COSGC Robotics Beacon Users Guide: Part 1, XBee Basics

The purpose of this guide is to walk the user through the basic hardware and software used for the Colorado Space Grant Robotics Challenge beacon system. It includes instructions on how to configure an XBee module to communicate with the Robotics Challenge beacon, how to use the signal from the beacon to determine the direction from a robot to the beacon, and how to build a version of the beacon that can be used as robots are developed and tested.

This guide assumes that you have a working knowledge of Arduino hardware (for example an Arduino Uno), and programming using the Arduino IDE.

1 Configuring and Testing XBee Modules

In order to configure and test the basic communication of your XBee modules, you will need the following hardware and software.

1.1 Hardware

- 2 XBee modules (either series 1 or series 3, but not series 2)
- 1 SparkFun FiO v3 [https://www.sparkfun.com/products/11520](https://www.sparkfun.com/products/11520)

1.2 Software


1.3 Hardware and Software Overview

This section gives a brief overview of the hardware and software used to configure and test the basic functionality of your XBee modules.

1.3.1 XBee Modules

An XBee is a wireless radio module, that will allow your robot to receive signals transmitted by the Robotics Challenge beacon. The beacon transmits a signal that will allow your robot to
determine a compass heading to the beacon. XBee’s can use different protocols to communicate:

- **IEEE 802.15.4** is a standard protocol for low rate wireless personal area networks. This is the protocol that we are currently using for the robotics beacon.
- **ZigBee** is an extension of the IEEE 802.15.4 standard that adds capability at the expense of greater complexity.
- **Digimesh** is another protocol similar to Zigbee in that it is another extension of the IEEE 802.15.4 standard, but it offers different functionality than that of ZigBee.

### 1.3.2 Hardware: XBee Explorer

XBee modules come with a number of interface pins that allow them to be powered and communicate with other devices, such as an Arduino. The simplest way to get working with an XBee module is to use an XBee Explorer, manufactured by SparkFun. The XBee Explorer is a standalone interface board that allows you to plug the XBee module directly into a USB port on a computer. We will use an XBee explorer to configure and test the XBee module before integrating the XBee on a robot.

### 1.3.3 Software: XCTU

In addition to the XBee Explorer, we also need software to talk to the XBee module. The manufacturer of XBees provides the XCTU software that allows us to do that.

### 1.3.4 Hardware: FiO

The SparkFun FiO v3 AT32U4 (or FiO for short) is an Arduino based board with an XBee socket. This allows us to use the FiO board as the interface between the microcontroller on a robot and the XBee, as well as to be the “brains” for the beacon system. Another benefit of the FiO is that they consume relatively little power. This allows them to be battery powered. Each comes with a connection for a lithium polymer battery and includes a charging circuit. In other words, you can charge the Li-Po by plugging both the board into a power source via USB as well as the Li-Po into the board.

### 1.4 Configuring an XBee series 3 to work with XBee series 1

For the Robotic Challenge, we are currently using the IEEE 802 15.4 protocol to communicate between the beacon and robots. IEEE 802 15.4 is the protocol used by Series 1 XBees, so if you have a Series 1 XBee, it comes ready to communicate using that protocol. Series 1 XBees are no longer being manufactured, which means that if you need to purchase a new XBee, you will most likely need to purchase a Series 3. The default communication protocol used by Series 3 XBees is not IEEE 802 15.4, however they can be configured relatively easily to use that protocol using an XBee explorer and the XCTU software. That means that if you have purchased a new XBee Series 3 module, you will need to update the firmware on your XBee to work with the IEEE 802 15.4 protocol that is used by Series 1 XBees.
If you have a Series 1 XBee, you can skip the rest of this section.

1.4.1 Procedure to Update XBee series 3 Firmware

1. Plug your XBee Series 3 module into the XBee socket on your XBee Explorer board (make sure that the tapered sides on the one end of the XBee line up with lines printed on the XBee Explorer.)
2. Plug your XBee Explorer into a USB port on your computer.
3. Launch the XCTU software.
4. Click on the “Add Device” button in XCTU, and select the communication port that the XBee Explorer is plugged into. Don’t change the Baud Rate or other options, and click “Finish”. XCTU will scan the selected port for a few seconds, then add an entry for your XBee module to the left hand pane in the XCTU window.
5. Select your XBee module in the left hand pane. XCTU should spend several seconds scanning the configuration of your XBee module, and then display that information in the right hand pane.
6. Select the “Update Firmware” button near the top of the right hand pane, and select the following settings: Product Family = XB3-24, Function Set = XBee3 802.15.4 TH, Firmware Version = 2006 (Newest), and click “Update”. A pop up window should show you the progress of the update procedure, and when it is finished, the configuration information in the right hand pane should reflect the change.

1.5 Testing XBee to XBee Communication

Once you have all your XBee modules “speaking” IEEE 802 15.4, it is good to test that they can communicate with each other, before we configure the XBees to receive signals from the robotics beacon.

We can use XCTU to test the communication between 2 XBees, but before we do, we will need to setup the Sparkfun FiO to allow the second XBee to be plugged into your computer.

1.6 Setting up the FiO

The Sparkfun FiO is an Arduino board with an integrated XBee socket, and is what we will use to receive data from the robotics beacon. It is also the board used to run the robotics beacon. Before you can use the FiO for the first time, you must add the FiO board to the list of boards your version of the Arduino IDE knows about. These instructions assume that you are using the Arduino IDE version 1.6.4 or later, and that you are using a computer running Windows 10, or MacOS. If you are running an earlier version of Windows, you may need to perform some additional steps to manually download drivers for the FiO, see the Sparkfun FiO Hookup Guide listed in the Reference section for detailed instructions.
1. Launch the Arduino IDE, and open the preferences (File->Preferences in Windows, Arduino->Preferences on a Mac.)

2. Near the bottom of the Preferences window, enter the following URL in the “Additional Boards Manager URLs” window: 
   https://raw.githubusercontent.com/sparkfun/Arduino_Boards/master/IDE(BoardManager/package_sparkfun_index.json then click “OK”

3. Open the Board Manager by selecting the Board menu, hovering over “Board”, and selecting “Board Manager”.

4. In the Board Manager window, scroll down to the “SparkFun AVR Boards”, and click Install.

5. You should now be able to select the “SparkFun FiO v3” from the “Board” menu.

**Important!** Before uploading code to the FiO, you must have selected the “SparkFun FiO v3” as your board in the Arduino IDE. Attempting to upload code the FiO with a different board selected will most likely “brick” the FiO (that is, put the FiO in a state where you will no longer be able to upload code to the FiO). If you do accidentally brick your FiO, never fear, there are instructions on how to recover a a bricked FiO available on SparkFun's FiO Hookup Guide.

1.7 Testing Basic XBee Communication

In order to test the communication between the two XBee modules, we need to plug them both into the computer, so that XCTU can talk to both modules at the same time. We will use the XBee explorer to plug one XBee in, and the FiO to plug the other in.

1.7.1 Loading Code onto the FiO

In order to allow XCTU to communicate with the XBee a simple sketch that tells the Fio to send data from the Xbee to the computer and vice versa. That way the Fio becomes a transparent medium through which data travels to and from the Xbee and Fio. The steps to accomplish this are as follows:

1. Attach the Xbee to the Fio. (Make sure the Xbee is facing the correct way! There’s an outline on the Fio so you can easily tell.)
2. Plug the Fio into a computer via USB.
3. Launch the Arduino IDE, and select the “SparkFun FiO v3” from the “Board” menu.
4. Open the “XbeeXCTU” sketch, found here: https://github.com/Colorado-Space-Grant-Consortium/Robotics-Archive/tree/master/2015_Robotics_Challenge/XbeeDiagnostics/XbeeXCTU
5. Make sure that the “Serial.begin(9600);” and “Serial.begin1(9600);” lines are uncommented, and the “Serial.begin(57600);” and “Serial.begin1(57600);” lines are commented out.
6. Compile and upload the “XbeeXCTU” sketch to the FiO.
You should now be able to use “Add Devices” to add the XBee connected to the FiO to the right hand pane in XCTU, along with XBee connected to the XBee Explorer.

1.7.2 Testing the XBees

With both XBee’s configured, they ought to be communicating with each other. Now is a good time to make sure of this working that we don’t run into issues later.

With both XBee’s plugged into the computer, in XCTU, click the terminal icon in the top right corner. Open serial communication by clicking the plug icon. You will have to do this for both XBee’s

With the Transmitter XBee selected, click the ‘add new frame to list’ button in the lower right. A window will pop-up. Click “create frame using ‘Frames Generator’ tool”.

![Image of XCTU terminal and frames generator window]

Create frame using ‘Frames Generator’ tool...
Select the proper settings as illustrated in the image below. The RF data doesn’t really matter, but make sure you put something there. Hit OK when you are finished.

Click ‘Send selected frame’ in the bottom right. You should notice a message appear in the Frames log that says a packet has been transmitted.
Now select the Receiving XBee by clicking on the tab at the top. You should notice a message in the Frames log that says a packet has been received. Congratulations! The XBee’s are now communicating with each other in API mode.

2 References

https://learn.sparkfun.com/tutorials/exploring-xbees-and-xctu/all#starting-with-x-ctu

https://learn.sparkfun.com/tutorials/pro-micro--fio-v3-hookup-guide