## What is Raspberry Pi?

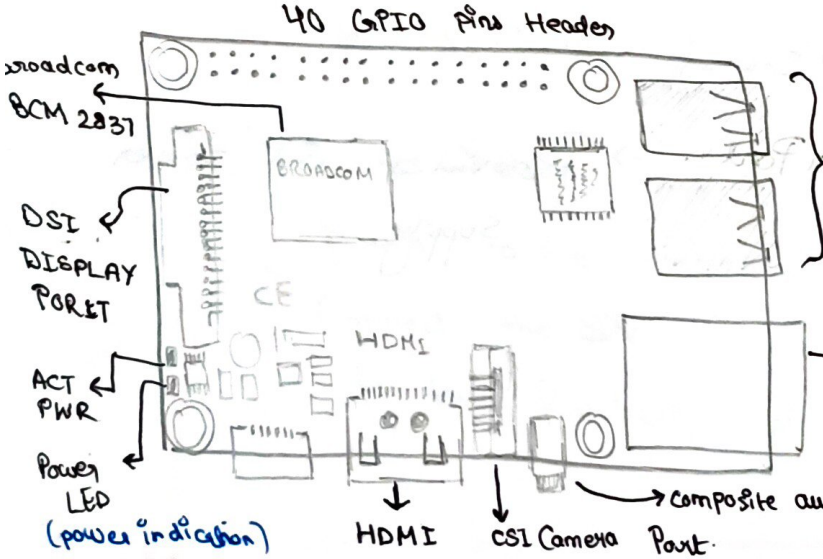
- it is a small single board computer. By connecting peripherals like keyboard, mouse, display to the R pi. it will act as a mini personal Computer.
- popularly used for real-time Image-/Video Processing, IoT based applications & Robotics applications.
- low power consumption.
- it is more than computer as it gives access to on chip-hardware

## OS for Raspberry Pi

- its default OS is Raspbian OS. Also they provide NOOBS OS for Raspberry Pi. We can install third party OS like Ubuntu, Archlinux, Windows 10 IOT core.
- Raspbian has GUI which includes tools for browsing, Python programming, office, games etc.

* We should use SD card (atleast 8 GB recommended) to store the OS.
* It has 40 GPIO's we can connect devices like LED, motors, sensors etc & can control them too.

## Raspberry Pi 3 Hardware Details




* HDMI - High Definition multimedia Interface. for computer monitors, TV etc.
* CSI Camera port - provides electrical connection between broadcom & pi camera.
* ACT PWR - shows the SD card availability
* A single card processor

## Peripherals in Raspberry 3.

↳ external devices you can connect to the Raspberry pi to extend its functionalities are—

- Camera → CSI port
- Sensors → GPIO pins or (I2C / SPI) protocols.
- Motors → via GPIO pins but using Motor drivers
- Displays → HDMI for monitors or GPIO's for small displ
- Keyboards/Mouse → via USB ports
- Storage → SD card for OS storage.

## Ports in Raspberry Pi 3 & their Ranges :-

- Camera Port. → 3.3V logic.
- GPIO ports → connects camera.
- Display Port (DSI) → connects LCD
- HDMI Port → outputs audio & video
- USB Ports → (4 x USB 2.0) connects peripherals
- Audio Jack → (3.5 mm)
- Ethernet Port. → wired internet.
- MicroSD Slot. → Storage
- Micro USB Power Port → Provides 5V power supply.

## Range of GPIO ports :-

40 pins of GPIO are divided into → GPIO pins
→ power pins
→ communication pins

Power pins - 3.3V - supply power to low-voltage peripherals
5V - for high-voltage peripherals.
Ground - ground connection for circuits

Communication pins-
- I2C → pins for connecting I2C devices
- SPI → for high speed devices connection
- UART → for serial communication

GPIO pins-
- 26 programmable GPIO pins that you can use as inputs/outputs
- example - Use GPIO17 for turning on an LED or reading data from a sensor.

## How can two motors be Supported on Raspberry pi ?

Step 1: Use a Motor driver. (eg- L298N motor driver)

Step 2: Connect the motors - each motor connects to one o the two output channels on the L298N module.

Step 3: Control pins :- Connect the IN1, IN2, IN3, IN pins of the L298N module to GPIO pins on the Raspberry Pi to control the motors ~~dee~~ direction

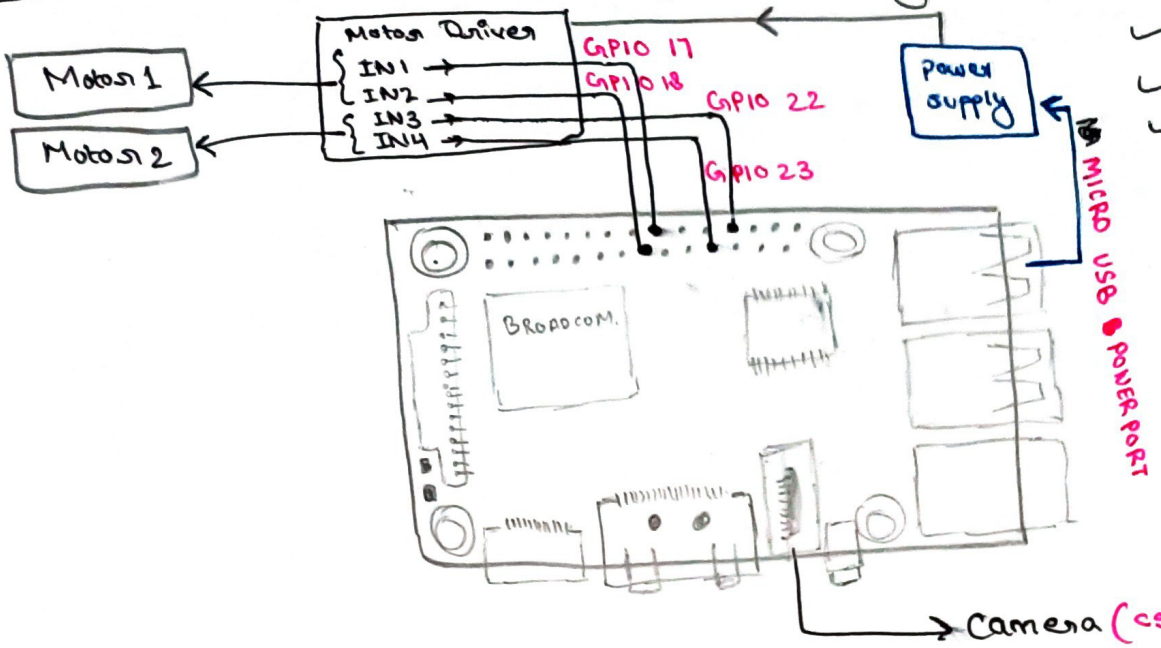
Step 4: Power Setup :- Use an external battery to supply power to the motors.
Connect the 5V & GND pins of the L298N to the Raspberry pi for communication

## Python Code Example-

```
import RPi.GPIO as GPIO
import time

# setup
GPIO.setmode (GPIO.BCM)
GPIO.setup (17, GPIO.OUT)  # Motor 1  IN1
GPIO.setup (18, GPIO.OUT)  # Motor 1  IN2
GPIO.setup (22, GPIO.OUT)  # Motor 2  IN3
GPIO.setup (23, GPIO.OUT)  # Motor 2  IN4

# Motor control
def motor1_forward():
    GPIO.output (17, GPIO.HIGH)
    GPIO.output (18, GPIO.LOW)

def motor1_backward():
    GPIO.output (17, GPIO.LOW)
    GPIO.output (18, GPIO.HIGH)

def motor2_forward():
    GPIO.output (22, GPIO.HIGH)
    GPIO.output (23, GPIO.LOW)

def motor2_backward():
    GPIO.output (22, GPIO.LOW)
    GPIO.output (23, GPIO.HIGH)

# Example usage.
motor1_forward()
motor2_backward()
time.sleep (2)
GPIO.cleanup()
```

## Schematic of our Platform using Raspberry Pi :-

- camera
- motor 1
- motor 2
- motor driver
- power supply
- power backup
- 




## What is PWM?

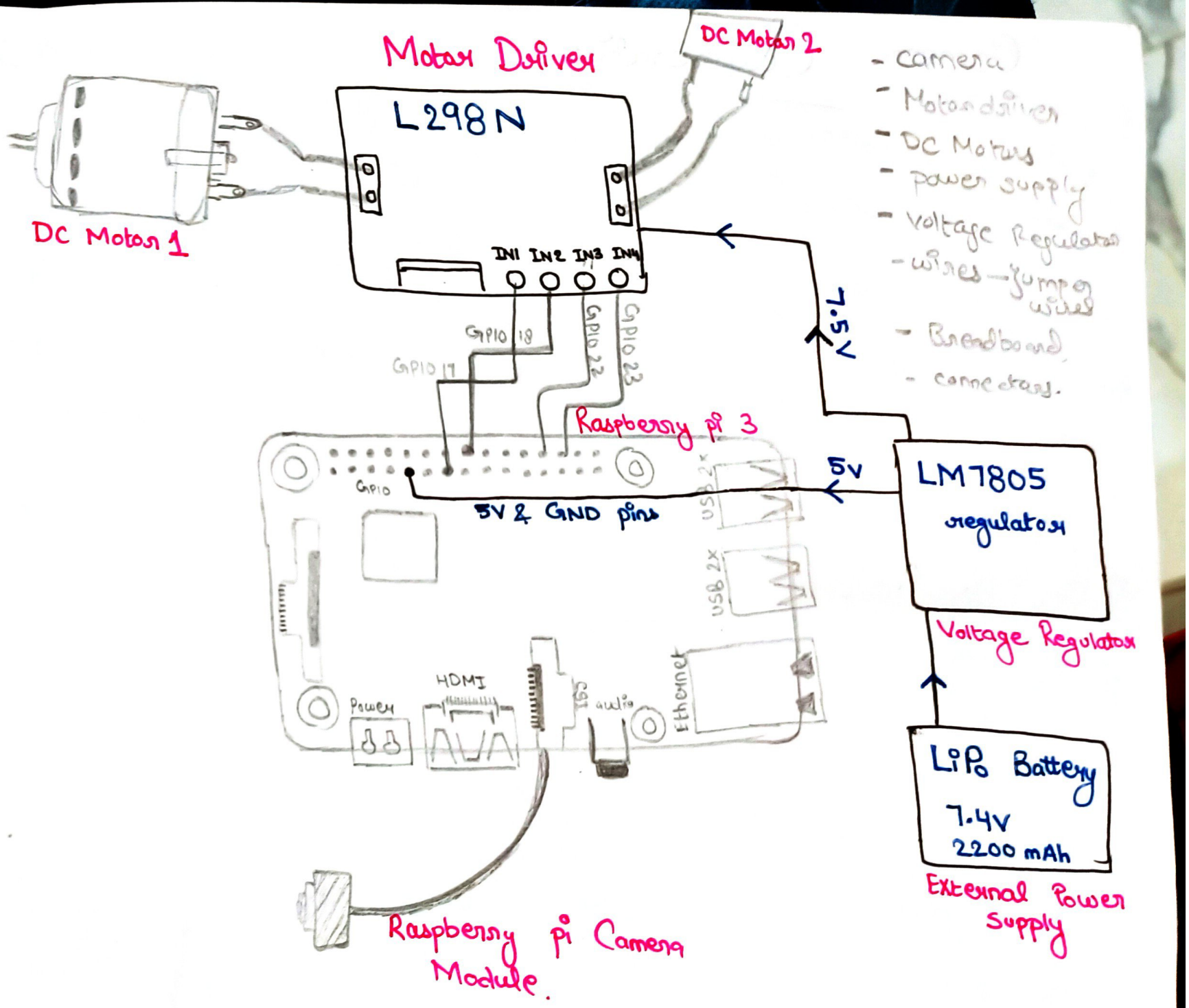
→ Pulse Width Modulation (PWM) is used to simulate varying levels of power by rapidly switching a signal between on & off states.

→ it controls the amt. of power supply to a device. it is commonly used to regulate the speed of a DC motor Control the brightness of LED's etc.

→ key words: Duty Cycle :- The % of one cycle in which the signal is 'on' a 50% duty cycle means its on half the time & off half time.

Frequency :-
↓
the rate at which PWM completes a cycle of on & off state.

* In our project we can use PWM to manage the speed of motors




Component List with their Part number :-

## Camera Calliberation -

↳ it is a process that determine the intrinsic & extrinsic parameters of a camera to correct lens distortions. & establish the relationship between the camera pixels coordinates & real world units.

This is essential for application requiring accurate measurements on 3D reconstructions.

## Types of Distortions -

① Radial Distortions - causes straight lines to appear curved, commonly seen as "barrel" or "fish eye" effect.

② Tangential Distortions - occurs when the lens is not perfectly parallel to the imaging plane, causing images to tilt.

By calliberating these distortions can be corrected, leading to more accurate representation of captured scenes.

## Performing Camera Calliberation on Laptop -

* install python
* install openCV using pip [pip install opencv-python]
* Capture Calliberation images - (for eg chessboard., capture multiple images)
* Run calliberation.
  - ↳ Utilize OpenCV functions to detect the pattern & compute the calliberation parameters.
  - ↳ OpenCV's documentation provides a detailed tutorial on this process. (docs.opencv.org — camera calliberation)

For radial distortion —

$$x_{distortion} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$

$$y_{distortion} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$

So for an undistorted pixel point at $(x, y)$ coordinates, its position on the distorted image will be $(x_{distorted}, y_{distorted})$

Tangential distortion occurs because the image taking lenses are not parallel to the imaging plane.

$$x_{distorted} = x + [2p_1 xy + p_2(r^2 + 2x^2)]$$

$$y_{distorted} = y + [p_1(r^2 + 2y^2) + 2p_2 xy]$$

So we have 5 distortion parameters which in Open CV are presented as one row matrix with 5 columns

$$distortion\_coefficients = (k_1 \; k_2 \; p_1 \; p_2 \; k_3)$$

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

↳ for unit conversion we use this formula.

$f_x$ & $f_y$ → camera focal lengths

$c_x$ & $c_y$ → optical centers expressed in pixels co-ordinates

The matrix containing these 4 parameters is referred to as camera matrix.

3 types of object calibration → Classical black white chessboard
→ symmetrical circle pattern
→ Asymmetrical circle pattern.