. Select the Port

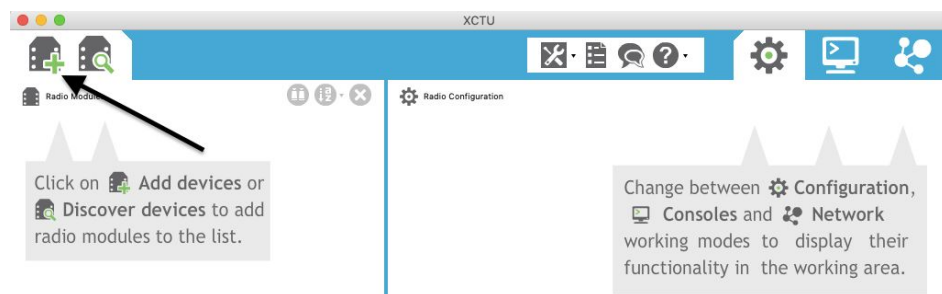


**Note:** On Mac, the port will appear slightly differently and will look like something shown as “usbmodem”

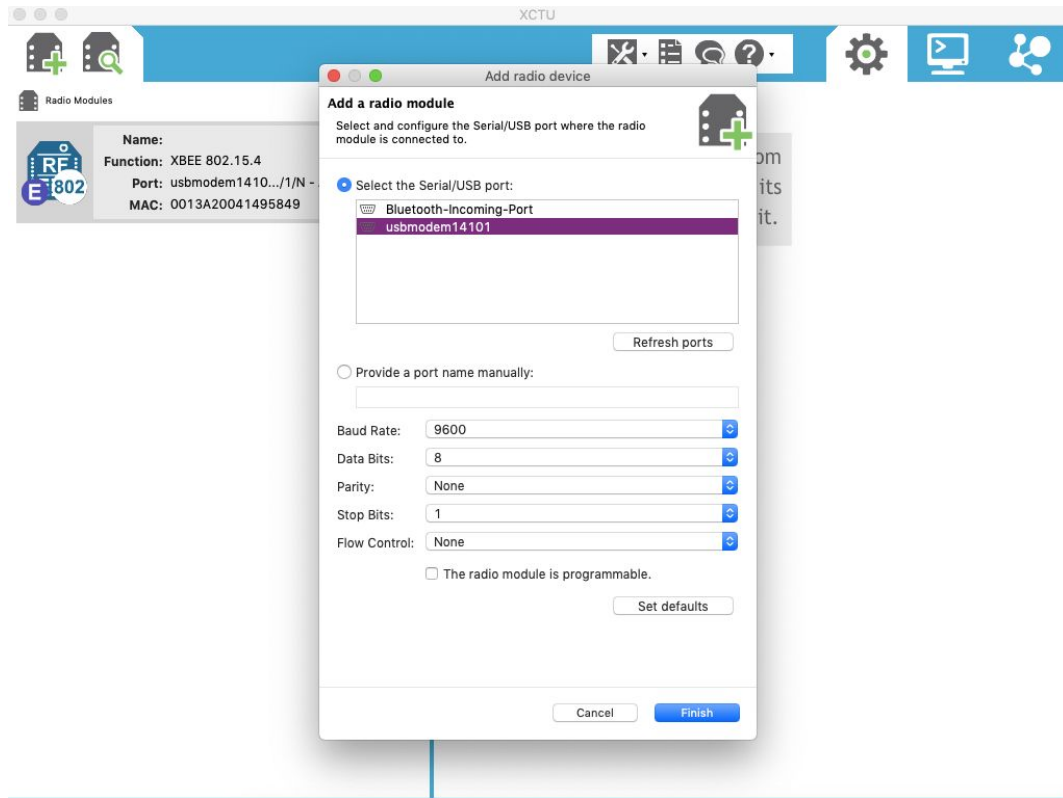
#### IV. XBee & XCTU

##### A. Steps to use XCTU and program the XBee:

1. Download XCTU with the link below:
  - a) <https://www.digi.com/products/iot-platform/xctu#productsupport-utilities>
2. Quit XCTU (there may be problems if XCTU is running while arduino is uploading)
3. Attach the XBee to the Fio
4. Plug the Fio into the computer using a USB cable
5. Make sure you have the “FirstRunXCTU” sketch downloaded
6. Select the board in Arduino “SparkFun Fio V3 3.3V/8MHz”
7. Upload the “FirstRunXCTU” sketch to the FIO
  - a) Comment out the part of the code that contains “57600”, and make sure the part that contains “9600” is not commented out
  - b) First hit “compile” (the check button) in the top left corner
  - c) Then hit “upload” (the arrow button) next to the “compile” button
8. Now, open XCTU and click on “Add Devices” in the top left corner



9. Select the “usbmodem140101” (Note, on Windows, the serial port will show as “com”)
  - a) Make sure the baud rate is set to 9600
  - (1) Select finish



Note: If you are using a Windows computer, make sure to always make the screen fullscreen, it will help you select the port.

- V. Now, refer to the companion XBee Guide for how to now configure the XBee (begins on page 14).



## VI. Troubleshooting the Sparkfun Fio v3 when it doesn't work

- A. <https://learn.sparkfun.com/tutorials/pro-micro--fio-v3-hookup-guide/troubleshooting-and-faq>

### Reset to Bootloader

We ship the Pro Micro with a modified version of the **Arduino Leonardo** bootloader, with one major enhancement. When a Leonardo (or any device using the "stock" bootloader) is externally reset, it goes back into the bootloader...and waits there eight seconds before it starts running the sketch. For some embedded projects, waiting eight seconds before a program runs isn't acceptable, so we modified the bootloader run time.



*Leonardo bootloader on reset functionality.*

When a Pro Micro is externally reset (by pulling the RST pin low), it'll only briefly (<750ms) start the bootloader before continuing on to the sketch. If you need the bootloader to run longer, **resetting twice quickly** will get the Pro Micro to enter bootloader mode for eight seconds.



*Pro Micro and Fio v3 reset functionality. Press reset twice, quickly to enter bootloader mode.*

Resetting the Pro Micro's, in particular, can be tricky, because there's no reset button. The 'RST' pins needs to be connected to ground to initiate a reset. This can be done with a small piece of wire, or an externally connected button.