

Introduction to Zigbee Mesh

Zigbee Point-point Communication

A Wireless Sensor Network (WSN) will be designed and implemented using Arduino / Raspberry pi and Xbee S2C module and IEEE 802.15.4/Zigbee standards. This whole network consists of end nodes, Router nodes and coordinator nodes. Xbee S2C modules will be used for this purpose.

Xbee S2Cs can connect and talk to each other over 2.4 GHz RF. As a start, we will be implementing a P-P communication where two nodes can talk to each other. One Xbee S2C module will be configured as End node and another one as coordinator. End node will be sending the data to the coordinator and coordinator will print that in a serial monitor.

At first the configuration of the modules. Connect two modules to your PC, open the XCTU software and hit the search devices. Add the selected device. After that update the Firmware. Select XB24C, Zigbee TH Reg and the newest firmware version. Reg stands for regular and TH stands for Through Hole

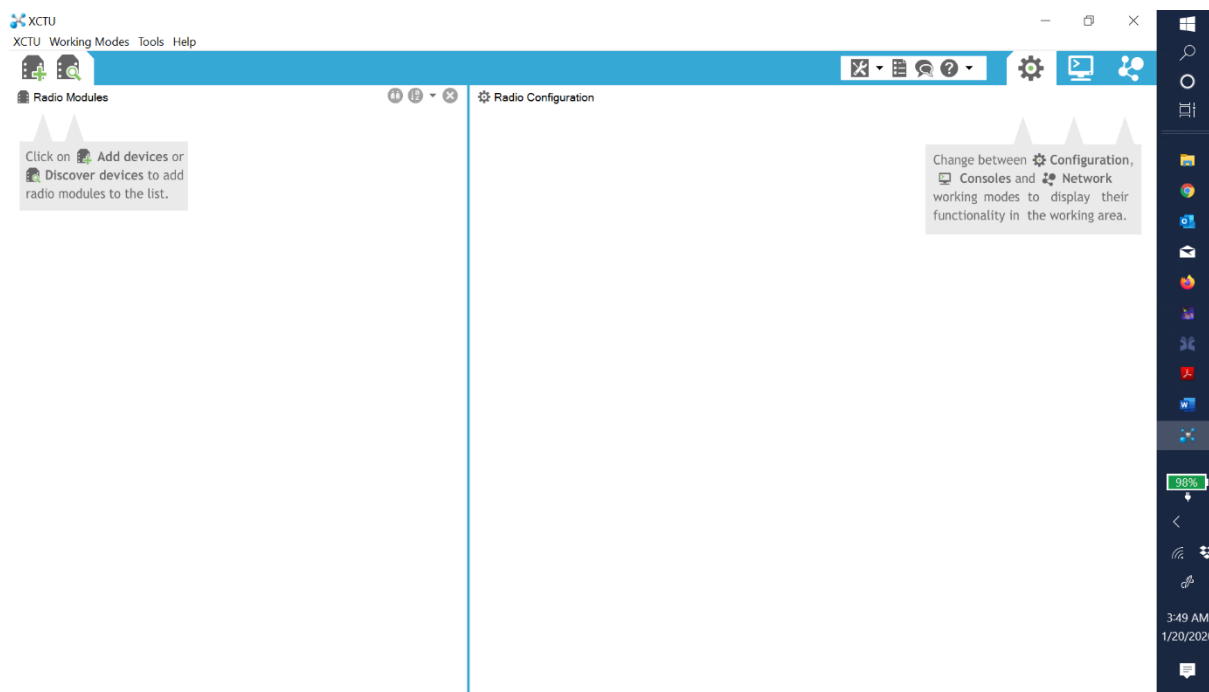


Figure 1 XCTU startup interface

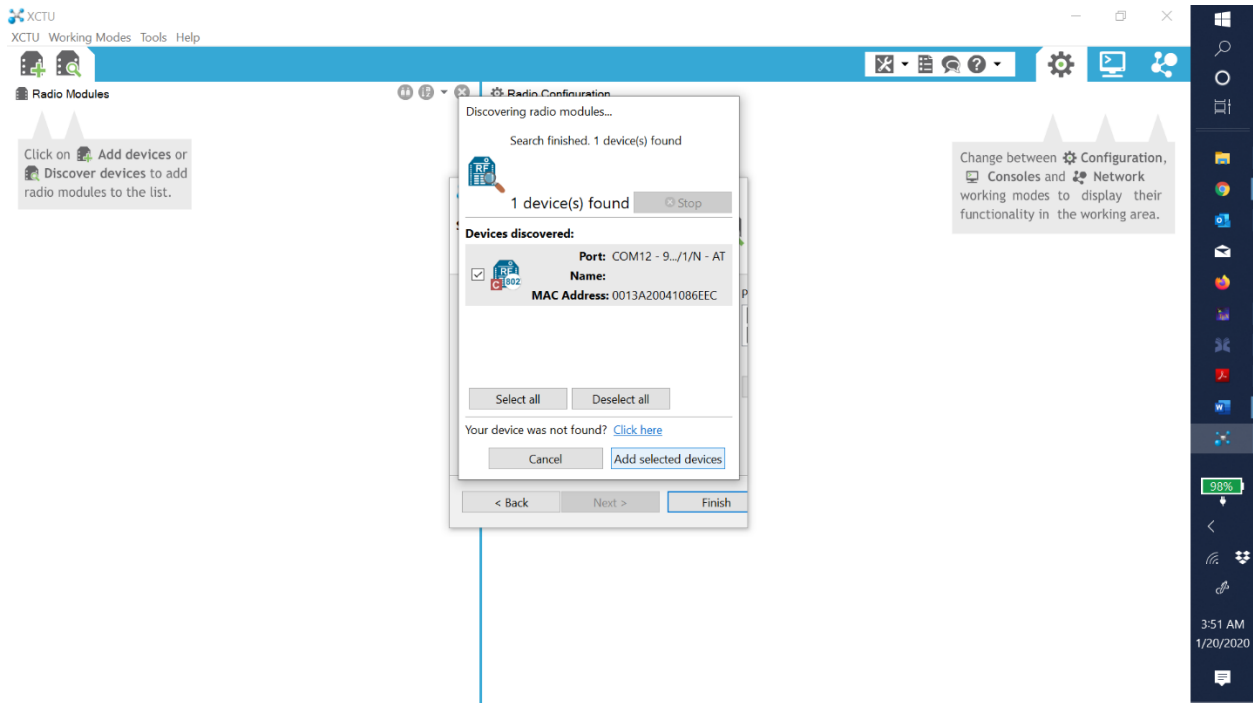


Figure 2 Search the connected device and add them

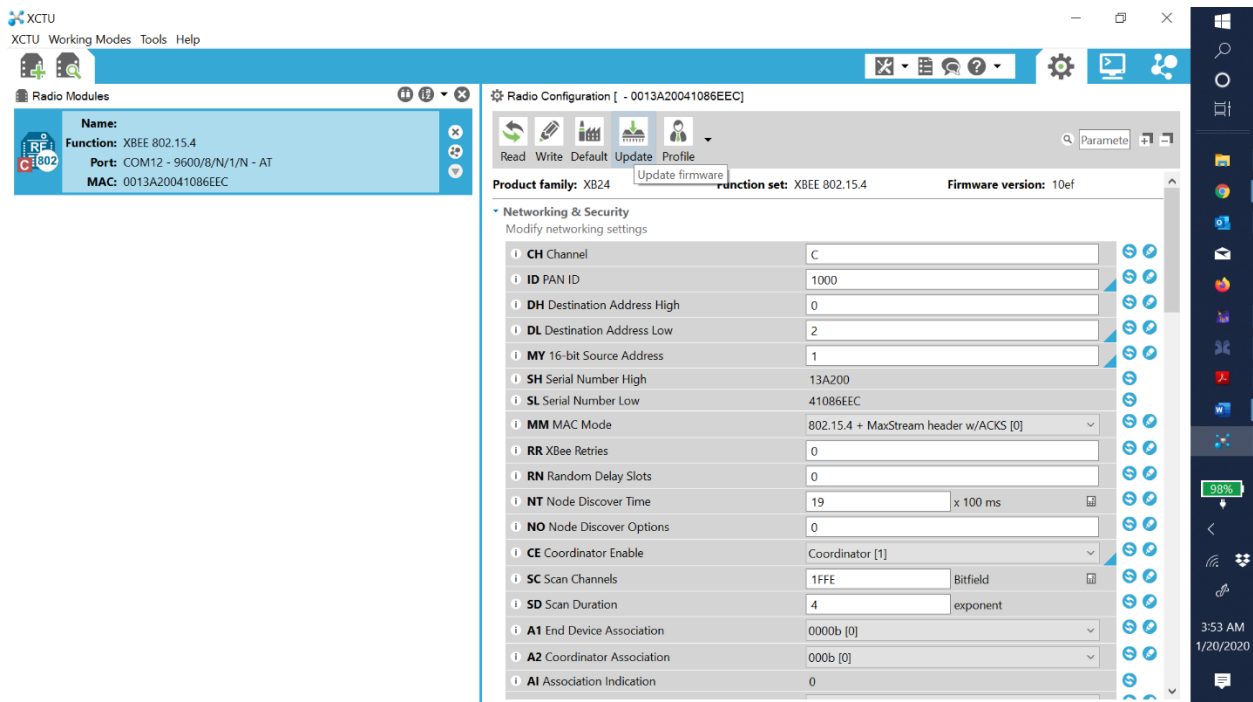


Figure 3 Update firmware as instructed above

Now, At first, we will make two Zigbee node talk to each other. We will use our PC as peripheral and chat with another Xbee using Zigbee point to point Communication over 2.4GHz frequency.

- There are two modes of Zigbee – 1. AT mode and 2. API Mode
- We will be working on AT mode.
- There are 3 types of nodes in Zigbee. 1. End Node 2. Router Node 3. Coordinator Node.

Now, we will set up one End Node and one Coordinator Node to communicate. The very basic settings of the End Node, Router and Coordinator are given below:

Nodes	Configuration
Coordinator	CE=1 SM=0
Router	CE=0 SM=0
End Node	CE=0 SM=1

Coordinator:

At first set the coordinator with CE=1 and SM=0. Give a unique PAN ID. This can be from 0 to FFFF hex. Make sure that, it is different from other groups. Now set the SC as 7FFF.

Give your preferred name like Coordinator 1 or something in NI.

For time being keep all other settings same as default.

Now write the changes.

End Node:

The end node settings are CE=0 and SM=1.

Most importantly, put the same unique PAN ID to the end node also. This enables the end node to find its coordinator. Keep the SC as 7FFF. Make JV channel verification enabled.

Keep other settings as it is. Now write the settings to the modules.

Now go to XCTU terminal. Close the serial communication for the modules.

Point to Point Communication:

Now, after closing the serial communication for both the end node and coordinator, type anything on the End node terminal. You can see that printed in RED in Coordinator terminal.

Now you can plug out your end node and plug in to a different remote PC. Open the XCTU terminal and close the serial communication without changing any of the settings.

You can test the communication to find whether it is okay or not. But if you type anything on the coordinator terminal, it will not be sent to the end node. Because the Coordinator is not set to the Broadcast mode.

Broadcast Mode:

Go to the settings of Coordinator and put Destination Address DH=0 and Destination Address DL =FFFF.

This will set the coordinator in broadcast mode.

If you write anything in coordinator terminal, it will be sent to all the nodes that are connected to the network.

Now try sending something like “Hello from Coordinator” and if you are successful then you can see that in your end node terminal in RED.

Connect Xbee to a Microcontroller:

Now we will be connecting our end node to a micro controller. In our case, it is Arduino nano. But, actually it can be any Arduino board or Raspberry Pi. This microcontroller is used to print some data continuously without any external human input. So, we can place that end node to a remote location within 50-60 m which is the range of our Xbee module.

At first program your Arduino nano to serial print “Hello from End Node 1” continuously with 5 sec delay. Run it on Arduino IDE to check the out put if it is okay or not.

If it is done, you are halfway through. Now connect the End node as like:

Arduino Xbee S2C

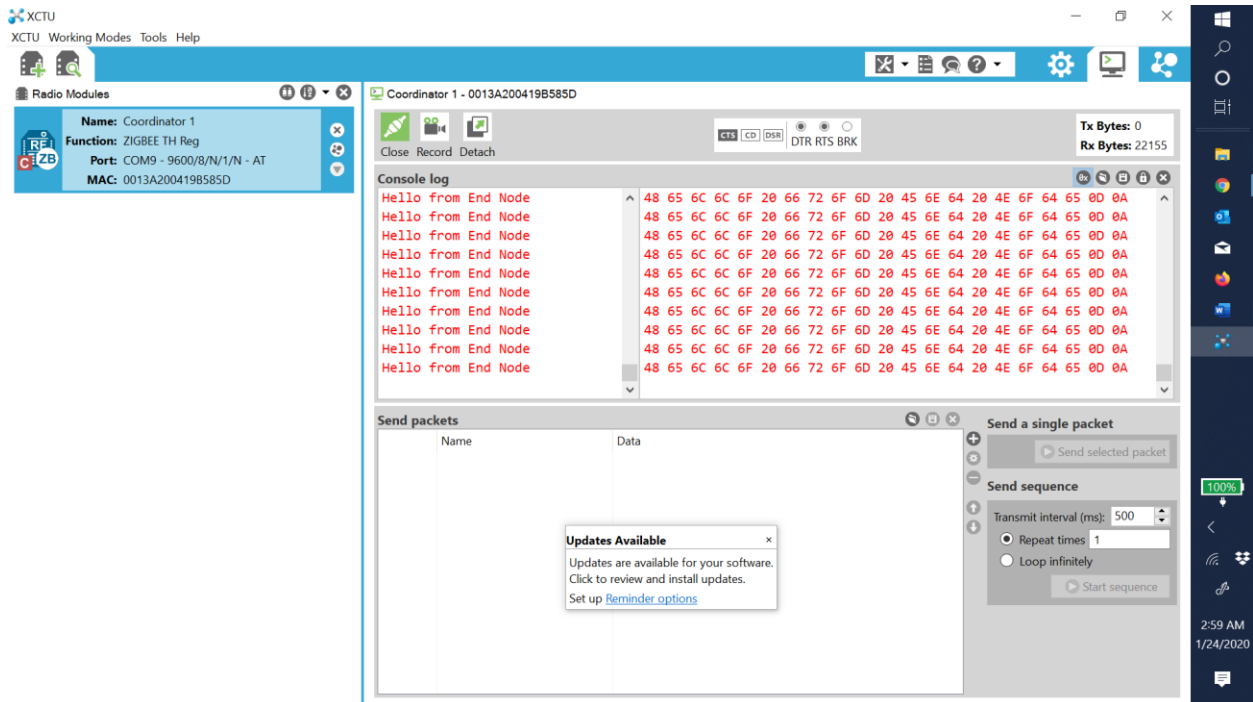
TX ➔ Din

RX ➔ Dout

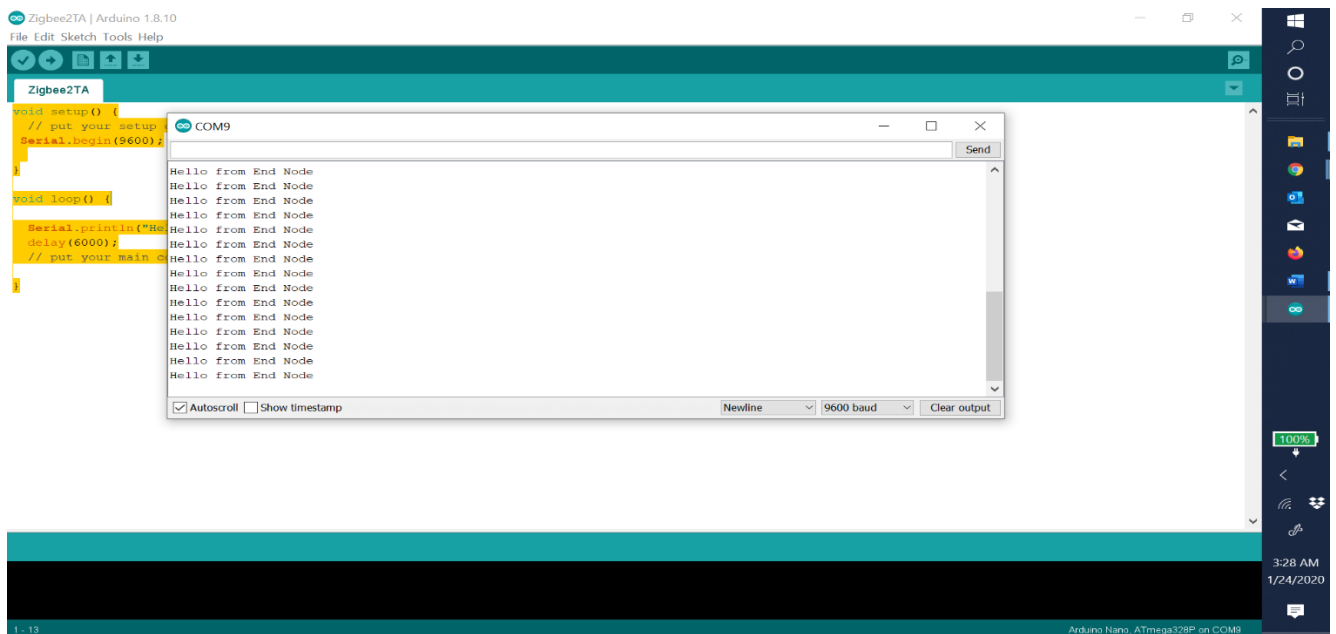
5V ➔ 5V_in

Gnd ➔ Gnd

Now power up the Connected Arduino with a power bank or any other 5V power source. And put it a bit away from the coordinator node. Now in the XCTU terminal you can see what you have coded to print.

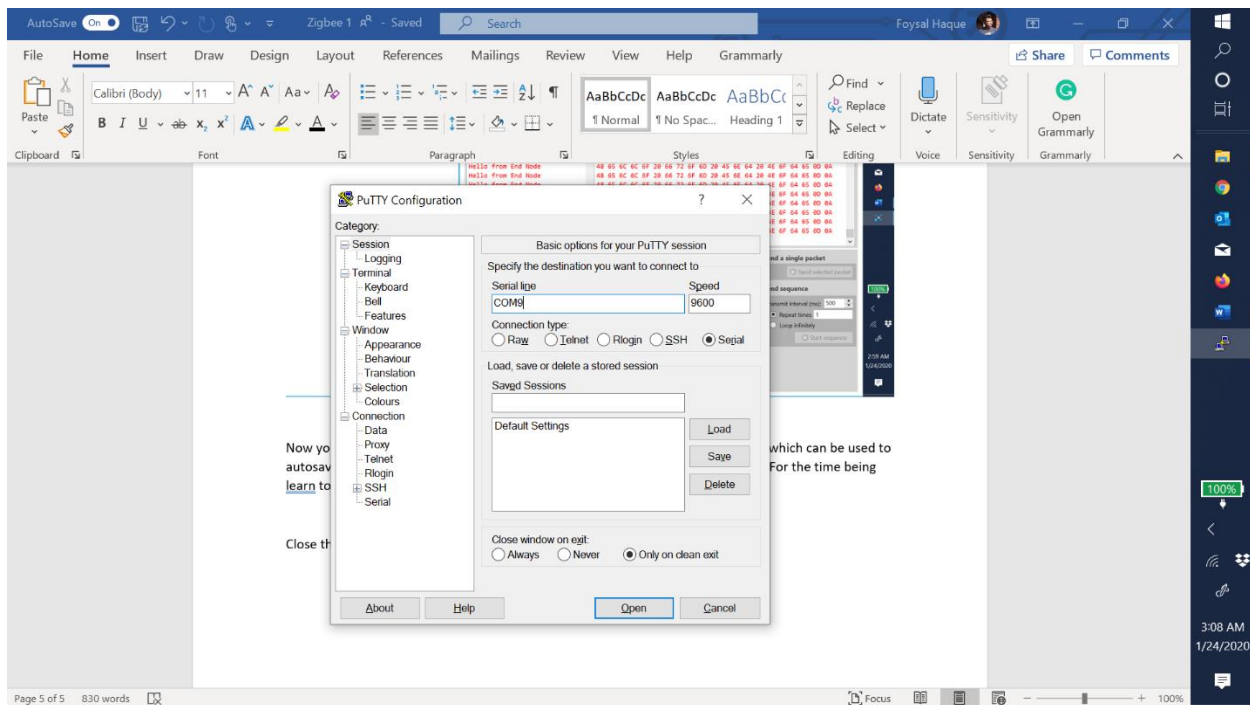


Now you can use any other Serial Monitor like serial monitor of Arduino for this purpose. Open Arduino IDE and select the 'Port' in which your coordinator is connected. It should be like COM1/COM5/COM8. Now if you open the serial monitor, you can see the same output as before.



We will be using Putty serial monitor which can be used to autosave the collected data from sensors in .CSV format. This will be used late on. For the time being, learn to use Putty. And see the same out put in that serial Monitor.

Close the XCTU & Arduino IDE and open Putty. Now select the option serial and put your serial line number as in the following figure and click ok.



You should be seeing the same output as like the XCTU terminal.

