Introduction to Microcontrollers

Experiment 2:

(KEY FOR ECE AND CYENG STUDENTS)

Communication with XBee Wireless Modules

Developed 2019 by S. Rowland and M. Eckels

Contents

1 Objectives

To learn to utilize XBee wireless modules to set up a server and client to transmit analog data.

2 Background

2.1 XBee Modules

2.1.1 What is an XBee?

XBee modules are embedded solutions providing wireless end-point connectivity to devices. These modules use the IEEE 802.15.4 networking protocol for fast point-to-multipoint or peer-to-peer networking. They are designed for high-throughput applications requiring low latency and predictable communication timing.

2.1.2 XBee Breakout Boards

The adapter allows the PC to configure, read, and write to the XBee module for wireless data transmission via USB cable. You can upload your program through the wireless controller module, or conduct wireless real-time data transmission. The adapter uses a FTDI232 USB-UART converter chip to ensure stable and reliable data transmission.

2.2 XCTU Software

XCTU is free software, provided by Digi (the manufacturer of XBee), which we will use to configure and manage XBees.

2.3 IR Range Sensors

The infrared range sensors used in this lab will allow the user to set a trigger distance with the trim pot on the IR board to determine at what distance the sensor will detect an object – a trigger event – or detect nothing. Most infrared sensors will give an analog output to give a variable range as an analog control signal, this board will only output high or low.

2.4 Arduino Serial Communication

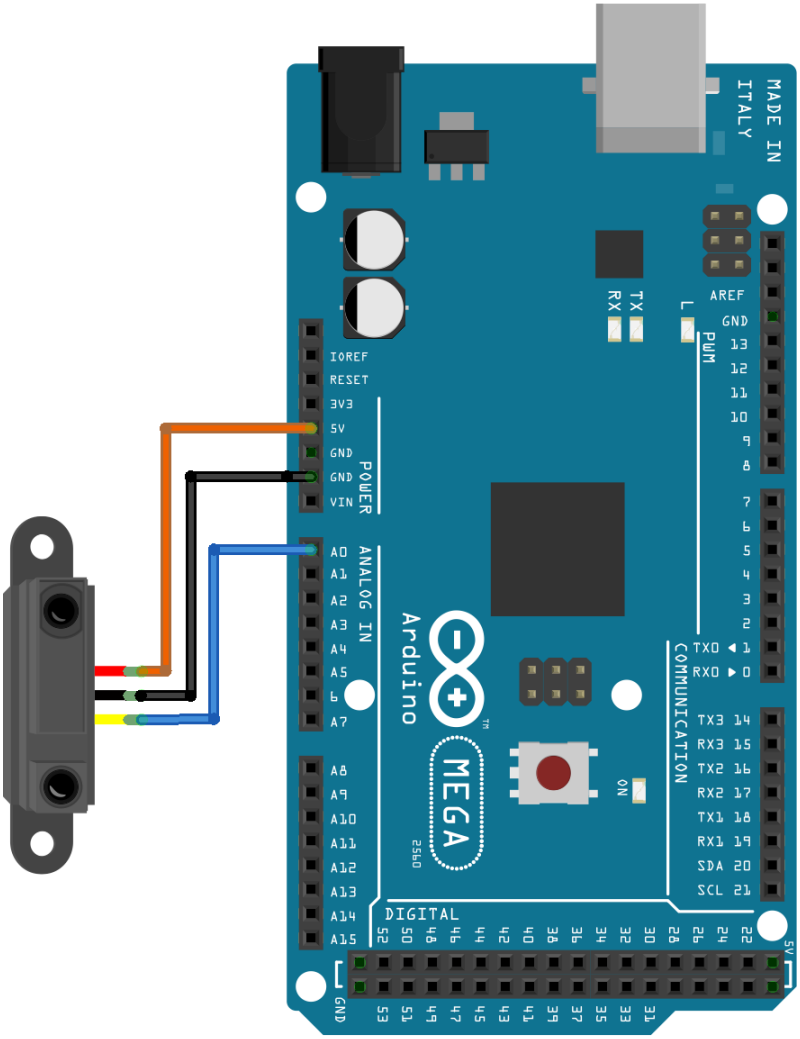
Serial communication on pins TX/RX uses TTL logic levels (5V or 3.3V depending on the board). Serial is used for communication between the Arduino board and a computer or other devices. All Arduino boards have at least one serial port (also known as a UART or USART): Serial. It communicates on digital pins 0 (RX) and 1 (TX) as well as with the computer via USB. The Arduino Mega has three additional serial ports: Serial1 on pins 19 (RX) and 18 (TX), Serial2 on pins 17 (RX) and 16 (TX), Serial3 on pins 15 (RX) and 14 (TX).

3 Procedures

3.1 Reading from the Range Sensor

3.1.1 Wiring the Arduino

To connect the Arduino to the IR sensor, connect the OUT pin on the IR sensor to the A0 pin on the Arduino board, then connect the GND pin on the IR sensor to the GND pin on the Arduino board, and finally connect the VCC pin on the IR sensor to the 5V pin on the Arduino board.



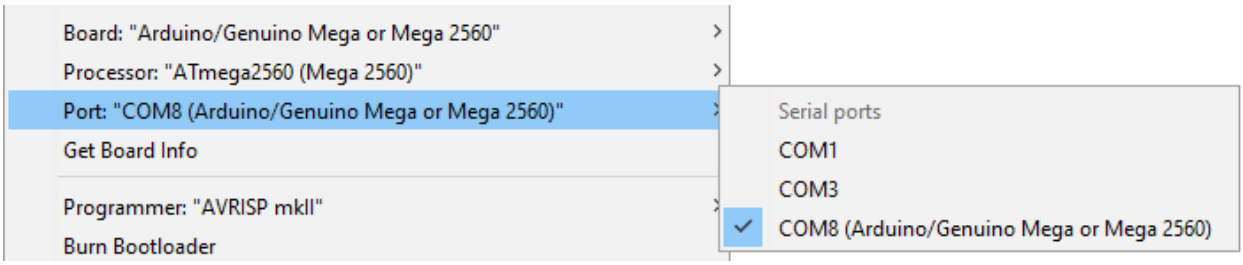
3.1.2 Programming Arduino

This example code reads from pin A0 on the Arduino board and prints the output state of the sensor to the default Serial 0 line; giving an output to the Serial Monitor in Arduino.

```
#define INPUTPIN A0

int value = 0;

void setup(){
  pinMode(INPUTPIN, INPUT);
  Serial.begin(9600);
}

void loop(){
  value = digitalRead(INPUTPIN);
  Serial.print("Value: ");

  if(value)
    Serial.print("NOTHING");
  else
    Serial.print("TRIGGER");

  Serial.println();
  delay(1000);
}
```

3.1.3 Testing the IR Sensor

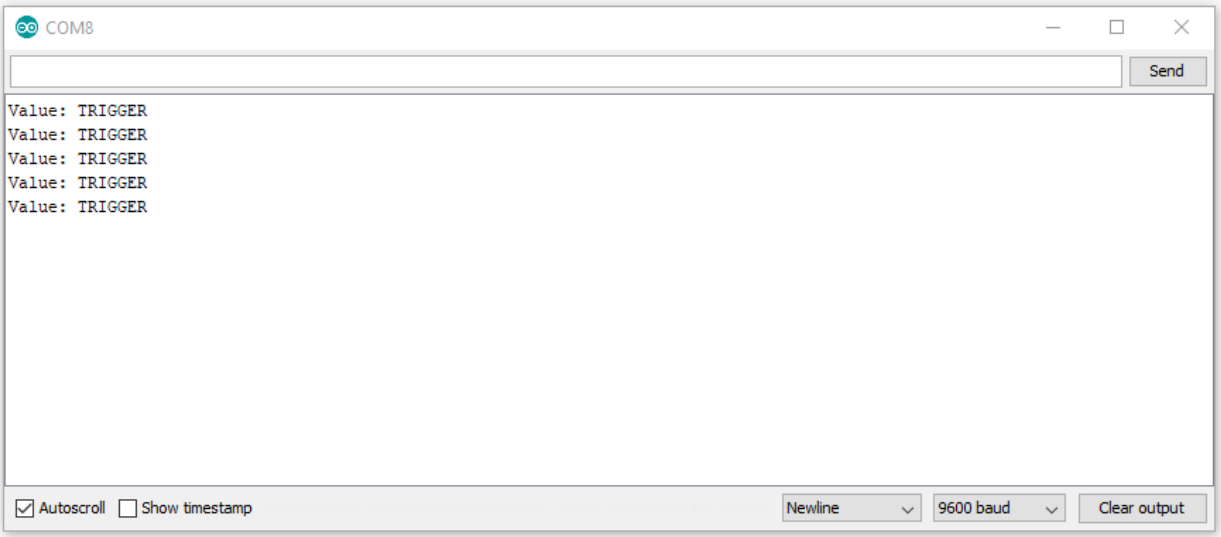
In a new Sketch in the Arduino IDE, enter the code, and select the COM device corresponding to the Arduino Mega microcontroller:



Once selected, upload the code to the Arduino:

Once uploaded, open the Serial Monitor to display the state of the infrared sensor:

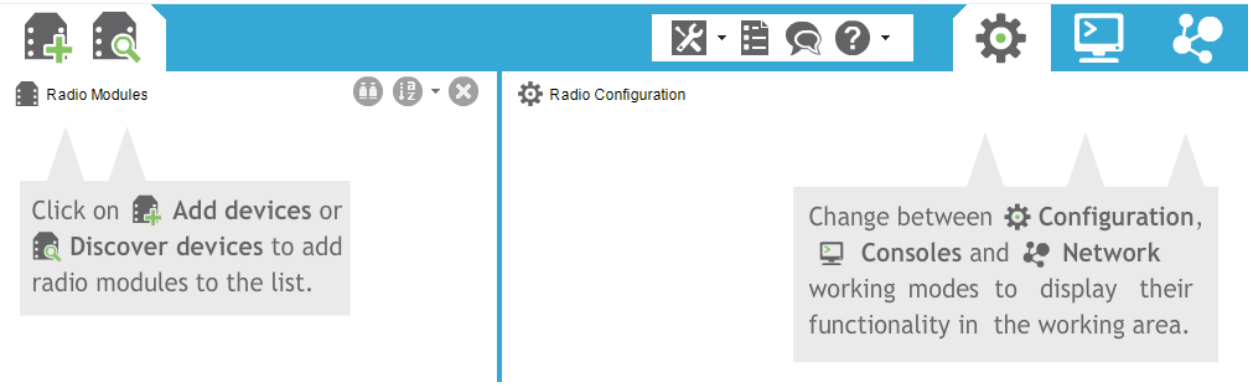
The Serial Monitor should now display the current state of the sensor updating at one second intervals:



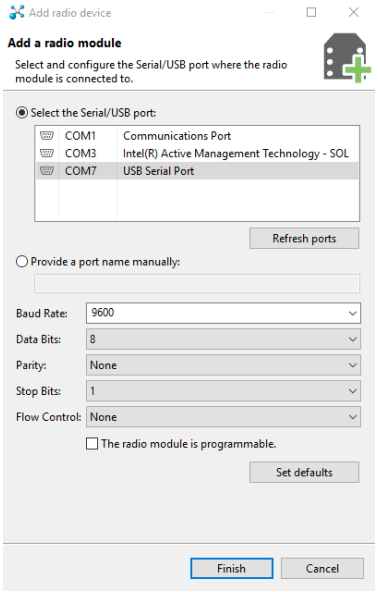
3.2 Configuring the XBee Modules

3.2.1 Adding an XBee Device in XCTU

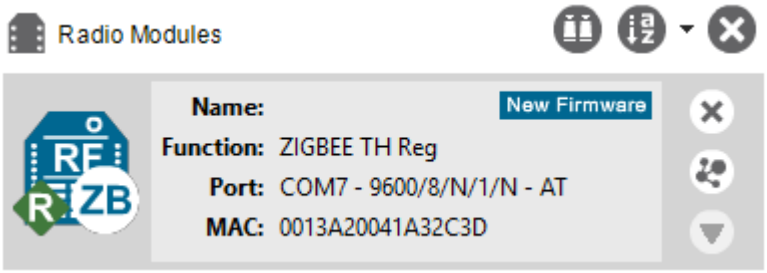
To begin configuring XBee devices, you must first open up XCTU on the computer, and plug in the XBee device with the USB FTDI Adapter and USB-Mini cable.



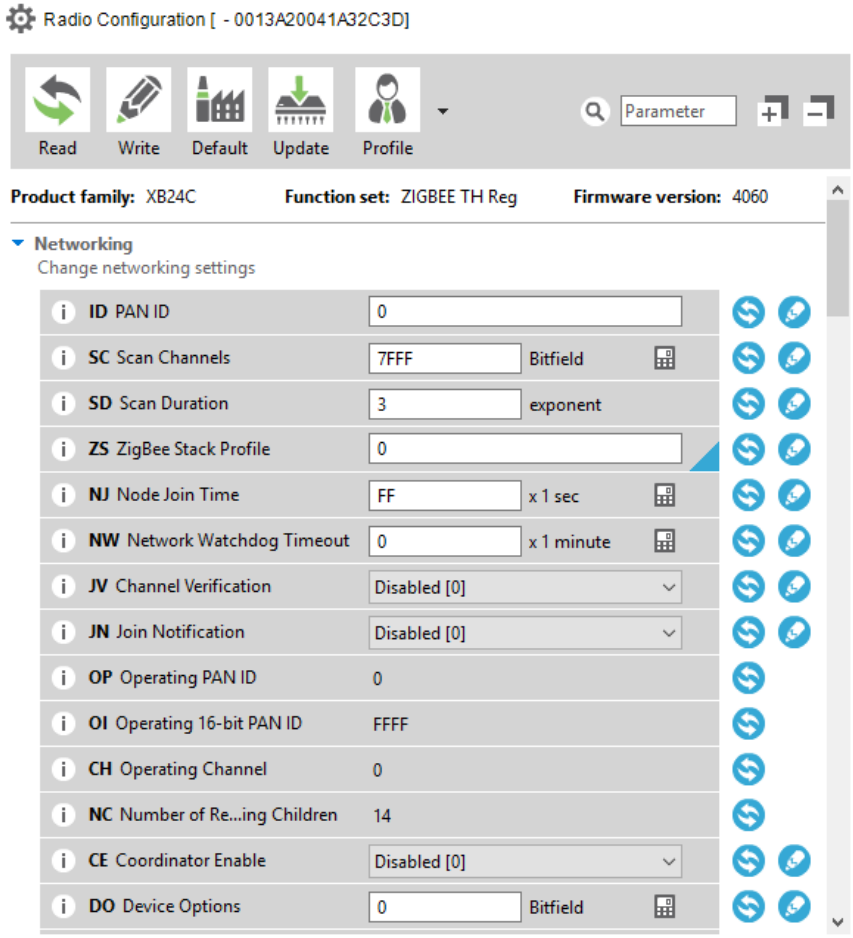
Once you are finished click the add devices button in XCTU in the top left corner. Click on the device that is called USB Serial Port and click Finish. The COM port name will differ between computers. If the device doesn't show up, then try to refresh the ports.



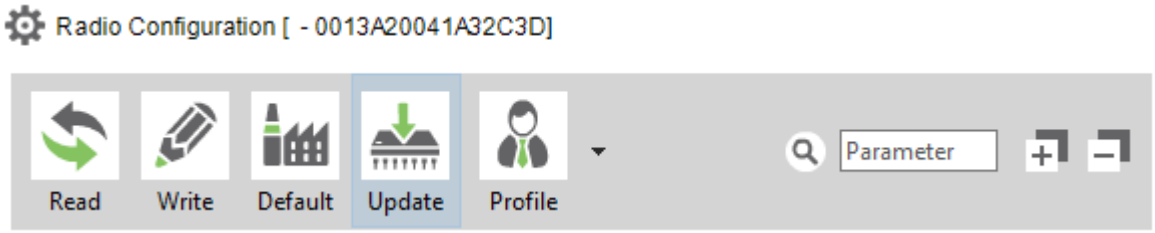
Once the device is loaded, its function configuration will need to be changed. Click on the newly added device to open its properties:



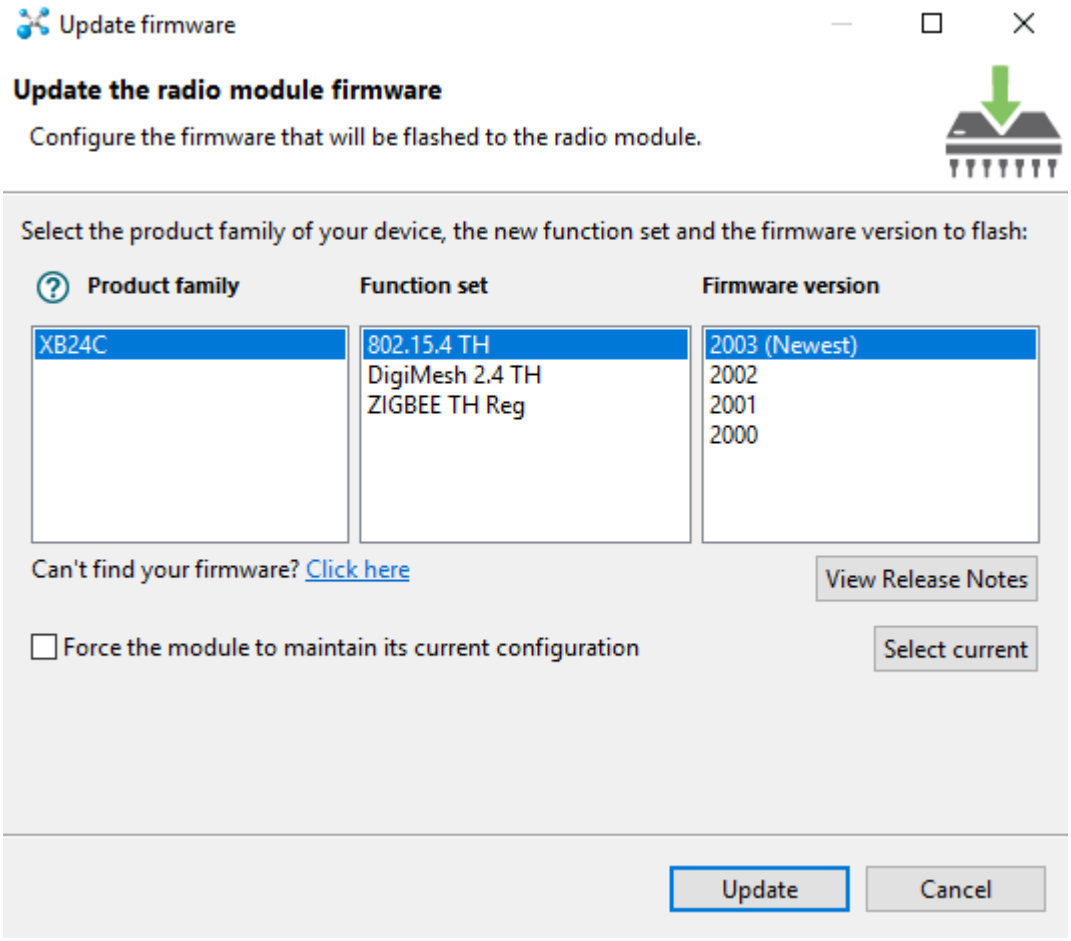
Once the data is loaded, the configuration can be modified in the right side of the XCTU window:



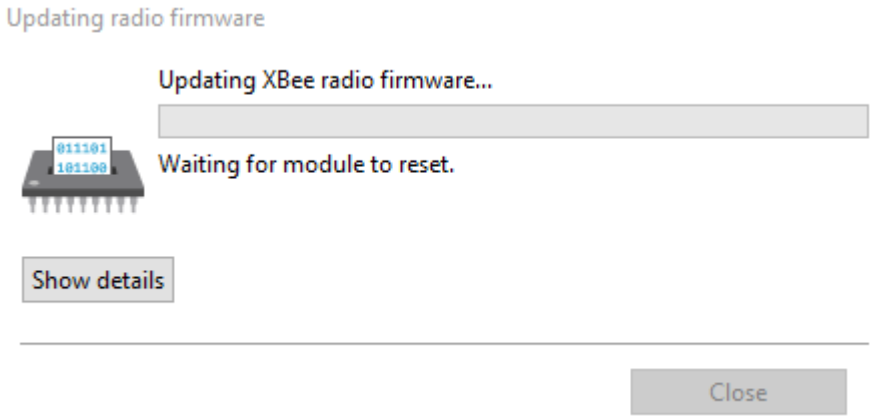
We will start with changing the function set of the XBee module. To do this click the update button in the top bar of the Radio configuration window:



Now with the firmware update window open, select the 802.15.4 TH function set with the newest firmware version, in this case 2003.



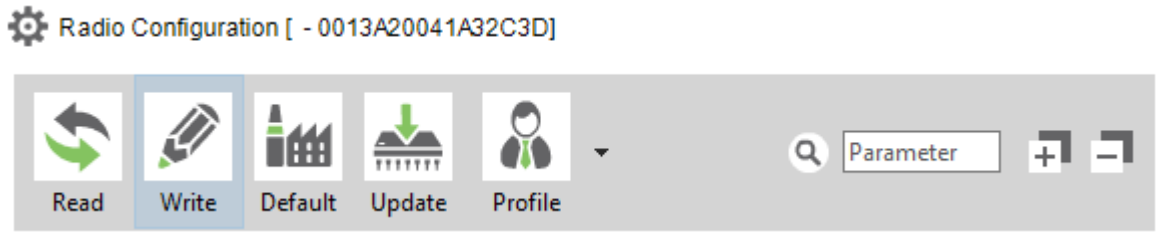
Once selected, click update. The update may take a few seconds:



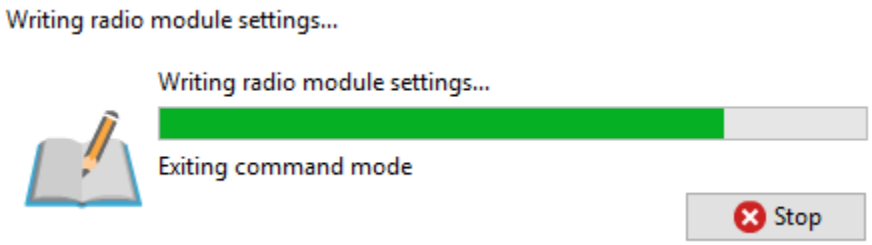
Once the module is updated, we only need to now change one value within the radio configuration window, the PAN ID or Personal Area Network identifier.

The default value is 3332. You will need to change this to a unique value for your group so that you will only communicate on your own unique PAN. To change this pick 4 hexadecimal characters and coordinate with the class to guarantee there will not be an overlap. Once selected, enter your new 4 character hex code in the PAN ID input box:

The green corner indicates that there is a value that is updated and needs to be written to the XBee. To write the change, click the write button in the top bar of the radio configuration window:

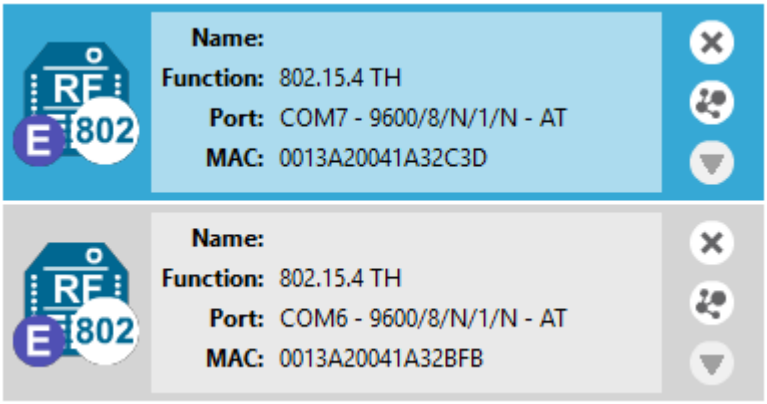


The write may take a few seconds:



Now that you have updated the XBee and changed the PAN ID to a unique value, this XBee is now ready to be used.

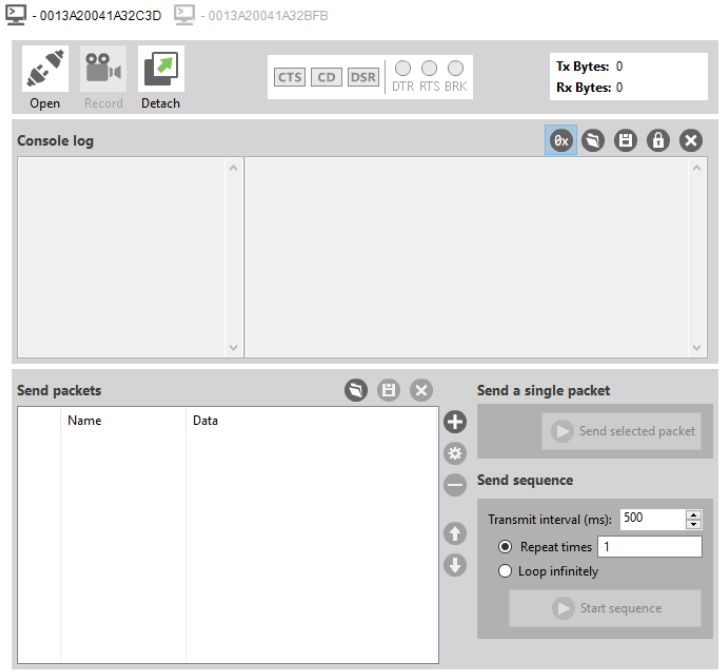
You will now need to repeat the previous steps within this section on the second XBee through XCTU to guarantee that both radio modules are on the same firmware version and have the same PAN identifier.



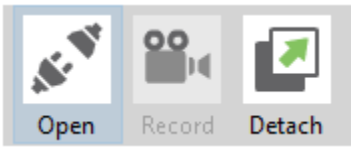
3.2.2 Testing the XBee Communication

To verify that your XBees are communicating properly, select one of the devices and click the terminal button at the top right of the screen:

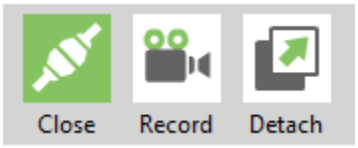
This window allows you to directly write characters into the XBee RX/TX lines:



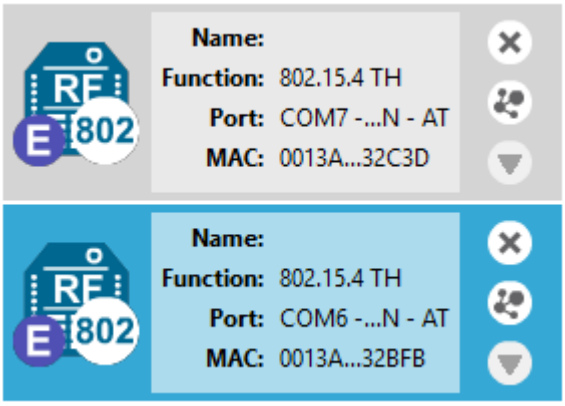
To start the communication, you will need to open both ports on both XBee devices. To do this click the Open button in the top left of the terminal window:



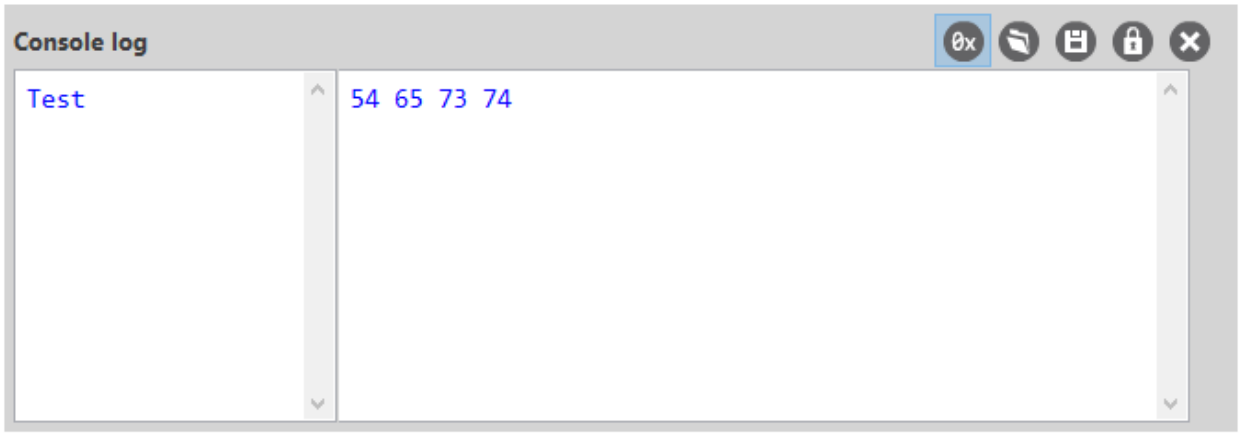
Once the port is opened, the icons will change:



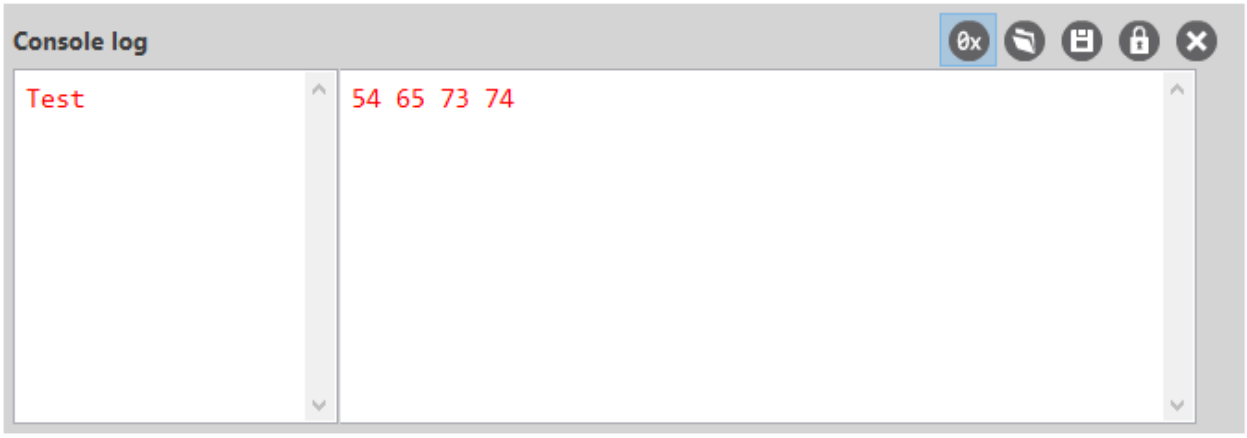
Repeat this again on the second device:



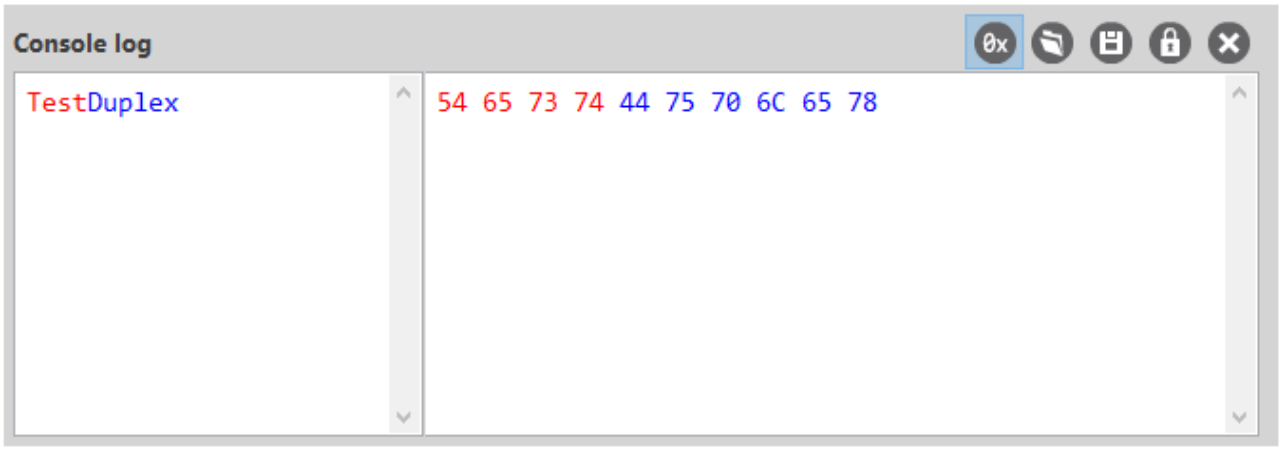
Once both are opened, go to the first window and enter some characters by clicking on the Console log and typing values:



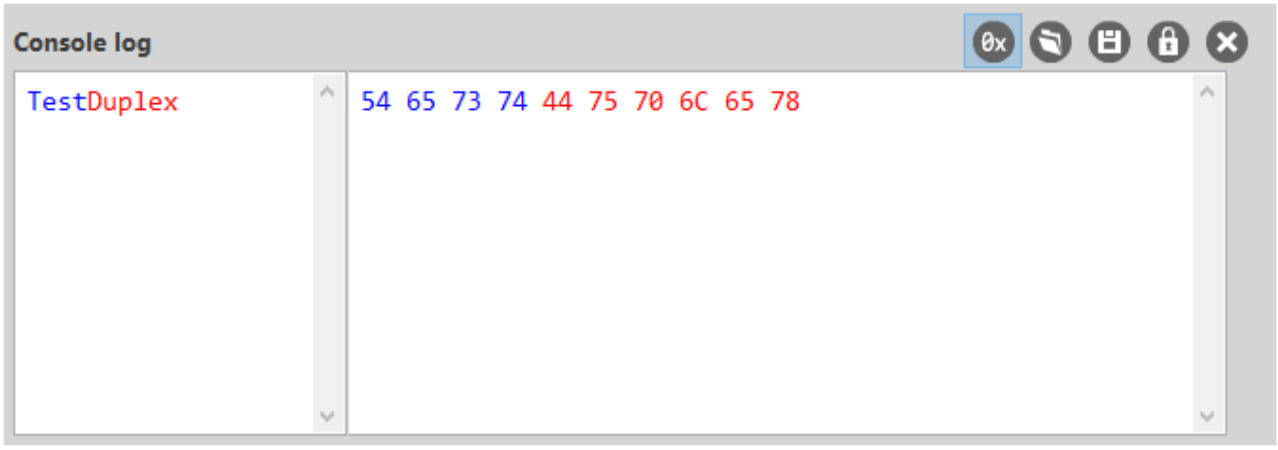
Now switch to the second device, to see if the values were transmitted properly:



Now on the second device, enter more characters to verify that there is duplex communication:



The device will display red text as received characters and blue text as transmitted characters. Switching back to the primary device will invert the colors of the text based on where the devices were transmitted from and received on:

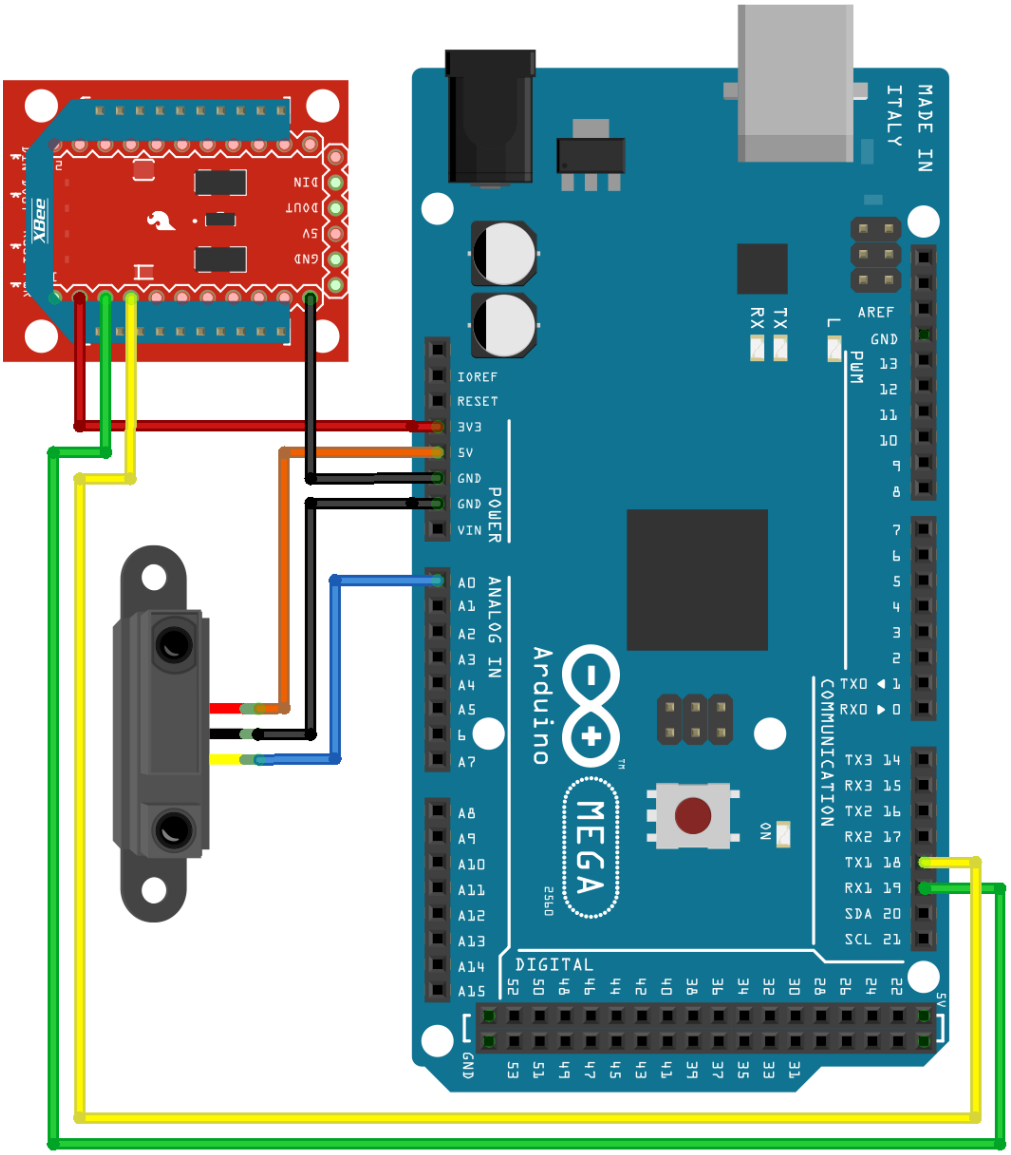


Now that you know that the XBee devices are communicating properly, you are now ready to set up a client and a server.

3.3 Setting up the XBee Server

3.3.1 Wiring the XBee to the Arduino Mega

To set up the server, we need to connect an Arduino Mega to both the XBee wireless module and to the IR distance sensor. Start by connecting the 3.3V pin on the XBee USB breakout to the Arduino 3.3V pin, then connect the VSS pin on the XBee breakout to the Arduino GND pin, then connect the DOUT pin on the XBee breakout to the Arduino RX1/PIN 19, and finally connect the DIN pin on the XBee breakout to the Arduino TX1/PIN 18.



3.3.2 Programming the Arduino Mega

This example code allows you to read from an input pin on the Arduino and print the output data over serial; in this case, the serial communication will be passed through the XBee devices.

```
#define INPUTPIN A0

int value = 0;

void setup(){
  pinMode(INPUTPIN, INPUT);
  Serial1.begin(9600);
}

void loop(){
  value = digitalRead(INPUTPIN);
  Serial1.print("Value: ");

  if(value)
    Serial1.print("NOTHING");
  else
    Serial1.print("TRIGGER");

  Serial1.println();
  delay(1000);
}
```

3.3.3 Starting the Server

Once the Arduino is programmed, wired, and is provided power the server will be running. To restart the server, press the reset button on the Arduino.

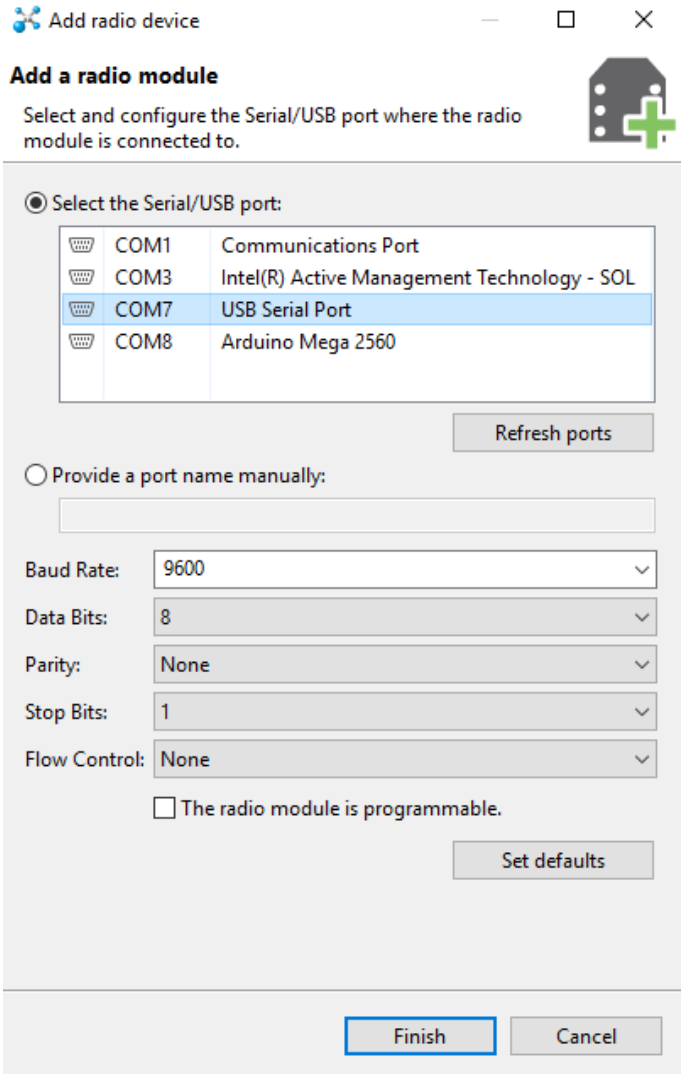
3.4 Setting up the XBee Client

3.4.1 Connecting to the XBee Module

Since the wireless communication is now configured on the XBee devices, the secondary XBee with the USB FTDI adapter only needs to be connected to the computer and opened as a COM device to view the data.

Once the XBee device is plugged in open XCTU and add a radio module:

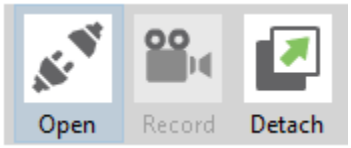
Connect as you did previously to configure the radio modules, select the device and press finish:



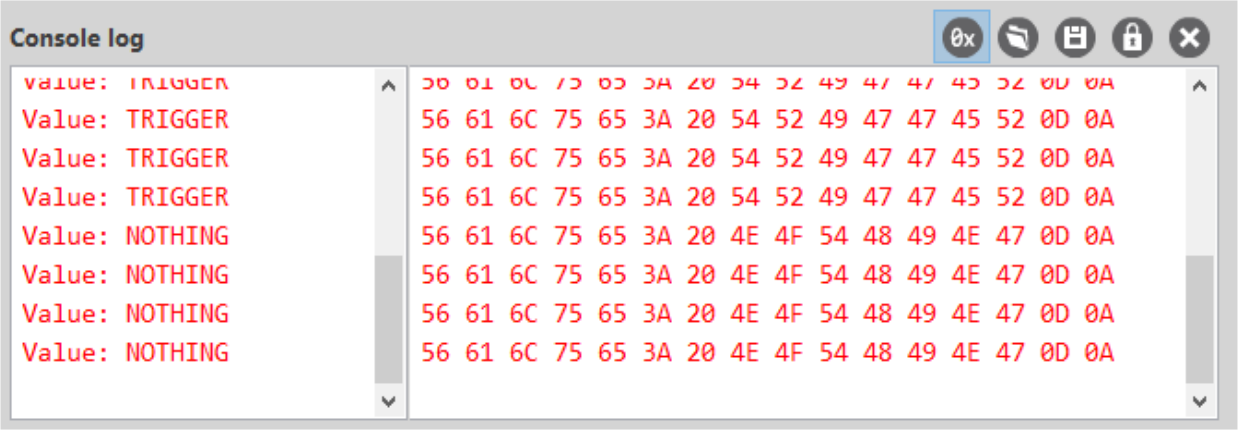
3.5 Reading the Data Wirelessly

Once the module is discovered and connected, open the terminal tab:

From the terminal tab, select open:



The client should now be connected to the server and should now be reading in data as shown below:



4 Study Questions & Deliverables

1. Provide a comprehensive report that demonstrates your completion of this laboratory assignment. Key sections to include in your report are "**Introduction and Background**", "**Methodologies**", and "**Results and Conclusions**" with inclusion of figures, tables, codes, and so forth in different sections.
2. What applications could this wireless server/client style architecture have?
3. Which devices that you have used might be using a technology similar to the XBee wireless modules?
4. What are some limitations of the XBee devices?

Instructor: Dr. Shayan (Sean) Taheri.

Note – Cheating and Plagiarism: Cheating and plagiarism are not permitted in any form and cause certain penalties. The instructor reserves the right to fail culprits.

Deliverable: All your responses to the assignment questions should be included in a single compressed file to be uploaded in the Gannon University (GU) – Blackboard Learn environment.

5 Equipment

Name	Quantity
Arduino Mega Microcontroller	1
USB-A to USB-B Cable	1
Male-to-Female Dupont Jumpers	12
Breadboard	1
XBee Wireless Module	2
XBee USB FTDI Breakout	2
USB-A to USB-Mini Cable	2
IR Distance Sensor	1