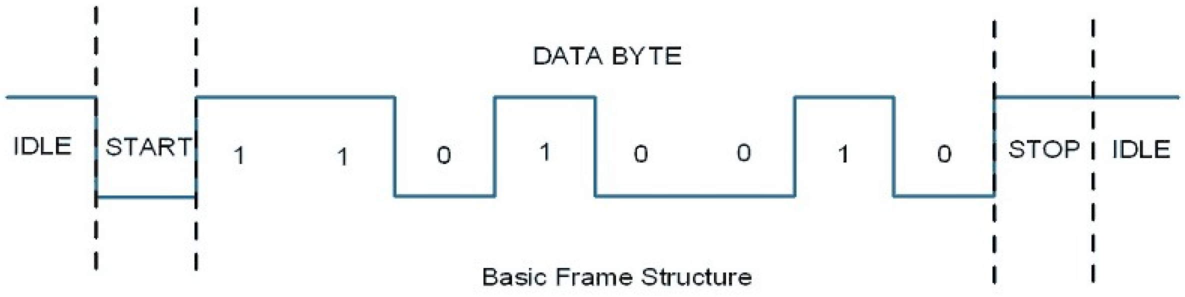
Introduction

UART (Universal Asynchronous Receiver/Transmitter) is a serial communication protocol in which data is transferred serially i.e. bit by bit. Asynchronous serial communication is widely used for byte oriented transmission. In Asynchronous serial communication, a byte of data is transferred at a time.

UART serial communication protocol uses a defined frame structure for their data bytes. Frame structure in Asynchronous communication consists:

- **START bit:** It is a bit with which indicates that serial communication has started and it is always low.
- **Data bits packet**: Data bits can be packets of 5 to 9 bits. Normally we use 8-bit data packet, which is always sent after the START bit.
- **STOP bit**: This usually is one or two bits in length. It is sent after data bits packet to indicate the end of frame. Stop bit is always logic high.



Usually, an asynchronous serial communication frame consists of a START bit (1 bit) followed by a data byte (8 bits) and then a STOP bit (1 bit), which forms a 10-bit frame as shown in the figure above. The frame can also consist of 2 STOP bits instead of a single bit, and there can also be a PARITY bit after the STOP bit.

Raspberry Pi UART

Raspberry Pi has two in-built UART which are as follows:

- **PL011 UART**
- **mini UART**

PL011 UART is an ARM based UART. This UART has better throughput than **mini** UART.

In Raspberry Pi 3, mini UART is used for Linux console output whereas PL011 is connected to the On-board Bluetooth module.

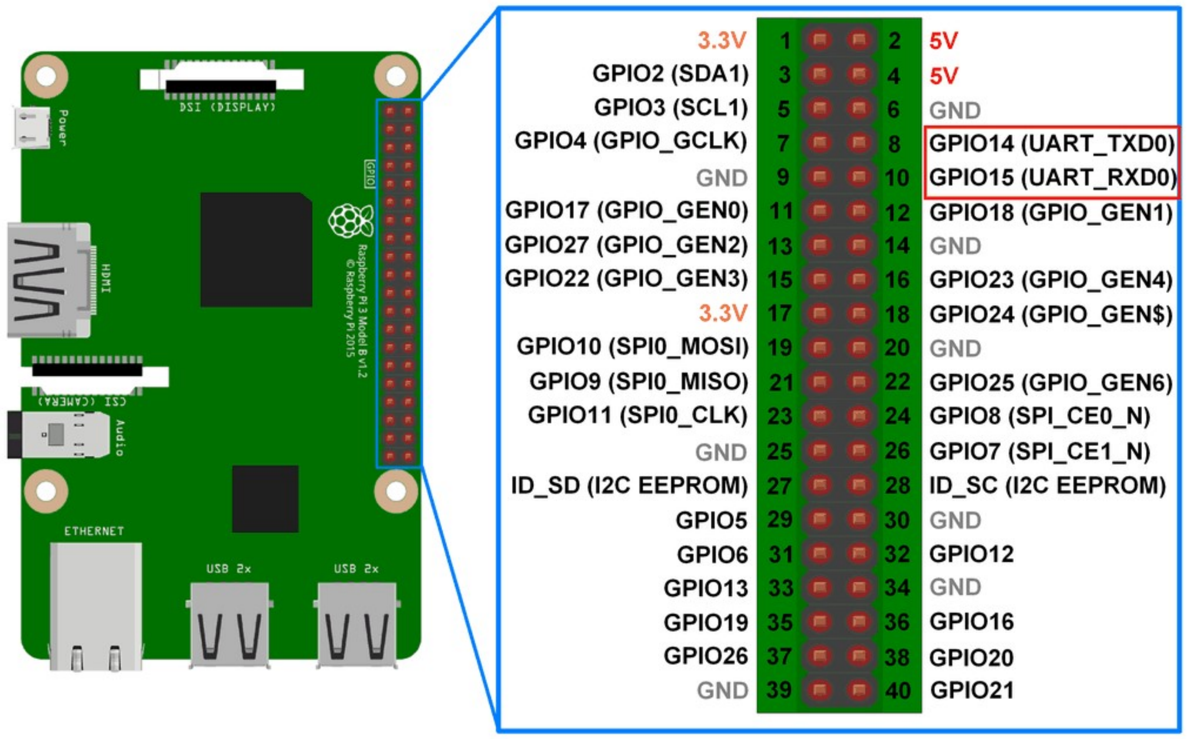
And in the other versions of Raspberry Pi, PL011 is used for Linux console output.

Mini UART uses the frequency which is linked to the core frequency of GPU. So as the GPU core frequency changes, the frequency of UART will also change which in turn will change the baud rate for UART. This makes the mini UART unstable which may lead to data loss or corruption. To make mini UART stable, fix the core frequency. mini UART doesn't have parity support.

The PL011 is a stable and high performance UART. For better and effective communication use PL011 UART instead of mini UART.

It is recommended to enable the UART of Raspberry Pi for serial communication. Otherwise, we are not able to communicate serially as UART ports are used for Linux console output and Bluetooth module.

Raspberry Pi 3 UART Pins


DSI (DISPLAY)
Power
GPIO
HDMI
Raspberry Pi 3 Model B v1.2
© Raspberry Pi 2015
CSI (CAMERA)
Audio
ETHERNET
USB 2x
USB 2x
3.3V 1 2 5V
GPIO2 (SDA1) 3 4 5V
GPIO3 (SCL1) 5 6 GND
GPIO4 (GPIO_GCLK) 7 8 GPIO14 (UART_TXD0)
GND 9 10 GPIO15 (UART_RXD0)
GPIO17 (GPIO_GEN0) 11 12 GPIO18 (GPIO_GEN1)
GPIO27 (GPIO_GEN2) 13 14 GND
GPIO22 (GPIO_GEN3) 15 16 GPIO23 (GPIO_GEN4)
3.3V 17 18 GPIO24 (GPIO_GEN$)
GPIO10 (SPI0_MOSI) 19 20 GND
GPIO9 (SPI0_MISO) 21 22 GPIO25 (GPIO_GEN6)
GPIO11 (SPI0_CLK) 23 24 GPIO8 (SPI_CE0_N)
GND 25 26 GPIO7 (SPI_CE1_N)
ID_SD (I2C EEPROM) 27 28 ID_SC (I2C EEPROM)
GPIO5 29 30 GND
GPIO6 31 32 GPIO12
GPIO13 33 34 GND
GPIO19 35 36 GPIO16
GPIO26 37 38 GPIO20
GND 39 40 GPIO21

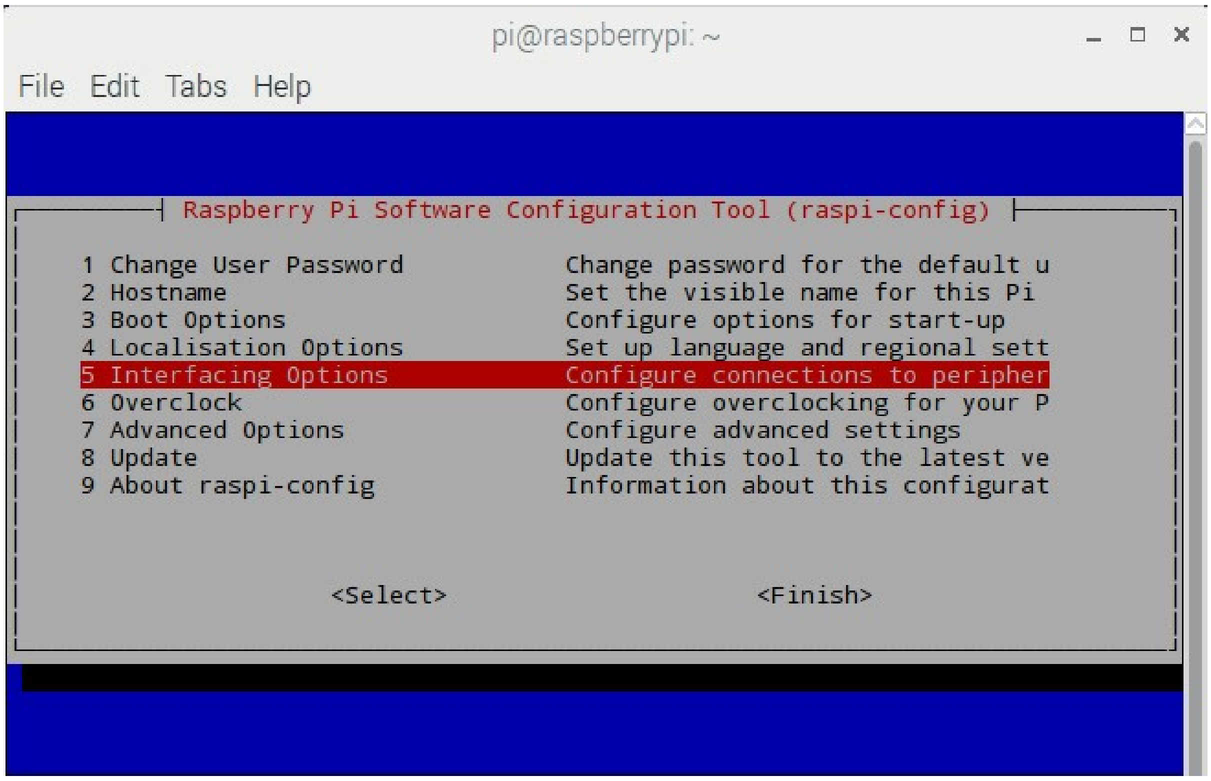
Raspberry Pi 3 UART Pins

Configure UART on Raspberry Pi

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

```
sudo raspi-config
```

Select -> **Interfacing Options**



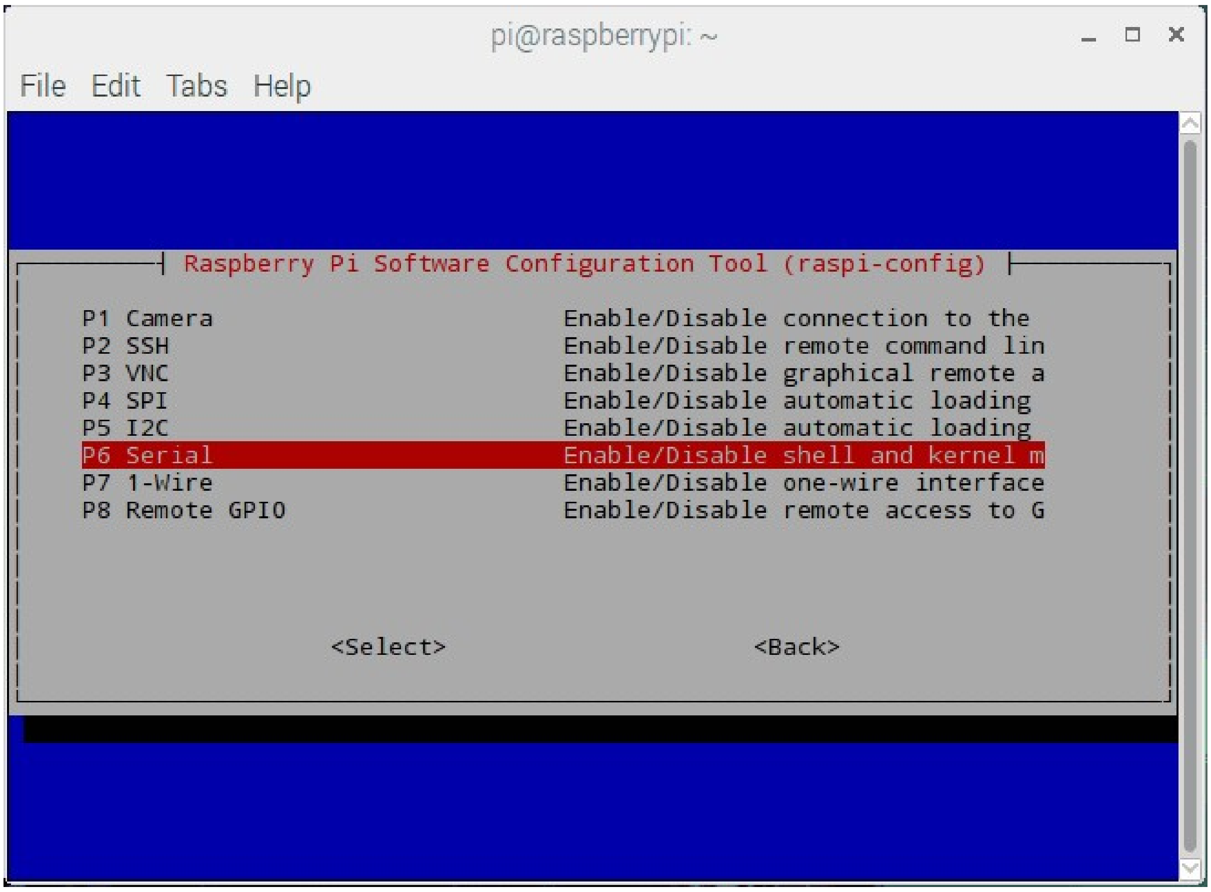
```
pi@raspberrypi: ~
File  Edit  Tabs  Help

┤ Raspberry Pi Software Configuration Tool (raspi-config) ├

    1 Change User Password         Change password for the default u
    2 Hostname                     Set the visible name for this Pi
    3 Boot Options                 Configure options for start-up
    4 Localisation Options         Set up language and regional sett
    5 Interfacing Options          Configure connections to peripher
    6 Overclock                    Configure overclocking for your P
    7 Advanced Options             Configure advanced settings
    8 Update                       Update this tool to the latest ve
    9 About raspi-config           Information about this configurat

                  <Select>                     <Finish>
```

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



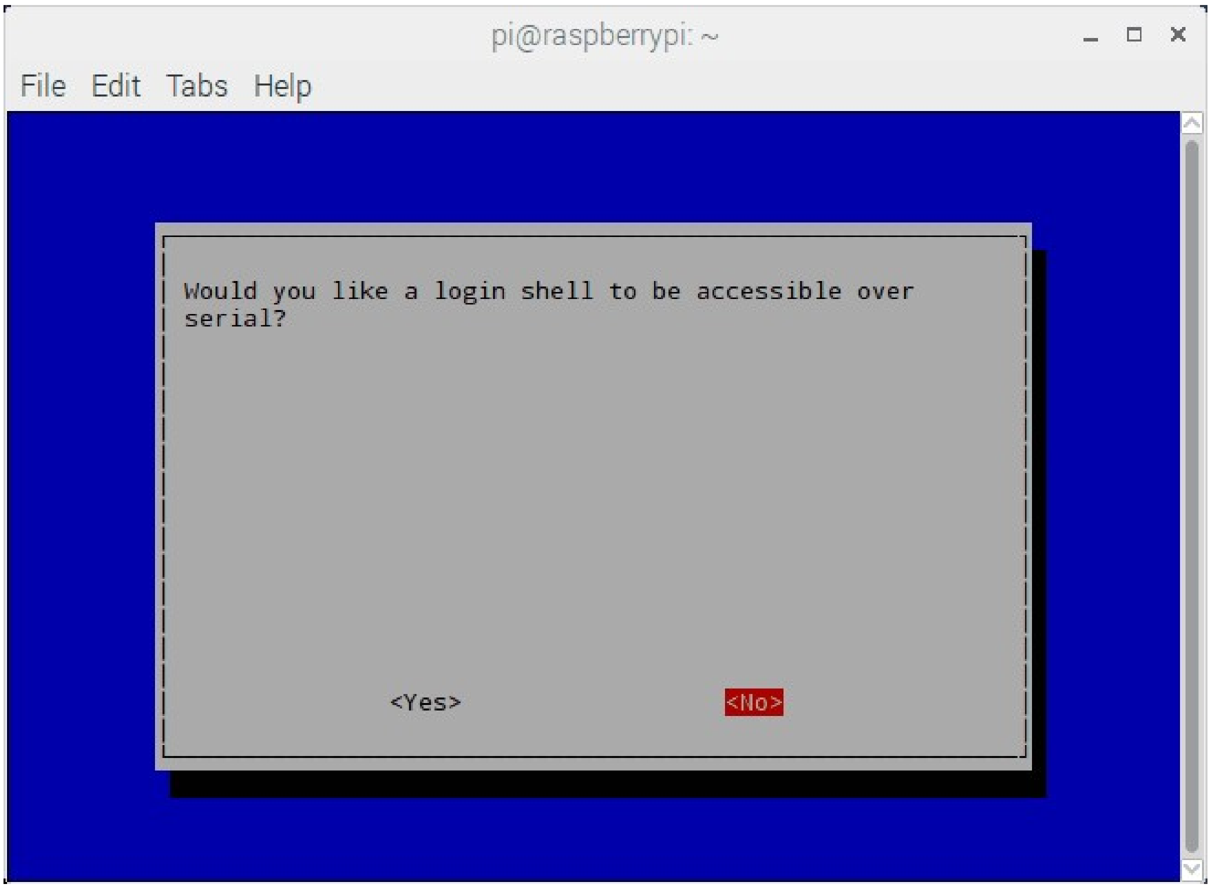
```
pi@raspberrypi: ~
File  Edit  Tabs  Help

┤ Raspberry Pi Software Configuration Tool (raspi-config) ├

    P1 Camera                      Enable/Disable connection to the
    P2 SSH                         Enable/Disable remote command lin
    P3 VNC                         Enable/Disable graphical remote a
    P4 SPI                         Enable/Disable automatic loading
    P5 I2C                         Enable/Disable automatic loading
    P6 Serial                      Enable/Disable shell and kernel m
    P7 1-Wire                      Enable/Disable one-wire interface
    P8 Remote GPIO                 Enable/Disable remote access to G

                  <Select>                     <Back>
```

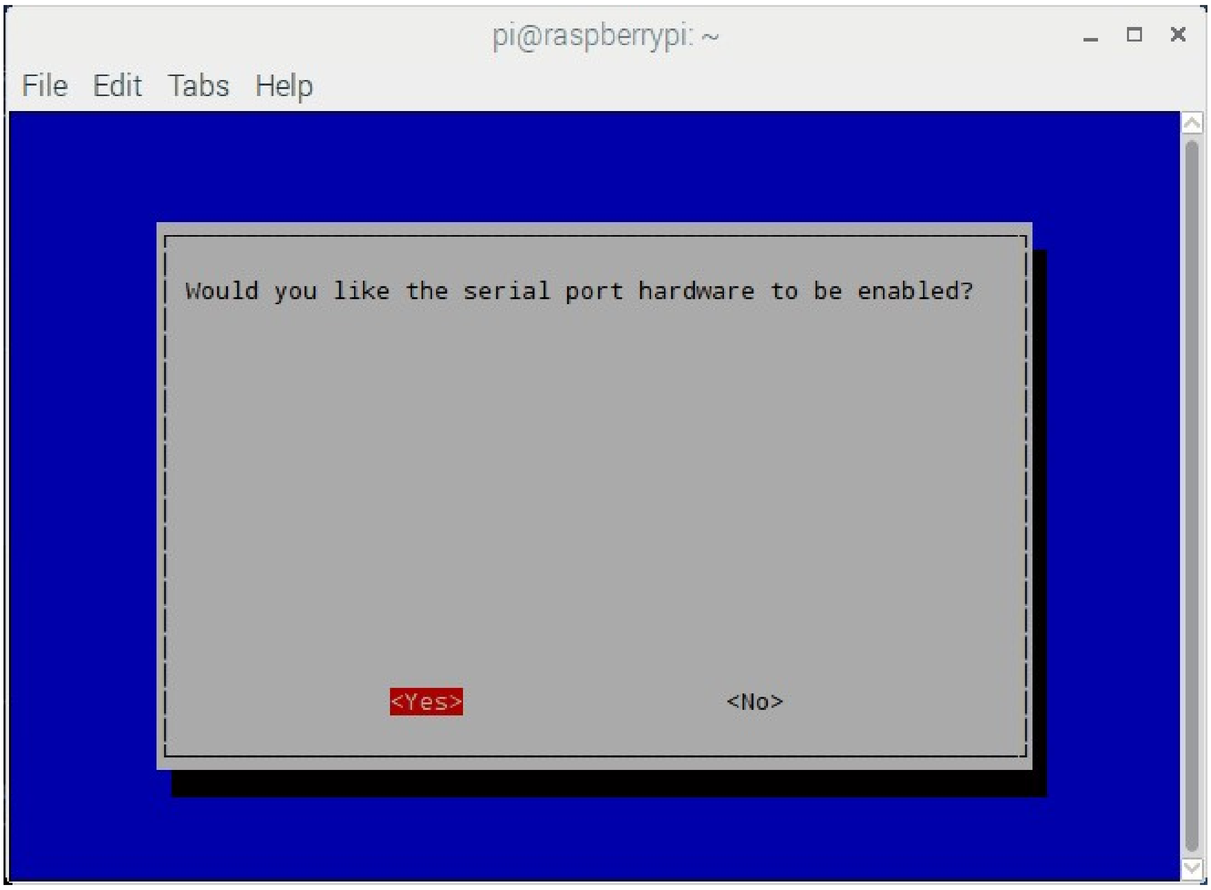
Then it will ask for login shell to be accessible over Serial, select **No** shown as follows.



```
Would you like a login shell to be accessible over
serial?

              <Yes>                    <No>
```

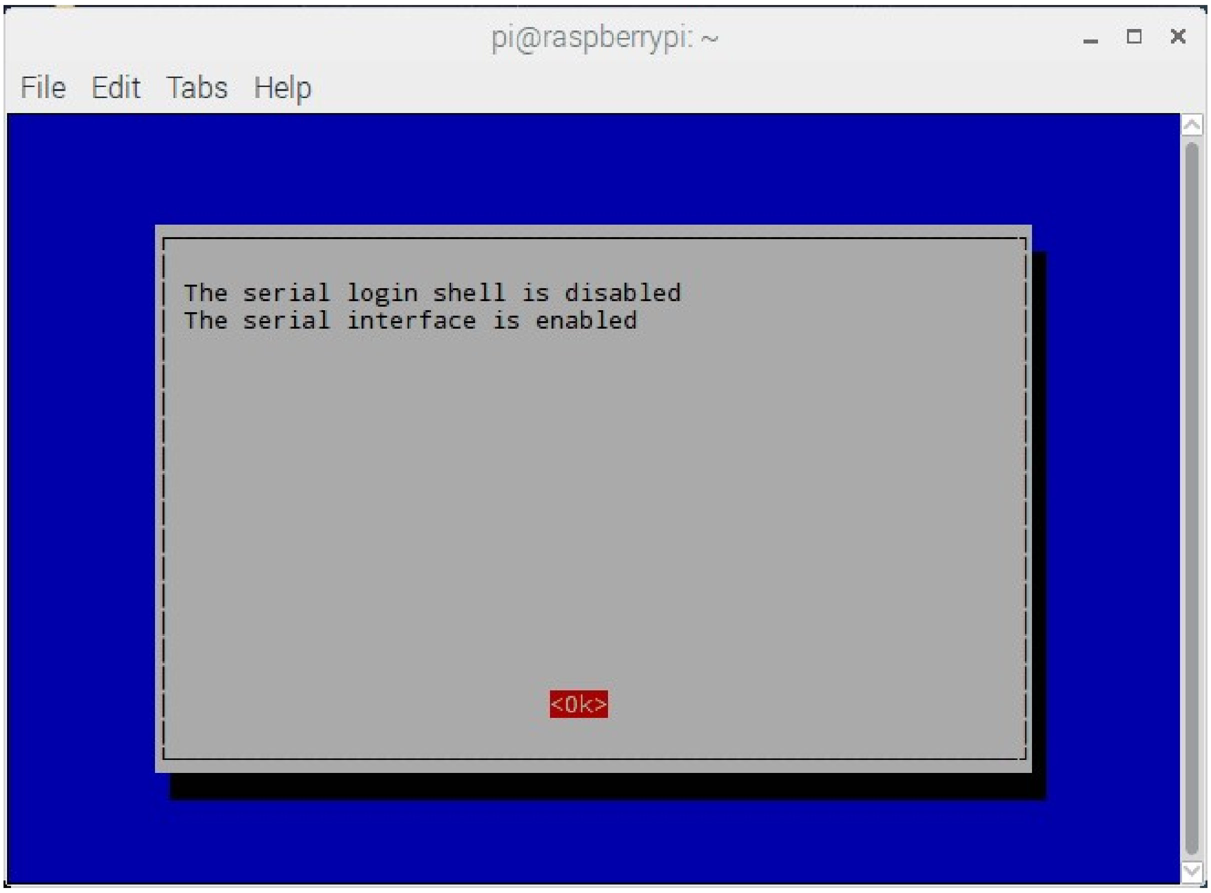
At the end, it will ask for enabling Hardware Serial port, select **Yes**,



```
Would you like the serial port hardware to be enabled?

          <Yes>                    <No>
```

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



```
The serial login shell is disabled
The serial interface is enabled
```

Then, reboot the Raspberry Pi.

Serial Port for UART Communication

By default, mini UART is mapped to UART pins (TX and RX) while PL011 is connected to on-board Bluetooth module on Raspberry Pi 3.

In previous version of Raspberry Pi models, PL011 is used for Linux Console output (mapped to UART pins) and there is no on-board Bluetooth module.

After making above configuration, UART can be used at UART pins (GPIO14 and GPIO15).

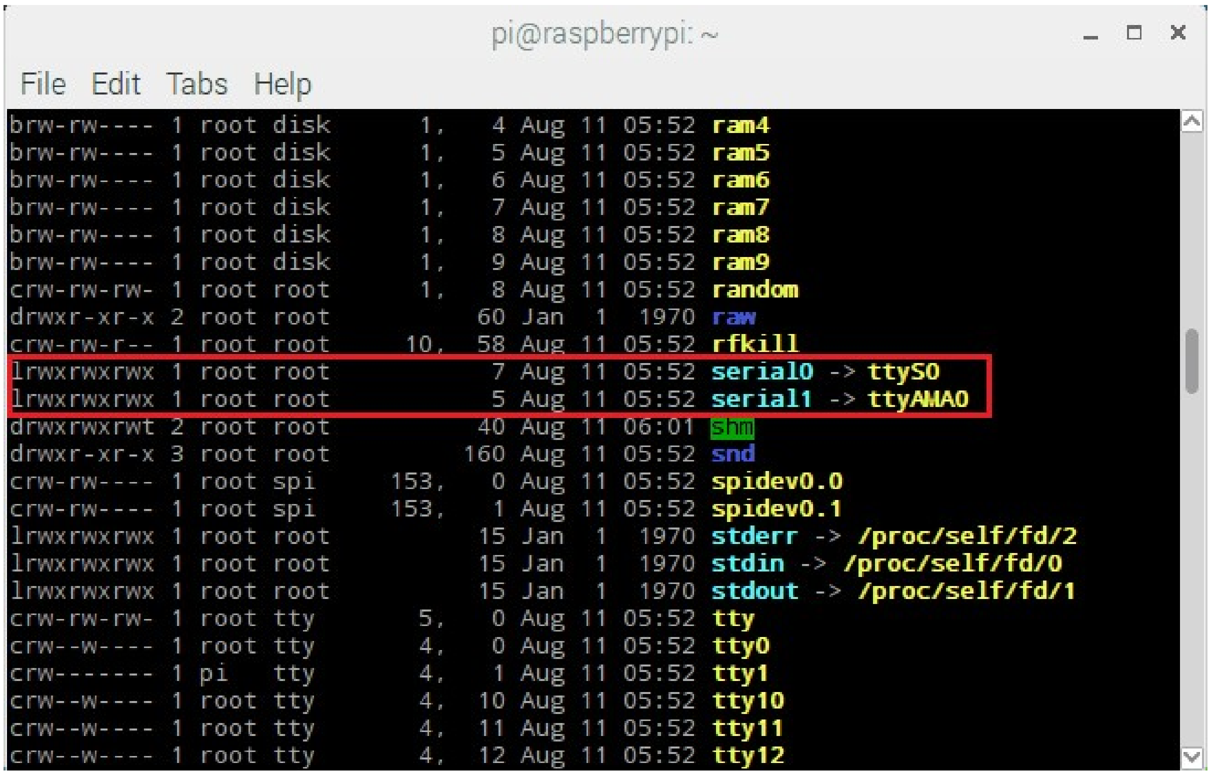
To access mini UART in Raspberry Pi 3, **ttyS0** port is assigned. And to access PL011 in Raspberry Pi 3 **ttyAMA0** port is assigned. But in other models of Raspberry Pi, there is only **ttyAMA0** port is accessible.

Hardware UART port i.e. GPIO14(TXD) and GPIO15 (RXD) is also known as **serial0** while other UART port which is connected to Bluetooth module is known as **serial1.**These names are created as serial aliases for Raspberry Pi version portability.

We can check which UART i.e. mini UART (ttyS0) or PL011 UART (ttyAMA0) is mapped to UART pins (GPIO14 and GPIO15). To check UART mapping, enter following commands.

```
ls -l /dev
```

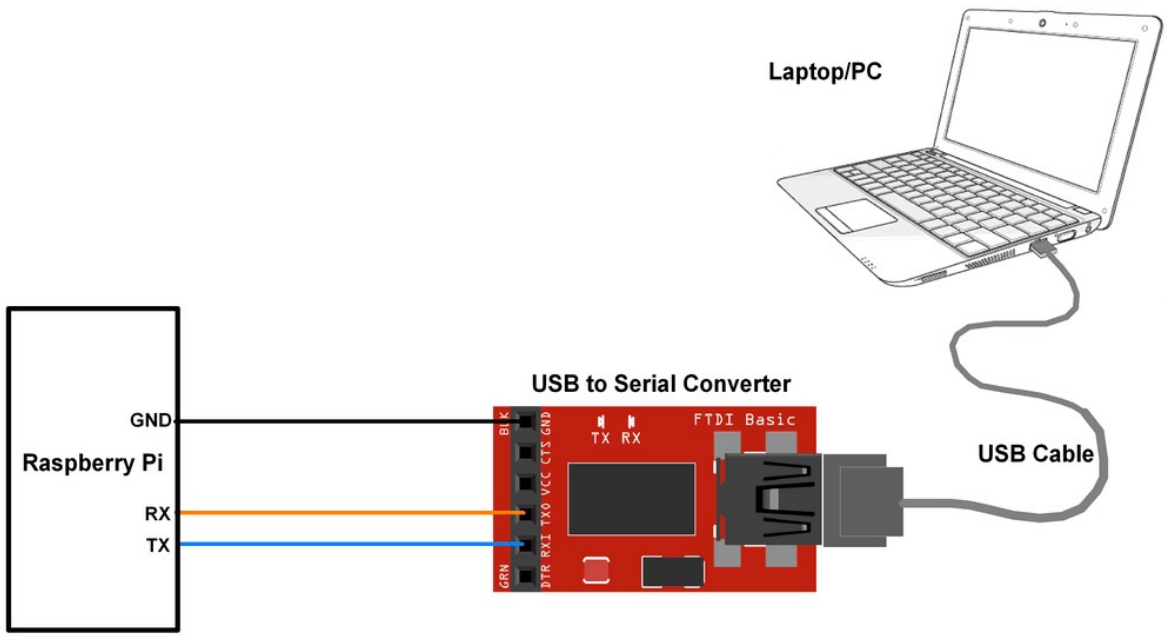
The UART mapping for /dev/ttyS0 and /dev/ttyAMA0 is shown below,



```
brw-rw---- 1 root disk      1,   4 Aug 11 05:52 ram4
brw-rw---- 1 root disk      1,   5 Aug 11 05:52 ram5
brw-rw---- 1 root disk      1,   6 Aug 11 05:52 ram6
brw-rw---- 1 root disk      1,   7 Aug 11 05:52 ram7
brw-rw---- 1 root disk      1,   8 Aug 11 05:52 ram8
brw-rw---- 1 root disk      1,   9 Aug 11 05:52 ram9
crw-rw-rw- 1 root root      1,   8 Aug 11 05:52 random
drwxr-xr-x 2 root root          60 Jan  1  1970 raw
crw-rw-r-- 1 root root     10,  58 Aug 11 05:52 rfkill
lrwxrwxrwx 1 root root           7 Aug 11 05:52 serial0 -> ttyS0
lrwxrwxrwx 1 root root           5 Aug 11 05:52 serial1 -> ttyAMA0
drwxrwxrwt 2 root root          40 Aug 11 06:01 shm
drwxr-xr-x 3 root root         160 Aug 11 05:52 snd
crw-rw---- 1 root spi     153,   0 Aug 11 05:52 spidev0.0
crw-rw---- 1 root spi     153,   1 Aug 11 05:52 spidev0.1
lrwxrwxrwx 1 root root          15 Jan  1  1970 stderr -> /proc/self/fd/2
lrwxrwxrwx 1 root root          15 Jan  1  1970 stdin -> /proc/self/fd/0
lrwxrwxrwx 1 root root          15 Jan  1  1970 stdout -> /proc/self/fd/1
crw-rw-rw- 1 root tty       5,   0 Aug 11 05:52 tty
crw--w---- 1 root tty       4,   0 Aug 11 05:52 tty0
crw------- 1 pi   tty       4,   1 Aug 11 05:52 tty1
crw--w---- 1 root tty       4,  10 Aug 11 05:52 tty10
crw--w---- 1 root tty       4,  11 Aug 11 05:52 tty11
crw--w---- 1 root tty       4,  12 Aug 11 05:52 tty12
```

Test serial communication in between Raspberry Pi and PC

To test that our Serial communication is working or not make the connections shown in below figure.



Raspberry Pi Interface with PC/Laptop for Serial Communication

Open Terminal on Laptop/PC to receive the data which will be transmitted from the Raspberry Pi.

Now, enter the following command to transmit data from Raspberry Pi terminal.

```
echo “Hello” > /dev/ttyS0
```

This command will output “Hello” string on UART port i.e. Tx pin and will display it on terminal application of PC/Laptop.

By default, mini UART is mapped to the GPIO14 (TX) and GPIO15(RX). While PL011 i.e. ttyAMA0 is connected to the on-board Bluetooth module.