LAB MANUAL - 1

Getting Started with Raspberry PI (Light Version Raspbian OS)

Disclaimer: The content is curated from online/offline resources and used for educational purpose only

Raspberry Pi recommends the use of Raspberry Pi Imager to install an operating system on to your SD card. You will need another computer with an SD card reader to install the image. Raspberry Pi Imager can be run on another Raspberry Pi, but also works on Microsoft Windows, Apple macOS, and Linux.

Step-1

Using Raspberry Pi Imager

Raspberry Pi have developed a graphical SD card writing tool that works on Mac OS, Ubuntu 18.04, and Windows called Raspberry Pi Imager; this is the easiest option for most users since it will download the image automatically and install it to the SD card.

Download the latest version of Raspberry Pi Imager and install it. If you want to use Raspberry Pi Imager from a second Raspberry Pi, you can install it from a terminal using ***sudo apt install rpi-imager***. Then:

Connect an SD card reader with the SD card inside.

Open Raspberry Pi Imager and choose the required OS from the list presented.

Choose the SD card you wish to write your image to.

Review your selections and click on the Write button to begin writing data to the SD Card.

Note:

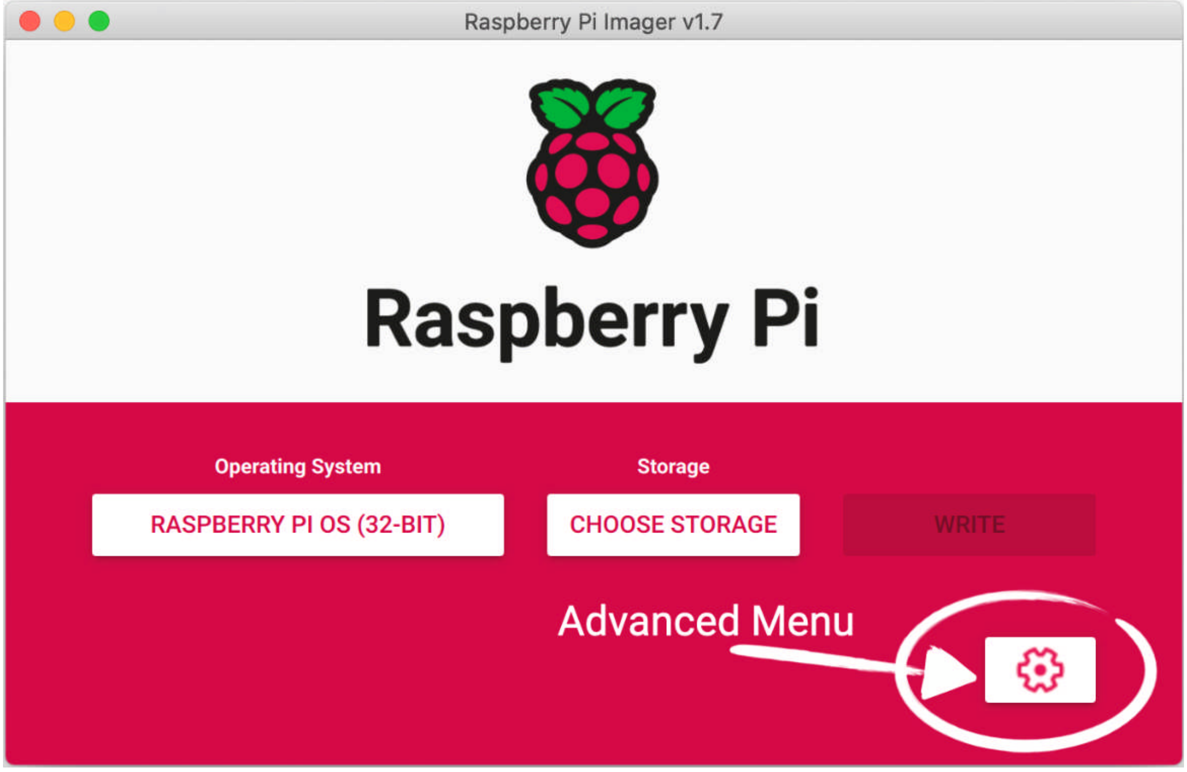
If using Raspberry Pi Imager on Windows 10 with controlled folder access enabled, you will need to explicitly allow Raspberry Pi Imager permission to write the SD card. If this is not done, the imaging process will fail with a "failed to write" error.

Disclaimer: The content is curated from online/offline resources and used for educational purpose only

You can now insert the SD card into the Raspberry Pi and power it up. When your Raspberry Pi boots for the first time a configuration wizard will run that allows you to set up your Raspberry Pi.

Advanced Options

When you have the Raspberry Pi Imager open, and after you have selected the operating system to install, a cog wheel will appear allowing you to open an "Advanced Options" menu if it is supported by the operating system. This menu lets you carry out tasks like enabling SSH, or setting your Raspberry Pi's hostname, and configuring the default user before first boot.



Amongst other things the Advanced Options menu is useful for when you want to configure a headless Raspberry Pi.

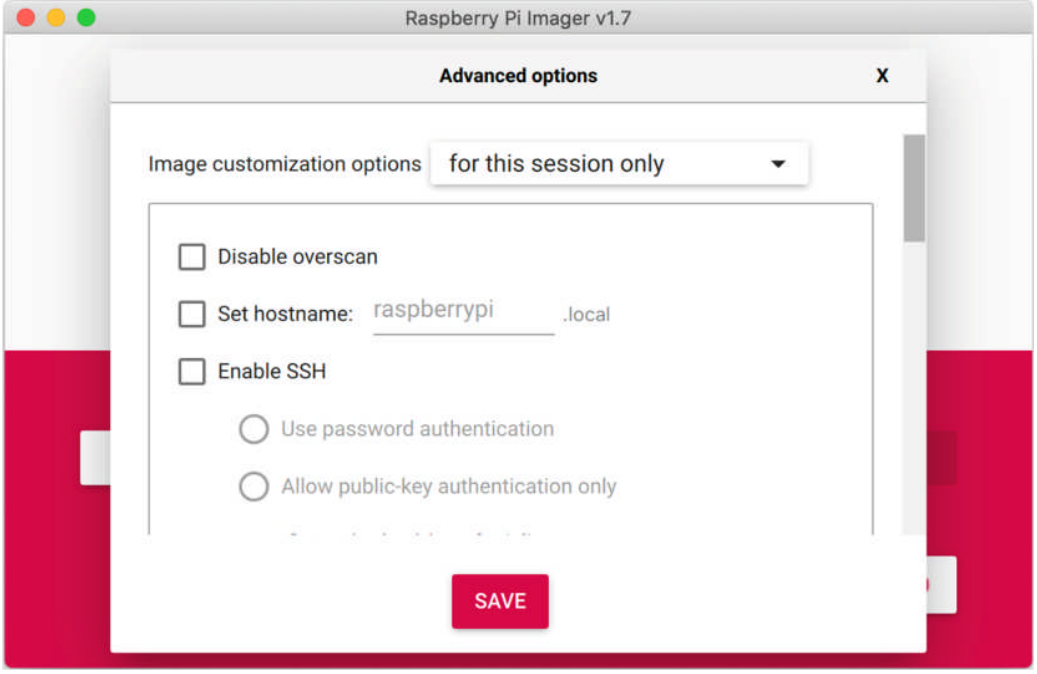


Image customization options: for this session only

□ Disable overscan

□ Set hostname: raspberrypi .local

□ Enable SSH

○ Use password authentication

○ Allow public-key authentication only

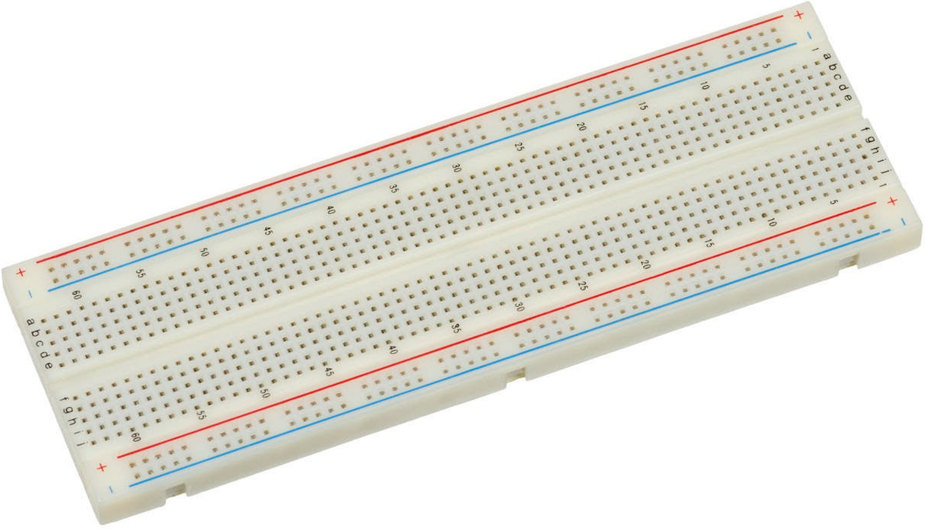
If you are installing Raspberry Pi OS Lite and intend to run it headless, you will still need to create a new user account. Since you will not be able to create the user account on first boot, you **MUST** configure the operating system using the Advanced Menu.

Downloading an Image

If you are using a different tool than Raspberry Pi Imager to write to your SD Card, most require you to download the image first, then use the tool to write it to the card. Official images for recommended operating systems are available to download from the Raspberry Pi website downloads page. Alternative operating systems for Raspberry Pi computers are also available from some third-party vendors.

You may need to unzip the downloaded file (.zip) to get the image file (.img) you need to write to the card.

What is a Breadboard?



Breadboard

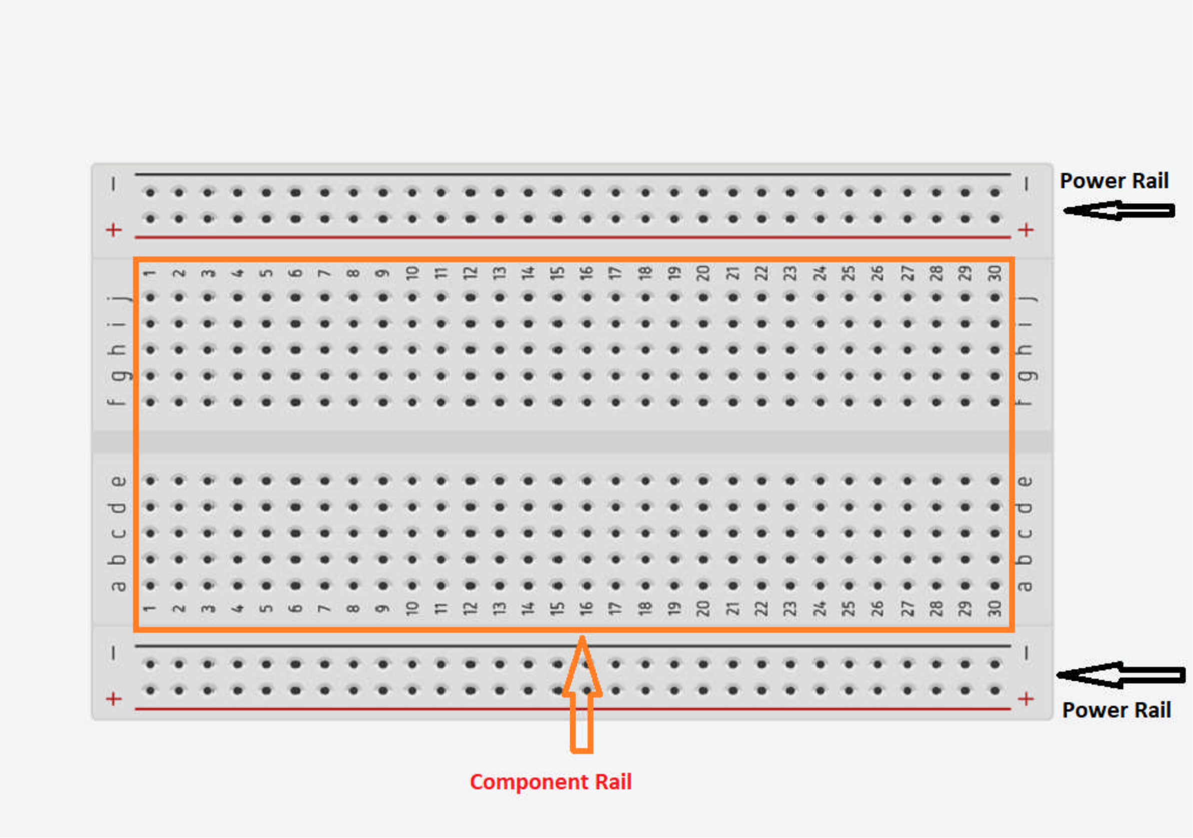
A breadboard is a solder less construction base used for developing a prototype and testing a circuit before building the final, permanent design on a PCB. It is a plastic board with tiny holes on it.

These small size hole are used for inserting our electronic components to make prototype of an electronic circuit. The connections are not permanent, so it is easy to remove component if you make a mistake, or just start over and do a new project.

Breadboard comes in different sizes.

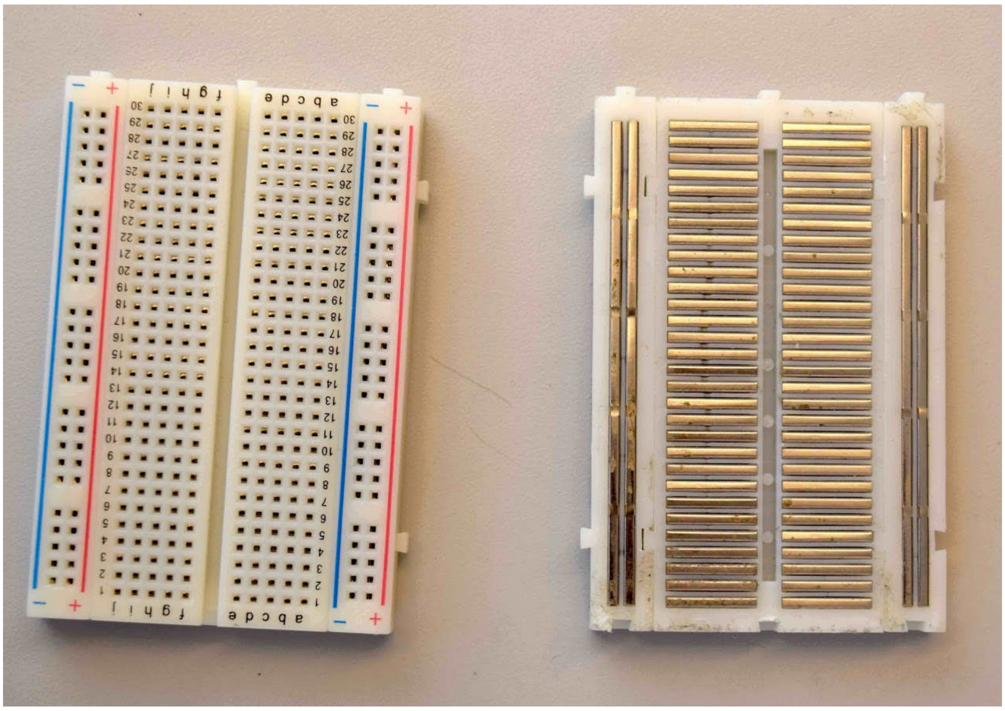
Breadboard Structure:

The internal structure of the Breadboard, can easily show you how and where to connect.



Front of a breadboard

The numbers and letters on a breadboard, helps to identify the holes. There are of two types rails - "Power Rail" and "Component Rail". The strip marked '+' or ' - ' are referred to as the "Power Rail". They are used to supply electrical power to the circuit when its connected to a battery pack or other external power supply.



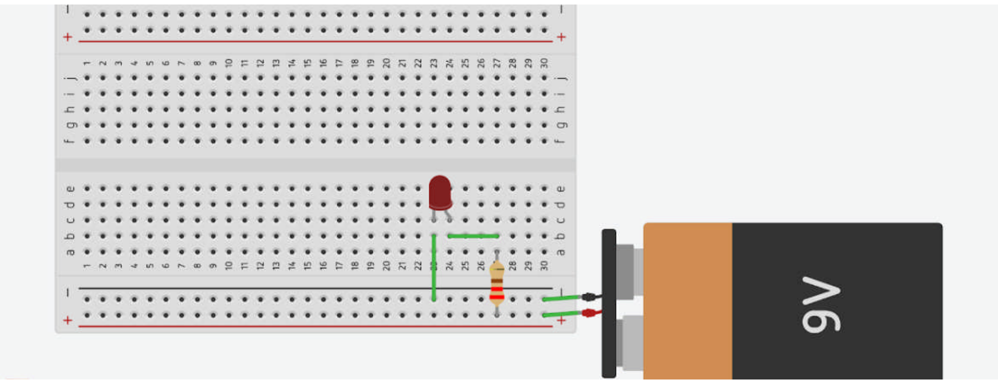
structure

Back of a breadboard

Inside a breadboard we can see number of small clips of metal. These metal clips helps to easily insert electronic components. There are four vertical metal clips at both ends. And at the middle part there are horizontal clips, these are used to place different components. While connecting +ve of battery at any point on the *bus/rails,* the entire line will become positive.

Step-2

Simple circuit on breadboard:



circuit

Series and Parallel Circuit on Breadboard:

Series Connection

A series circuit is a circuit in which component are connected end to end so that the circuit will have only one path through which electric current flows.

In series connection voltage gets divided.

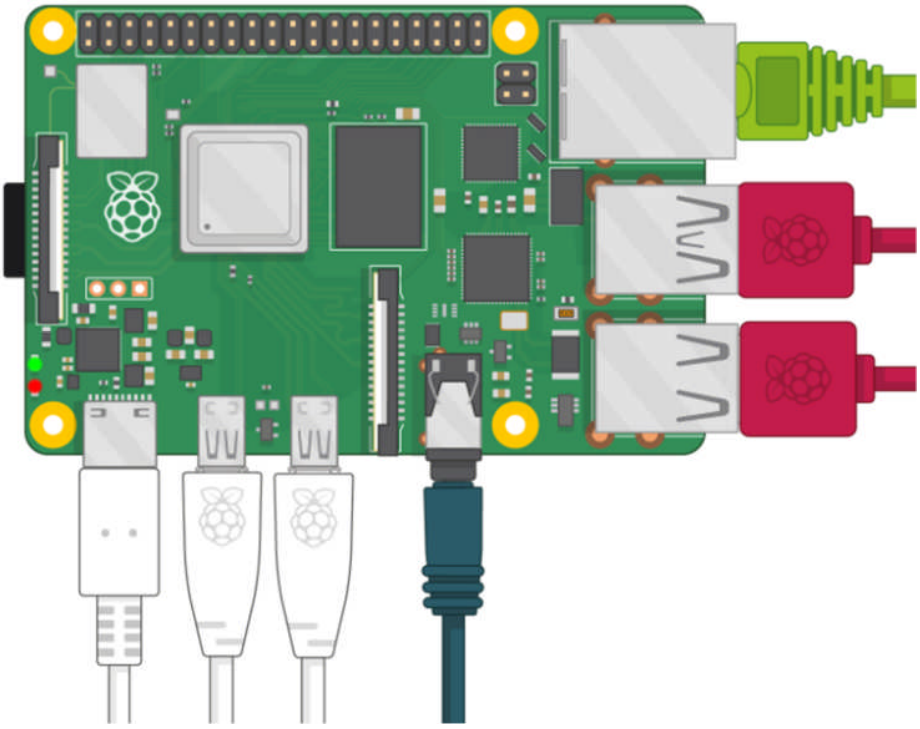
Parallel Connection

Components are said to be connected in parallel when the end of each of the component have a common point or junction and the other ends are also connected to a common point or junction.

In parallel connection current gets divided.

Step- 3

Complete Setup:-



Connection

Display Raspberry Pi OS

After successfully connecting check out Raspberry OS on Display.

Click on the raspberry icon and select preference option.

Go into configuration window and enable all interface like i2c ,vnc etc.

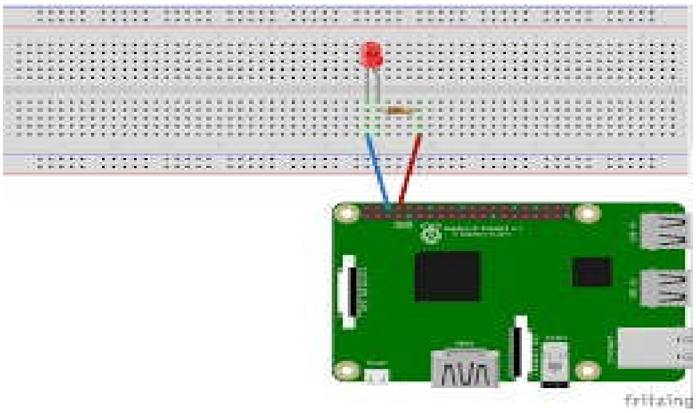
Display

Step-4

Blink an LED With Raspberry-pi

Procedure-

- Take breadboard, led and resister.
- Design a circuit for blinking LED using a breadboard.
- Connect the board with a jumper wire and resister with raspberry-pi any gpio pin 17
- Create a new file by clicking New.
- Save the new file by clicking Save. ...
- Enter the code given into the next slide.
- Save the file and run the code by clicking on Run.
- The LED should be flashing on and off.



raspberry board

Code for led blink-

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
LED_PIN = 17
GPIO.setup(LED_PIN, GPIO.OUT)
try:
while True:
GPIO.output(LED_PIN, GPIO.HIGH)
time.sleep(1)
GPIO.output(LED_PIN, GPIO.LOW)
time.sleep(1)
except KeyboardInterrupt:
GPIO.cleanup()
```