Connecting Hardware Peripherals to Raspberry-Pi board

Raspberry-Pi setup through SSH (Headless Configuration of Raspberry-Pi-3 (VNC, Putti))

How to Connect a Raspberry-Pi to the Laptop or PC Display

Required devices:

- Raspberry Pi
- Ethernet Cable
- Laptop/ PC
- SD Card with Raspbian
- Micro USB Cable

How does it Work?

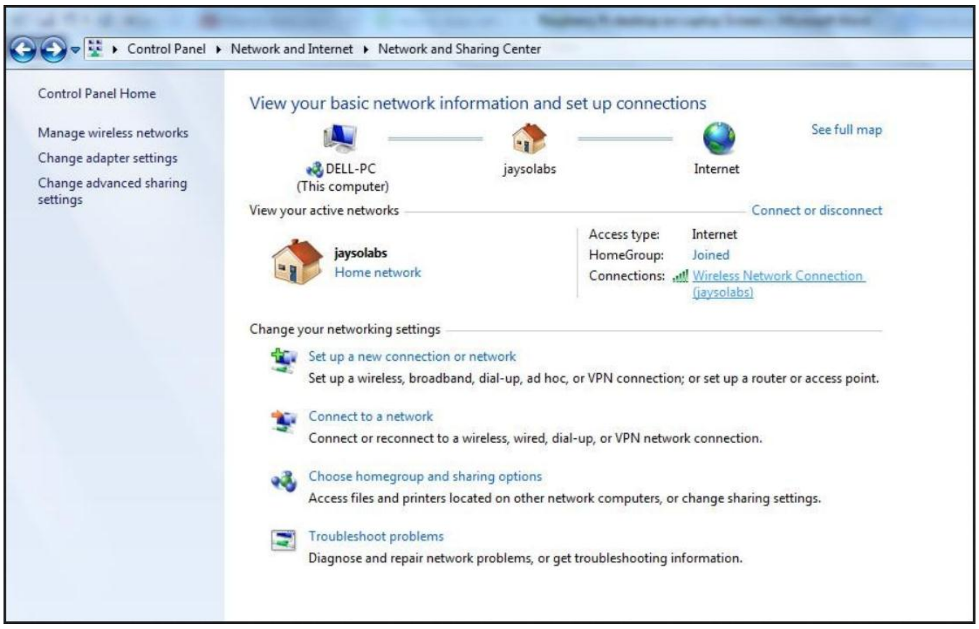
- To connect a Raspberry Pi to a laptop or PC display, you can simply use an Ethernet cable.
- The Raspberry Pi's desktop GUI can be viewed through the laptop or PC display using a 100mbps Ethernet connection between the two.
- We used VNC server software to connect the Raspberry-Pi to our laptop or PC.
- Installing the VNC server on your Raspberry-Pi allows you to see the Raspberry Pi's desktop remotely.

Setting up your Raspberry Pi

- Install Raspbian OS on blank SD card.
- Insert this SD card into Raspberry-Pi board.
- Connect micro USB cable to power the Raspberry-Pi.

Sharing Internet Over Ethernet in Window OS

This step explains how you can share your laptop or PC with the Raspberry Pi via Ethernet cable.



View your basic network information and set up connections

View your active networks

jaysolabs
Home network

Access type: Internet
HomeGroup: Joined
Connections: Wireless Network Connection (jaysolabs)

Change your networking settings

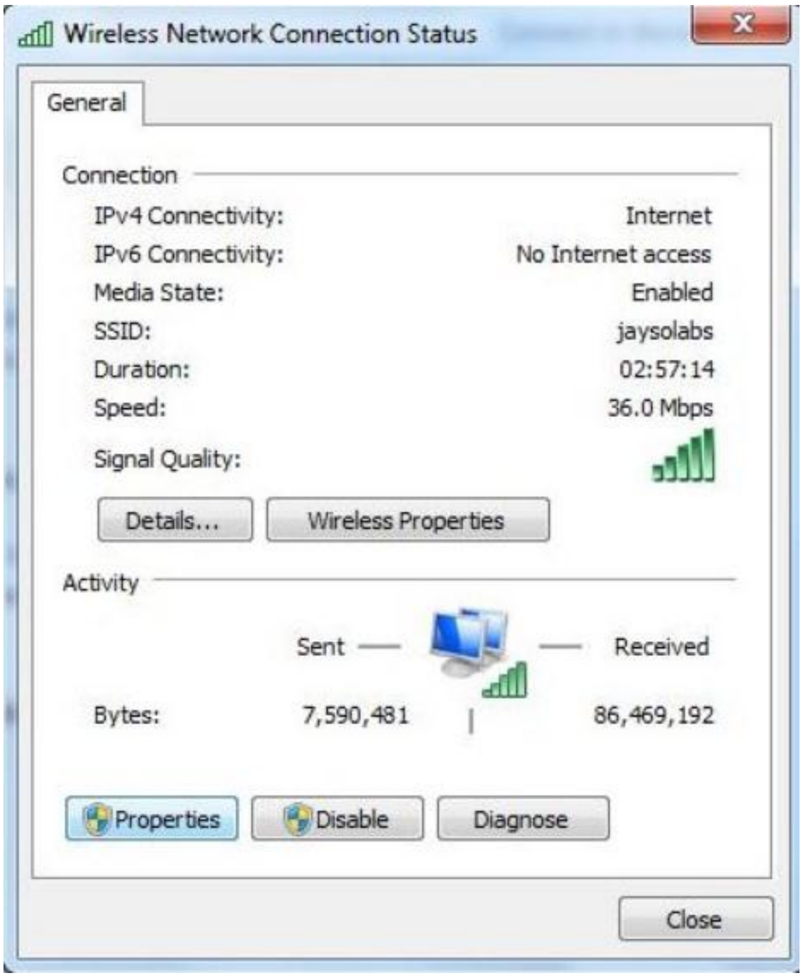
Set up a new connection or network
Set up a wireless, broadband, dial-up, ad hoc, or VPN connection; or set up a router or access point.

Connect to a network
Connect or reconnect to a wireless, wired, dial-up, or VPN network connection.

Choose homegroup and sharing options
Access files and printers located on other network computers, or change sharing settings.

Troubleshoot problems
Diagnose and repair network problems, or get troubleshooting information.

- To share internet with multiple users over Ethernet, go to Network and Sharing Center.
- Then click on the WiFi network
- Double click on Wireless area connection
- Click on Properties (shown below)


Wireless Network Connection Status

General

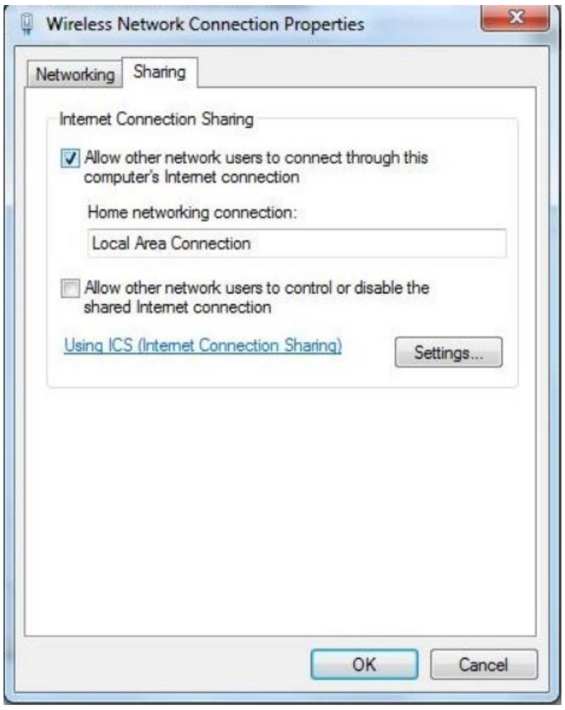
Connection

IPv4 Connectivity:	Internet
IPv6 Connectivity:	No Internet access
Media State:	Enabled
SSID:	jaysolabs
Duration:	02:57:14
Speed:	36.0 Mbps
Signal Quality:	

Activity

	Sent	Received
Bytes:	7,590,481	86,469,192

- Go to “Sharing” tab and click on “Allow other network users to connect”.


Wireless Network Connection Properties

Networking Sharing

Internet Connection Sharing

☑ Allow other network users to connect through this computer's Internet connection

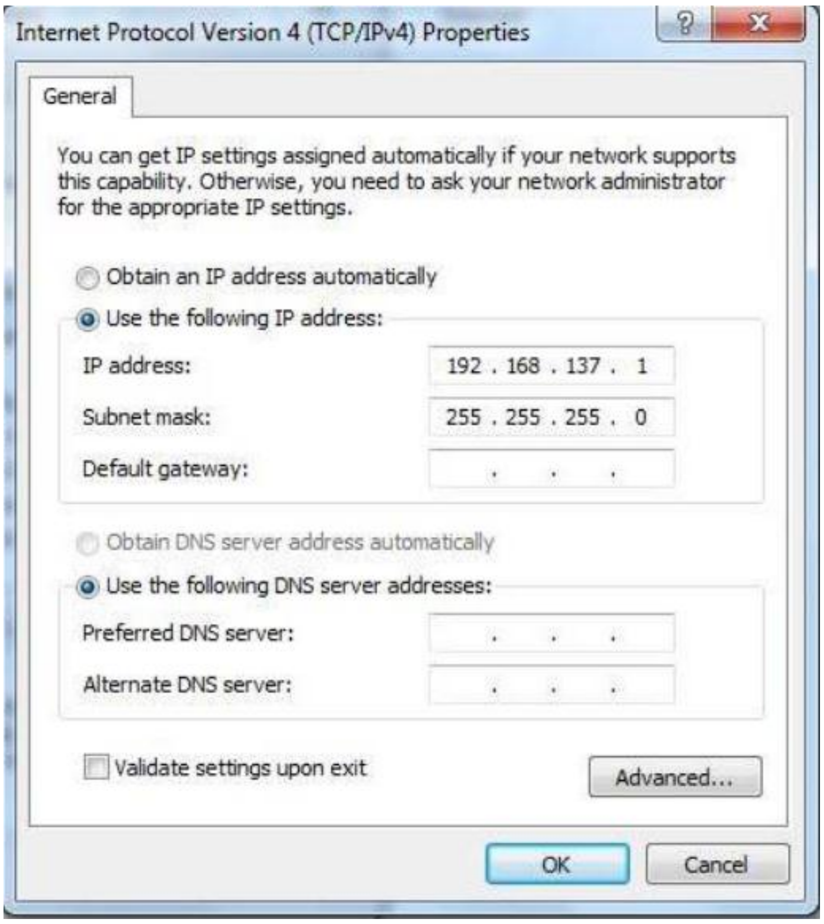
Home networking connection:

Local Area Connection

☐ Allow other network users to control or disable the shared Internet connection

Using ICS (Internet Connection Sharing)

- After this, make sure that the networking connection is changed to “Local Area Connection”
- Now, to check the IP assigned to the network established, click on the new local area connection link created:


Internet Protocol Version 4 (TCP/IPv4) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

- ○ Obtain an IP address automatically
- ● Use the following IP address:

IP address:	192 . 168 . 137 . 1
Subnet mask:	255 . 255 . 255 . 0
Default gateway:	. . .

- ○ Obtain DNS server address automatically
- ● Use the following DNS server addresses:

Preferred DNS server:	. . .
Alternate DNS server:	. . .

☐ Validate settings upon exit

- Now open command prompt.
- Ping the broadcast address of your IP. (Type) E.g. : ping 192.168.137.1
- Stop the ping after 5 seconds.
- To get the IP address of Raspberry Pi in the established network, use the Software “Advance IP Scanner”. It is free software.

Setting up the VNC Server to Connect Your Raspberry Pi to the Laptop or PC Display

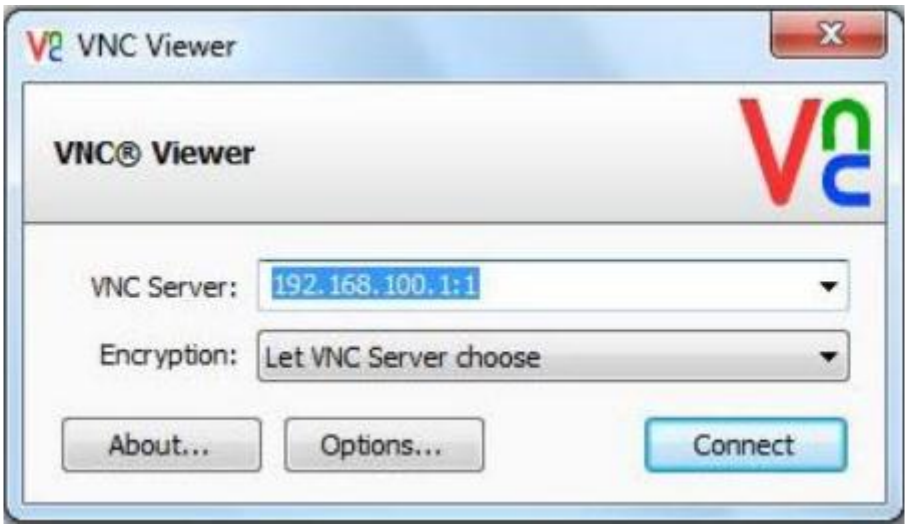
- First install VNC server and Putty on your laptop/ PC
- Open Putty Software, and enter login ID: pi and Password: raspberry.
- After that, enter commands into Putty i.e,

```
$ sudo apt-get update
$ sudo apt-get install tightvncserver
$ vncserver :1
```

- You will be prompted to enter and confirm a password.
- This will be asked only once, during first time setup.
- Enter an 8 digit password.
- Note that this is the password you will need to use to connect to your Raspberry Pi remotely.

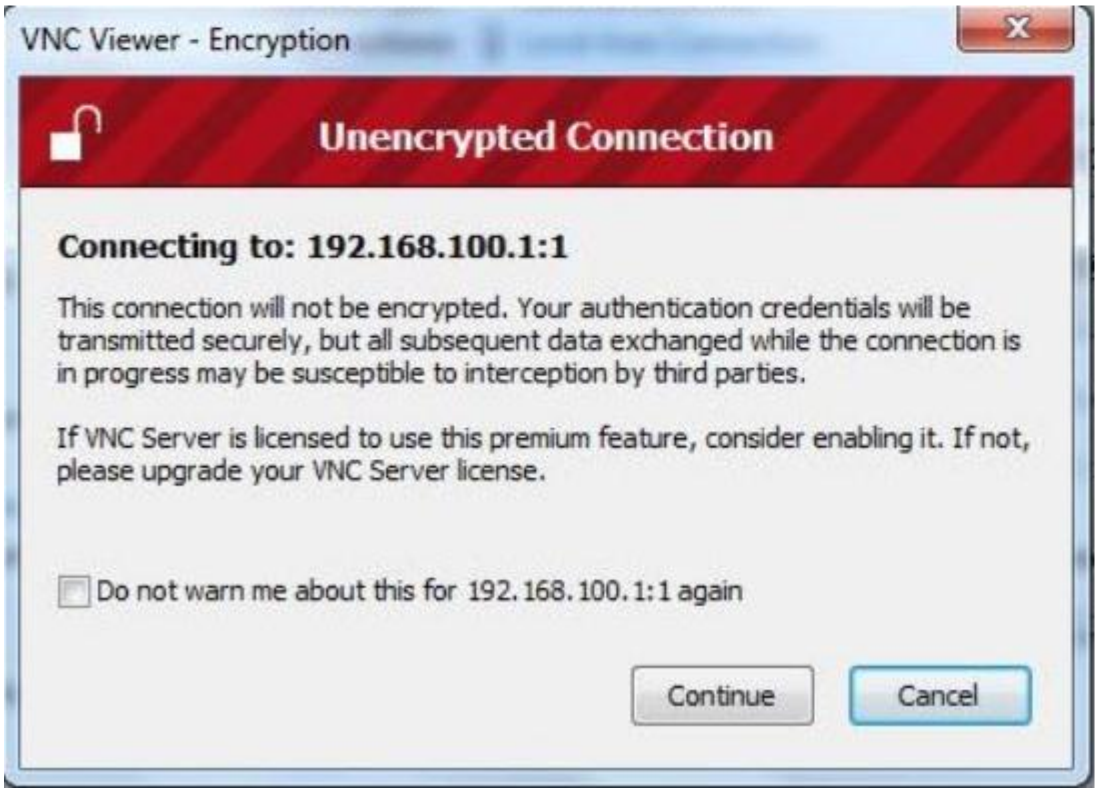
Setting Up the Client Side (Laptop or PC)

- Download VNC client and install it.
- When you first run VNC viewer, you will see following screen



VNC Server: 192.168.100.1:1
Encryption: Let VNC Server choose

- Enter the IP address of your Raspberry Pi given dynamically by your laptop and append with : 1 (denoting port number) and press connect.
- You will get a warning message, press 'Continue':



Unencrypted Connection

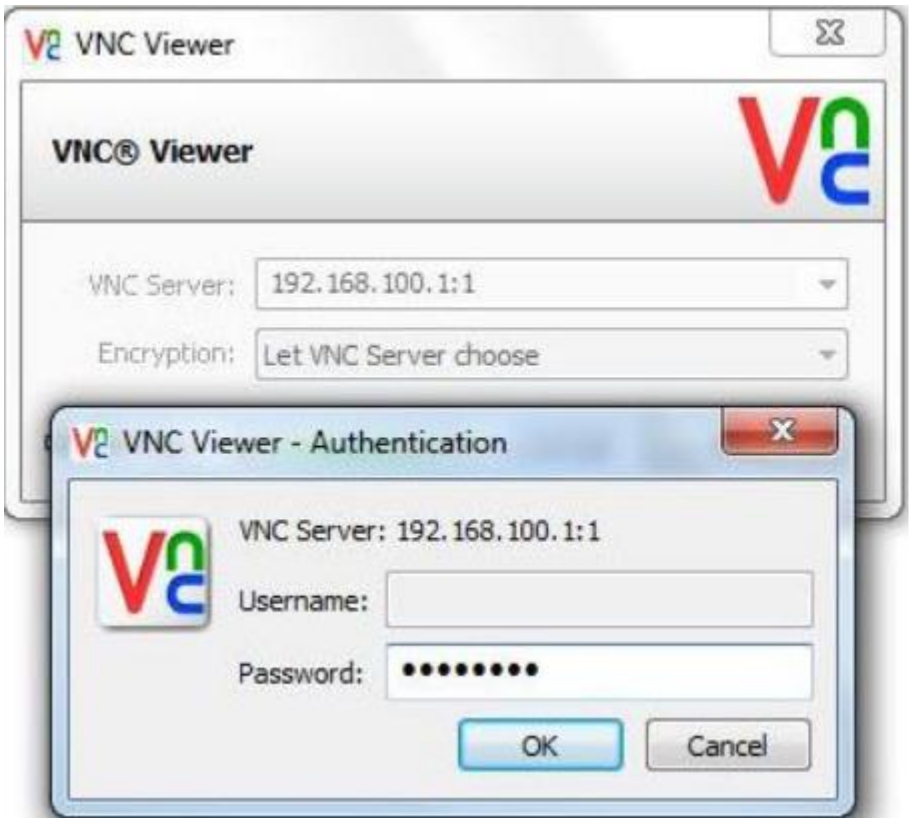
Connecting to: 192.168.100.1:1

This connection will not be encrypted. Your authentication credentials will be transmitted securely, but all subsequent data exchanged while the connection is in progress may be susceptible to interception by third parties.

If VNC Server is licensed to use this premium feature, consider enabling it. If not, please upgrade your VNC Server license.

☐ Do not warn me about this for 192.168.100.1:1 again

- Enter the 8 digit password which was entered in the VNC server installation on your Raspberry Pi:


VNC Viewer

VNC® Viewer

VNC Server: 192.168.100.1:1

Encryption: Let VNC Server choose

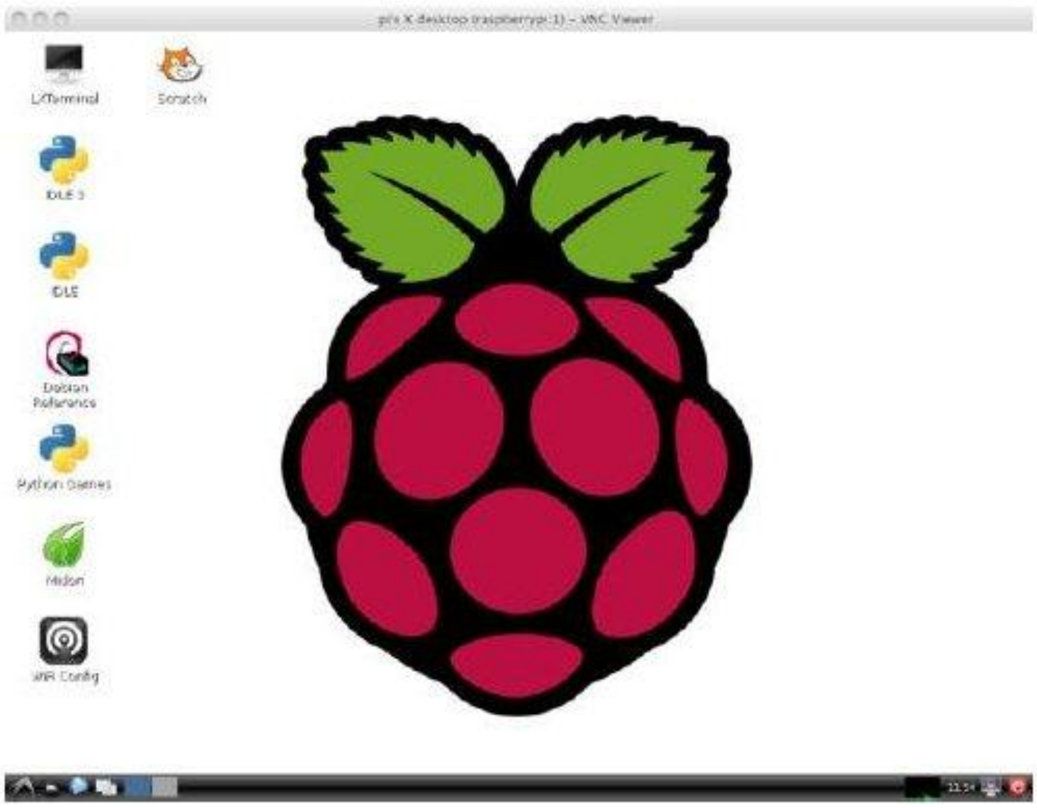
VNC Viewer - Authentication

VNC Server: 192.168.100.1:1

Username:

Password: ●●●●●●●●●

- Finally, the Raspberry Pi desktop should appear as a VNC window.
- You will be able to access the GUI and do everything as if you are using the Pi's keyboard, mouse, and monitor directly.



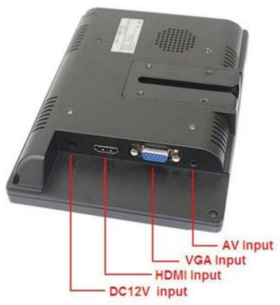
Raspberry-Pi setup using mouse, keyboard and monitor

To work with Raspberry-Pi, we have to connect some peripherals to it. They are as follows

- A monitor with power cable and data cable
- A micro USB power supply
- A wired keyboard and mouse, or a wireless keyboard and mouse
- A micro SD card

Monitors - HDMI

- There are several different types of monitor that you can use with the Raspberry Pi:



- Most modern television sets and monitors have an HDMI port, and are the easiest to get working with the Raspberry Pi.
- You can use an HDMI cable to connect the Raspberry Pi directly to the television or monitor.

VGA

- Some old monitors have a VGA port.
- These can be trickier to use as you'll need an HDMI-to-VGA converter, which can change digital video to analogue video.
- A simple port adapter won't work.

Power supplies

- For Raspberry Pi 3, it's recommended to use a 5V, 2.5A power supply.

Mobile device charger

- Many mobile devices are powered using a 5V micro USB charger.
- These can often be used to power the Raspberry Pi, although it's worth checking that they provide sufficient voltage and current (5V / 1.2 - 2.5A).

Keyboard and mouse

Wired keyboard and mouse

- Any standard USB keyboard and mouse can be used with the Raspberry Pi. These plug and play devices will work without any additional driver. Simply plug them into the Raspberry Pi and they should be recognized when it starts up.

Bluetooth keyboard and mouse

- Bluetooth keyboards and mice can work with the Raspberry Pi, but your success rates will vary depending on the model and manufacturer. It's best to consult the manufacturer's documentation to see whether or not a device is compatible with the Raspberry Pi.

SD cards

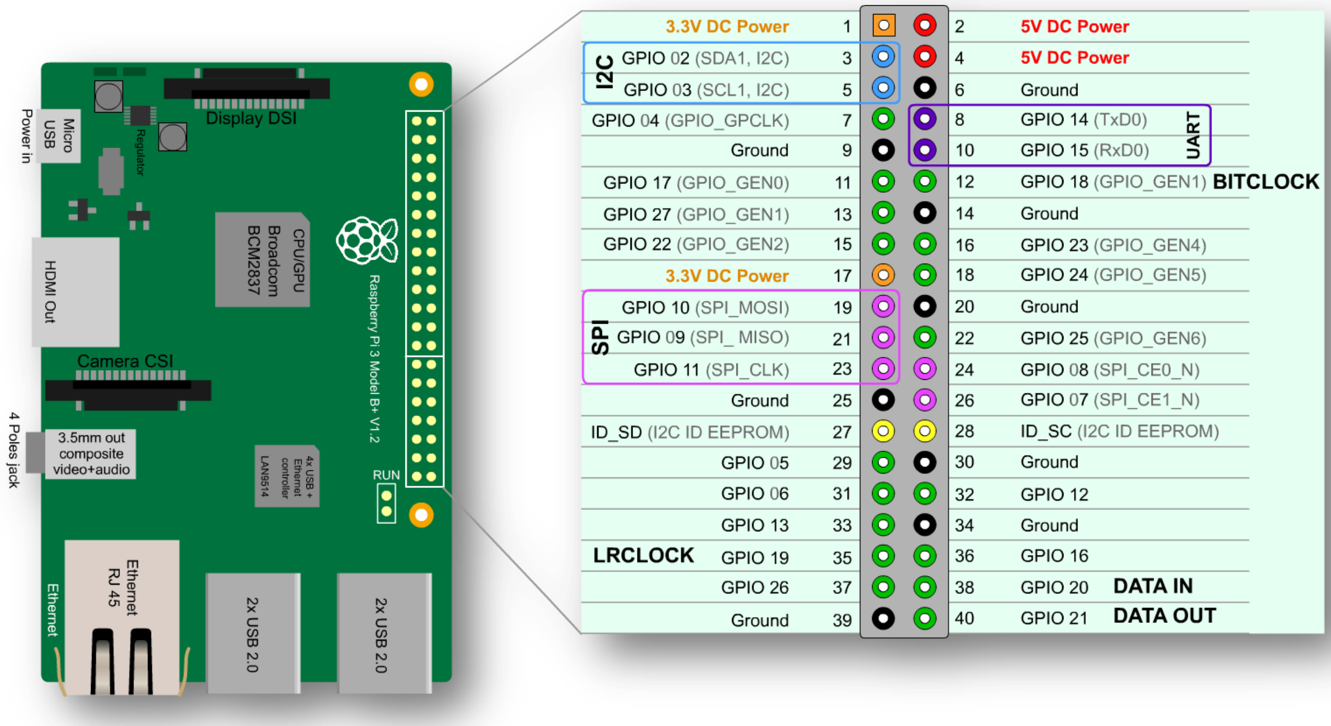
- The latest version of Raspbian, the default operating system recommended for the Raspberry Pi, requires an 8GB (or larger) micro SD card.
- Not all SD cards are made equal, and some have higher failure rates than others.
- Any 8GB SD card will work, although you'll need to follow the software setup guide to learn how to load an operating system onto the card.

Understanding GPIO pins on Raspberry Pi board and its use in program

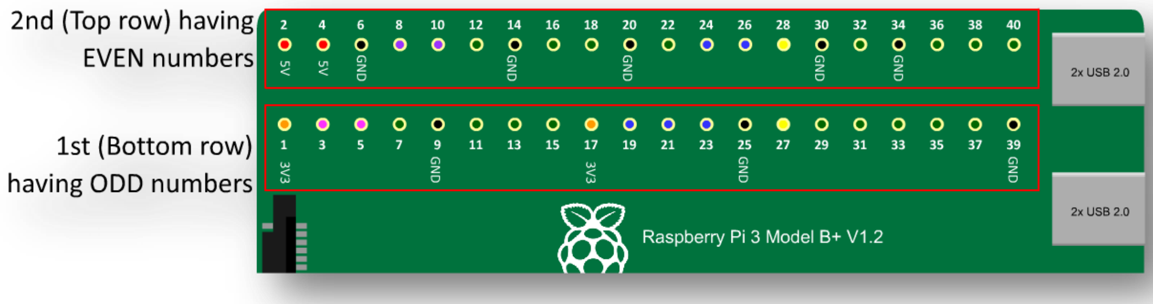
Aim/Objectives:

- To understand the GPIO pins of Raspberry-Pi 3
- To program the GPIO pins of Raspberry-Pi 3 using Python

Introduction:



- Raspberry Pi 3 Model B is the latest version of raspberry pi board.
- It is released on 29 February.
- The above figure shows the Raspberry Pi 3 Model B and It's GPIO pins
- General-purpose input/output (GPIO) is a generic pin on an integrated circuit or computer board whose behavior—including whether it is an input or output pin—is controllable by the user at run time.



- There are 40 pins available on board of Raspberry pi 3 model B.
- The Pins are arranged in a 2×20 fashion as shown in the figure above
- Out of these, 26 pins are GPIO pins
- As you can observe, the numbers to the pins are given in zigzag manner.
- The first (bottom) row starts with number '1'. So the pins in this row have odd numbers i.e. from 1 to 39.
- The 2nd (Top) row starts with number '2'. So the pins in this row have even numbers i.e. from 2 to 40.
- Out of 40 pins,
    a. 26 pins are GPIO pins,
    b. 8 pins are Ground (GND) pins,
    c. 2 pins are 5V power supply pins
    d. 2 pins are 3.3V power supply pins
    e. 2 pins are not used
- Now if you're coming to the Raspberry Pi as an Arduino user, you're probably used to referencing pins with a single, unique number.
- In Raspberry Pi there are two different numbering schemes for referencing Pi pin numbers:
    1. Broadcom chip-specific pin numbers (BCM)
    2. Physical pin numbers (BOARD)
- You're free to use either number-system.
- The programs require that you declare which scheme you're using at the very beginning of your program.
- In a program, at a time, you can use **only one** number scheme.

Broadcom chip-specific pin numbers (BCM)

- BCM - Broadcom pin number, commonly called "GPIO", these are the ones you probably want to use with RPi.GPIO
- The parameter used for this system is (GPIO.BCM).
- This is a lower level way of working - it refers to the channel numbers on the Broadcom SOC.
- To use this system, you have to always work with a diagram describing which channel number goes to which pin on the RPi board.
- Your script could break between revisions of Raspberry Pi boards.
- In this system
    a. 26 GPIO pins are named as GPIO 01 to GPIO 26

Physical Numbering System (BOARD)

- This system uses physical - Numbers corresponding to the pin's physical location on the header
- The numbers printed on the board are physical numbering system.
- The parameter used for this system is (GPIO.BOARD).
- The advantage of using this numbering system is that your hardware will always work, regardless of the board revision of the RPi.
- You will not need to rewire your connector or change your code.
- In this system
  a. 26 GPIO pins are named between 0 to 40

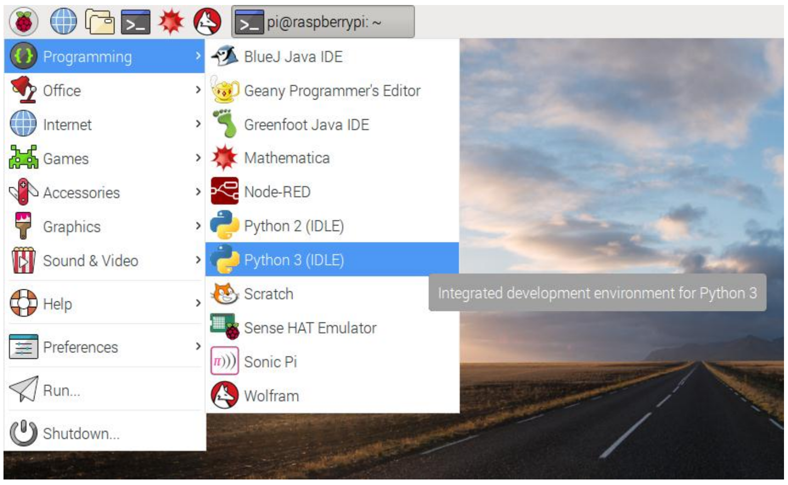
The below table summarizes the pinout of Raspberry-Pi in both the number systems.

Sr No.	Pins		BOARD Pin No	BCM Pin No
1	3.3V		1, 17	1, 17
2	5V		2, 4	2, 4
3	Ground		6,9,14,20,25,30,34,39	6,9,14,20,25,30,34,39
4	UART	TXD	8	GPIO 14
		RXD	10	GPIO 15
5	I2C1	SDA1	3	GPIO 2
		SCL1	5	GPIO 3
	I2C0	SDA0	27	ID_SD
		SCL0	28	ID_SC
6	SPI0	MOSI 0	19	GPIO 10
		MISO 0	21	GPIO 9
		SCLK 0	23	GPIO 11
		CE 0	24	GPIO 8
	SPI1	MOSI 1	38	GPIO 20
		MISO 1	35	GPIO 19
		SCLK 1	40	GPIO 21
		CE 1	26	GPIO 7

The Python IDLE shell and command line

To use the Python IDLE IDE for programming in Raspberry-Pi use the following

- Open **Python 3** from the main menu:



- Or open terminal window and type the command *sudo idle 3.5* and press enter
- Install all libraries required for Buzzer as given above.
- Write the program as per algorithm given below
- Save with **Ctrl + S** and run with **F5**.
- See output on Python Shell or Terminal Window.

Raspberry Pi GPIO programming using Python

- The Raspberry Pi is often used in conjunction with other hardware to create interesting electronic projects.
- The Pi 3 comes with 40 GPIO pins that you can use to interface with various hardware devices—for both receiving data from them or for writing data to them.
- To do this, we have to program the GPIO pins. To do this, special libraries in Python are used. To include these libraries in the program, the command used is 'import'
- This way, we can write applications to both read and also to control devices, i.e., turn them on and off, etc.
- The default operating system used in Raspberry-Pi is Raspbian.
- The Python package used for Raspberry Pi GPIO programming is RPi.GPIO. It is already installed in Raspbian.
- If you are using any other operating system, the package can be installed by using the following command:

  $ sudo pip install RPi.GPIO
- There are **important 8 steps** in the programming of Raspberry-Pi using Python as follows

1. Import the RPi.GPIO library using the following command

   import RPi.GPIO as GPIO
2. Import the Time library using the following command

```
import time
```

3. Set numbering scheme to be used. The method used for this is GPIO.setmode(). We will use physical number scheme. So the method is written as

   GPIO.setmode(GPIO.BOAD)

4. Set the pin mode as INPUT or OUTPUT using the commands
   ```
   GPIO.setup(channel, GPIO.IN)
   GPIO.setup(channel, GPIO.OUT)
   ```
5. Read input using following command
   ```
   GPIO.input(pin no)
   ```
6. Write output using following comman
   ```
   GPIO.output(pin no, state)
   ```
7. Give delay using command using following command
   ```
   time.sleep(1) # delay for 1 second
   ```
8. Clean up GPIO and exit using following commands
   ```
   GPIO.cleanup()
   print(“Exiting...”)
   ```

- You must clean up the pin set-ups before your program exits otherwise those pin settings will persist, and that might cause trouble when you use the same pins in another program.
- The Pi ‘expresses its displeasure’ with a warning.
- To clean up the entire set of pins, invoke GPIO.cleanup().
- If you want only a few pins to be cleaned up, then the pin numbers should be provided as GPIO.cleanup (channel_list).
- Anyway, you can suppress the warning messages by calling GPIO.setwarnings (False).
- Save the program with proper name. The file is saved with extension ‘.py’.
- The IDE named ‘IDLE’ used for programming is an interpreter and not a compiler. So to run the python program, we need to give the super user permission as follows.

Studying Connectivity and Configuration of Raspberry Pi board with basic peripherals (LEDs)

Aim/Objectives:

- To understand the concept of Led bar
- To understand the common anode & common cathode configuration.
- To interface LED bar with Raspberry Pi.
- Generate various patterns on LED bar.

Software:

- Raspbian OS
- IDLE editor

Hardware Modules:

- Raspberry Pi Board
- LED bar
- Monitor

Theory:

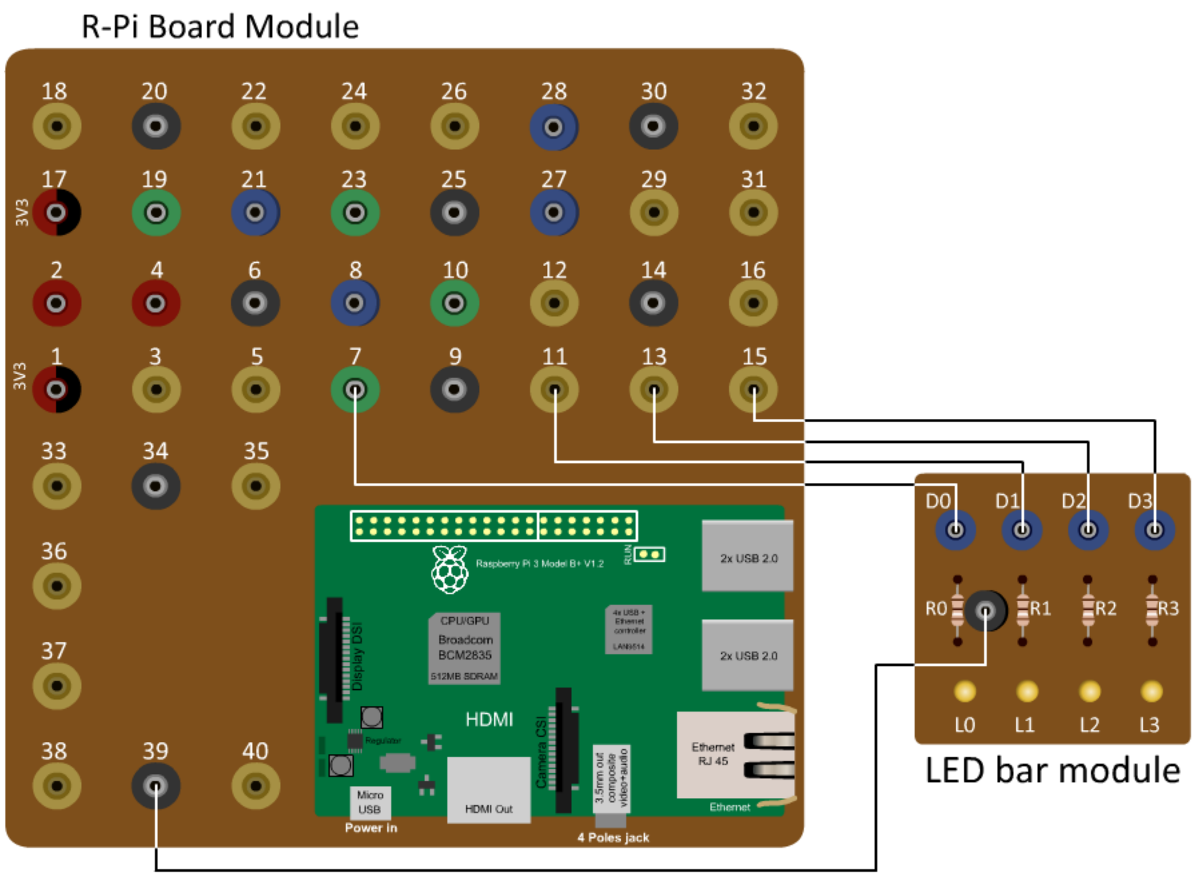
Introduction to "LED"

- LED is a Light Emitting Diode.
- Light emitting diode is a two lead semiconductor light source. It is a p-n junction diode, which emits light when it is activated.
- When a suitable voltage is applied to the leads, electrons are able to recombine with electron holes within the device, and the color of light (corresponding to the energy of photon) is determined by the energy band gap of the semiconductor.
- It has two terminals named as 'anode (+ve)' and 'cathode (-ve)'.
- Battery is connected to these two terminals.
- When LED is forward biased, it emits light.
- In LED bar number of LEDs are connected in series (in our case 8 LEDs are connected)
- LED bar has two configurations as
- Common Anode: In this, anode terminal of all the LEDs are made common and connected to the VCC (+5v). By controlling cathode terminal we can make LED ON or OFF (current sourcing).
- Common Cathode: In this, cathode terminal of all the LEDs are made common and connected to the Ground (0v). By controlling anode terminal we can make LED ON or OFF (current sinking).

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.

Interface diagram:



Steps for assembling circuit:

- Connect led bar module pins from D0- D3 to Raspberry Pi GPIO pins 7, 11, 13, 15 respectively.
- Connect led bar module pin COM to the GND pin of Raspberry-Pi module.

Procedure:

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

Algorithm:

- Import GPIO and Time library
- Set mode i.e. GPIO.BOARD
- Set GPIO 8 pins as a Output pin
- Print message “ON”
- After 1 second time delay, Make all the leds ON one by one
- Print message “OFF”
- After 1 second time delay, Make all the leds OFF one by one

Observation:

- See output on Command prompt or Python shell.