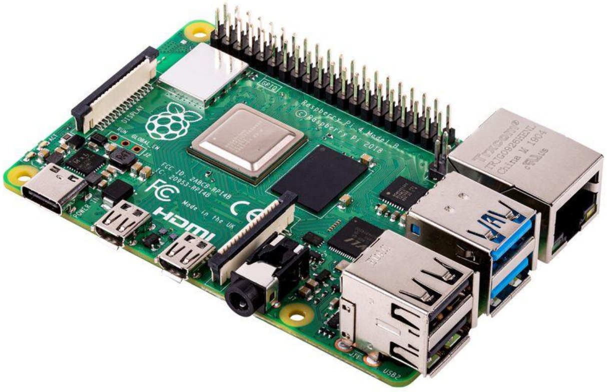
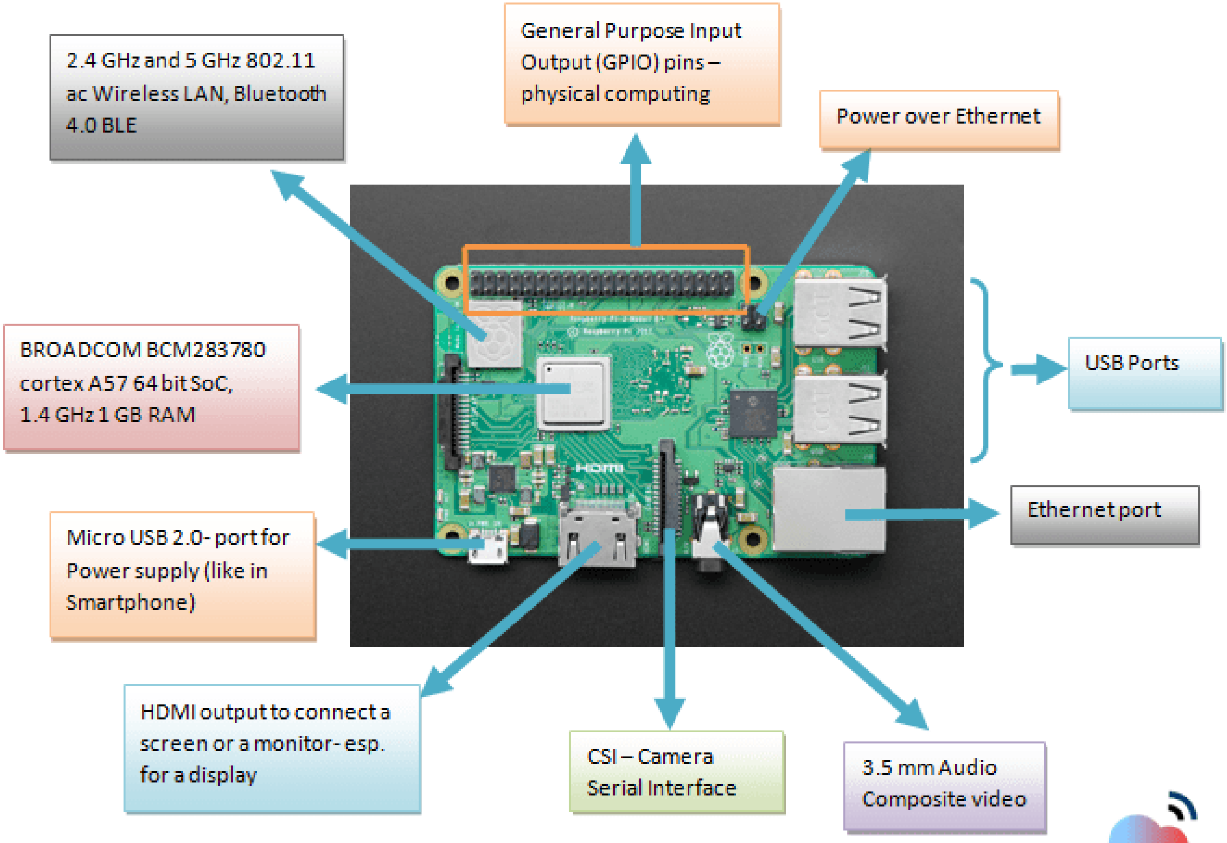


# RASPBERRY PI WITH THE INTERNET OF THINGS - IOTEDU

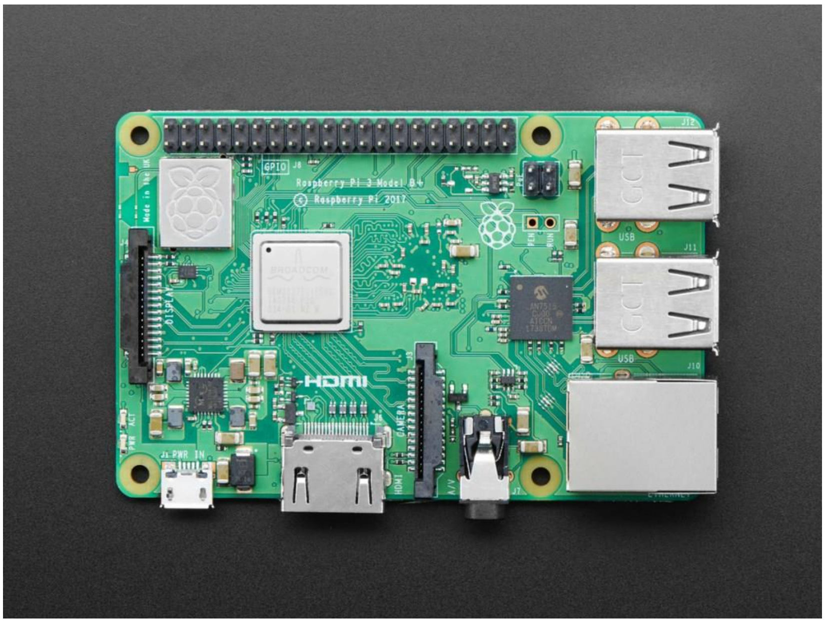
Chapter 1 – Introduction to Raspberry Pi

# 1. An Introduction to the Raspberry Pi


2.4 GHz and 5 GHz 802.11 ac Wireless LAN, Bluetooth 4.0 BLE
General Purpose Input Output (GPIO) pins – physical computing
Power over Ethernet
BROADCOM BCM283780 cortex A57 64 bit SoC, 1.4 GHz 1 GB RAM
USB Ports
Ethernet port
Micro USB 2.0- port for Power supply (like in Smartphone)
HDMI output to connect a screen or a monitor- esp. for a display
CSI – Camera Serial Interface
3.5 mm Audio Composite video

**Anatomy of Raspberry Pi 3B+ Model**

Iot4beginners



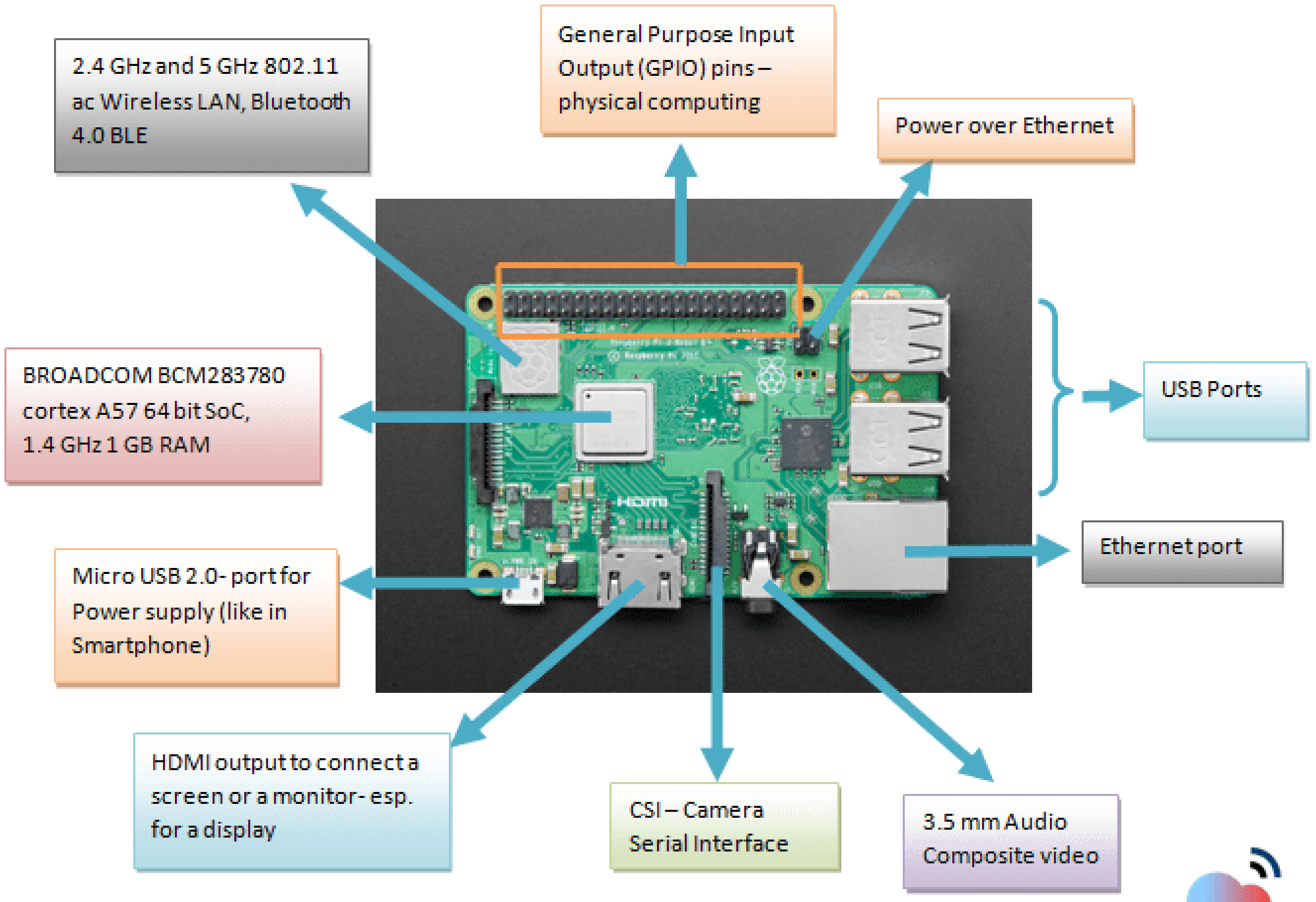
Raspberry Pi 3B+ Model

The Raspberry Pi is also known as RPi is a small, “Single Board Computer” (SBC) developed in the United Kingdom by a charity called Raspberry Pi Foundation. Their main aim is to educate basic computer science in developing countries. They had released several generations of Raspberry Pi.

- Raspberry Pi Model B (First generation) – February 2012
- Raspberry Pi Model A – February 2013
- Raspberry Pi Compute Model- April 2014
- Raspberry Pi Model B+ – July 2014
- Raspberry Pi 2 – February 2015
- Raspberry Pi Zero- November 2015
- Raspberry Pi 3  Model B – February 2016
- Raspberry Pi Zero W – February 2017
- Raspberry Pi 3 Model B+ – March 2018

The Raspberry Pi was first released on 29 February 2012 with an introductory price of US$35. The System On Chip (SoC) used is Broadcom BCM2837B0 and compatible with plenty of Operating Systems such as Linux, FreeBSD, NetBSD, OpenBSD, Plan 9, RiscOS and Windows 10 IoT Core. The Raspberry Pi 3+ uses a Broadcom BCM2837B0 SoC with a 1.4 GHz 64-bit quad-core ARM Cortex-A53 processor, with 512 KB shared L2 cache. Besides, it comes with a 40 pin General Purpose Input Output (GPIO) connector. The Keyboard and mouse can be connected with the USB, and other components such as sensors can be connected with the GPIO pins. The RPi has no built-in real-time clock, so it cannot track the time of the day. It comes with lots of accessories like Camera, SenseHat ( has lots of built-in sensors), Gertboard, Official Display and so on.

## Anatomy of Raspberry Pi 3 B+ model


2.4 GHz and 5 GHz 802.11 ac Wireless LAN, Bluetooth 4.0 BLE
General Purpose Input Output (GPIO) pins – physical computing
Power over Ethernet
BROADCOM BCM283780 cortex A57 64 bit SoC, 1.4 GHz 1 GB RAM
USB Ports
Ethernet port
Micro USB 2.0- port for Power supply (like in Smartphone)
HDMI output to connect a screen or a monitor- esp. for a display
CSI – Camera Serial Interface
3.5 mm Audio Composite video

**Anatomy of Raspberry Pi 3B+ Model**

Iot4beginners

# HARDWARE SPECIFICATIONS

The Raspberry Pi 3B+ model was launched in March 2018, comes with a CPU, GPU, USB ports, 40 pin header (GPIO pins), WiFi, Bluetooth, POE (Power over Ethernet). The CPU has an Arithmetic Logic Unit (ALU) and a control unit and the primary storage is the Memory. A GPU (Graphics Processing Unit) is nothing but a specialized circuit that is very efficient at processing images and computer graphics, better than a CPU. This kind of design that packs all the components required to power a computer onto a single chip is called **System on Chip (SoC)**. Nowadays, SoCs are implemented into Smartphones and Tablets. The System on Chip has both CPU and GPU combined together in this model to give enormous speed compared to the previous versions. At the back side of the Raspberry Pi, there is a slot allocated for the Secondary Storage, where you will be installing your OS.

## CPU and GPU

A CPU is always considered to be a brain of a computer, and it works same for Raspberry Pi too. It has a 64-bit capacity and is responsible for **arithmetic and logical units**. It also comes with a control unit. The GPU ie, the Graphics Processing Unit is very efficient in **image calculation and processing**. The Broadcom video core cable is included in the device for video games.

## Clock Speed and Memory

The Raspberry Pi comes with a quad-core ARM Cortex A53 and a RAM memory (**Primary**) of 1 GB. It consists of a clock speed of 1.4 GHz Broadcom BCM2837B0. At the back side of the Raspberry Pi, there is an additional slot is given for the **secondary** memory, this takes in a micro SD card, similar to which all the cameras and smartphones use for their additional storage. You will be installing the OS in the SD card. It's better if you use 16 GB or 32

GB memory, because the OS will be taking more space. If there is anything wrong with the Raspberry Pi, you can replace the SD card with new OS installed. Or else, you can keep many SD cards with installed OS, so that you no need to download and copy in the card again.

## LEDs

There are two LEDs in the device, one is red and another one is green. The red LED indicates the on/off of the Pi whereas, the green LED reads the software in the SD card. So, when you have the OS installed in secondary memory ie, in the given slot, the green LED blinks, if not then there is a fault in your SD card. You can simply replace the card with new OS installed in that.

## Micro USB 2.0 Power source Connector

The RPi consists of a micro USB for power source, which you use for smartphones. It provides 5 V power to the board and it draws 170 to 200 mA which is more power than the previous version of the model.

## Other specifications

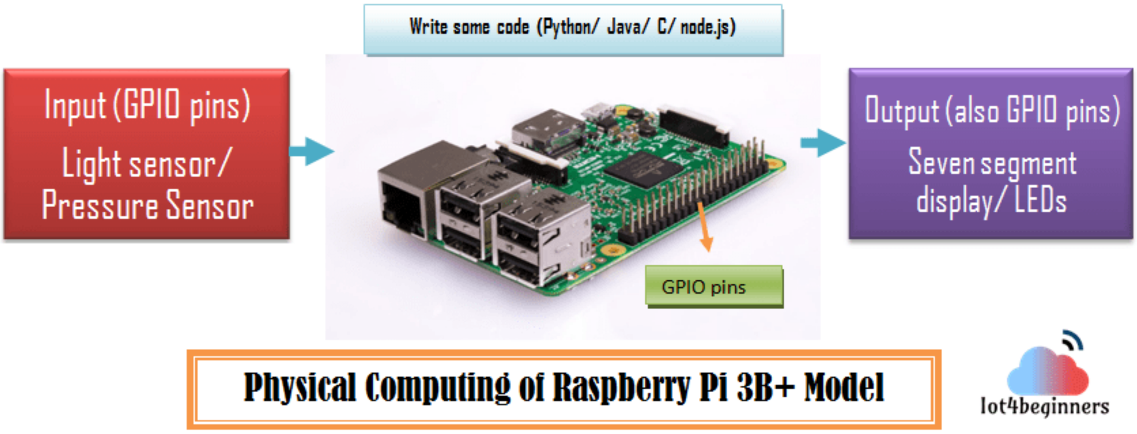
- The Raspberry Pi consists of an **Ethernet**, through which you can connect with LAN cable to your modem/ computer/ laptop.
- There are **4 USB ports** in this model to provide a connection to the peripherals such as mouse, keyboard, etc
- An **HDMI** output port in the RPi is used for the display when connected to monitor or screen.
- **Audio output socket and video composite** reside in a single 4 pole 3.5 mm near the HDMI output.
- It also has a **Camera Serial Interface** where you can connect the Raspberry Pi Camera into that via cable. It is available in the market for $20- $35.
- PoE (power over Ethernet) is an additional feature incorporated in this device that lacks in B model.
- The Raspberry Pi 3 B+ measures 85mm length and 56mm width same as in the B model.

# What is a GPIO Pin?

Building “things” (Systems) that can respond to the world around them is called physical computing. The idea behind this physical computing is to make the “things” to communicate is to use the **Sensors**. These sensors cannot be directly communicated with the computer/laptops that we use everyday. There are special computers like Microcontrollers – Arduino and processors like Raspberry Pi which can communicate with those sensors by the special input-output pins called the **GPIO pins (General Purpose Input Output).** These pins interact with the sensors with a wide variety of input sources or the data provided by the sensor and produce an output. The output can be produced as a sound through the buzzer, start or change of speed of a motor, blink an LED or even a seven segment display.

The Raspberry Pi 3B+ model has 40 GPIO pins, unlike the Arduino the Pi has digital I/O pins. In the earlier models there were only 26 pins.

# Physical Computing of GPIO Pins



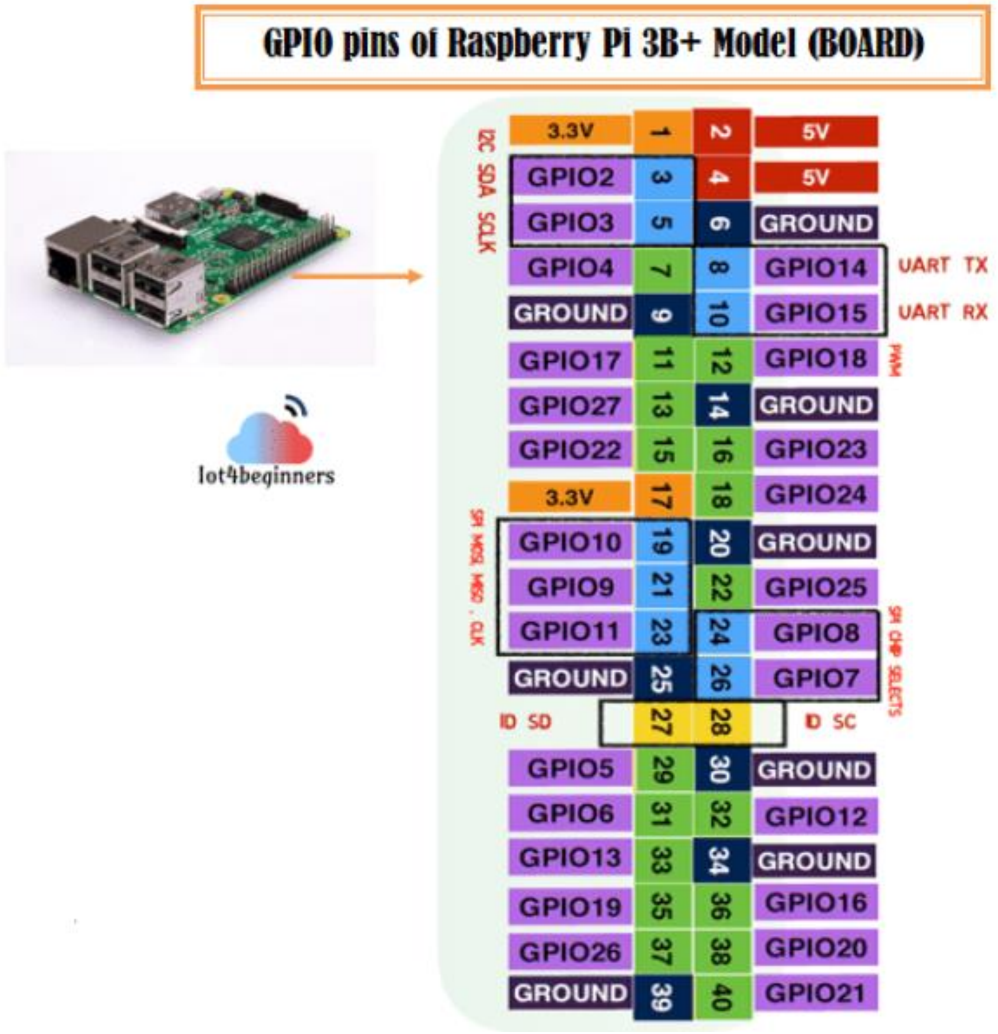
Physical Computing of GPIO pins

Before we go in detail with each of the pin, we first understand how the physical computing is setup in Raspberry Pi using these GPIO pins.

- First thing, you should figure out what kind of input and output devices you need to compute. A light sensor or a pressure sensor can be input and whereas you need an output device to implement the results. So the output device maybe a seven-segment display or just an LED to indicate the results.
- Understand the interface of the device like how it gives output and how it receives the input. Each pin has specific purpose, some need to be a certain voltage, etc.
- Then connect it to the computer, say Raspberry Pi in our case and understand which GPIO pin you had to use based on the interface whether input or output.
- Finally, write the code- configure the pin number in the code which you have chosen, when it should look for the input and at what time it should give the output. You can choose any language to write the code.

## Roles of the GPIO pins

There are two numbering schemes of these GPIO pins. They are 1. BOARD and 2. BCM. The difference between these schemes is, BOARD signifies using the physical pin numbers on the Raspberry Pi P1 connector and the BCM signifies the Broadcom SoC channel designation. We will use the **BOARD numbering scheme**.



| | | | | |
|---|---|---|---|---|
| 3.3V | 1 | 2 | 5V | |
| GPIO2 | 3 | 4 | 5V | |
| GPIO3 | 5 | 6 | GROUND | |
| GPIO4 | 7 | 8 | GPIO14 | UART TX |
| GROUND | 9 | 10 | GPIO15 | UART RX |
| GPIO17 | 11 | 12 | GPIO18 | PWM |
| GPIO27 | 13 | 14 | GROUND | |
| GPIO22 | 15 | 16 | GPIO23 | |
| 3.3V | 17 | 18 | GPIO24 | |
| GPIO10 | 19 | 20 | GROUND | |
| GPIO9 | 21 | 22 | GPIO25 | |
| GPIO11 | 23 | 24 | GPIO8 | |
| GROUND | 25 | 26 | GPIO7 | |
| ID SD | 27 | 28 | ID SC | |
| GPIO5 | 29 | 30 | GROUND | |
| GPIO6 | 31 | 32 | GPIO12 | |
| GPIO13 | 33 | 34 | GROUND | |
| GPIO19 | 35 | 36 | GPIO16 | |
| GPIO26 | 37 | 38 | GPIO20 | |
| GROUND | 39 | 40 | GPIO21 | |

GPIO pins- Raspberry Pi

This is how the pins are numbered on the Raspberry Pi board. If you notice that one side of the pins, numbered even, in which some of the pins are ground pins. These are the negative terminal of the battery ie, Raspberry Pi, which will act as the 0 V or the neutral end of the circuit. There are 26 bi-directional pins that are configured as input or output which take two states HIGH (3.3 V) or LOW (0 V).

There two kind of power pins ie, it will act as positive terminal

- Pin number 2 and 4 – draw 5 V from the Pi's micro USB, it can draw up to 300 mA
- Pin number 1 and 17- draw 3.3 V and can draw only up to 50 mA

There are some pins allocated for serial communication

SPI (Serial Peripheral Interface)- 5 pins ( 19,21,23,24,26)

- GPIO10- MOSI (Master Out Slave In)
- GPIO9- MISO (Master In Slave Out)
- GPIO11 – CLK
- GPIO7 and GPIO8- CHIP select

UART (Universal Asynchronous Receiver/Transmitter)- pin number 8 and 10 – to convert parallel interface and serial interface

- GPIO14 – UART transmitter
- GPIO15- UART receiver

I2C Inter Integrated Circuit – there are two pins for data and clock.

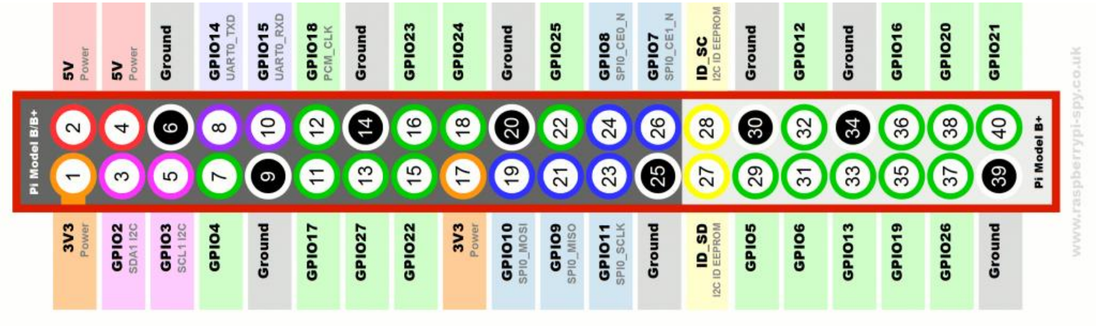
- GPIO2 – SDA (for data)

GPIO3- SCLK- clock

Pin number 27 and 28 can be used to connect to other hardware boards and configure the GPIO as per that need.

Pin number 12 can be used for **Pulse Width Modulation** functionality.

## Difference between BCM and BOARD pin numbering in Raspberry Pi



There are two kinds of Input and Output pin numbering for the Raspberry pi. One is the BCM and the other is BOARD. Basically, these pin numberings are useful for writing python script for the Raspberry Pi.

**GPIO BOARD**– This type of pin numbering refers to the number of the pin in the plug, i.e, the numbers printed on the board, for example, P1. The advantage of this type of numbering is, it will not change even though the version of board changes.

**GPIO BCM**– The BCM option refers to the pin by “Broadcom SOC Channel. They signify the Broadcom SOC channel designation. The BCM channel changes as the version number changes.

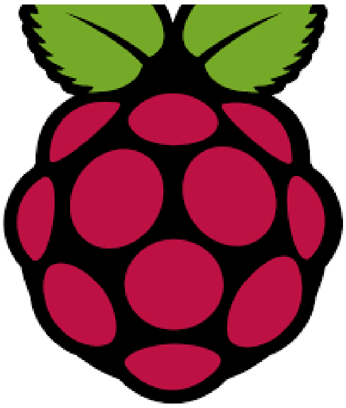
Note: It is very important to wire the GPIO pins with limited resistors to avoid serious damage to the Raspberry Pi. LEDs must have resistors to limit the current passing through them. The motors should not be connected to the GPIO pins directly.

## How to use the GPIO pin numbers in Python?

I have used a simple LED blink python program with Raspberry Pi. Here the GPIO.BOARD is the type of pin numbering which was explained earlier. The number 11 is the pin for LED and considered as an output.

```
import RPi.GPIO as GPIO
from time import sleep
GPIO.setmode (GPIO.BOARD)
GPIO.setup (11,GPIO.OUT)
while True:
   GPIO.output(11,True)
   time.sleep(2)
   GPIO.output(11,False)
   time.sleep(2)
```

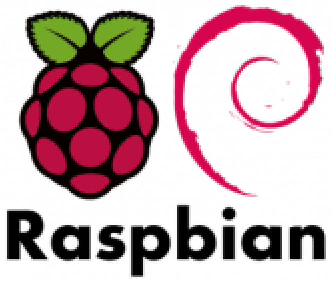
# 2. Operating Systems (OS) available for Raspberry Pi



If you buy the Raspberry Pi Kit which comes with preloaded **NOOBS** on an SD card, then it will make it easier to download and install the OS for your Pi because the NOOBS is an OS manager needed for installation and easy setup of the OS on PI device. We can install the OS manually on an empty SD card too. Here is a list of Operating Systems that is suitable for your Raspberry Pi.

## Raspbian

Raspbian – Official standard operating system based on Linux Debian. It is the best Operating System for Raspberry Pi 3, and **Raspbian Buster is the apt OS for Raspberry Pi 4.** Raspbian is very simple to use, you just have to dump an image available on the official website of Raspberry Pi. Though Raspbian is a light-weight OS that is based on Debian distribution, it is specifically designed for Raspberry Pi, also it comes with pre-installed software such as Thonny Editor for Python, Libre Office, email client and many more.



You can update the software by using the Linux command,

sudo apt-get update

You will find guidelines and installation easily on their official website.

## Windows IoT Core

If you bought the Pi 3 for coding and programming specifically, then the Windows 10 IoT Core is the best option to go with. Don't get confused with the name because the Pi will not support the same Windows 10 version we use on our desktop PC, but uses a slightly simpler and slimmed down version of the OS.

Despite the lack of a classic Windows GUI, Windows IoT Core has the advantage above all that it works seamlessly with Microsoft's in-house technologies such as the Universal Windows Platform (UWP) as well as Azure Cloud, Office 365 and other cloud and server services. It also allows the relatively smooth use of Windows hardware on the Pi. This makes the Raspberry Pi fit for the Internet of Things according to Microsoft reading.

## Pidora

Pidora – Interesting Raspbian alternative with Fedora

Famous alternative for the Fedora OS for Raspberry Pi. It is based on the Fedora build for ARM V6 Architecture. It uses Fedora 20 package set. Many people stick to the Raspbian OS but the new and advanced OS is giving tough competition to the Raspbian. The new version came up in 2014. They can be useful for headless mode rather than having a monitor to connect to the OS like in Rasbian. It also comes with all the necessary software which is useful for Raspberry Pi.

## Arch Linux ARM

base Archlinux – In contrast to Raspbian, Archlinux relies on current software updates. Unfortunately, system stability can suffer.

The Arch Linux ARM has a simple base Linux, that will shape it according to your needs and full control over your Pi. It is basically a distribution of Linux for ARM Architecture. The kernel and software support is focussed on hard-float ARMv6 and ARMv7, soft-float ARMv5te, and ARMv8 AArch64. The update is done vial small packages daily. To conclude, this OS can be used for full control over the system.

## OpenELEC

OpenElec – Recommended MediaCenter-Distribution with automatic updates. Open Embedded Linux Entertainment Center is a small Linux based **Just Enough Operating System** (JeOS) built from scratch as a platform to turn your computer into a **Kodi media center**.

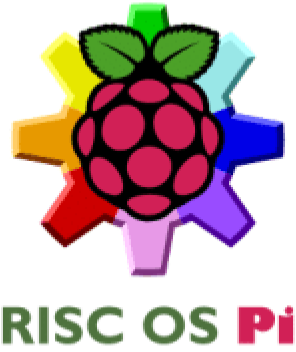
As it is designed to be lightweight, it uses very little system resources for processor or memory. With support for Intel's HD Graphics, NVIDIA's GeForce & ION platform, AMD's Radeon & Fusion platform as well as Broadcom's

Crystal HD chip, it can support high definition content on machines with low-powered processors by offloading video content to the supported graphics cards and decoders.
For Raspberry Pi 1 and 2: Click here.

# OSMC

Raspbmc – Simple and powerful XBMC-Mediacenter. Its tagline says, "OSMC is a free and open-source media center built for the people, by the people." The OSMC is based on the Debian Linux and Kodi Media Center. OSMC runs on all models of Raspberry Pi prior to the Pi 4. They recommend you use a Class 10 SD card, a stable power supply, and at least the Pi Model 2 for the best experience.

# RISC OS – Non-Linux based operating system



The concept of RISC OS is something that is non-Linux and works on its own. It is unique rather than other traditional OS for Raspberry Pi. It has a super responsive Universal Storage Platform. The RISC OS Pico is suitable for Pi users. A minimum of 2 GB memory in the SD card is a pre-requisite to run this OS on Pi. If you are a Pi enthusiast, then you should try this OS. It is not built on Linux and does not support the electronic prototype projects, but it is a whole new concept of OS which works on its own.

## Other OS available for Raspberry Pi:

New Linutop OS – It is an ideal Operating System for professional use like Internet Kiosk, Embedded Systems, Digital Signage, etc.

Kali Linux – Successor of the well-known BackTrack Hacking Distribution

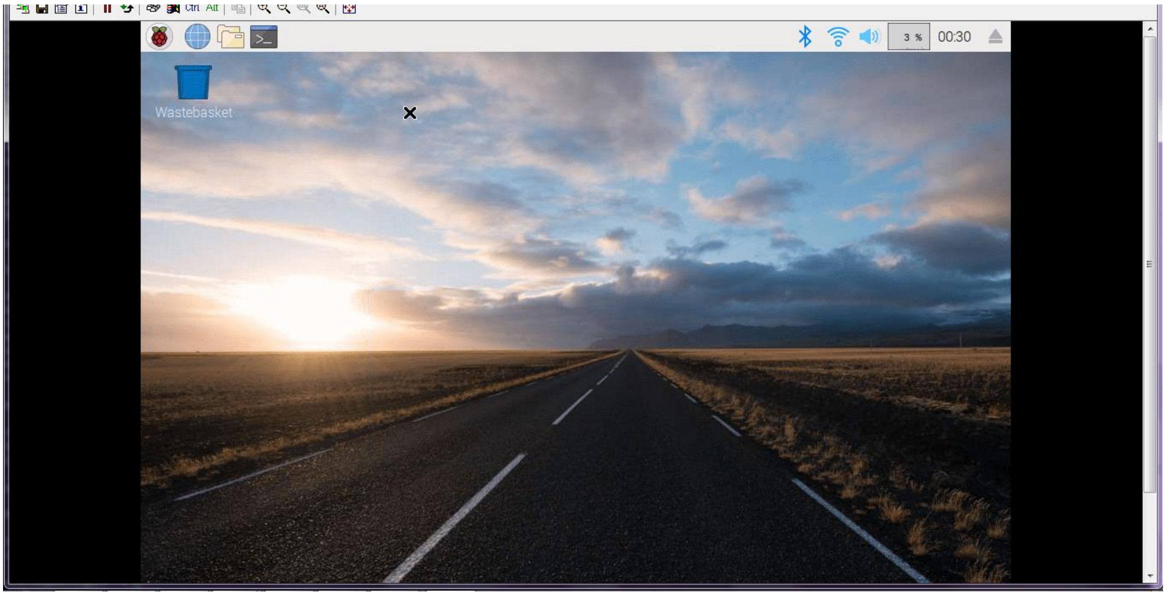
Razdroid – Android for the Raspberry

Currently still unstable RetroPi – Emulated with your Pi over 15 retro consoles

### Raspbian – the best

The Raspbian Is the Best All-Around Operating System, is the "official" operating system of the Raspberry Pi and because of that, it's the one most people will want to start with. It is a version of **Linux** built specifically for the Raspberry Pi.

# 3. Headless Raspberry Pi Setup



In my previous article, I had written about Raspberry Pi and the use of the GPIO pins. You can set up the Pi using a monitor connected to the Pi using an HDMI cable and the keyboard, mouse connected to RPi's USB port. Just download the Raspbian OS in an SD card and put in the slot given in the backside of the Raspberry Pi and BAM!! Do not worry, if you don't have a monitor, keyboard, etc for this. It is so simple to set up the Raspberry Pi and access on your laptop. Only thing, you must have a LAN (Ethernet) cable and modem with the internet. **When you do not use a monitor or keyboard to run a Pi, it is called as headless.**

## About Raspbian

Raspbian is a **free operating system based on Debian optimized for the Raspberry Pi hardware**. An operating system is the set of basic programs and utilities that make your Raspberry Pi run. However, Raspbian provides more than a pure OS: it comes with over 35,000 packages, pre-compiled software bundled in a nice format for easy installation on your Raspberry Pi. The Raspbian is free software and a community funded.

# Things you needed to setup Raspberry Pi

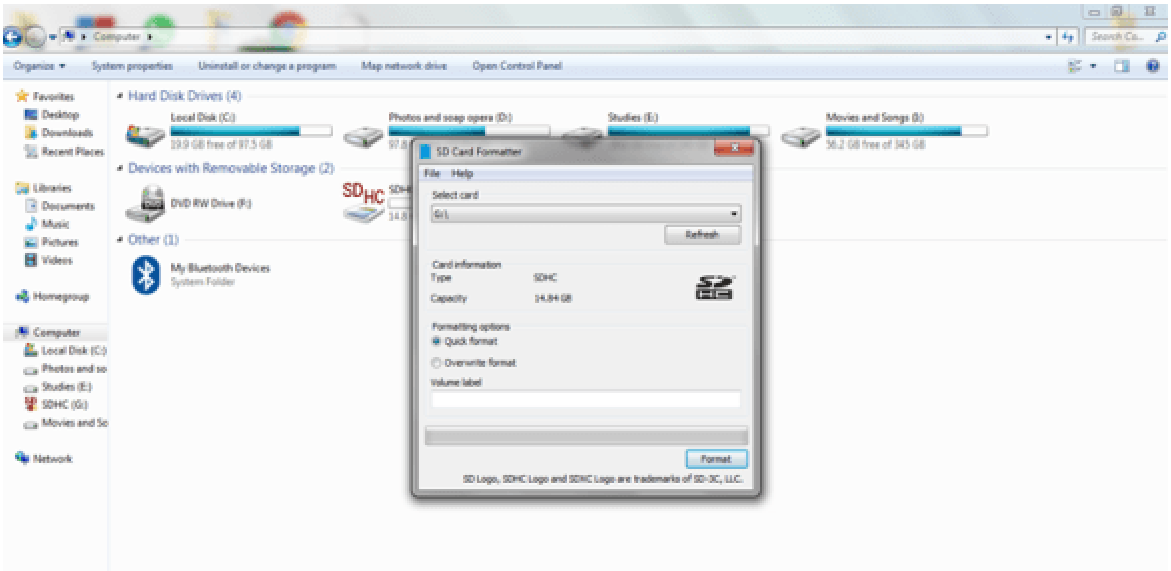
- Raspberry Pi
- Ethernet cable
- Router
- SD card (maybe 16 GB)
- SD card reader
- Micro USB 2.0 Power source Connector ( smartphone charger to power RPi)

# Softwares needed

- SD Card Formatter – https://www.sdcard.org/downloads/formatter_4/
- Win32 DiskImager – https://sourceforge.net/projects/win32diskimager/
- Advanced IP Scanner / Fing app – https://www.advanced-ip-scanner.com/
- Putty- https://www.putty.org/
- TightVNC Viewer – https://www.tightvnc.com/download.php
- Raspbian Image – https://www.raspberrypi.org/downloads/raspbian/

**Follow these steps to install the headless Raspberry Pi successfully.**

# Step 1 : Formatting the SD Card using the SD Card Formatter



SD Card Formatter

Select card: G:\

Card information
Type: SDHC
Capacity: 14.84 GB

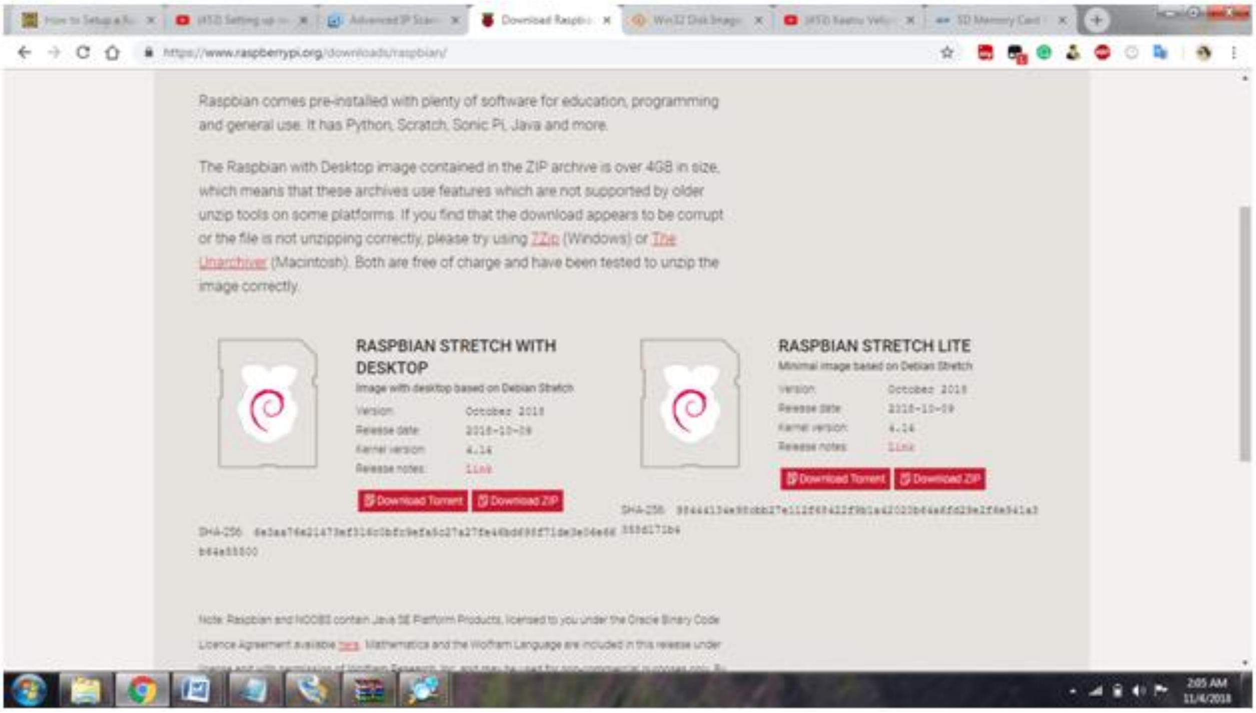
Formatting options
● Quick format
○ Overwrite format

Volume label

SD Logo, SDHC Logo and SDXC Logo are trademarks of SD-3C, LLC.

Install the SD card formatter from the link given above. Insert the SD card in the card reader and format it.

## Step 2: Download the Raspbian Image



https://www.raspberrypi.org/downloads/raspbian/

Raspbian comes pre-installed with plenty of software for education, programming and general use. It has Python, Scratch, Sonic Pi, Java and more.

The Raspbian with Desktop image contained in the ZIP archive is over 4GB in size, which means that these archives use features which are not supported by older unzip tools on some platforms. If you find that the download appears to be corrupt or the file is not unzipping correctly, please try using 7Zip (Windows) or The Unarchiver (Macintosh). Both are free of charge and have been tested to unzip the image correctly.

RASPBIAN STRETCH WITH DESKTOP

Image with desktop based on Debian Stretch

| | |
|---|---|
| Version: | October 2018 |
| Release date: | 2018-10-09 |
| Kernel version: | 4.14 |
| Release notes: | Link |

SHA-256: [illegible]

RASPBIAN STRETCH LITE

Minimal image based on Debian Stretch

| | |
|---|---|
| Version: | October 2018 |
| Release date: | 2018-10-09 |
| Kernel version: | 4.14 |
| Release notes: | Link |

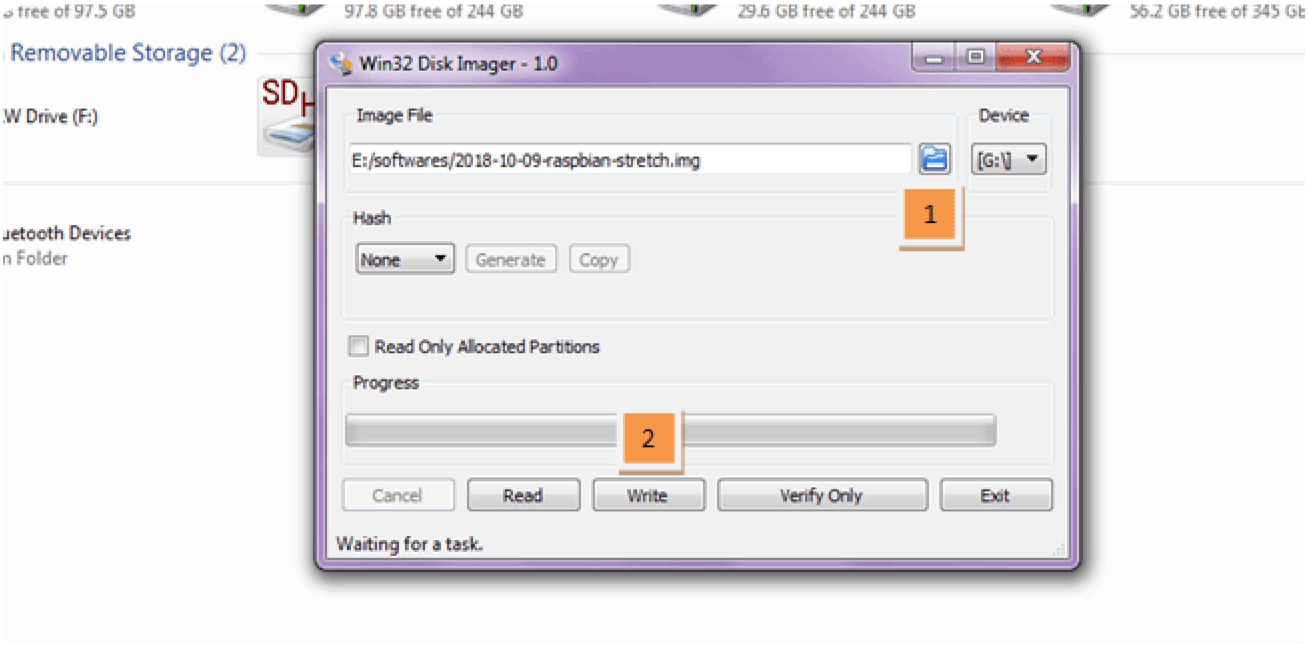
SHA-256: [illegible]

Note: Raspbian and NOOBS contain Java SE Platform Products, licensed to you under the Oracle Binary Code Licence Agreement available here. Mathematica and the Wolfram Language are included in this release under

Download the Raspbian Image from the given link. You can choose either Raspbian Stretch Lite version or with the desktop from the options. If you are **choosing Raspbian stretch with deskto**p, you will have access to desktop and icons ie. you will have access to GUI. If you are **choosing the Raspbian stretch lite version**, you will be able to boot the command line. Also, the size of this file is less and uses fewer resources. But for beginners, use the desktop version.

It will be downloaded in a Zip file, unzip it and you will find an image file. Meanwhile, download the Win32 DiskImager in your laptop.

# Step 3: Write the Image file to the SD card using Win32 DiskImager



Win32 Disk Imager - 1.0

Image File: E:/softwares/2018-10-09-raspbian-stretch.img

Device: [G:\]

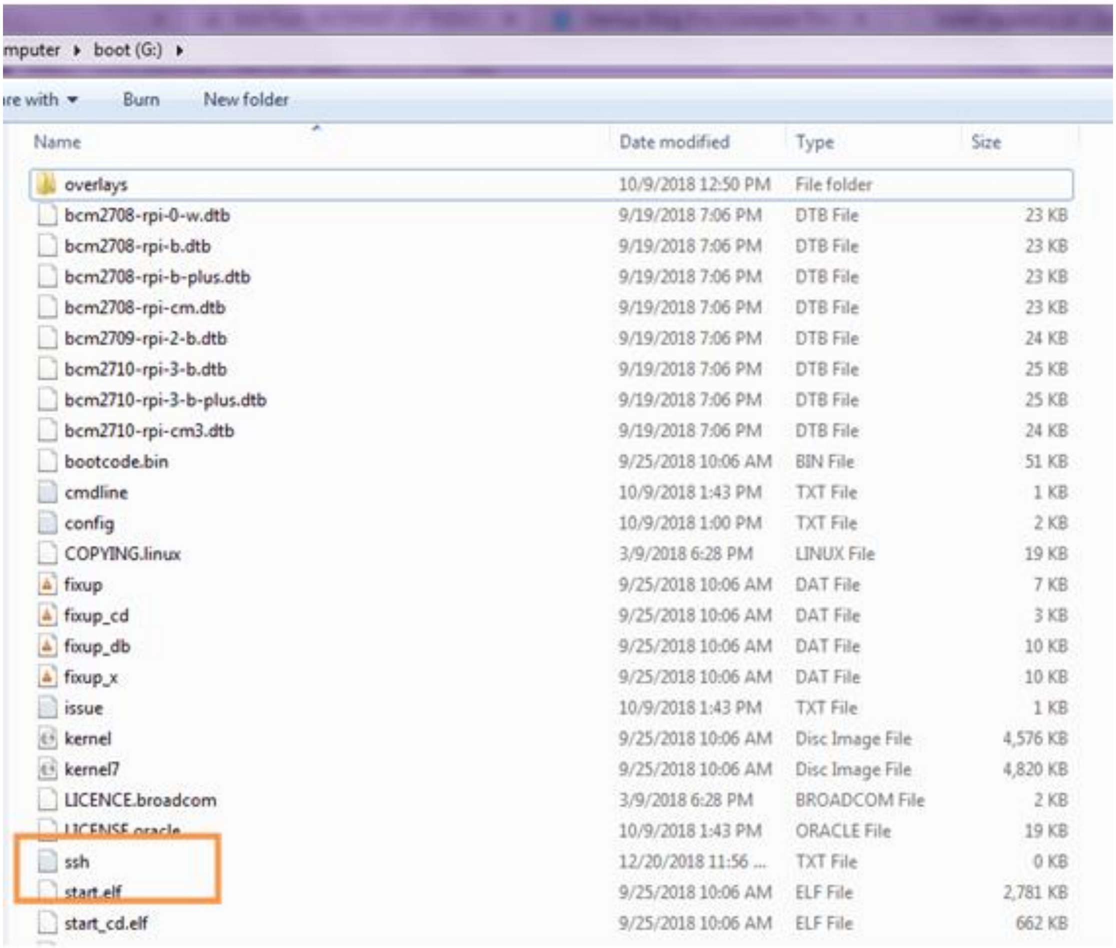
Hash: None

□ Read Only Allocated Partitions

Waiting for a task.

Open the Win32 DiskImager and click the icon (1). Select the image you need to write to the SD card and click Write (2) option. This takes 5 – 10 minutes to be done.

## Step 4: Adding SSH file to the SD card



| Name | Date modified | Type | Size |
|---|---|---|---|
| overlays | 10/9/2018 12:50 PM | File folder | |
| bcm2708-rpi-0-w.dtb | 9/19/2018 7:06 PM | DTB File | 23 KB |
| bcm2708-rpi-b.dtb | 9/19/2018 7:06 PM | DTB File | 23 KB |
| bcm2708-rpi-b-plus.dtb | 9/19/2018 7:06 PM | DTB File | 23 KB |
| bcm2708-rpi-cm.dtb | 9/19/2018 7:06 PM | DTB File | 23 KB |
| bcm2709-rpi-2-b.dtb | 9/19/2018 7:06 PM | DTB File | 24 KB |
| bcm2710-rpi-3-b.dtb | 9/19/2018 7:06 PM | DTB File | 25 KB |
| bcm2710-rpi-3-b-plus.dtb | 9/19/2018 7:06 PM | DTB File | 25 KB |
| bcm2710-rpi-cm3.dtb | 9/19/2018 7:06 PM | DTB File | 24 KB |
| bootcode.bin | 9/25/2018 10:06 AM | BIN File | 51 KB |
| cmdline | 10/9/2018 1:43 PM | TXT File | 1 KB |
| config | 10/9/2018 1:00 PM | TXT File | 2 KB |
| COPYING.linux | 3/9/2018 6:28 PM | LINUX File | 19 KB |
| fixup | 9/25/2018 10:06 AM | DAT File | 7 KB |
| fixup_cd | 9/25/2018 10:06 AM | DAT File | 3 KB |
| fixup_db | 9/25/2018 10:06 AM | DAT File | 10 KB |
| fixup_x | 9/25/2018 10:06 AM | DAT File | 10 KB |
| issue | 10/9/2018 1:43 PM | TXT File | 1 KB |
| kernel | 9/25/2018 10:06 AM | Disc Image File | 4,576 KB |
| kernel7 | 9/25/2018 10:06 AM | Disc Image File | 4,820 KB |
| LICENCE.broadcom | 3/9/2018 6:28 PM | BROADCOM File | 2 KB |
| LICENSE.oracle | 10/9/2018 1:43 PM | ORACLE File | 19 KB |
| ssh | 12/20/2018 11:56 ... | TXT File | 0 KB |
| start.elf | 9/25/2018 10:06 AM | ELF File | 2,781 KB |
| start_cd.elf | 9/25/2018 10:06 AM | ELF File | 662 KB |

ssh

So now you have written the image file into the SD card and you will numerous files. In the recent Raspbian OS, SSH is not enabled so you have to create a file and add to that. Simply you can create a new file in the notepad and save it as "ssh" and copy it to the SD card. This flags the Pi to enable the SSH system on the next boot.

## Step 5 : Start Booting the Pi

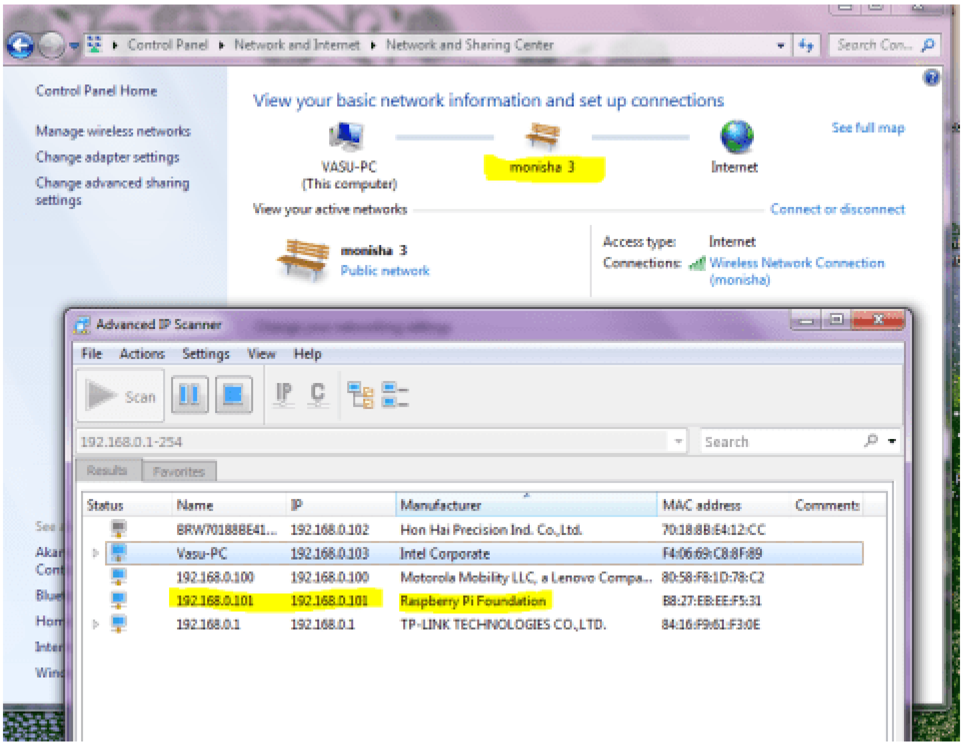
Boot the Pi

Right now you have the Raspbian OS installed in your SD card. Put it in the slot given back of of the Pi and connect the Power connector and the Ethernet cable to the modem with internet. Both the green and red LEDs should be blinking. Now download the TightVNC viewer, Putty and Advanced IP Scanner.

## Step 6: Find the IP address of you Raspberry Pi

You can find your Raspberry Pi's IP address by using an app called Fing or you can download Advanced IP Scanner.

*Important note: The network of which your software / app using to find the IP address of your Pi should be same as the network of the Raspberry Pi which is connected via Ethernet cable to the modem.*



monisha 3

Access type: Internet

Connections: Wireless Network Connection (monisha)

192.168.0.1-254

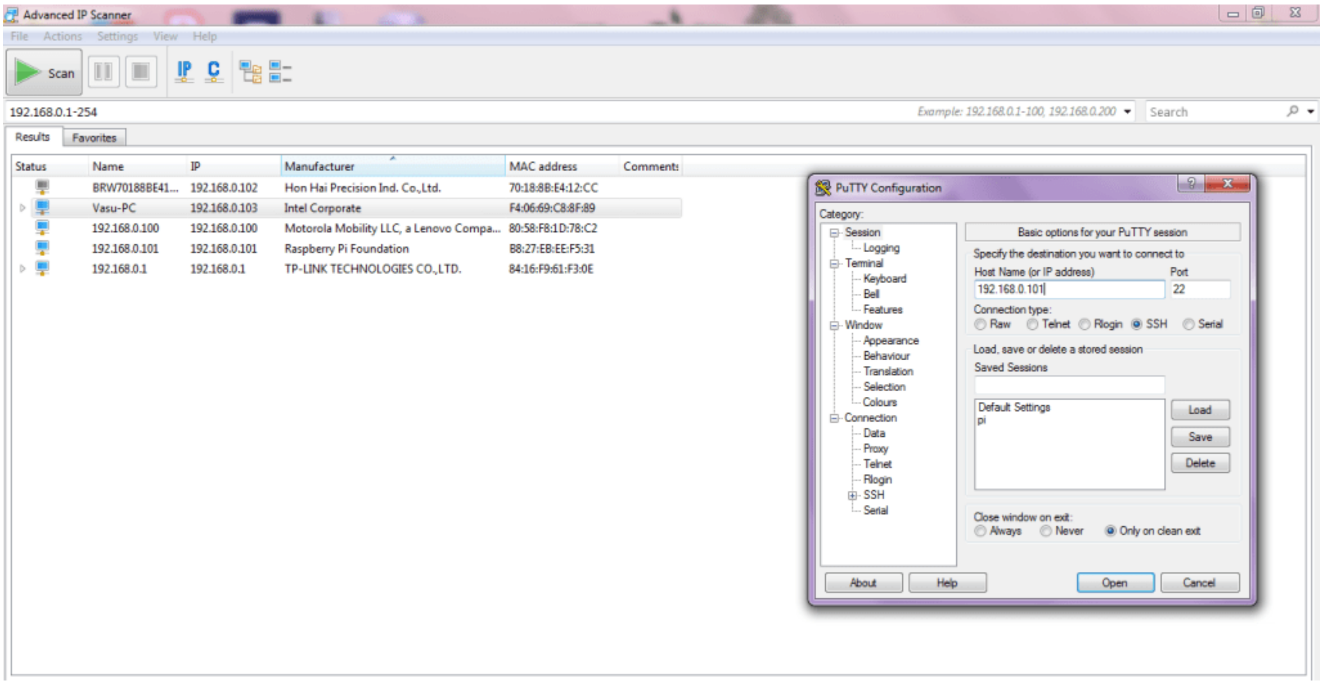
| Status | Name | IP | Manufacturer | MAC address | Comments |
|---|---|---|---|---|---|
| | BRW70188BE41... | 192.168.0.102 | Hon Hai Precision Ind. Co.,Ltd. | 70:18:8B:E4:12:CC | |
| | Vasu-PC | 192.168.0.103 | Intel Corporate | F4:06:69:C8:8F:89 | |
| | 192.168.0.100 | 192.168.0.100 | Motorola Mobility LLC, a Lenovo Compa... | 80:58:F8:1D:78:C2 | |
| | 192.168.0.101 | 192.168.0.101 | Raspberry Pi Foundation | B8:27:EB:EE:F5:31 | |
| | 192.168.0.1 | 192.168.0.1 | TP-LINK TECHNOLOGIES CO.,LTD. | 84:16:F9:61:F3:0E | |

Advanced Scanner

## Step 7: SSH into the Raspberry Pi

I use putty for enabling the SSH in the Pi. The default credentials are

- **username : pi**
- **password : raspberry (this can be changed in the configuration)**



192.168.0.1-254

Example: 192.168.0.1-100, 192.168.0.200

Search

| Status | Name | IP | Manufacturer | MAC address | Comments |
|---|---|---|---|---|---|
| | BRW70188BE41... | 192.168.0.102 | Hon Hai Precision Ind. Co.,Ltd. | 70:18:8B:E4:12:CC | |
| | Vasu-PC | 192.168.0.103 | Intel Corporate | F4:06:69:C8:8F:89 | |
| | 192.168.0.100 | 192.168.0.100 | Motorola Mobility LLC, a Lenovo Compa... | 80:58:F8:1D:78:C2 | |
| | 192.168.0.101 | 192.168.0.101 | Raspberry Pi Foundation | B8:27:EB:EE:F5:31 | |
| | 192.168.0.1 | 192.168.0.1 | TP-LINK TECHNOLOGIES CO.,LTD. | 84:16:F9:61:F3:0E | |

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): 192.168.0.101 Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

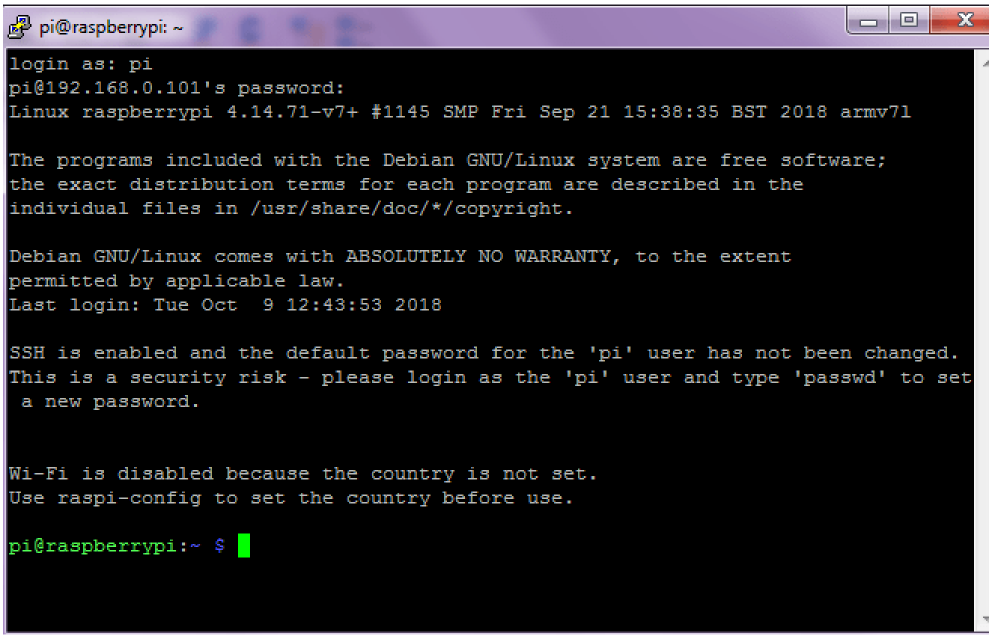
Saved Sessions

Default Settings
pi

Close window on exit: ○ Always ○ Never ● Only on clean exit

putty

Enter the IP address of the Raspberry Pi in the putty and enable the X11 forwarding. Click Open. Give the username as **pi** and password as **raspberry**.



```
login as: pi
pi@192.168.0.101's password:
Linux raspberrypi 4.14.71-v7+ #1145 SMP Fri Sep 21 15:38:35 BST 2018 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Oct  9 12:43:53 2018

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
 a new password.


Wi-Fi is disabled because the country is not set.
Use raspi-config to set the country before use.

pi@raspberrypi:~ $
```

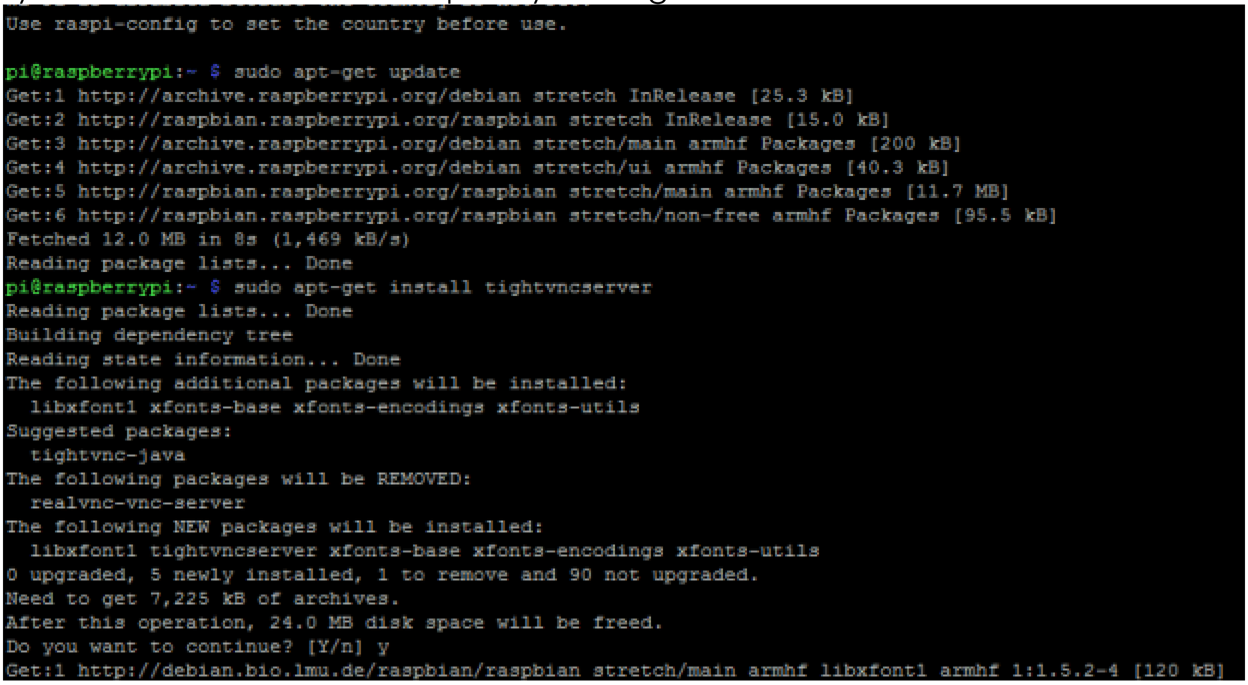
Update the Raspberry Pi with the following linux shell commands. Install the tight vnc server in the Raspberry Pi.

sudo apt-get update

sudo apt-get install tightvncserver

# Step 8 : View in the tightVNC viewer

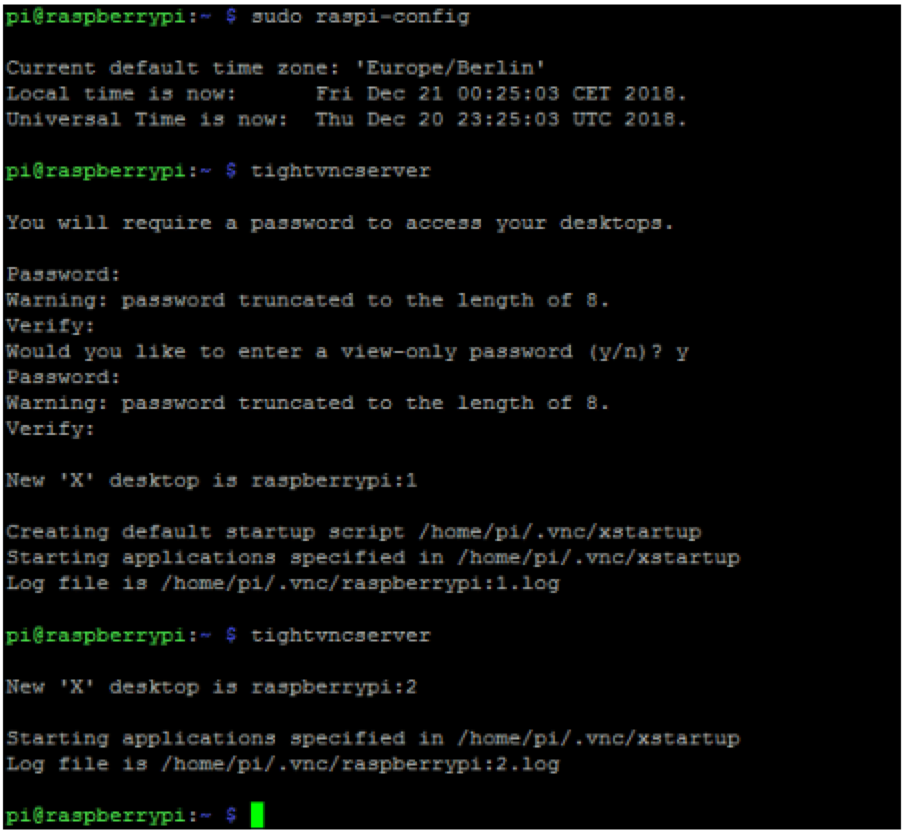
As you had already downloaded and installed the tightVNC viewer in your system. Now install in the Raspberry Pi using the above command.



```
Use raspi-config to set the country before use.

pi@raspberrypi:~ $ sudo apt-get update
Get:1 http://archive.raspberrypi.org/debian stretch InRelease [25.3 kB]
Get:2 http://raspbian.raspberrypi.org/raspbian stretch InRelease [15.0 kB]
Get:3 http://archive.raspberrypi.org/debian stretch/main armhf Packages [200 kB]
Get:4 http://archive.raspberrypi.org/debian stretch/ui armhf Packages [40.3 kB]
Get:5 http://raspbian.raspberrypi.org/raspbian stretch/main armhf Packages [11.7 MB]
Get:6 http://raspbian.raspberrypi.org/raspbian stretch/non-free armhf Packages [95.5 kB]
Fetched 12.0 MB in 8s (1,469 kB/s)
Reading package lists... Done
pi@raspberrypi:~ $ sudo apt-get install tightvncserver
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libxfont1 xfonts-base xfonts-encodings xfonts-utils
Suggested packages:
  tightvnc-java
The following packages will be REMOVED:
  realvnc-vnc-server
The following NEW packages will be installed:
  libxfont1 tightvncserver xfonts-base xfonts-encodings xfonts-utils
0 upgraded, 5 newly installed, 1 to remove and 90 not upgraded.
Need to get 7,225 kB of archives.
After this operation, 24.0 MB disk space will be freed.
Do you want to continue? [Y/n] y
Get:1 http://debian.bio.lmu.de/raspbian/raspbian stretch/main armhf libxfont1 armhf 1:1.5.2-4 [120 kB]
```

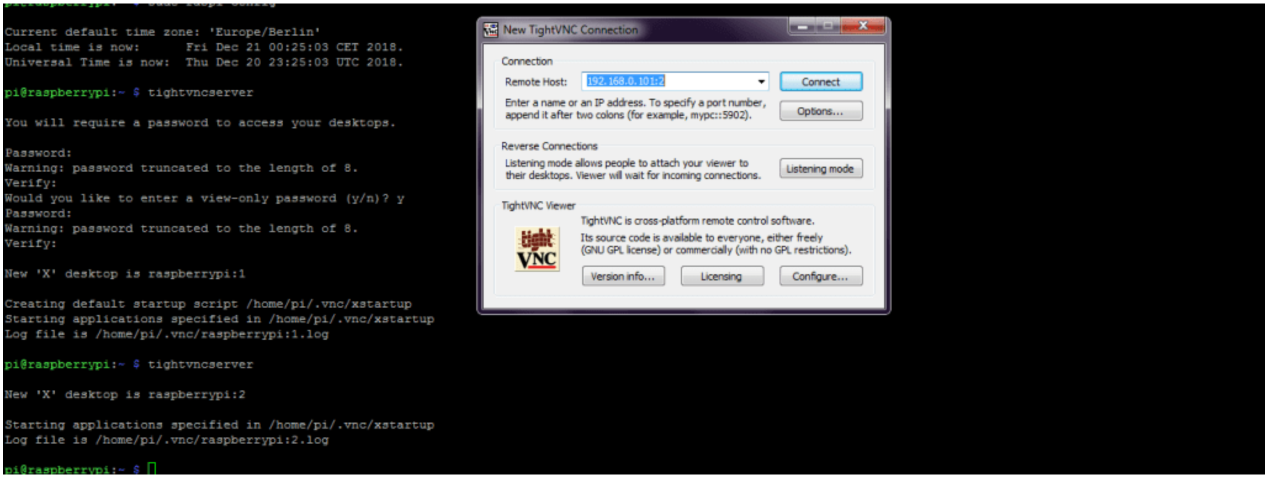
tightvncserver

It asks to set up a password for tightVNC server.



```
pi@raspberrypi:~ $ sudo raspi-config

Current default time zone: 'Europe/Berlin'
Local time is now:      Fri Dec 21 00:25:03 CET 2018.
Universal Time is now:  Thu Dec 20 23:25:03 UTC 2018.

pi@raspberrypi:~ $ tightvncserver

You will require a password to access your desktops.

Password:
Warning: password truncated to the length of 8.
Verify:
Would you like to enter a view-only password (y/n)? y
Password:
Warning: password truncated to the length of 8.
Verify:

New 'X' desktop is raspberrypi:1

Creating default startup script /home/pi/.vnc/xstartup
Starting applications specified in /home/pi/.vnc/xstartup
Log file is /home/pi/.vnc/raspberrypi:1.log

pi@raspberrypi:~ $ tightvncserver

New 'X' desktop is raspberrypi:2

Starting applications specified in /home/pi/.vnc/xstartup
Log file is /home/pi/.vnc/raspberrypi:2.log

pi@raspberrypi:~ $ 
```

Open the tightVNC viewer in your system and give your password. You will be getting a dialog box like the image given below where you have to give your IP address and the number which is given in your putty. For example–> **IP address:X.** In this case, the IP address is 192.168.0.101 and the number is 2. So it is, 192.168.0.101:2. Click Connect.



```
Current default time zone: 'Europe/Berlin'
Local time is now:      Fri Dec 21 00:25:03 CET 2018.
Universal Time is now:  Thu Dec 20 23:25:03 UTC 2018.

pi@raspberrypi:~ $ tightvncserver

You will require a password to access your desktops.

Password:
Warning: password truncated to the length of 8.
Verify:
Would you like to enter a view-only password (y/n)? y
Password:
Warning: password truncated to the length of 8.
Verify:

New 'X' desktop is raspberrypi:1

Creating default startup script /home/pi/.vnc/xstartup
Starting applications specified in /home/pi/.vnc/xstartup
Log file is /home/pi/.vnc/raspberrypi:1.log

pi@raspberrypi:~ $ tightvncserver

New 'X' desktop is raspberrypi:2

Starting applications specified in /home/pi/.vnc/xstartup
Log file is /home/pi/.vnc/raspberrypi:2.log

pi@raspberrypi:~ $ 
```

New TightVNC Connection

Connection

Remote Host: 192.168.0.101:2

Enter a name or an IP address. To specify a port number, append it after two colons (for example, mypc::5902).

Reverse Connections

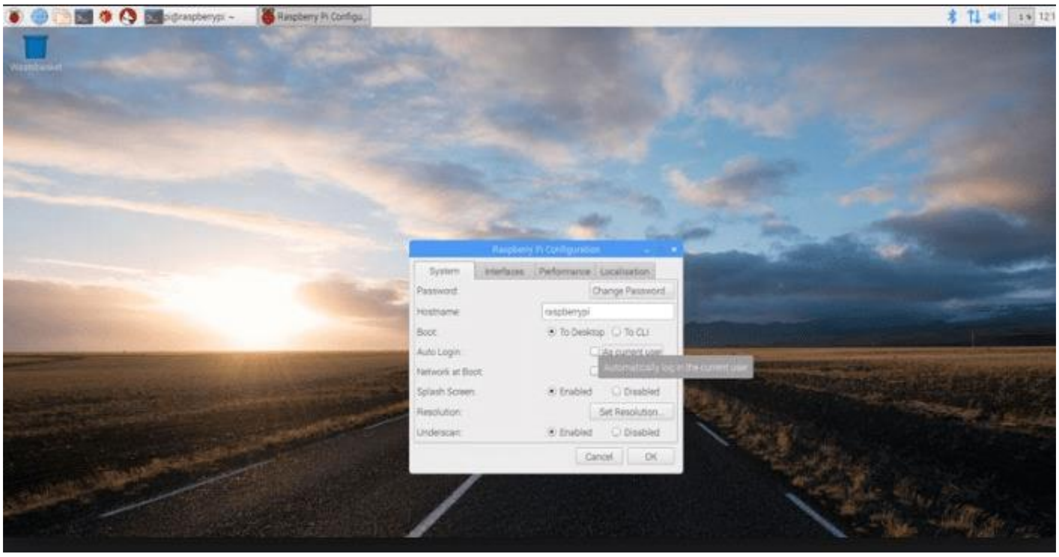
Listening mode allows people to attach your viewer to their desktops. Viewer will wait for incoming connections.

TightVNC Viewer

TightVNC is cross-platform remote control software.

Its source code is available to everyone, either freely (GNU GPL license) or commercially (with no GPL restrictions).

You will get a display like this, but you can change that to wifi mode. You can simply do that by clicking on the data symbol on the right top corner and select your wifi name and enter password.



Password:

Hostname: raspberrypi

Boot: ● To Desktop ○ To CLI

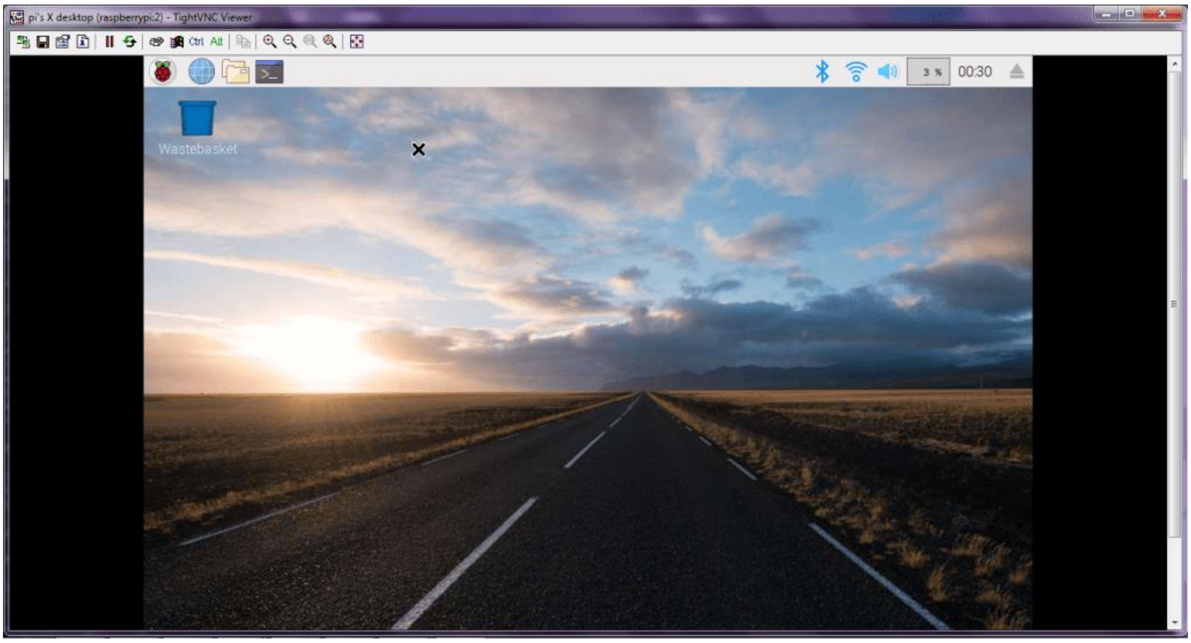
Auto Login: □ As current user

Network at Boot: □

Splash Screen: ● Enabled ○ Disabled

Resolution:

Underscan: ● Enabled ○ Disabled



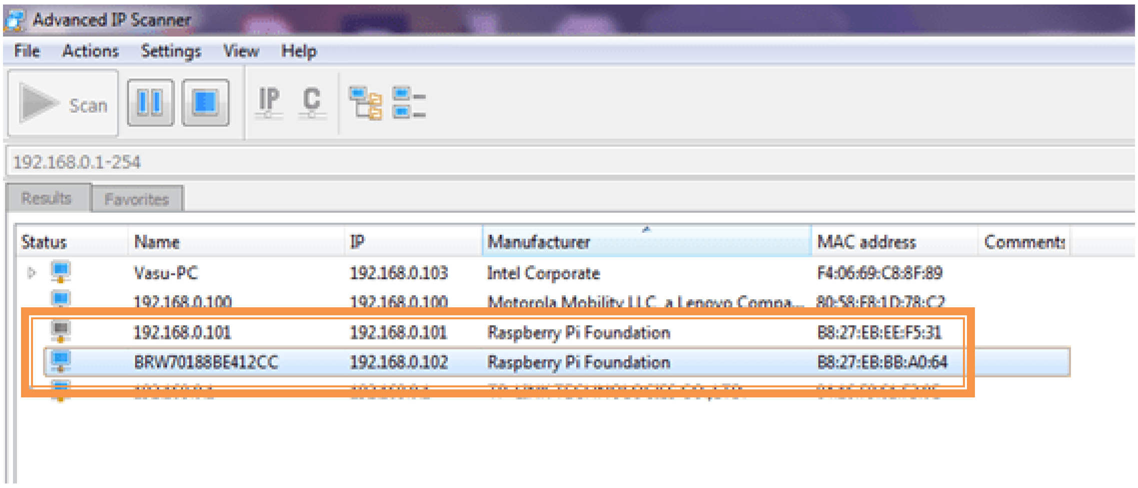
# Configure the Raspberry Pi

You must configure the Raspberry Pi and set up the location in order to access the WiFi.

```
sudo raspi-config
```

# Final Step

So now you have enabled the WiFI. You can **remove the ethernet cable from the Pi and start from the Step 6**. Scan for the IP address and you will find a new one.



192.168.0.1-254

| Status | Name | IP | Manufacturer | MAC address | Comments |
|---|---|---|---|---|---|
| | Vasu-PC | 192.168.0.103 | Intel Corporate | F4:06:69:C8:8F:89 | |
| | 192.168.0.100 | 192.168.0.100 | Motorola Mobility LLC, a Lenovo Compa... | 80:58:F8:1D:78:C2 | |
| | 192.168.0.101 | 192.168.0.101 | Raspberry Pi Foundation | B8:27:EB:EE:F5:31 | |
| | BRW70188BE412CC | 192.168.0.102 | Raspberry Pi Foundation | B8:27:EB:BB:A0:64 | |
| | [illegible] | [illegible] | [illegible] | [illegible] | |

You can find the new IP address of the Raspberry Pi without the LAN cable. Use the IP address for Putty and the TightVNC viewer.

Everytime when you are opening the putty, check for the updates by applying the command:

```
sudo apt-get update
```

When you are disconnecting the Pi, apply the command;

```
sudo reboot
```

Summary:

1. Download and install Putty, TightVNC viewer
2. Download and install SD card formatter and Advanced IP Scanner
3. Download Raspbian Image
4. Write the Raspbian image in SD card using Win32 DiskImager
5. Insert the SD card in the Pi and power up
6. Connect the ethernet cable to the router
7. Open Advance IP Scanner and search for the Pi's IP address
8. Open Putty and enter the IP address
9. Install tightvncserver in putty

# How to make an LED blink with Raspberry Pi?

## Getting started with Raspberry Pi

This section deals with the very basic circuit from our electronics. From our childhood electronics textbooks till engineering books, the basic circuit which contains LED will be demonstrated. Maybe in our childhood we learned using a battery and a simple resistor and now we are going to learn using Raspberry Pi. The main objective of this article is to produce an output from the Raspberry Pi using the LED flashing on and off.

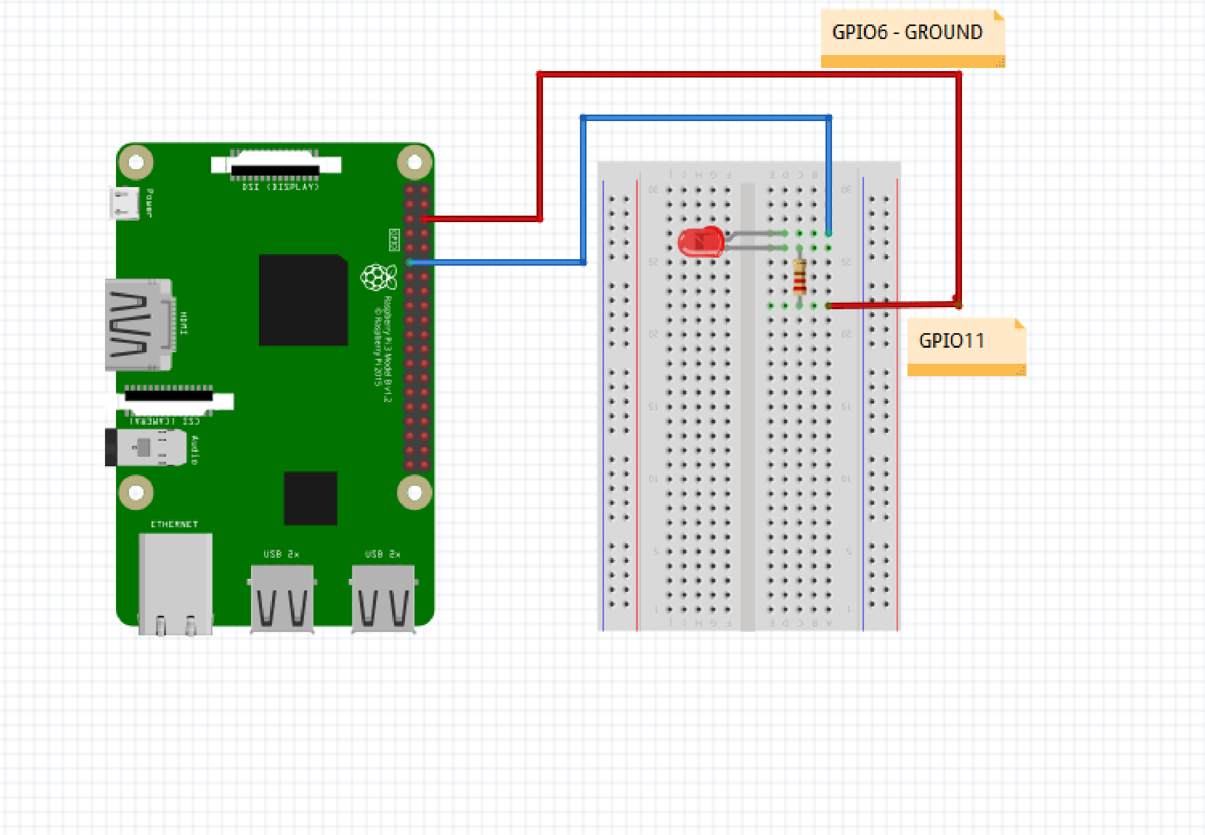
## What is an LED?

LED ( Light Emitting Diode) is a light source when a suitable voltage applied across the leads, the LED releases energy in the form of light that is photons. You must make sure that the voltage applied across the LED should not be too high, or else the LED will be damaged and also the legs are connected across the circuit properly.

## Things you need

- Breadboard
- Light Emitting Diode
- Resistor > 68 ohms
- Raspberry Pi
- Male to female wires

# Circuit Diagram



LED circuit design

# Explanation

The Raspberry Pi has both positive and negative terminal like a battery. The negative terminal will be connected to the GPIO pin 6 which is a ground pin. We will make one of the bi-directional pin as an output, when the output is high it will act as a positive terminal of the battery. This battery will have a voltage of 3.3 V. When the output is low, the positive terminal is at 0 volts so no current flows.

Note: Use a resistor which is higher that 68 ohms because the voltage across the LED should not be high as it damages the LED.

- Insert an LED into a breadboard.
- Connect the resistor which is more than 68 ohms across the longer end of the LED.
- Connect the GPIO pin 11 to the shorter end of the LED.
- Connect the GPIO pin 6 (ground) to the other end of the resistor.

# Python code and explanation

```
import RPi.GPIO as GPIO
from time import sleep
GPIO.setmode (GPIO.BOARD)
GPIO.setup (11,GPIO.OUT)
while True:
   GPIO.output(11,True)
   time.sleep(2)
   GPIO.output(11,False)
   time.sleep(2)
```

You can use any programming language for the coding part like Python, wiring Pi, C, node.js, etc. So, I am using Python here. Here is the full code for the LED flashing ON/OFF. You can save that with extension .py

```
import RPi.GPIO as GPIO
from time import sleep
```

The two lines above from the code explains, importing the libraries you will need RPi.GPIO to control the GPIO pins. Importing time to control how long the pin will be ON or OFF.

```
GPIO.setmode (GPIO.BOARD)
GPIO.setup (11,GPIO.OUT)
```

GPIO.BOARD indicates the numbering scheme you are using. The line (11,GPIO.OUT) means you are using GPIO pin 11 as output pin.

```
while True:
GPIO.output(11,True)
time.sleep(2)
GPIO.output(11,False)
time.sleep(2)
```

while is used to continue as long as the program runs.

GPIO.output(11,True) sets the output to high and remain that way for 2 seconds ( sleep(2)). The same is set to low and remain that way for 2 seconds. If you need time-lapse for 5 seconds, you can change that in time.sleep(5). You can save this by any name with .py and run at Python IDLE in your Raspbian OS orelse you can run this at the command line:

```
sudo python ledblink.py
```