Steps on how to set-up the Arduino Uno, Raspberry Pi and Configure the Xbee S2 transceiver modules in AT mode

Contents

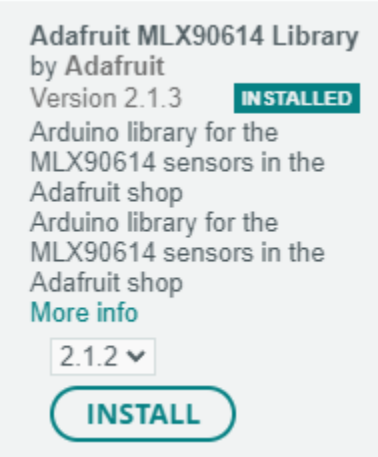
Arduino Set-up

Step 1)

Download and install the current version of the Arduino IDE from https://www.arduino.cc/en/software

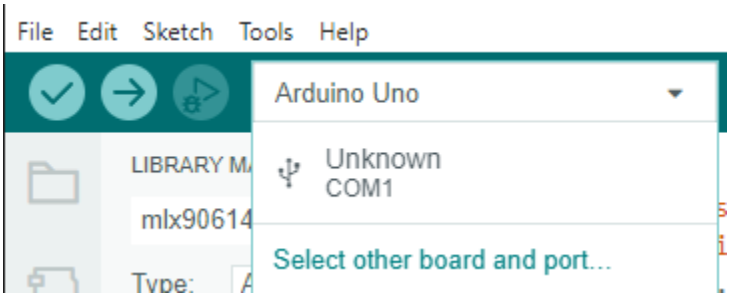
Step 2)

After opening the Arduino IDE, navigate to the Library Manager under Tools > Manage Libraries. Search for MLX90614 and install the “Adafruit MLX90614 Library by Adafruit” using the current version.



Step 3)

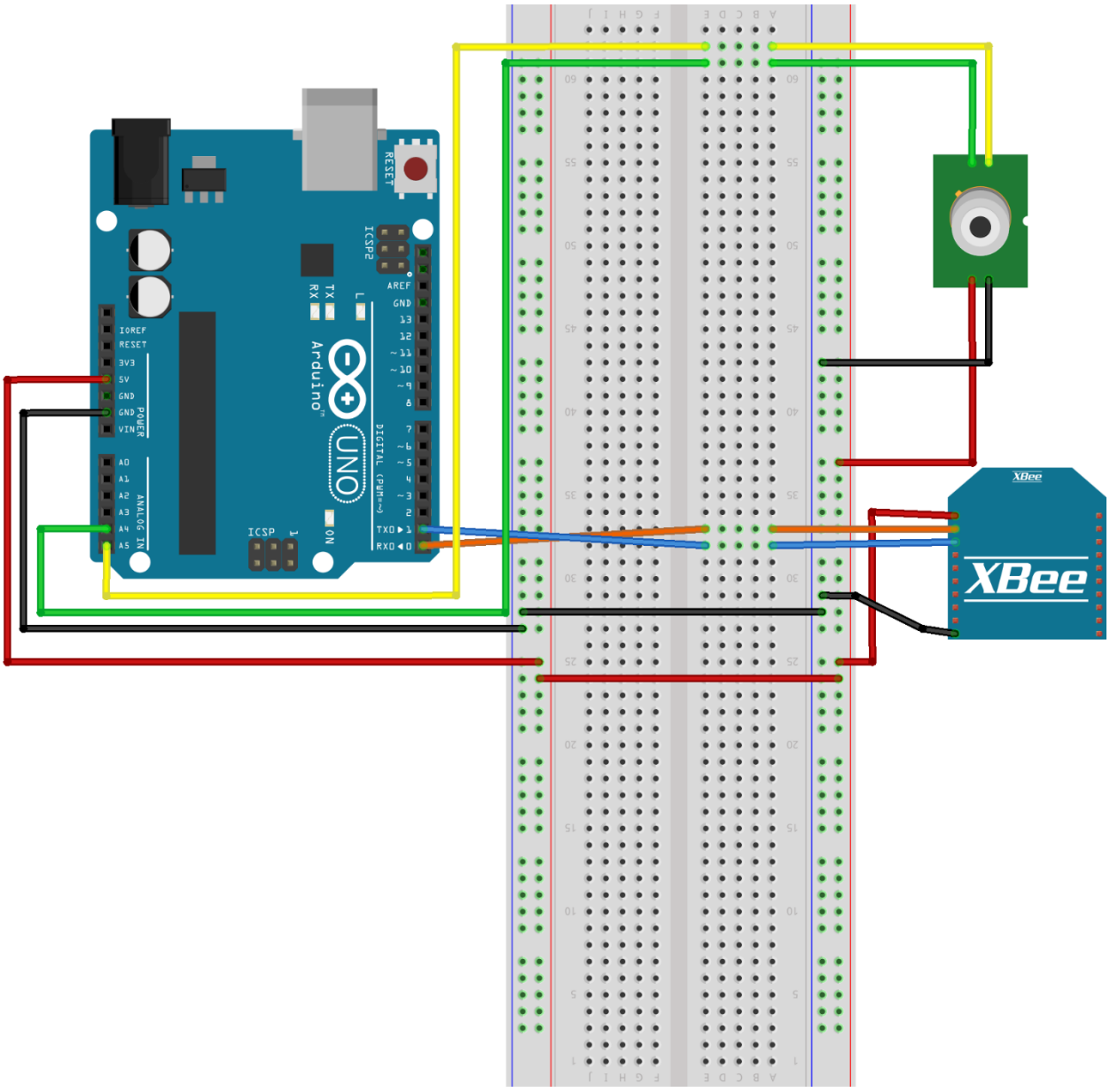
Plug the Arduino uno into the PC via included usb cable. Head to device manager and look under the COM ports to find out the COM port the Arduino uno is using. Back in the IDE select the correct COM port the uno is connected to

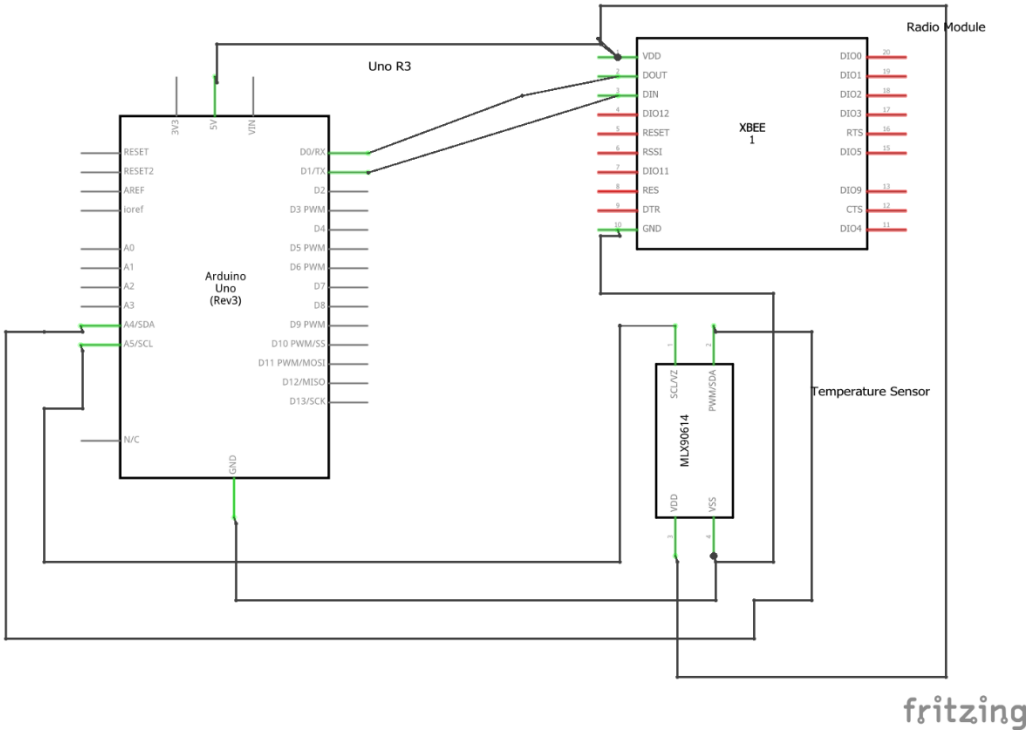


Step 4)

Upload the opened sketch to the board and after it has successfully uploaded, unplug the uno from the computer

Step 5) Following the below diagram, connect the wires following the connections.





Step 6) Finally, connect the Arduino uno to a power source, such as either a battery pack or back to the computer and the uno will begin reading and broadcasting temperature values to a listening device

Raspberry Pi Set up

For the sake of this set-up, this set-up will follow under the assumption the reader has already followed the initial raspberry set-up instructions that come with a brand new pi and will assume the reader has already installed the pi OS, set-up wifi, and are able to connect to the pi board over SSH.

Step 1)

First connect to the board through your preferred method of SSH. And then enter the following command

sudo raspi-config

Navigate to interface options > enter, Navigate to serial > enter, and select no to the login shell and then yes to enabling the serial hardware port.

Step 2)

Next we will copy the two included files over into a new folder we create. Open a new terminal or PowerShell window and navigate to the directory where the files we will transfer are located.

Using the following commands

First create a folder using the following command

mkdir Dawson

and then following the scp command in this format **scp** [source files] [user]@[host]:[path]

scp Receiver.py Emailer.py pi@[ip_of_pi]:Dawson/

Step 3)

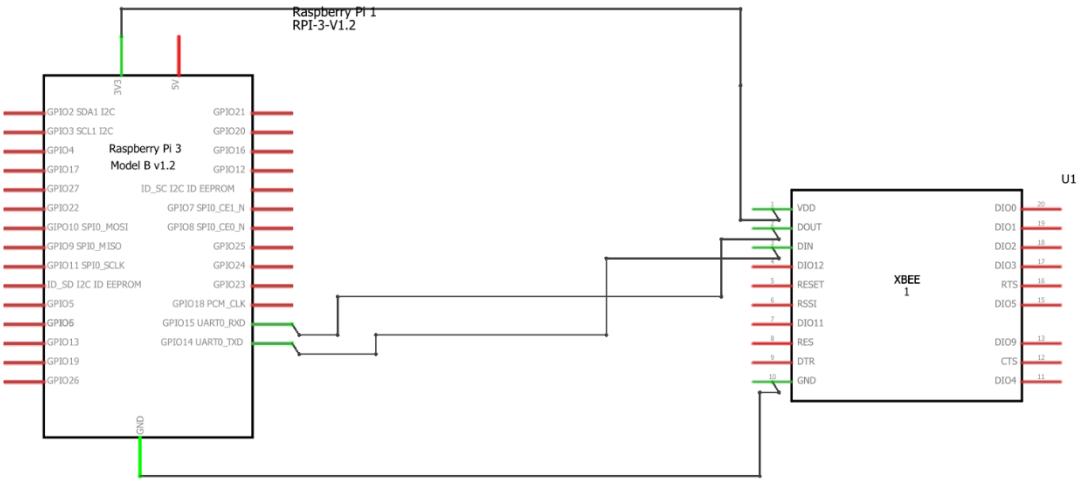
To run the scripts we have copied over, back in our SSH connection to the pi, navigate to our newly created Dawson folder and check to ensure the files were copied over using the **ls** command. Once we have confirmed that they were successfully copied over, we can then run the command

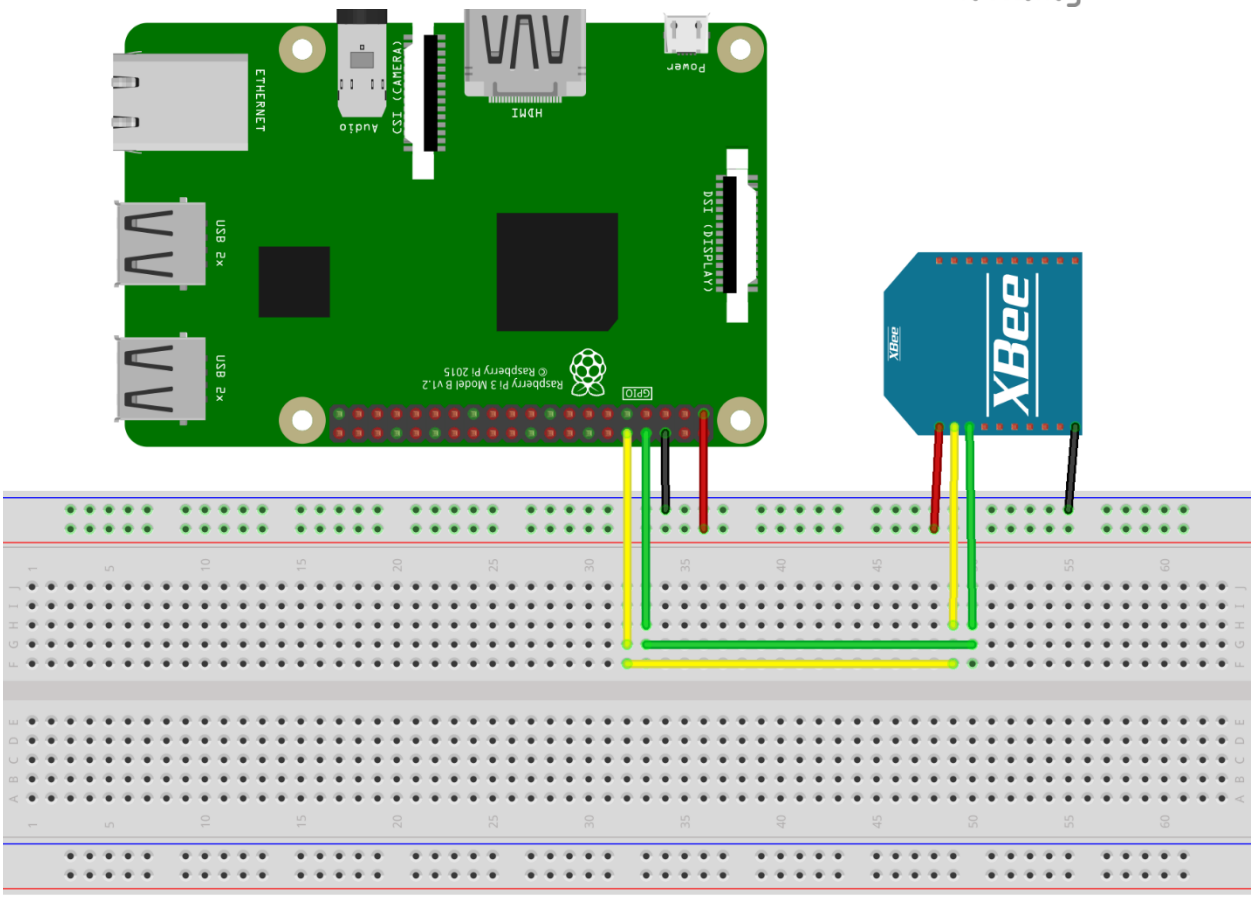
python3 Receiver.py to execute the script when we are ready.

Step 4)

The final step is connecting the pi and xbee module following the below schematic and diagram.

Once that is finished you are ready to run the script and start receiving any data being transferred from the Arduino uno


Raspberry Pi 1
RPI-3-V1.2
Raspberry Pi 3
Model B v1.2
U1
XBEE
1
fritzing


ETHERNET
Audio
CSI (CAMERA)
HDMI
Power
USB 2x
USB 2x
DSI (DISPLAY)
GPIO
XBee
fritzing

Xbee S2 Configuration with explorer boards

This part of the readme will cover the configuration of the xbee modules so that they can receive and send data through AT mode. The steps outlined below were taken and followed from here: https://www.electronicwings.com/sensors-modules/xbee-module

Setup will require 1 explorer board and 1 xbee module at minimum

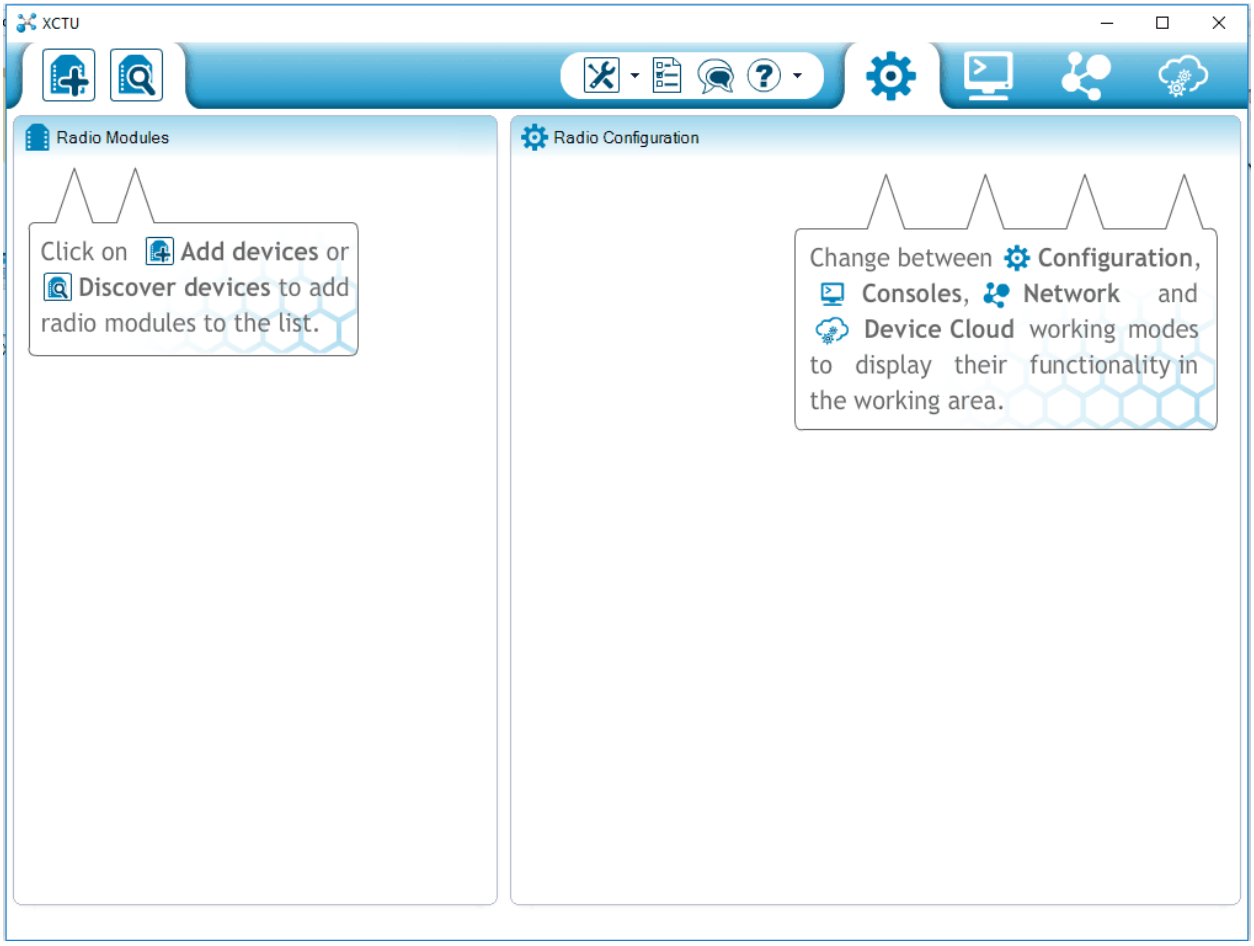
Step 1)

Download and install X-CTU https://www.digi.com/products/embedded-systems/digi-xbee/digi-xbee-tools/xctu#productsupport-utilities

With one of the xbee modules plugged into the explorer board, use the accompanying usb adapter cable to connect it to the computer.

Step 2)

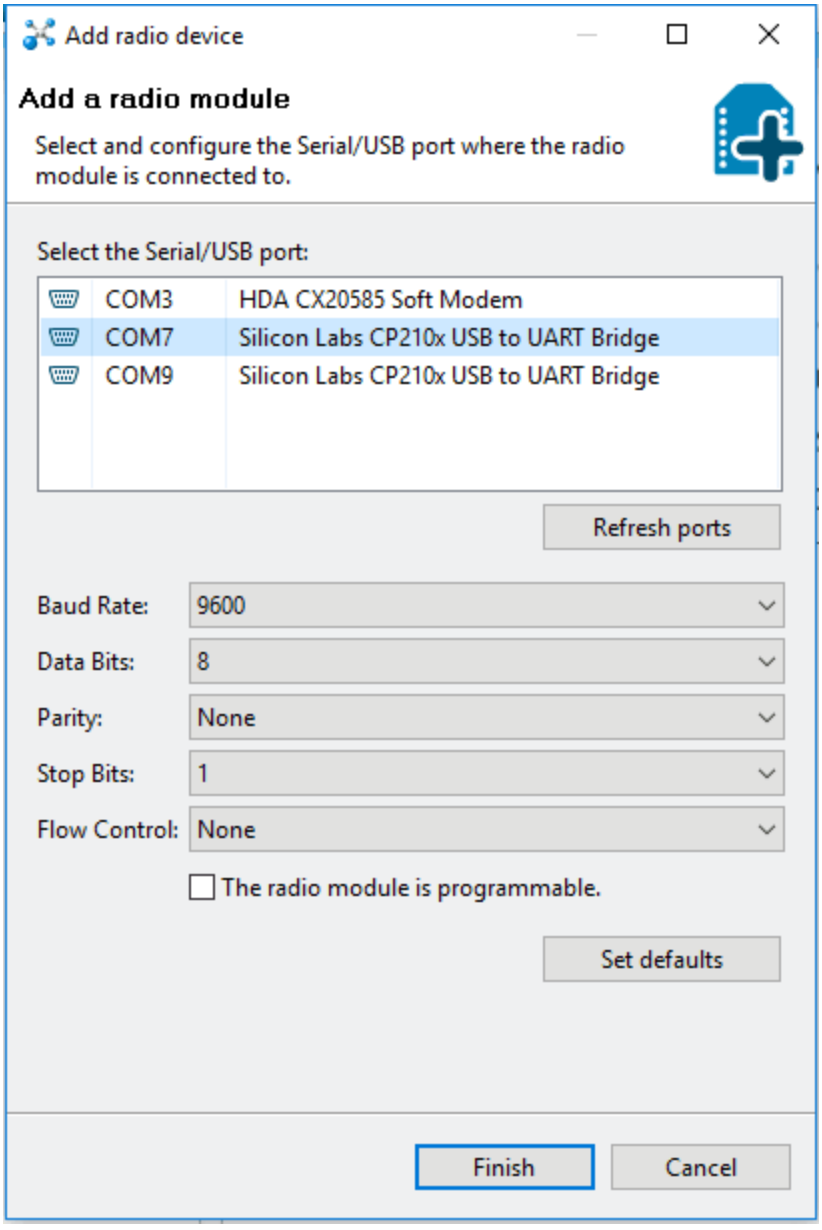
With the software open select the add radio button in the upper left



The **Add radio device** window will pop up in which we need to select USB port to which we have attached XBee devices. As shown in below snap.
Set Baud rate which is 9600 by default.

Set values of Data bits, parity and Stop bits.

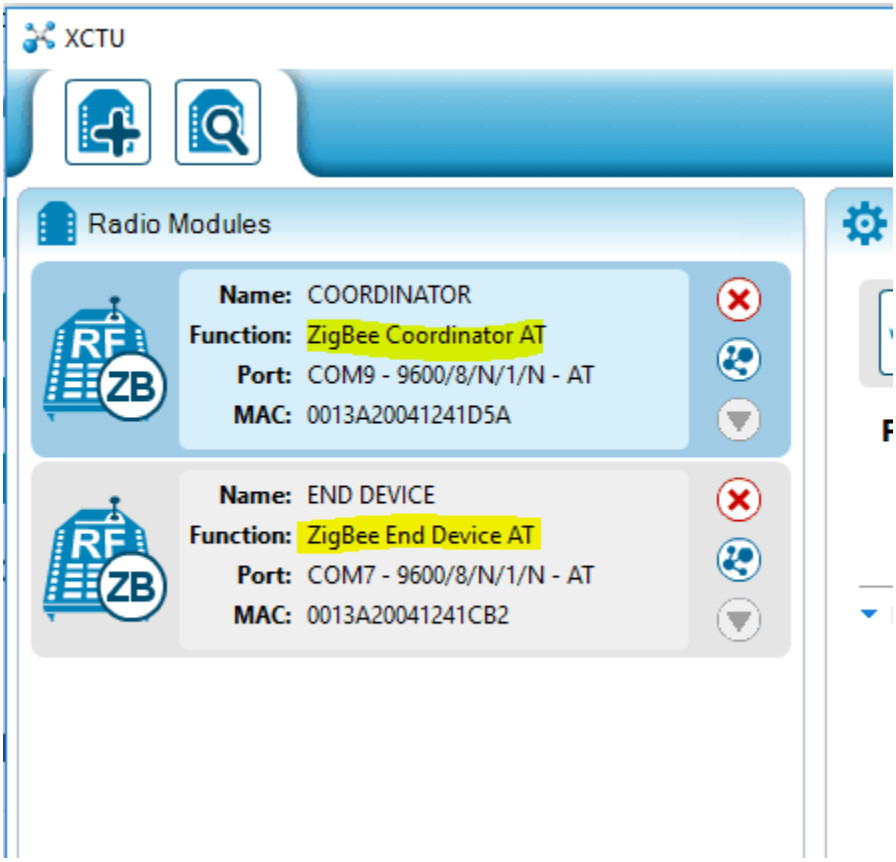
For sake of simplicity keep all default parameters as shown in figure and click on finish. It will add selected XBee device. Do same procedure to add second XBee device.



Now both devices will appear in upper left corner of X-CTU and their parameter value at right hand side as shown in below figure.

- Here we have configured one XBee device as Coordinator.
- And another one as End Device.

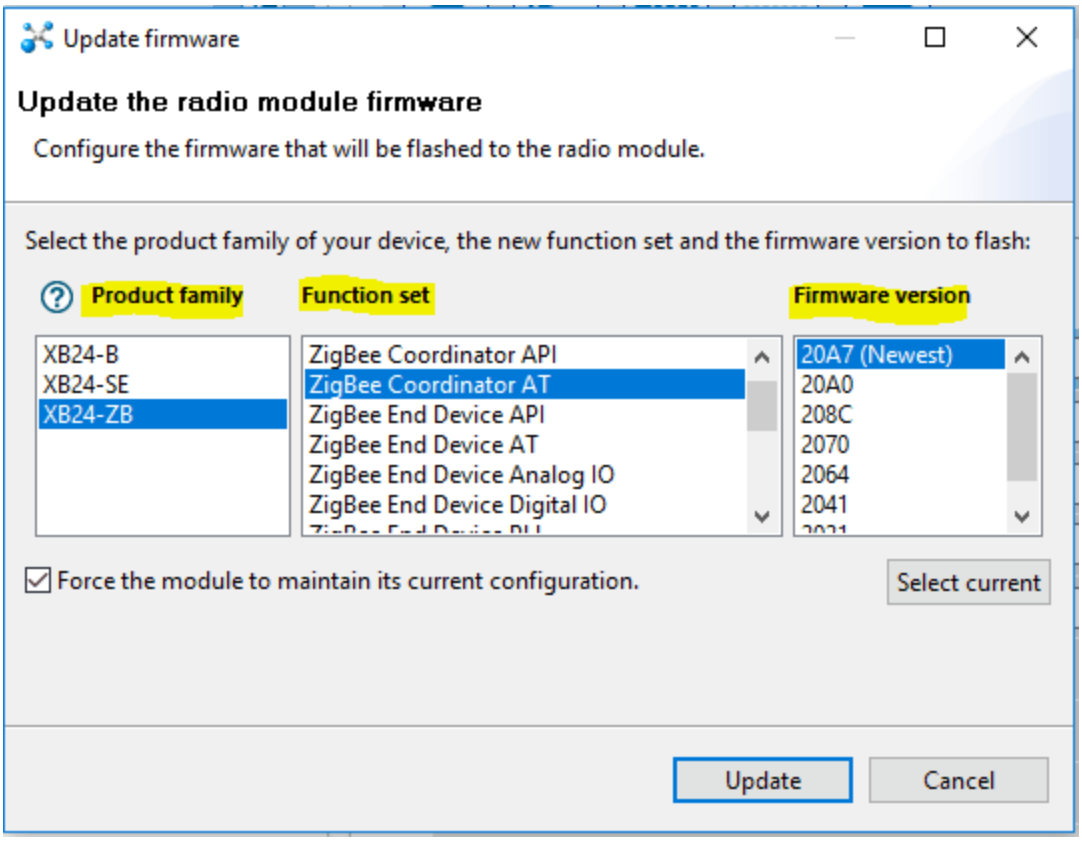
Both are configured in transparent (AT command) mode.



To configure XBee device in another operating mode and function set click on update

symbol besides the upper right hand side. Update Firmware window will pop up as shown in below figure, where we can select,

- Product family of XBee device.
- Function set to select required operating mode and function type of XBee device.
- And firmware version.

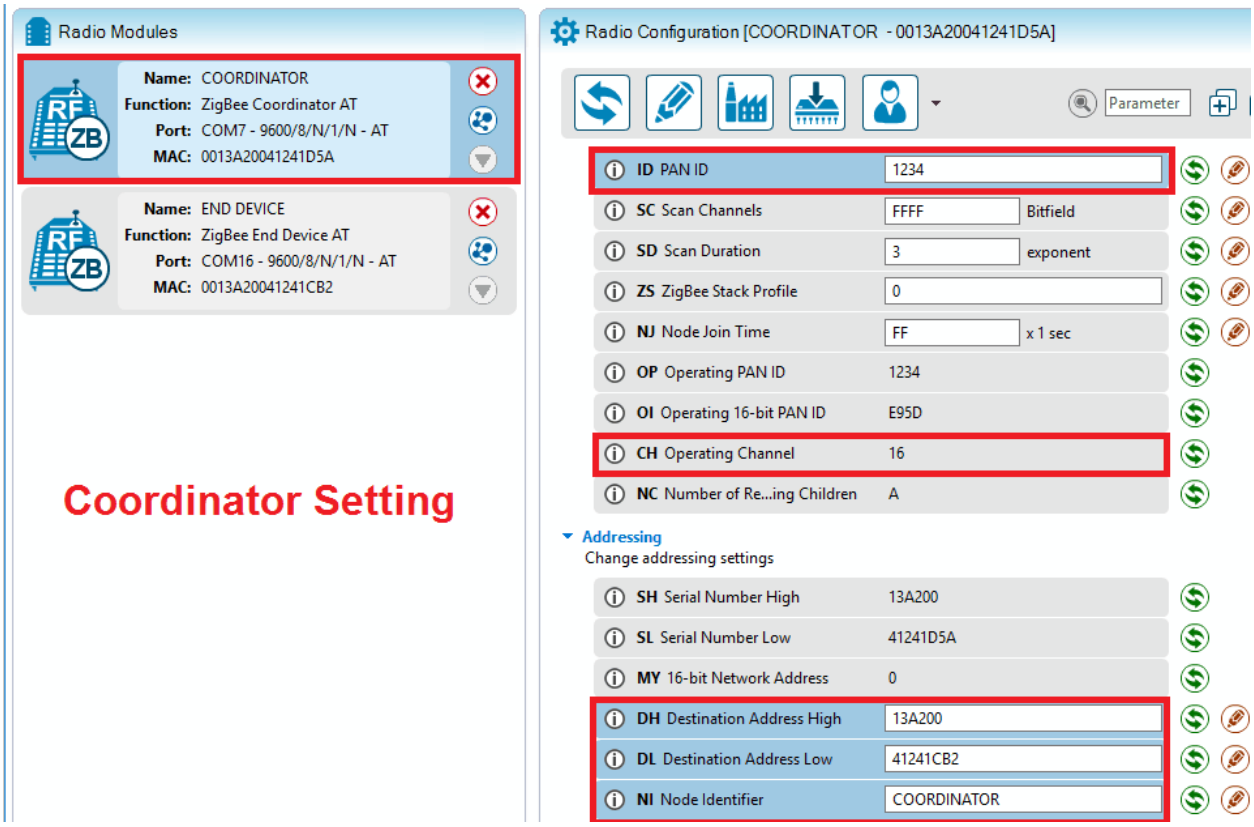


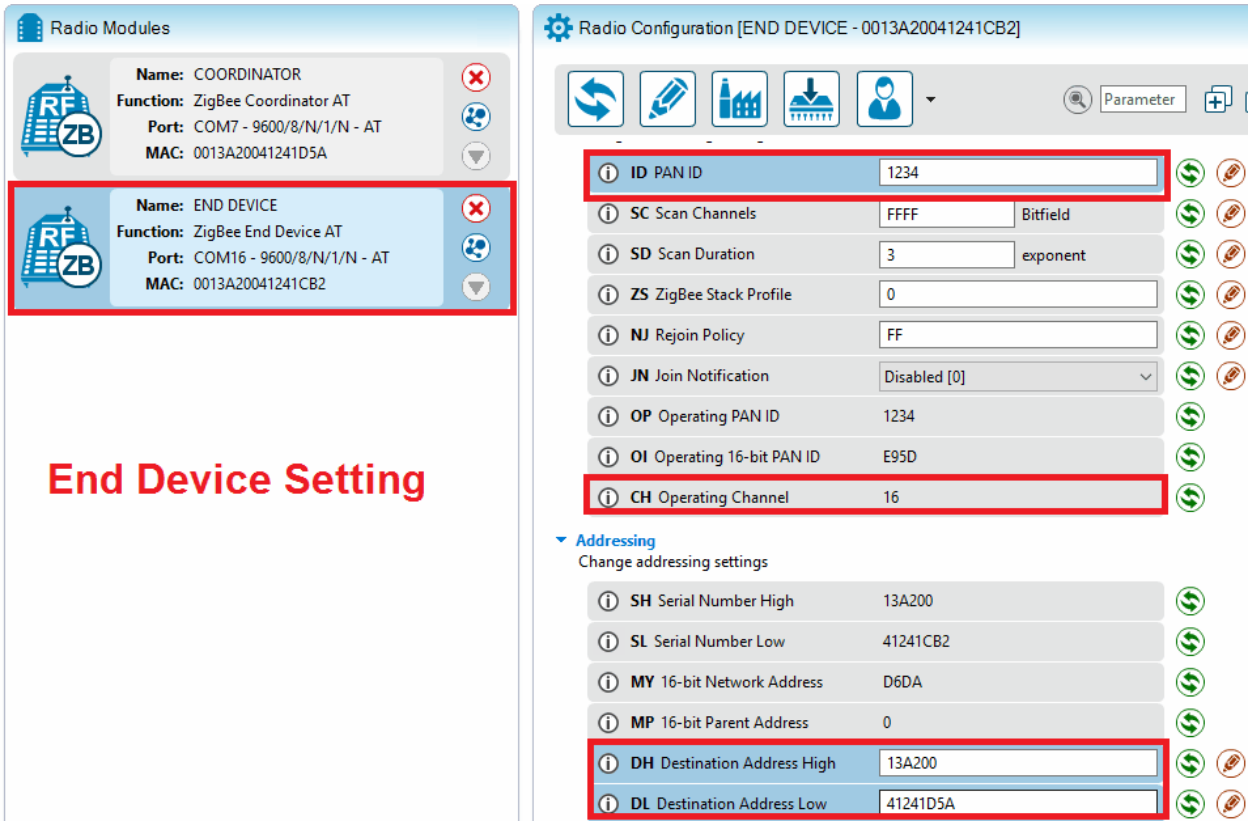
After updating XBee devices with correct firmware, we must ensure at least three parameters of XBee devices given below

- Baud rate should be same for each device.
- PAN ID should be unique for all XBee devices in same n/w.
- Operating Channel should be same for all devices in same n/w.

Now let's set XBee parameters i.e. PAN **ID**, Channel (**CH**) and Communicate in Unicast mode using long 64-bit addressing mode for both XBee devices as shown in below figure.

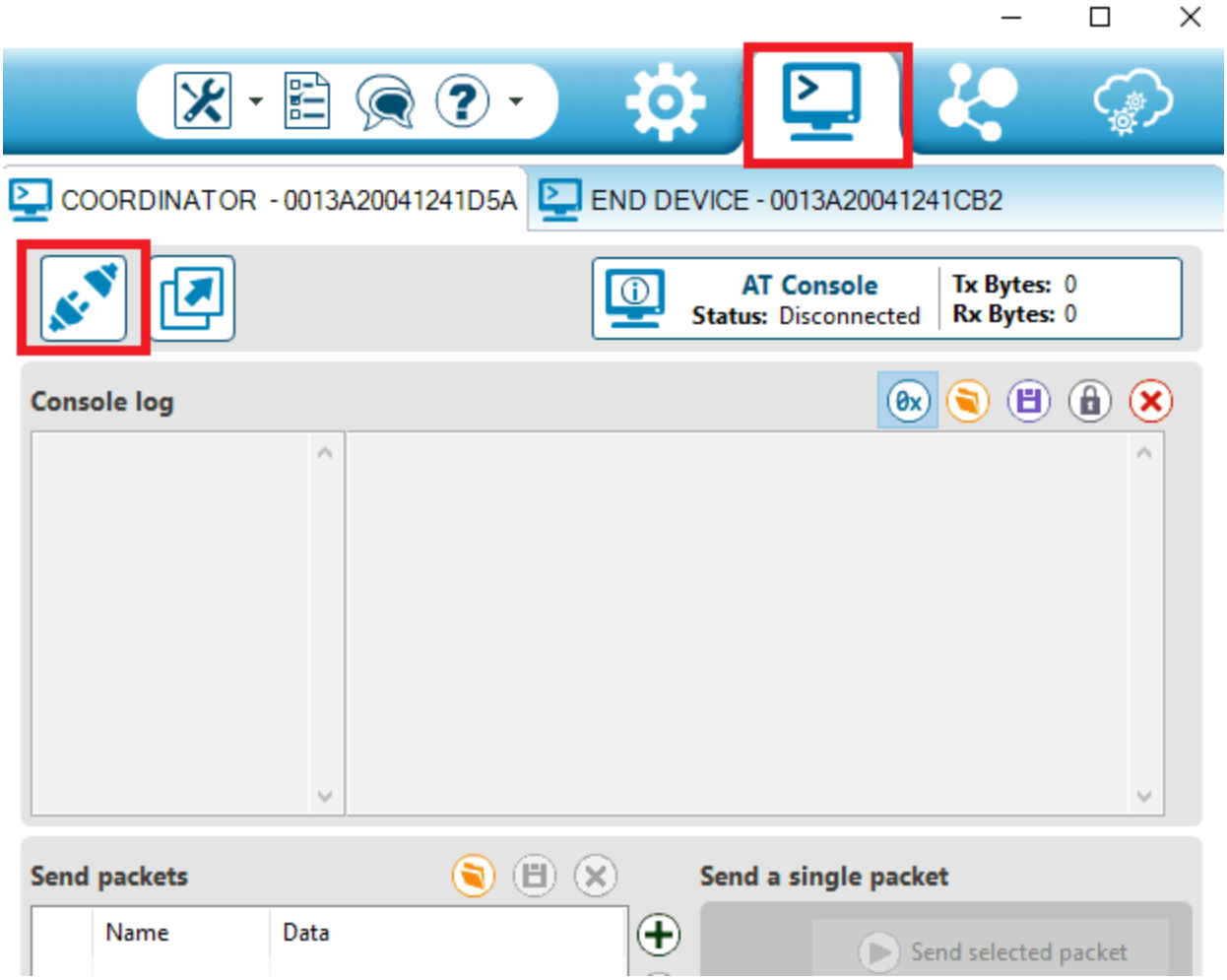
- Here we are setting PAN **ID** = 1234. Coordinator will select channel.
- For long 64-bit addressing mode we are setting Coordinator Source Address (**SH** &**SL**) parameters in End Device Destination Address (**DH** &**DL**) parameters and vice versa.
- Don't forget to keep Baud rate same of devices in the same network.


Radio Modules
Name: COORDINATOR
Function: ZigBee Coordinator AT
Port: COM7 - 9600/8/N/1/N - AT
MAC: 0013A20041241D5A
Name: END DEVICE
Function: ZigBee End Device AT
Port: COM16 - 9600/8/N/1/N - AT
MAC: 0013A20041241CB2
Coordinator Setting
Radio Configuration [COORDINATOR - 0013A20041241D5A]
Parameter
ID PAN ID 1234
SC Scan Channels FFFF Bitfield
SD Scan Duration 3 exponent
ZS ZigBee Stack Profile 0
NJ Node Join Time FF x 1 sec
OP Operating PAN ID 1234
OI Operating 16-bit PAN ID E95D
CH Operating Channel 16
NC Number of Re...ing Children A
Addressing
Change addressing settings
SH Serial Number High 13A200
SL Serial Number Low 41241D5A
MY 16-bit Network Address 0
DH Destination Address High 13A200
DL Destination Address Low 41241CB2
NI Node Identifier COORDINATOR

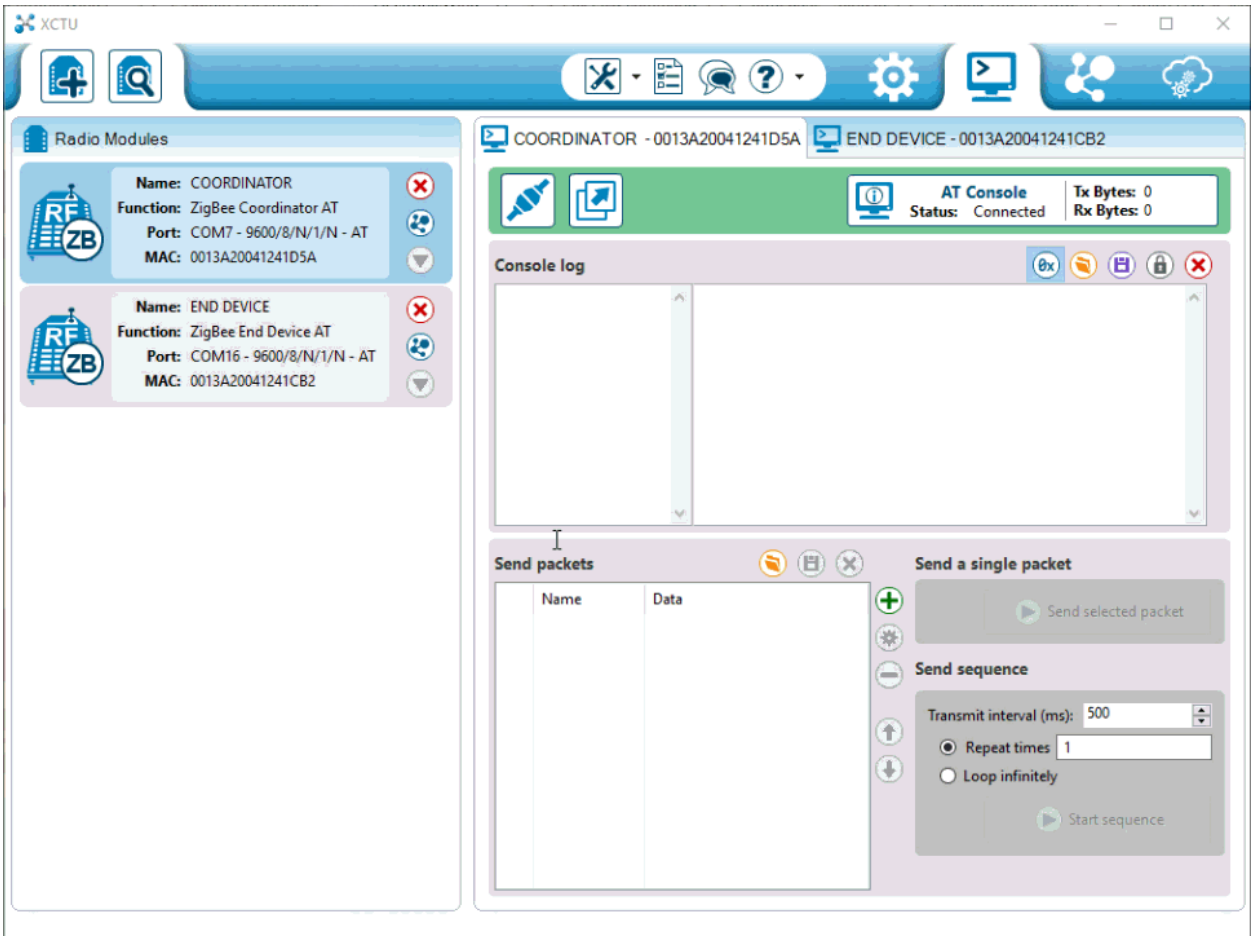

Radio Modules
Name: COORDINATOR
Function: ZigBee Coordinator AT
Port: COM7 - 9600/8/N/1/N - AT
MAC: 0013A20041241D5A
Name: END DEVICE
Function: ZigBee End Device AT
Port: COM16 - 9600/8/N/1/N - AT
MAC: 0013A20041241CB2
End Device Setting
Radio Configuration [END DEVICE - 0013A20041241CB2]
Parameter
ID PAN ID 1234
SC Scan Channels FFFF Bitfield
SD Scan Duration 3 exponent
ZS ZigBee Stack Profile 0
NJ Rejoin Policy FF
JN Join Notification Disabled [0]
OP Operating PAN ID 1234
OI Operating 16-bit PAN ID E95D
CH Operating Channel 16
Addressing
Change addressing settings
SH Serial Number High 13A200
SL Serial Number Low 41241CB2
MY 16-bit Network Address D6DA
MP 16-bit Parent Address 0
DH Destination Address High 13A200
DL Destination Address Low 41241D5A

Now to test communication, open the console by clicking

symbol shown in upper right corner of X-CTU window and click on **open** to open console as shown in below figure.



Now type message from Coordinator console log, it will appear at end device console log and vice versa as seen in below GIF.



Step 3)

After confirming both devices are able to communicate, you are finished and able to connect them to the Arduino uno and Raspberry Pi devices.