Understanding the connectivity of Raspberry Pi board with X-bee module. Writing a network application for communication between two devices using X-bee

Objectives:

- To study how to interface X-bee with Raspberry Pi
- To study wireless communication using two X-bees

Software:

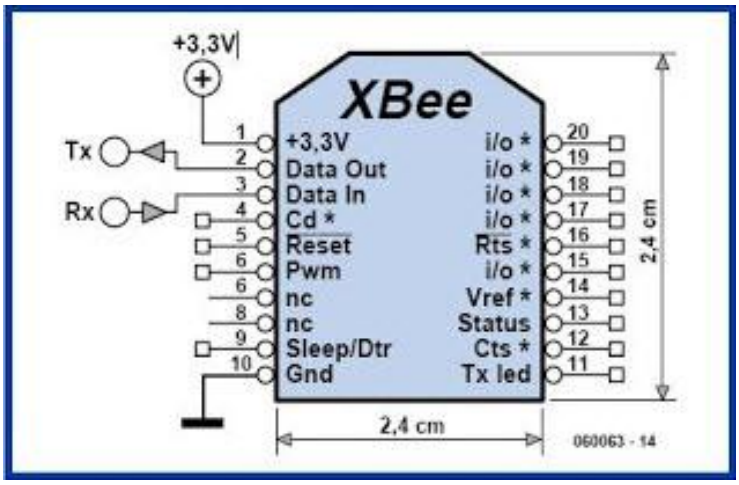
- Raspberry OS,
- IDLE IDE for Python
- XCTU for configuring X-bee

Hardware Modules:

- Raspberry pi
- Two X-bee modules
- FRC cable

Theory:

- X-Bee is a low-power wireless mesh network standard, operating in the 2.4 GHz range.
- XBee is not same as ZigBee—instead, XBee is a brand of radio communication modules (made byDigi) that can support a number of protocols, including ZigBee, 802.15.4, WiFi, etc.
- Its range is 10 to 100 meters.
- X-Bee is often used in home automation products, though it is not the only option.



Configuration of X-bee modules:

To establish communication between two X-bees, they need to configure.

Step 1:

- To configure XBee module, connect XBee Explorer shield to a computer via USB.

Fig. 1: XBee Explorer shield

Step 2:

Here we set the Channel, PAN ID, and Address values for the X-bee

Channel:

- The channel calibrates the operating frequency within the 2.4GHz 802.15.4 band.
- The XBees must be on the same channel to communicate with one another.

PAN ID (Personal Area Network ID):

- The X-Bees must share the same PAN ID to communicate with one another. You can choose a value between 0 and 0xFFFF.

Addressing:

- Each X-Bee has a Source address (referred to as “MY address”) and a Destination address (which has an upper half, Destination High or DH, and a lower half, Destination Low or DL).
- The XBee’s destination address specifies to which source address it can send data.
- You can specify a universally unique address by using the 64-bit address printed on the back of the module, use a shorter 16-bit address (unique within a network), or use a string of text (e.g., “Alice’s radio”).

Additionally, each XBee in a network plays one of the following roles,

Coordinator:

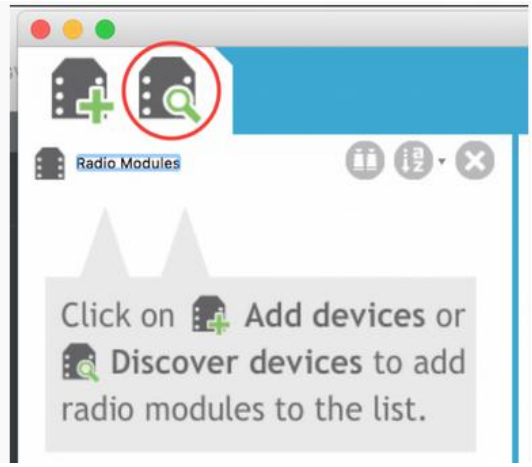
- Each network has exactly one Coordinator, which serves as the root of the network tree.

Router:

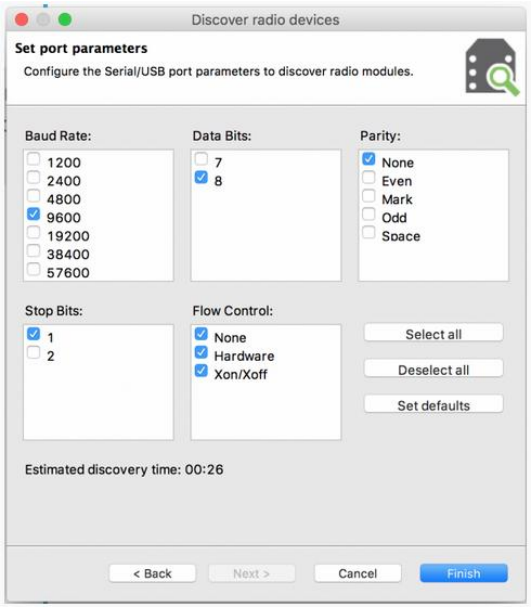
- A network can have multiple Routers; these can forward information to end devices and also run application functions.

Steps to configure the X-bee

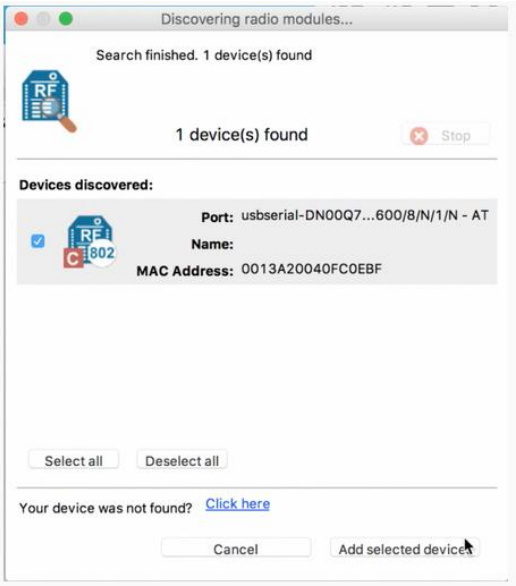
- Download and install XCTU. It is available for both Windows and Mac.
- Plug first XBee into an Explorer module, and connect it to computer's USB port via a USB cable.
- Open XCTU and click "Discover devices."



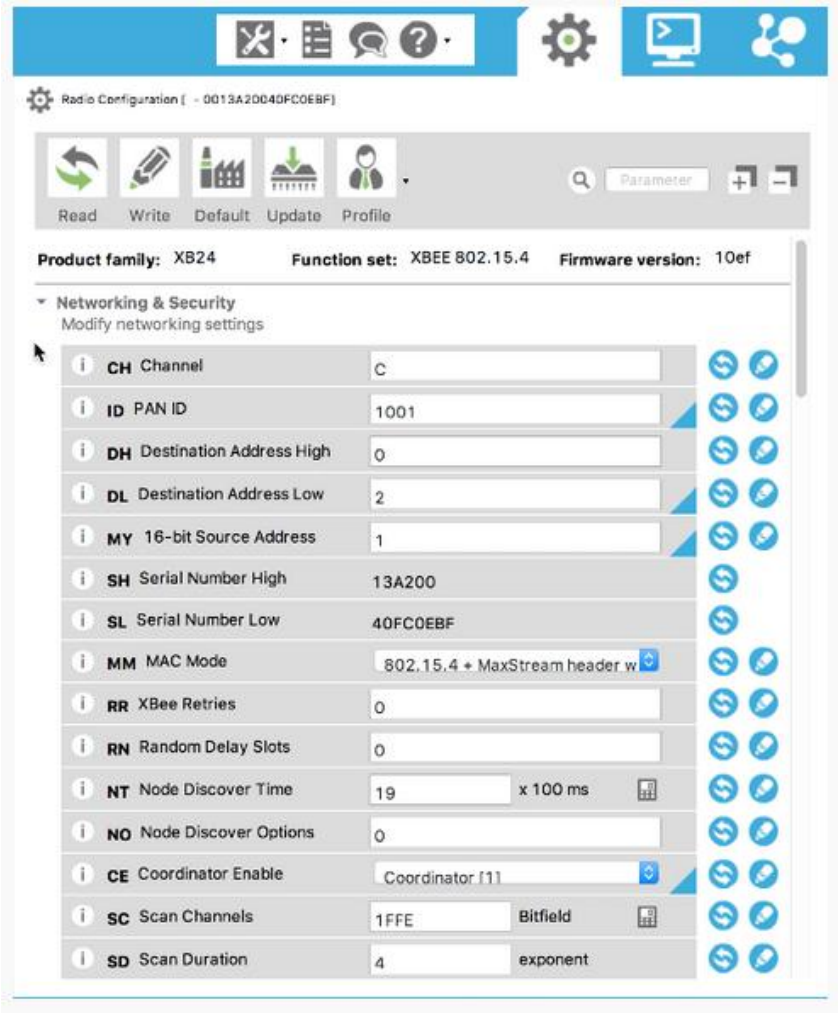
- Select the port to be scanned. Then on the next page, select the settings as shown below. Click "Finish."



- Your device should appear on the "Devices discovered" list.
- Click "Add selected devices" for your module.



- With the Gear icon selected, click the radio module in the left-hand menu.
- This should open up a long list of settings.



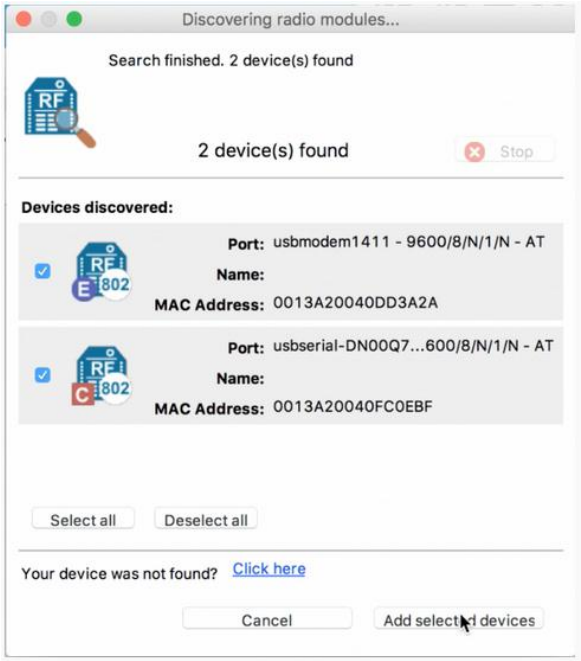
- We can set the X-bee as Coordinator or Router.
- Plug the first X-bee on the Explorer. Set the first X-bee as Coordinator and do the settings as per given in the below table.
- Now unplug the first X-bee and plug the second X-bee on the Explorer. Set the second X-bee as Router and do the settings as per given in the below table.

	Co-ordinator	Router
	X-bee-1	X-bee-2
DH	0	0
DL	FFFF	0
PAN ID	1234	1234
JV channel verification	Disable	Enable
CE Coordinator Enable	Enable	Disable

- After you've made all your changes, click "**Write**."

Run the Communication Test

- Plug one XBee (on its Explorer module) into a USB port (via USB cable).
- Plug the other XBee (on its Explorer module) into another USB port.
- In XCTU, scan for devices. You should see both devices available.
- Select both of them and click "Add selected devices."



- Click one of the modules in the left-hand column. Now select the Console icon to view the console. Click "Open."

- Repeat for the other module, opening up a console.
- Type some text in one of the console. You should see the result echoed back in the other console.

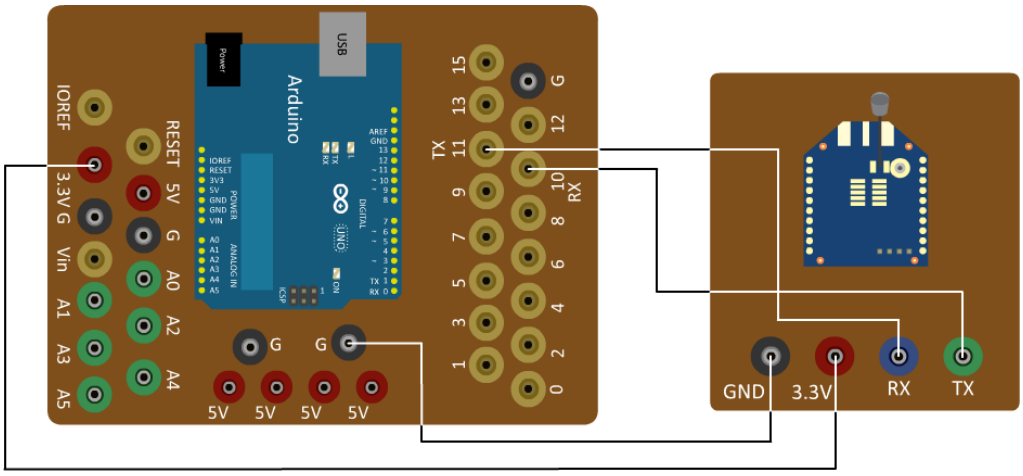
Safety precautions:

- Raspberry-Pi provides 3.3V and 5V VCC pins
- Raspberry-Pi operates on 3.3V.
- Various sensors and actuators operate on different voltages.
- Read datasheet of a given sensor or an actuator and then use appropriate VCC pin to connect a sensor or an actuator.
- Ensure that signal voltage coming to the Raspberry-Pi from any sensor or actuator does not exceed 3.3V.
- If signal/data coming to Raspberry-Pi is greater than 3.3V then use voltage level shifter module to decrease the incoming voltage.
- The Raspberry-Pi is a costly device, hence you should show the circuit connections to your instructor before starting your experiment.

Set Up Your Circuits

For Arduino:

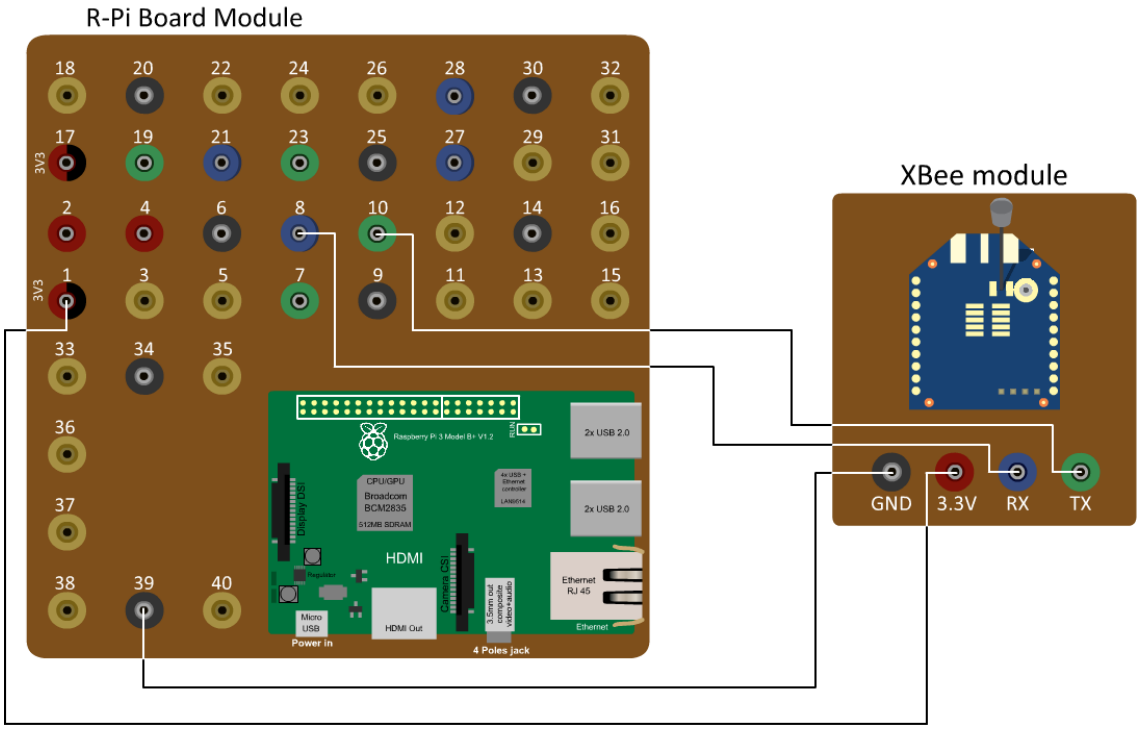
- We will connect one X-bee to Arduino Microcontroller.
- This will act as sender
- The interfacing diagram of X-bee with Arduino is as shown below



Connections with Arduino-

Arduino Pins	X-bee Pins
10	Data Out (Pin-2) (Tx)
11	Data In (Pin-3) (Rx)
3.3v	Vcc or 3.3v (Pin-1)
G	Gnd(Pin-10)

For Raspberry Pi:



Steps for assembling the circuit:

R-Pi-3 Pins	X-bee Pins
10	Data Out (Pin-2) (Tx)
8	Data In (Pin-3) (Rx)
3.3v	Vcc or 3.3v (Pin-1)
G	Gnd(Pin-10)

Procedure:

- Write the program as per the algorithm given below.
- Save program.
- Run using run module

How to enable UART port:

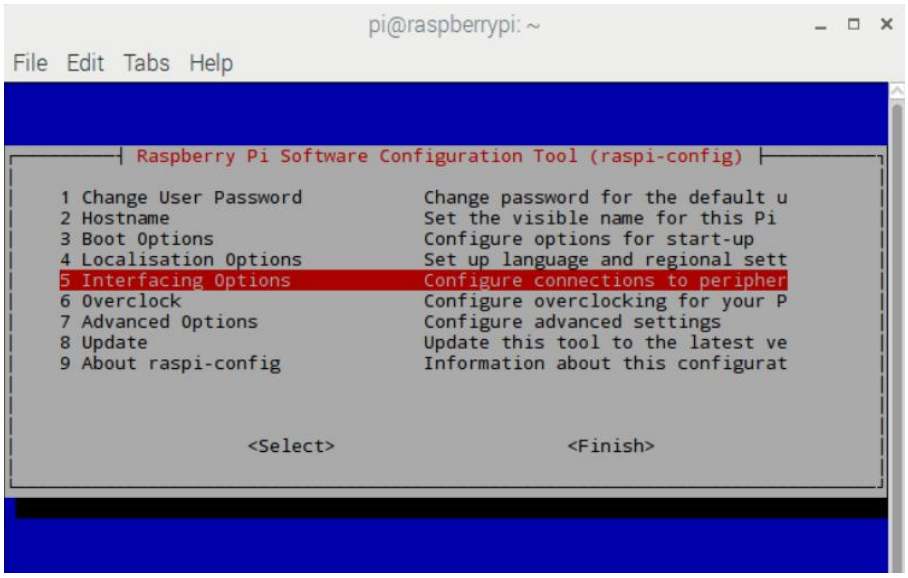
Step 1:

- In Raspberry Pi, enter following command in Terminal window to enable UART,

```
sudo raspi-config
```

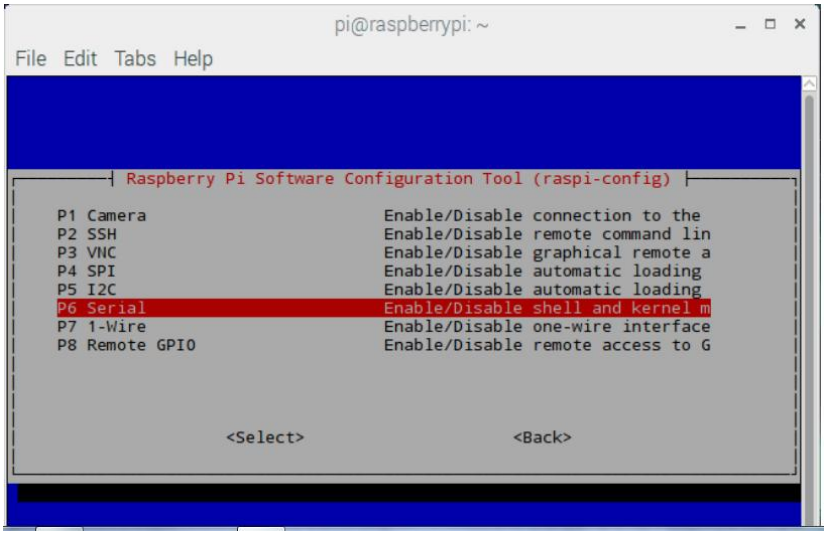
Step 2:

Select -> **Interfacing Options**



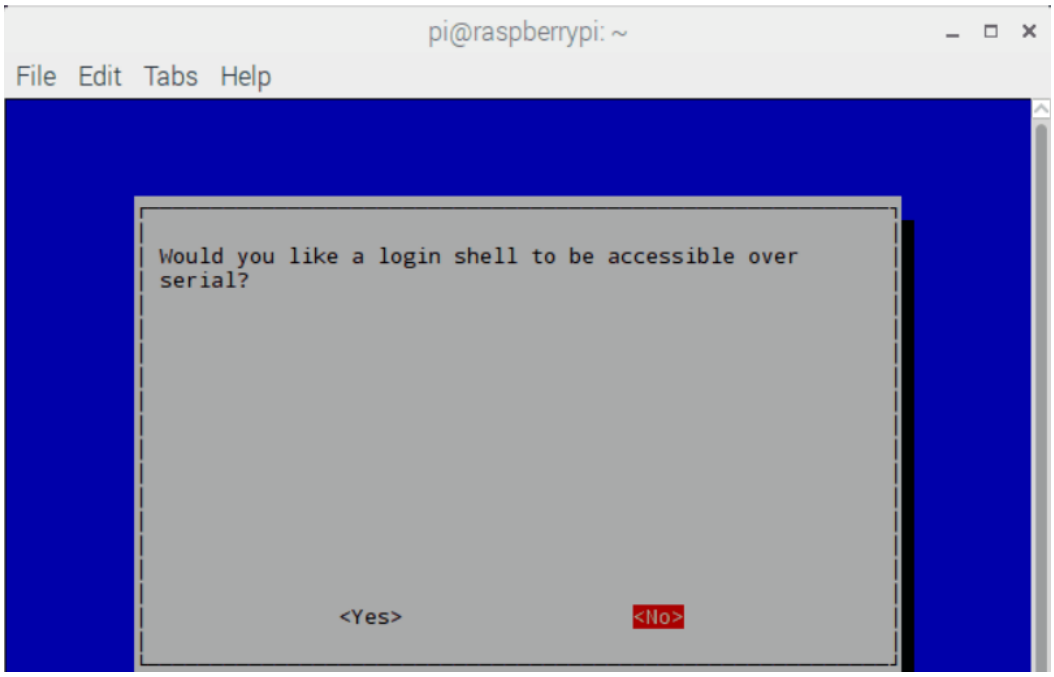
Step 3:

- After selecting Interfacing option, select **Serial** option to enable UART



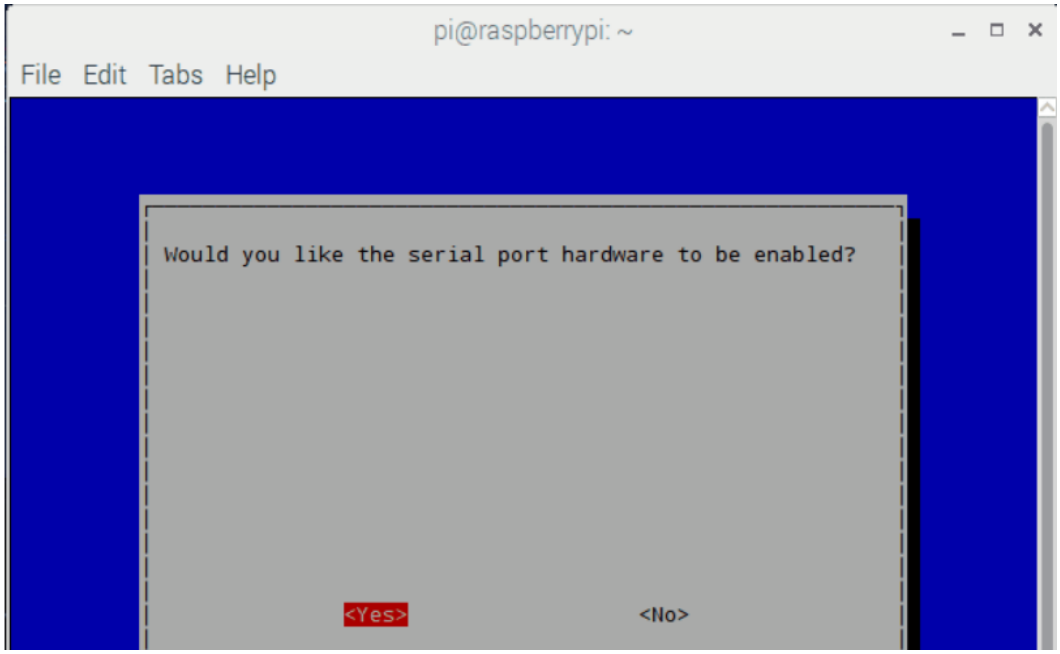
Step 4:

- Then it will ask for login shell to be accessible over Serial, select **No** shown as follows.



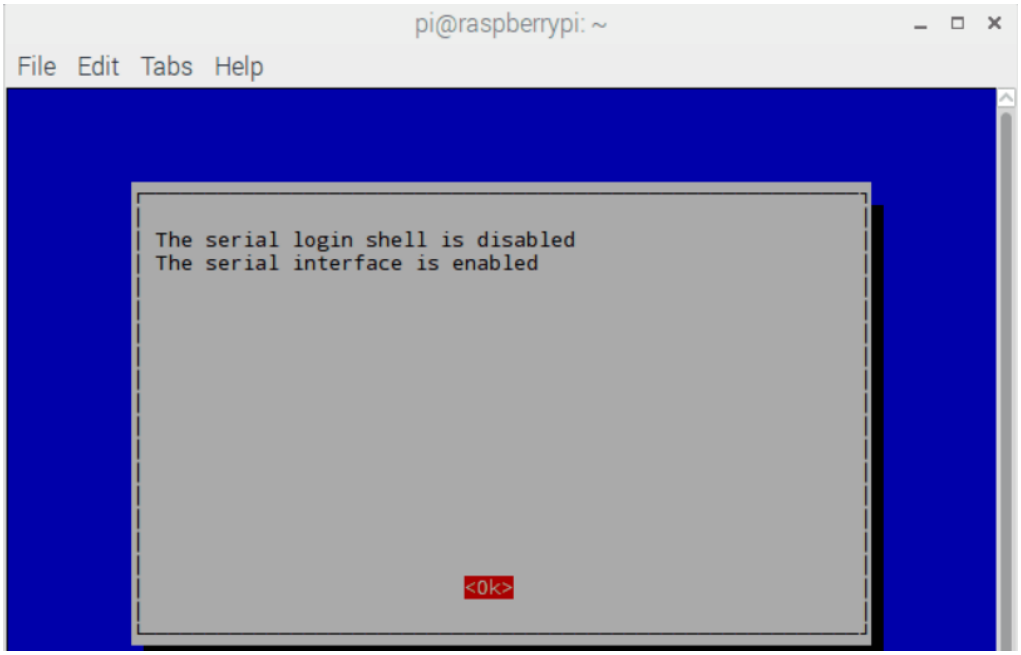
Step 5:

- At the end, it will ask for enabling Hardware Serial port, select **Yes**,



Step 6:

- Finally, our UART is enabled for Serial Communication on RX and TX pin of R-Pi-3



Step 7:

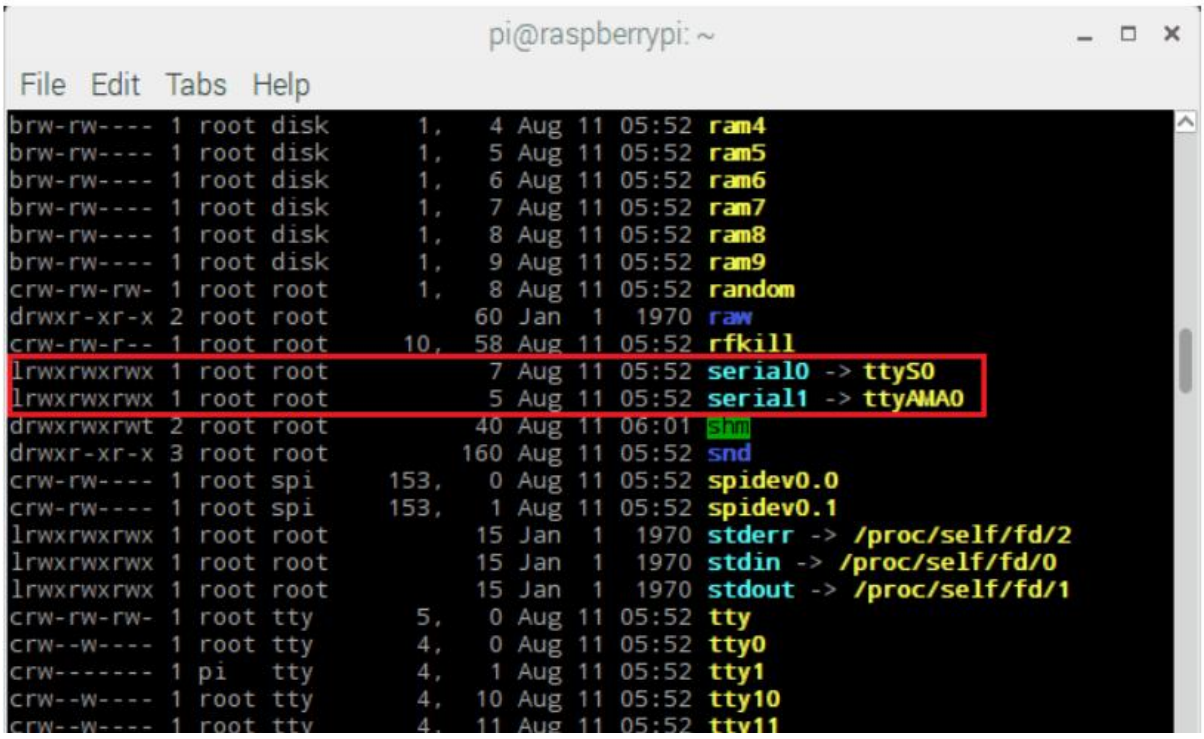
- Then, reboot the Raspberry Pi.
- Sudo reboot

Step 8:

- To check UART mapping, enter following commands.
- Enter command,

```
ls -l /dev
```

- The UART mapping for /dev/ttyS0 and /dev/ttyAMA0 is shown below,



Study of different CPU Frequency Governors. Writing an application to change CPU frequency of Raspberry-Pi board

Aim/Objectives:

- To understand the concept of CPU Frequency Governance

CPU Frequency Governor

- The governor defines the power characteristics of the system CPU, which in turn affects CPU performance.
- Each governor has its own unique behavior, purpose, and suitability in terms of workload.

Types of CPU Frequency Governors:

Performance:

- The Performance governor forces the CPU to use the highest possible clock frequency.
- Clock frequency will be statically set, and will not change.
- No power saving benefit.
- Only suitable for hours of heavy workload

Power Save:

- The Power-save governor forces the CPU to use the lowest possible clock frequency.
- Clock frequency will be statically set, and will not change.
- This governor offers maximum power savings, but at the lowest CPU performance.

On-Demand:

- The On-Demand governor is a dynamic governor that allows the CPU
    a. To achieve maximum clock frequency when system load is high, and
    b. To achieve minimum clock frequency when the system is idle

User Space:

- The Userspace governor allows the user to set the CPU frequency for User programs.

CPU Frequency of Raspberry Pi 3:

- Minimum CPU frequency: 700 MHz
- Maximum CPU frequency: 1200 MHz (1.2 GHz)
- Raspberry pi 3 having On-Demand CPU Frequency Governor

Steps to change the CPU Frequency of Raspberry Pi 3:

- For changing cpu frequency of Raspberry Pi 3 we know the current frequency of Raspberry Pi-3
- Open the LX-Terminal of Raspberry Pi-3
- Type the following command
  cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_cur_freq
- Press Enter key
- You will get the current speed in kHz.
- Divide this speed by 1000 to get in MHz.

- For changing current cpu frequency open config.txt file using following command
  sudo nano /boot/config.txt
- Press Enter key
- It will open the configuration file
- Uncomment the arm_freq and change its value (Not more than 1200) because CPU frequency of Raspberry pi 3 is minimum 700 MHz and Maximum 1.2 GHz.
- Then press Ctrl+O and then enter for writing the changes in file.
- Open the LX-Terminal in Raspberry pi-3
- Reboot Raspberry pi using following command
  sudo reboot
- After Restarting Raspberry Pi 3 open LX-Terminal and check the current frequency of Raspberry Pi 3 using step 3.