ASSIGNMENT NO B3

TITLE : Write a network application for communication between two devices using Zigbee.

PROBLEM STATEMENT : Understanding and connectivity of Raspberry-Pi /Beagle board with a Zigbee module. Write a network application for communication between two devices using Zigbee.

LEARNING OBJECTIVE :

- To understand unctionalities of various single board embedded platforms fundamentals.
- Develop application for Communication between more raspberrypi hardware.

LEARNING OUTCOMES : The students will be able to

- Perform the connectivity with Raspberry-Pi, Beagle board, Arduino and other micro controller.
- Implement transmitter and receiver program using python by using zigbee device.

S/W PACKAGES AND HARDWARE APPARATUS USED : Python, two Raspberrypi devices, Zigbee device, Latest Version of 64 bit Operating Systems,Open Source Fedora-GHz. 8 G.B. RAM, 500 G.B.
HDD, 15"Color Monitor, Keyboard, Mouse.

THEORY :

- ZigBee Communication Using Raspberry Pi

ZigBee is a communication device used for the data transfer between the controllers,computers, systems, really anything with a serial port. As it works with low power consumption, the transmission distances is limited to 10–100 meters line-of-sight. ZigBee devices can transmit data over long distances by passing data through a mesh network of intermediate devices to reach more distant ones. ZigBee is typically used in low data rate applications that require long battery life and secure networking. Its main applications are in the field of wireless sensor network based on industries as it requires short-range low-rate wireless data transfer. The technology defined by the ZigBee specification is intended to be simpler and less expensive than other wireless networks.

- Zigbee Technology

Zigbee communication is specially built for control and sensor networks on IEEE 802.15.4 standard for wireless personal area networks (WPANs), and it is the product from Zigbee alliance. This communication standard defines physical and Media Access Control (MAC) layers to handle many devices at low-data rates. These Zigbee‟s WPANs operate at 868 MHz, 902-928MHz and 2.4 GHz frequencies. The date rate of 250 kbps is best suited for periodic as well as intermediate two way transmission of data between sensors and controllers.

Zigbee is low-cost and low-powered mesh network widely deployed for controlling and monitoring applications where it covers 10-100 meters within the range. This communication

system is less expensive and simpler than the other proprietary short-range wireless sensor networks as Bluetooth and Wi-Fi. Zigbee supports different network configurations for master to master or master to slave communications. And also, it can be operated in different modes as a result the battery power is conserved. Zigbee networks are extendable with the use of routers and allow many nodes to interconnect with each other for building a wider area network

- Zigbee Architecture
  Physical Layer: This layer does modulation and demodulation operations up on transmitting and receiving signals respectively. This layer‟s frequency, date rate and number of channels are given below.
  MAC Layer: This layer is responsible for reliable transmission of data by accessing different networks with the carrier sense multiple access collision avoidance (CSMA). This also transmits the beacon frames for synchronizing communication.
  Network Layer: This layer takes care of all network related operations such as network setup, end device connection and disconnection to network, routing, device configurations, etc.
  Application Support Sub-Layer: This layer enables the services necessary for Zigbee device object and application objects to interface with the network layers for data managing services. This layer is responsible for matching two devices according to their services and needs.
  Application Framework: It provides two types of data services as key value pair and generic message services. Generic message is a developer defined structure, whereas the key value pair is used for getting attributes within the application objects. ZDO provides an interface between application objects and APS layer in Zigbee devices. It is responsible for detecting, initiating and binding other devices to the network.Zigbee Operating Modes and Its Topologies.

Zigbee two way data is transferred in two modes: Non-beacon mode and Beacon mode. In a beacon mode, the coordinators and routers continuously monitor active state of incoming data hence more power is consumed. In this mode, the routers and coordinators do not sleep because at any time any node can wake up and communicate. However, it requires more power supply and its overall power consumption is low because most of the devices are in an inactive state for over long periods in the network.

CONCLUSION :

In this way we have studies and successfully performed communication between two devices using Zigbee.

Process for connecting two devices:

Hardware Requirements

1.1 x Raspberry Pi with Raspbian Installed in it
2.2 x XBee Pro S2C modules (any other model can be used)
3.1 x XBee explorer board (optional)
4.1 x Xbee Breakout board (optional)
5.USB cables

6.LEDs

It is assumed that your Raspberry Pi is already flashed with an operating system. If not, follow the Getting started with Raspberry Pi tutorial before proceeding. Here we are using **Rasbian Jessie installed Raspberry Pi 3**.

Here External Monitor using HDMI cable is used as display to connect with Raspberry Pi. If you don't have monitor, you can use SSH client (Putty) or VNC server to connect to Raspberry pi using Laptop or computer. Learn more about setting up Raspberry Pi headlessly here.

Configuring XBee Modules using XCTU

As we have learnt in previous tutorial of ZigBee Introduction that the XBee module can act as a Coordinator, Router or an End device but it need to be configured to work in desired mode. So, before using the **XBee modules with Raspberry Pi**, we have to configure these modules using XCTU software.

To connect XBee module with the laptop, a USB to serial converter or specifically designed explorer board is used. Just hook up the XBee module to the Explorer board and plug it with the laptop using USB cable.

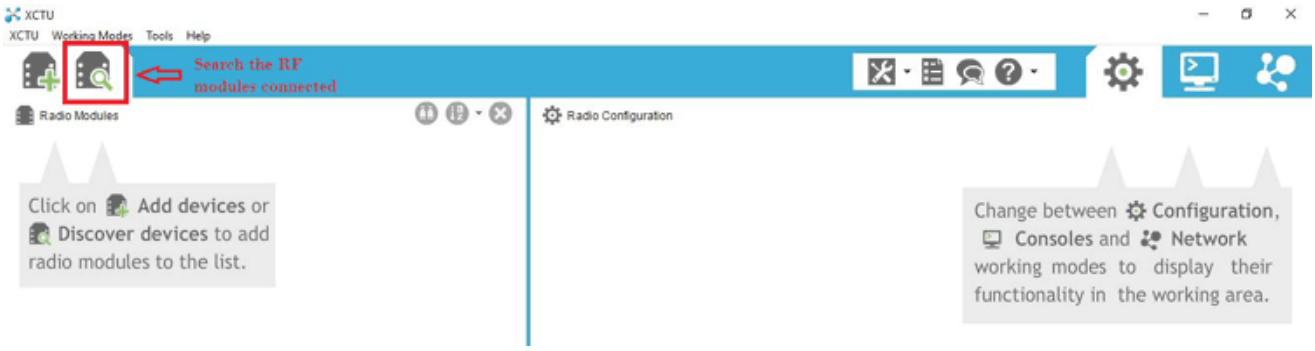
If you don't have any converter or explorer board, then an **Arduino board can be used as a USB to serial device** which can easily communicate with the XBee and laptop. Just upload blank sketch in Arduino board and now it can behave like a USB to Serial converter.

Configuring XBee Modules:

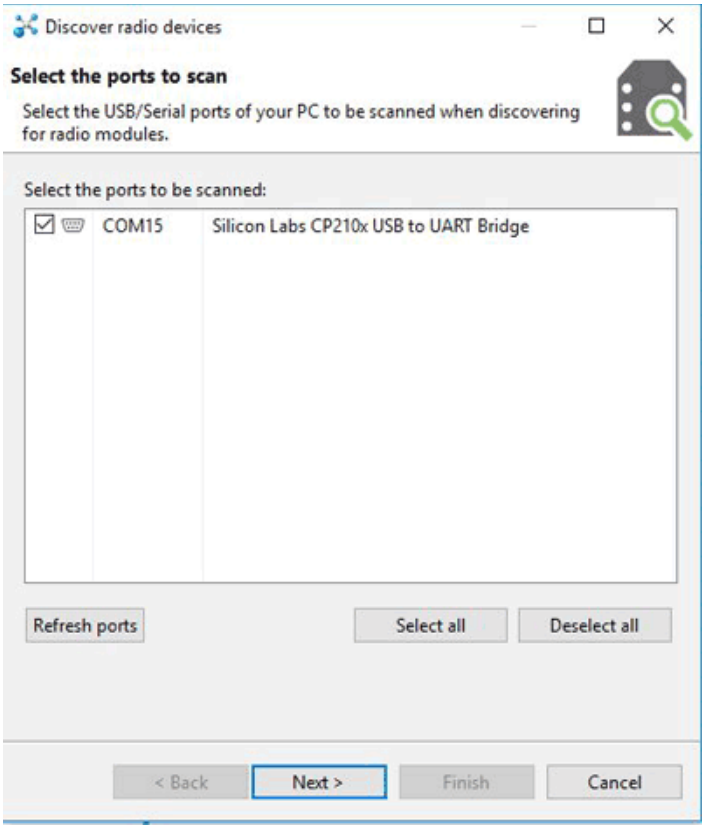
Here in this tutorial, an **Explorer board is used to configure the XBee modules**.

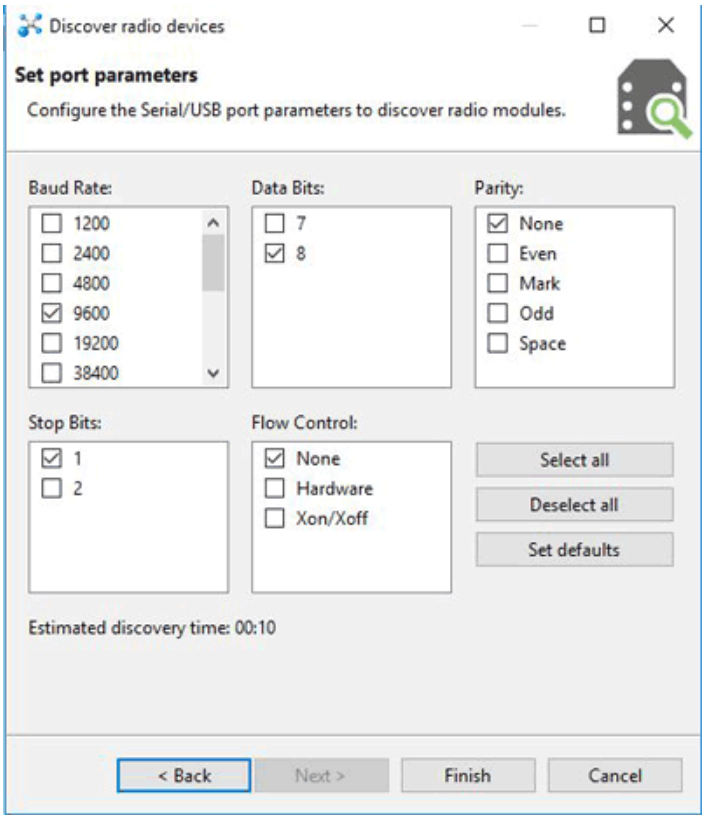
Download the XCTU software from this link and install it. After downloading and installing the XCTU software, open it and make sure your XBee module is properly connected. Check the COM port of the Arduino board in device manager.

1. Now, click on the search button. This will show you all the RF devices connected with your laptop. In our case, it will show only one XBee module.

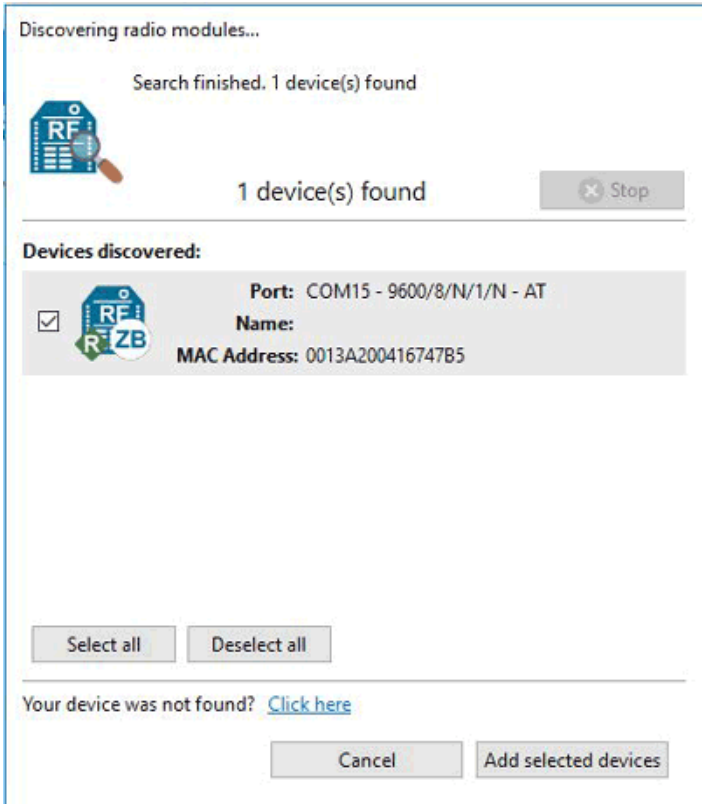


2. Select the Serial port of the Explorer board/Arduino board and click on Next.



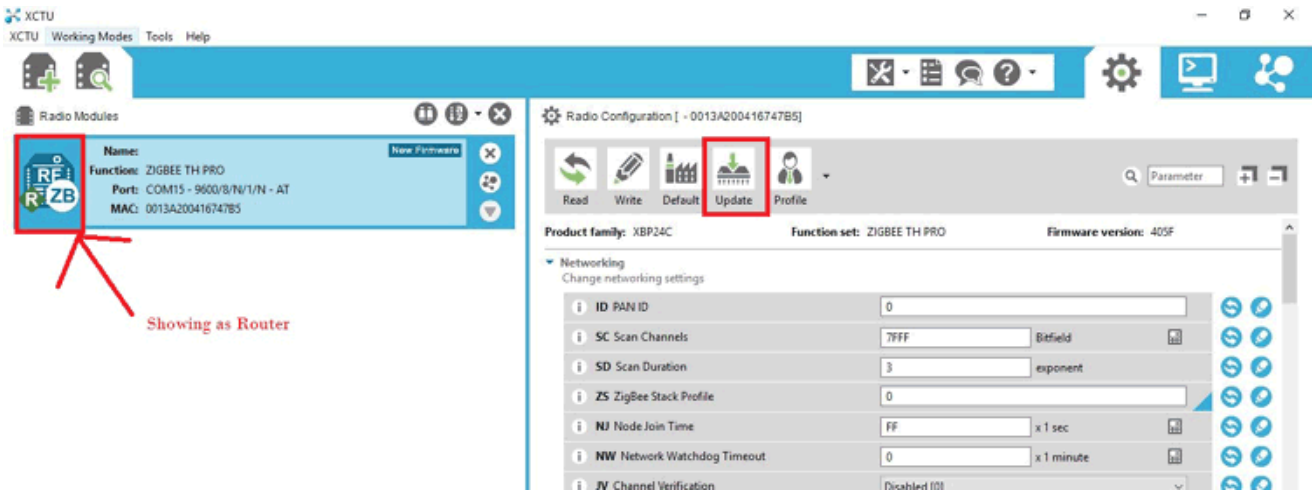


4. Select the Discovered device and click on **Add selected device**. This process will add your XBee module to XCTU dashboard.

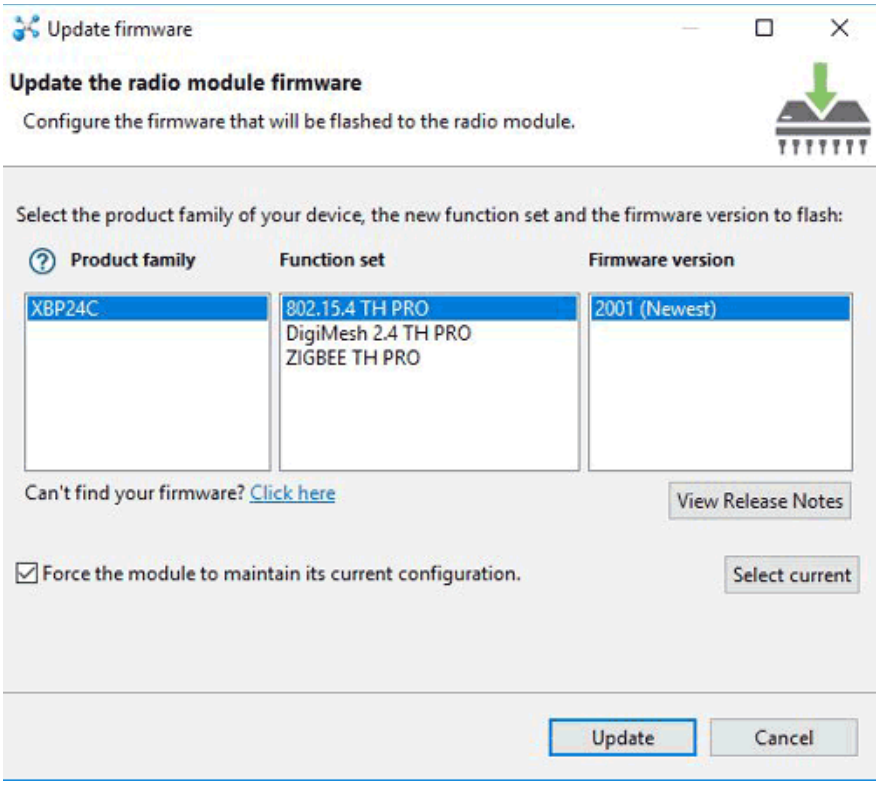


5. Now, you can configure your XBee module in this window. You can use either AT commands or put the data manually. As you can see, there is **R showing on the left panel which means XBee is in router mode. We have to make it Coordinator for the transmitter part.**

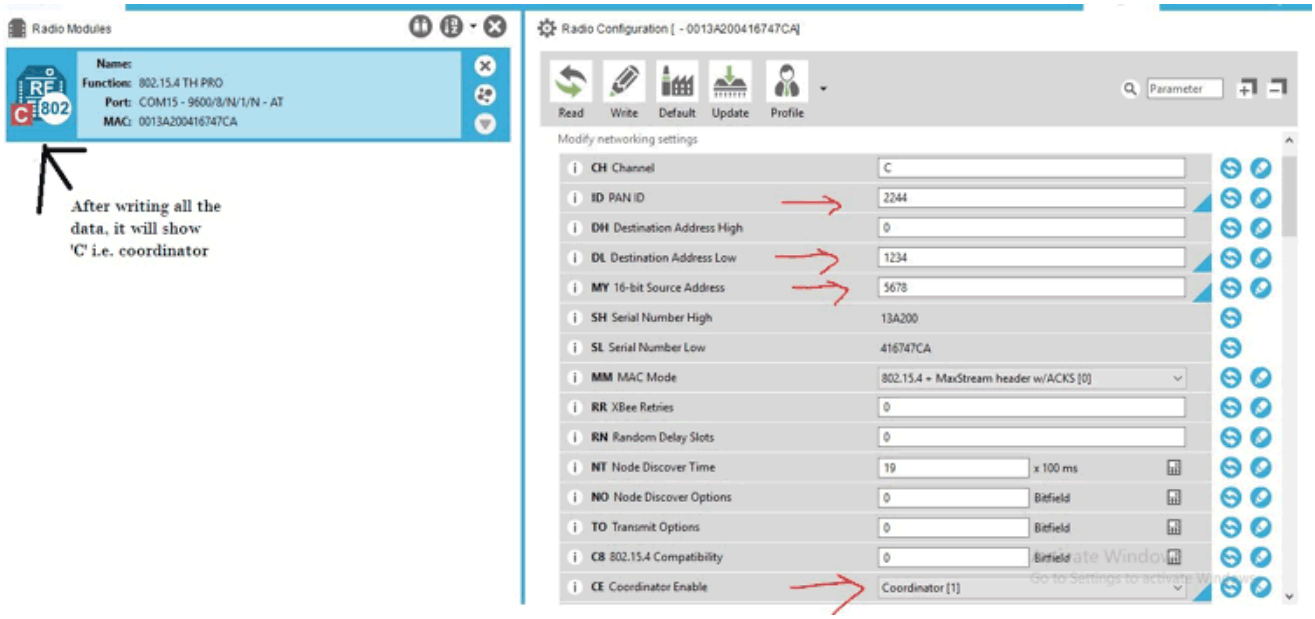
First, update the Firmware by clicking on the Update firmware.



6. Choose the Product family of your device which is available on back of your XBee module. Select function set and firmware version as highlighted below and **click on Update.**



7. Now, you have to give ID, MY and DL data to make connection with other XBee. ID remains same for both the modules. Only MY and DL data interchange i.e. **MY for the receiver XBee becomes DL of the transmitter XBee (coordinator) and DL for the receiver XBee becomes MY of the transmitter XBee.** Make CE as **Coordinator** and then hit the **Write** button. As shown below.



	ATDL	***ATMY***	*ATID*
XBee 1 coordinator	*1234*	*5678*	*2244*
XBee 2 end device	*5678*	*1234*	*2244*

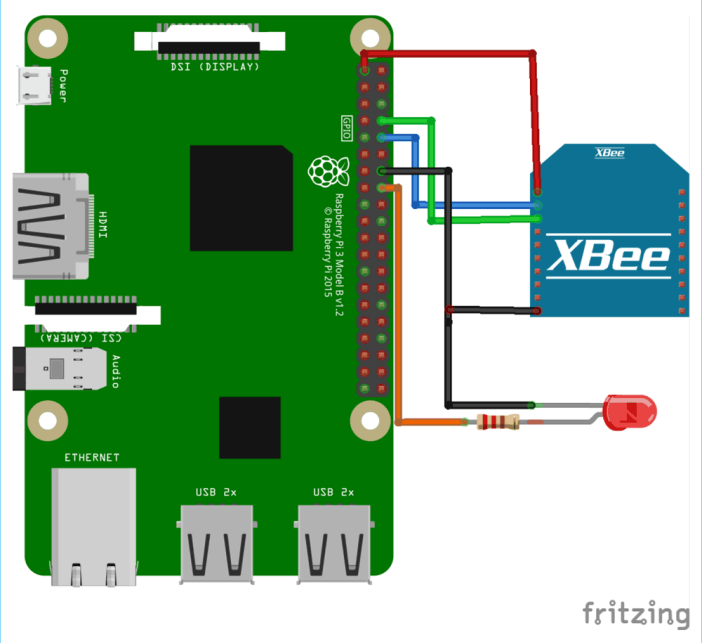
8. After writing the above data to the transmitter part, plug out it from the explorer board and plug in the second XBee module in it. Repeat the same process as above only changes are the DL, MY,

and CE. **As we will make the second XBee as End device so in CE drop down menu, select the End device and hit the Write button.**

9. Now, our XBee modules are ready to interface with the Raspberry Pi. We will **connect the transmitter XBee to the laptop and receiver XBee with the Raspberry Pi.** Then give commands to the receiver part using laptop. laptop.

Circuit Diagram for Receiver Part

Connections for **interfacing ZigBee module with Raspberry PI** are shown in the circuit diagram.



OUTPUT:

Testing the wireless XBee communication using Raspberry Pi

Now, we all set to **test our XBee transmitter and receiver**. To give command to the transmitter part, we will use XCTU's console terminal. Click on the Console icon near the settings option. Then, click on Open button to connect the XBee to the laptop.

Enter 'a' in Console log

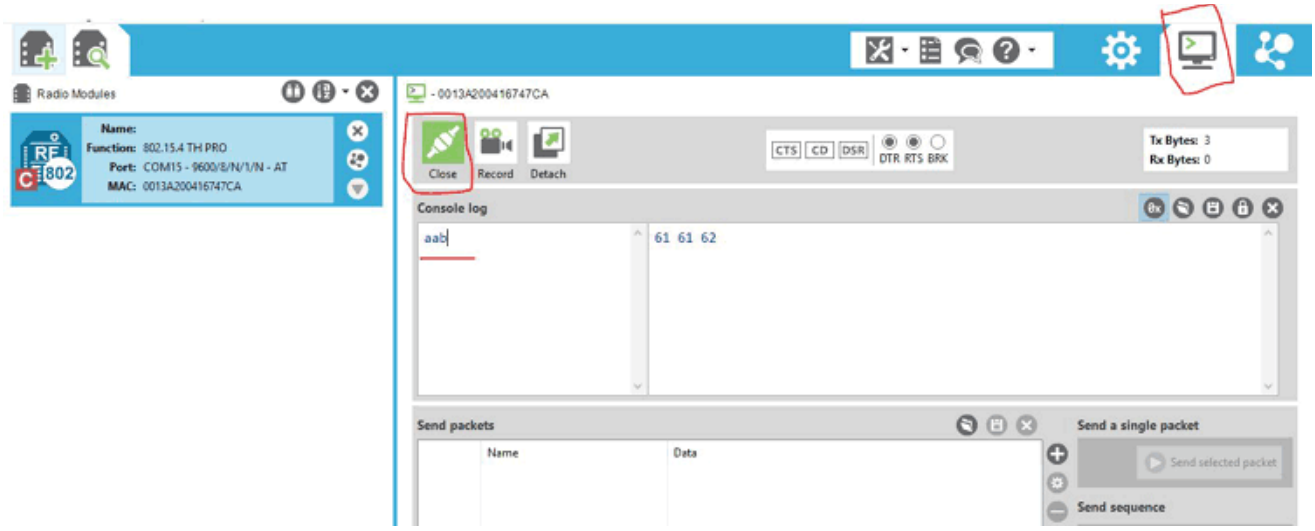
Code

```
#!/usr/bin/env python
import time
import serial
import RPi.GPIO as GPIO
```

```
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(23,GPIO.OUT)
ser = serial.Serial(
    port='/dev/ttyS0',
    baudrate = 9600,
    parity=serial.PARITY_NONE,
    stopbits=serial.STOPBITS_ONE,
    bytesize=serial.EIGHTBITS,
    timeout=1
)
counter=0

while 1:
    #ser.write(str.encode('Write counter: %d \n'%(counter)))
    #time.sleep(1)
    #counter += 1
    x=ser.readline().strip()
    print(x)
    if x == 'a':
        GPIO.output(23,GPIO.HIGH)
        time.sleep(3)
    else:
        GPIO.output(23,GPIO.LOW)
```

. You will see that LED will turn ON for 3 seconds and then it turn OFF.



In this way you can also connect the transmitter XBee to the Arduino board as described in the and make the Raspberry Pi and Arduino to communicate with each other.