



Experiment 3.3

Student Name: Nabha Varshney

Branch: B.E.-CSE

Semester: 6th

Subject Name: Internet of Things Lab

UID: 20BCS4995

Section/Group: 20BCS704-A

Date of Performance: 02/05/2023

Subject Code: 20CSP-358

1. Aim:

Study the Implementation of Zigbee Protocol using Raspberry Pi/Arduino.

2. Objective:

- Learn about interfacing.
- Learn about IoT programming.

3. Requirements:

- 2 XBee Pro S2 Transceiver
- 2 UART to USB adapter board
- 1 USB Cord

4. Procedure:

XBee wireless transceivers provide a quick, easy way to add wireless communication to any system. This page will outline how to set up two XBee Pro Series 2 transceivers for communication with each other.

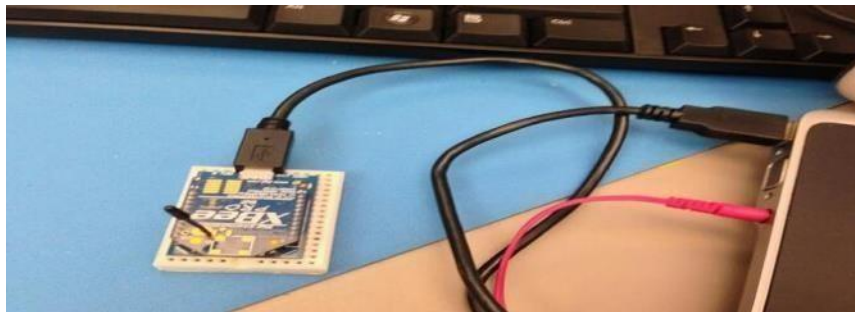
XBee Wireless Communication Setup

[Step 1: Download X-CTU Software](#)

The X-CTU software is free to download and provides a simple interface to configure and update your XBee transceivers. With this software firmware updates are a breeze and configuration is simple. The software can be downloaded from Digi's website.

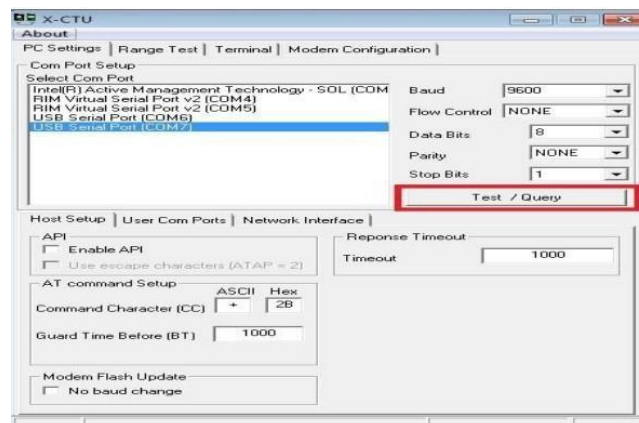
Step 2: Put together your XBee breakout board

The XBee transceivers have a 2mm pin spacing, which does not allow them to be plugged into a standard 0.1-inch breadboard. There are, however, several different breakout boards available that allow the transceiver to be inserted into a breadboard. The various adapter boards also allow for connection through USB or serial to your computer. The breakout board used here has a UART to USB conversion circuit and allows the XBee to be connected to the computer and XCTU software easily.



Step 3: Configure 1st XBee as a coordinator

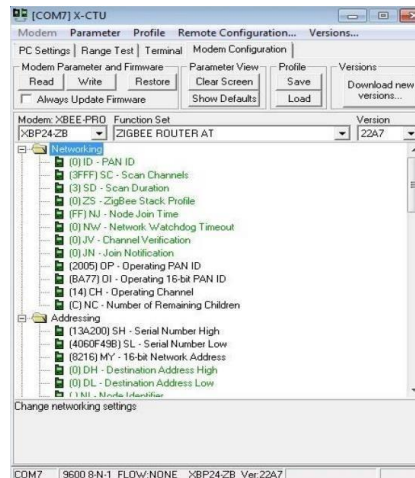
When opening the X-CTU software you should see a window like that shown. After selecting the proper COM port click the Test/Query button.



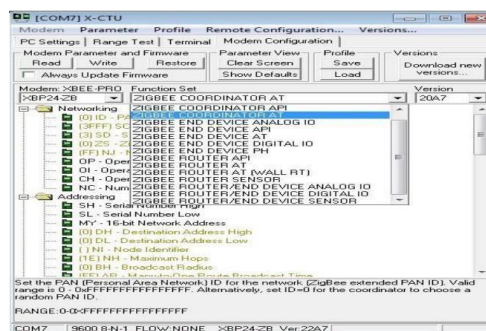
After selecting the Test/Query button, you should see a dialogue box like the one below. You will want to record the serial number shown, as you will need it in a couple minutes.



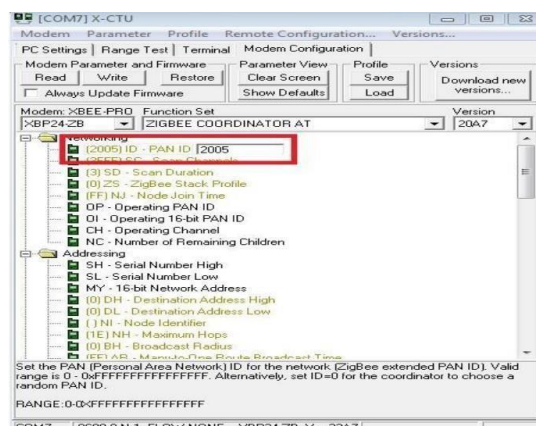
After recording the serial number you can click OK. Next, select the Modem Configuration tab at the top of the window. Once here, select the read button. This will bring up the current configuration for the connected XBee and will be similar to the following:



Once here, you want to select Zigbee Coordinator AT in the function set drop down menu.

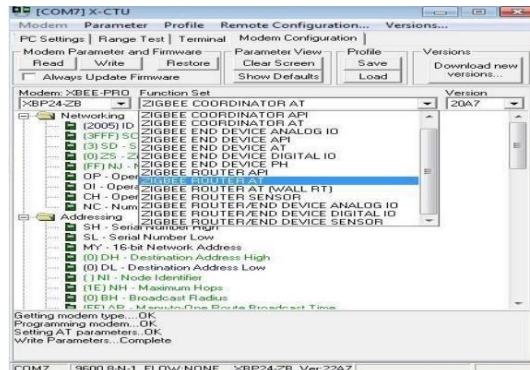


After selecting the coordinator function set, you will need to set the PAN ID. This can be any four-digit number and allows the XBees to distinguish between modules in their network and those from other networks. Once you have added the wanted PAN ID click the “Write” button. This will update and configure the XBee. Once this is done you can disconnect the XBee and plug in the second XBee.

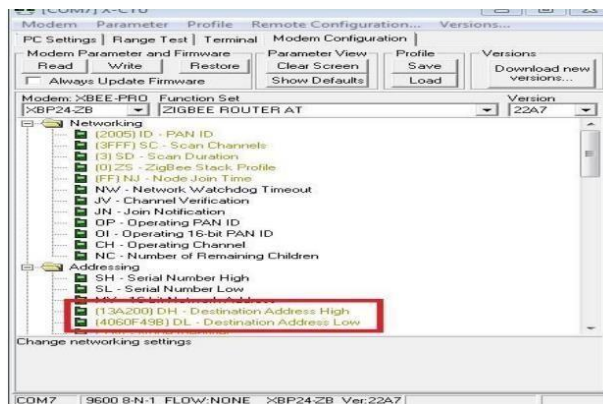


Step 4: Configure 2nd XBee as Router

To configure the 2nd XBee, you will follow the same process as for the coordinator with one difference. In the PC settings tab, again, click the “Test/Query” button and record the serial number. Then in the Modem configuration tab, click the read button to load the current configuration of the XBee and set the PAN ID to the same ID used for the coordinator. The only change will be the function set you choose. For the second XBee we will set this as Zigbee Router AT.



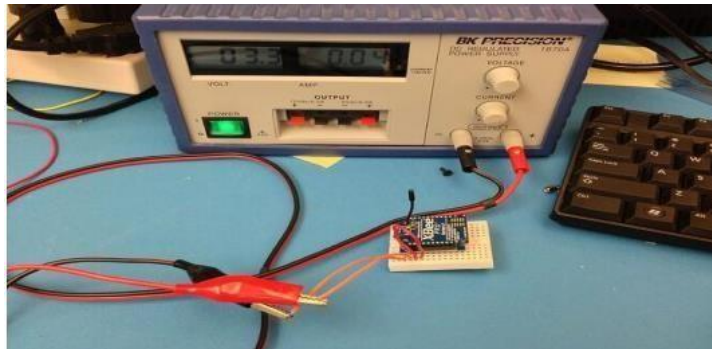
After setting the PAN ID and the function set for the router you will put the recorded serial numbers to use. Enter the first 6 digits of the coordinators serial number into the Destination Address High field and the rest of the serial number into the Destination Address Low field. Then select the “Write” button to update the configuration settings for the router. Once this is done updating you can disconnect the router and reconnect the coordinator to the computer.



After re-connecting the coordinator to the computer. You will again go into the Modem Configuration tab and click the read button. You will then want to set the Destination Address to the serial number of the Router XBee in the same manner as you just did for the Router. Once done you will again click the write button to update the coordinator XBee settings. After the write process is complete you are ready to use your XBees and communicate wirelessly

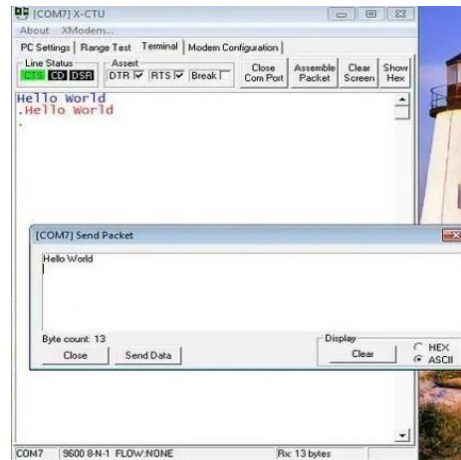
Step 5: Test the configuration

A simple test can be done to be sure the two XBees are communicating properly. You can connect either one of the XBees to the computer. Then, connect the second XBee to 3.3V power and connect the Dout and Din pins together. This will cause the XBee to automatically retransmit any data it receives.



When you have both XBees connected go to the Terminal tab in the X-CTU window. Whatever you type in the terminal window will appear in blue font and whatever is received will appear in red font. If the XBees are configured correctly every character you type should be mirrored in red.





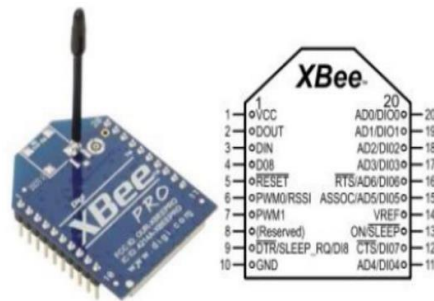
If everything you type is being reflected back in red, you are successfully transmitting and receiving with your XBees!! You are now ready to use them however you wish. You can connect them to any microcontroller and transmit data through the UART peripheral or you can connect each XBee to a different computer and have a chat application. There are many more possibilities that are now at your fingertips with your working XBees.

About Zigbee

Zigbee is a wireless communication protocol targeted for battery-powered devices (it has both low power and low cost). It generally operates in the 2.4GHz range (although there are geographic variations), and supports data ranges from 20 to 250 kbits/s. The transmission distance though, is small compared to the likes of LoRa. It is 10 to 100 m, whereas LoRa can transmit over a few kilometers. Another thing to note is that Zigbee communication doesn't work very well if there is no line of sight between transmitter and receiver.



These work with UART and therefore, it is fairly easy to interface them with Arduino. It is important to look at the pinout of XBee though, to understand which are the UART pins



The DOUT and DIN pins in the figure above are the UART pins (TX and RX). They can be connected to two digital pins of Arduino (if you plan to use SoftwareSerial), or else to pins 0 and 1 of Arduino respectively (if you plan to use HW Serial). Please note that you won't be able to read print statements from the Arduino on the Serial Monitor if you use Hardware Serial for Zigbee interface.

Configuring the XBee modules

The XBee modules (transmitter and receiver) need to be configured using the X-CTU Software. It can be downloaded from XTCU6.3, this software is provided by DigiKey, and they have given a detailed configuration guide. Therefore, there is no point of me reinventing the wheel here.

There's another one by Sparkfun that is adapted to the newer version of the X-CTU software.

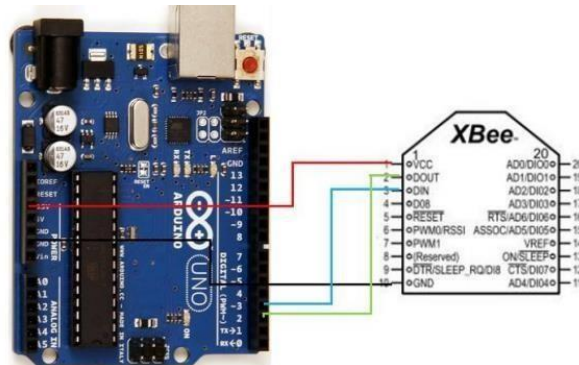
Please note that the two XBee modules that intend to communicate with each other should belong to the same series.

Here are a few things to note about the configuration –

- You will need a breakout board or an Explorer with a USB to UART converter for this configuration.
- The PAN ID (Personal Area Network ID) has to be the same for the devices that want to communicate with each other.
- One module needs to be set as the transmitter and the other as the receiver (this is determined by the CE field).
- Note the baud rate that you set. This will be used in the Arduino code, when configuring the Serial communication with XBee.

Circuit Diagram

Once your XBee is configured, you can connect it to the Arduino via the breakout/Explorer board. In that case, the pinout will be slightly different depending on which board/ Explorer you are using. Here we will assume you are connecting the XBee directly to Arduino Uno, in which case, the connections will be –



As you can see, we have connected Vcc to 3.3V on Arduino, GND to GND, DOUT (TX) to pin2, which will act as RX on the Arduino, and DIN (RX) to pin 3, which will act as TX on the Arduino.

5. Code/Program-

On the transmitting side, the code will be –

```
#include <SoftwareSerial.h>
SoftwareSerial xbeeSerial(2,3); //RX, TX

void setup() {
    Serial.begin(9600);
    xbeeSerial.begin(9600);
}

void loop() {
    if(Serial.available() > 0){
        char input = Serial.read();
        xbeeSerial.print(input);
    }
}
```

As you can see, whatever is sent by the user on the Serial Monitor is sent to the XBee module, and it will be received on the receiving side. The code for the receiving side is –

```
#include <SoftwareSerial.h>
SoftwareSerial xbeeSerial(2,3); //RX, TX

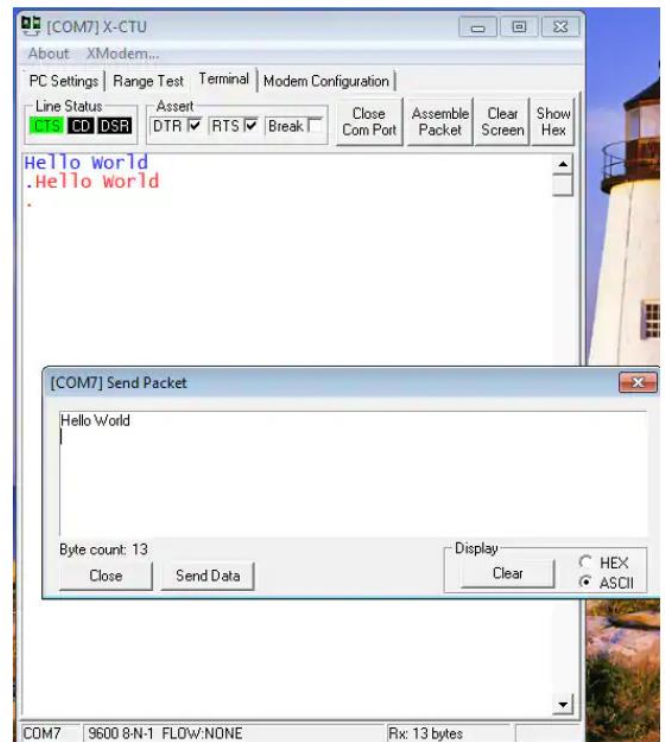
void setup() {
    Serial.begin(9600);
    xbeeSerial.begin(9600);
}

void loop() {
    if(xbeeSerial.available() > 0){
```



```
char input = xbeeSerial.read();
    Serial.print(input);
}
}
```

6. Output



Result-

In this experiment we learn how implement the Zigbee Protocol using Raspberry Pi/Arduino. And study different operations of Zigbee protocol. And how to put together the Xbee breakout board.

Learning Outcomes(What I have learnt):

- Learnt how to use Xbee Pro S2 transceiver with Arduino.
- Learnt how to use the software(X-CTU) using Zigbee protocol on Arduino.
- Learnt how to use Zigbee Protocol using code on Arduino.