

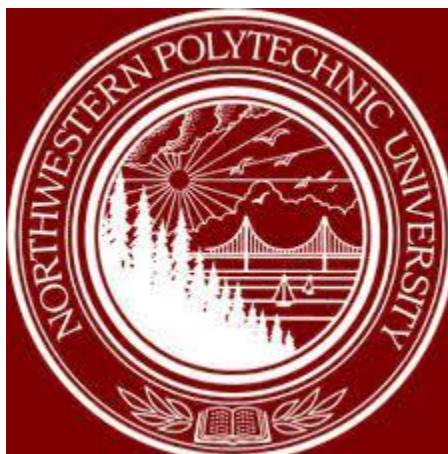
# Controlling DC motor using RASPBERRY PI 3 GPIO



---

Aditi Vaidya

vaidya19598@mail.npu.edu



Northwestern Polytechnic University Fremont, CA Copyright © 2017, All Rights Reserved .

# Table of Contents

## 1 Introduction

1.1 Objective

1.2 Requirement

1.3 Principle

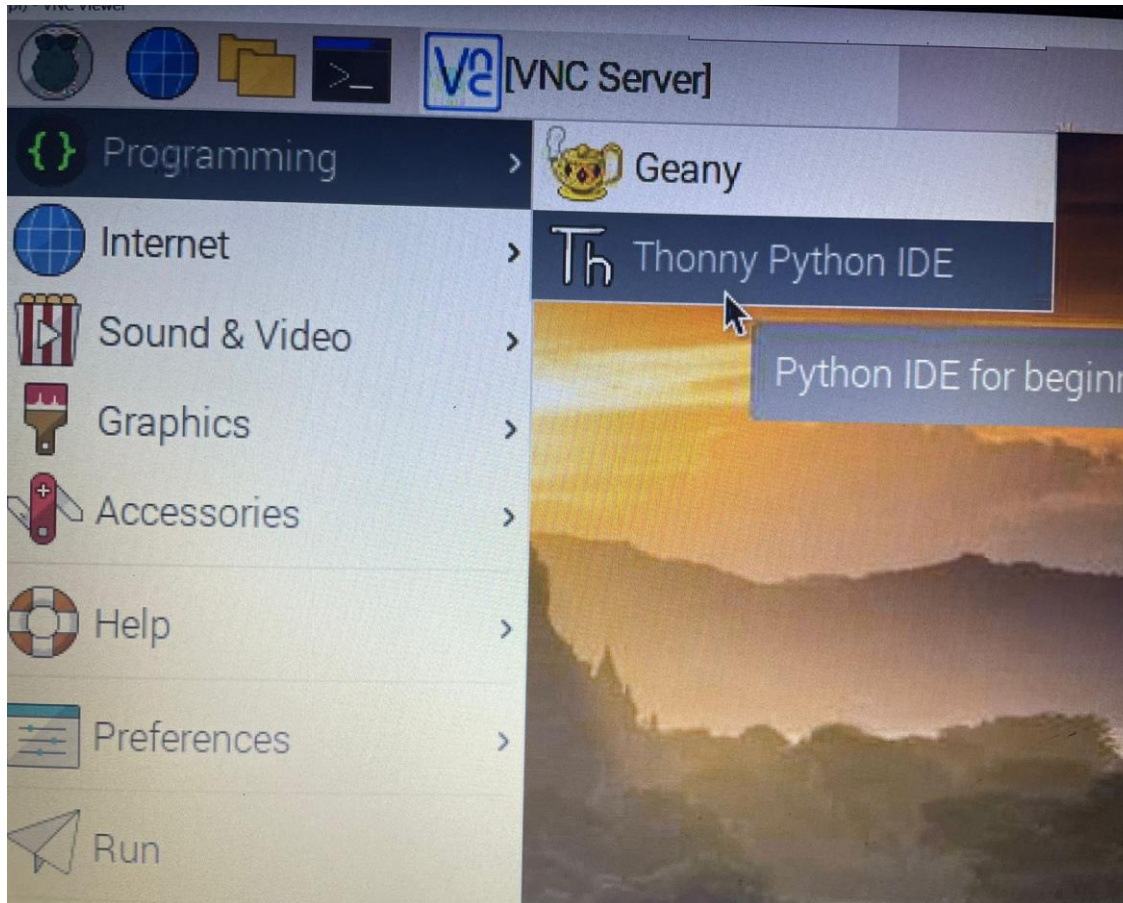
## 2 Hardware connections

## 3 Working

3.1 Programming

## 1.1 Objective

The main objective is to drive a DC motor . Since the DC Motor needs a larger current, for safety purpose, here we use the Power Supply Module to supply motors. We achieve this with Python programming using the Raspberry Pi 3 model B development board with inbuilt thony ide.

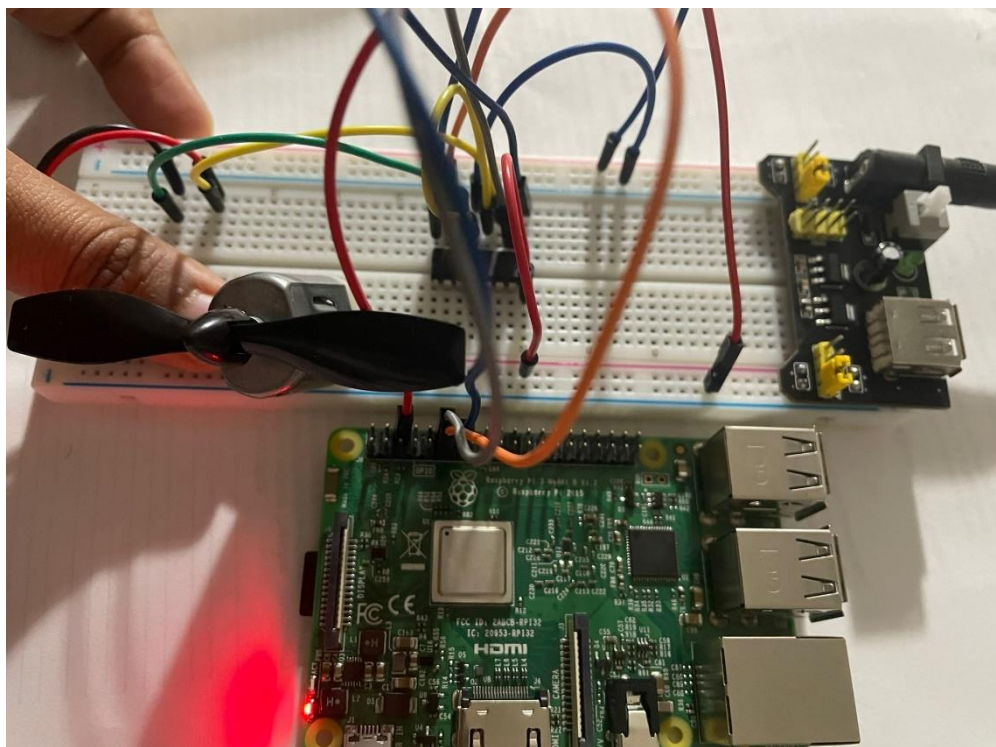
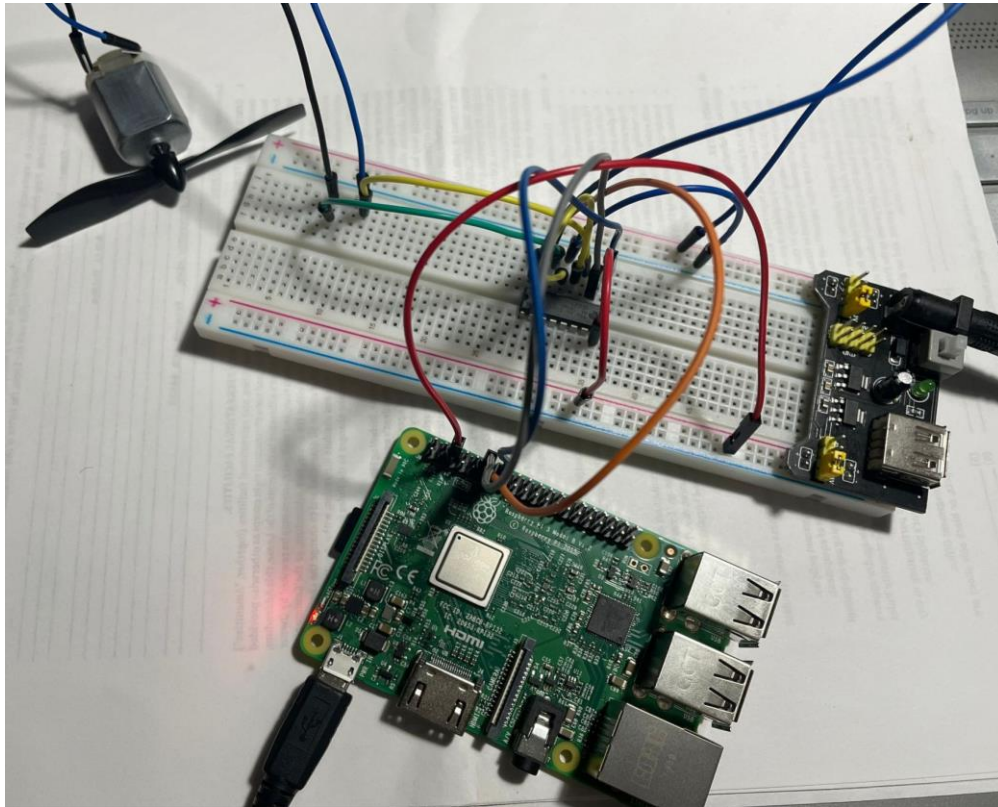


## 1.2 Requirement

- ✓ 1 \* Raspberry Pi
- ✓ 1 \* Breadboard
- ✓ 1 \* L293D
- ✓ 1 \* Power Module
- ✓ 1 \* Resistor (1K $\Omega$ )
- ✓ Jumper wires

### 1.3 Principle

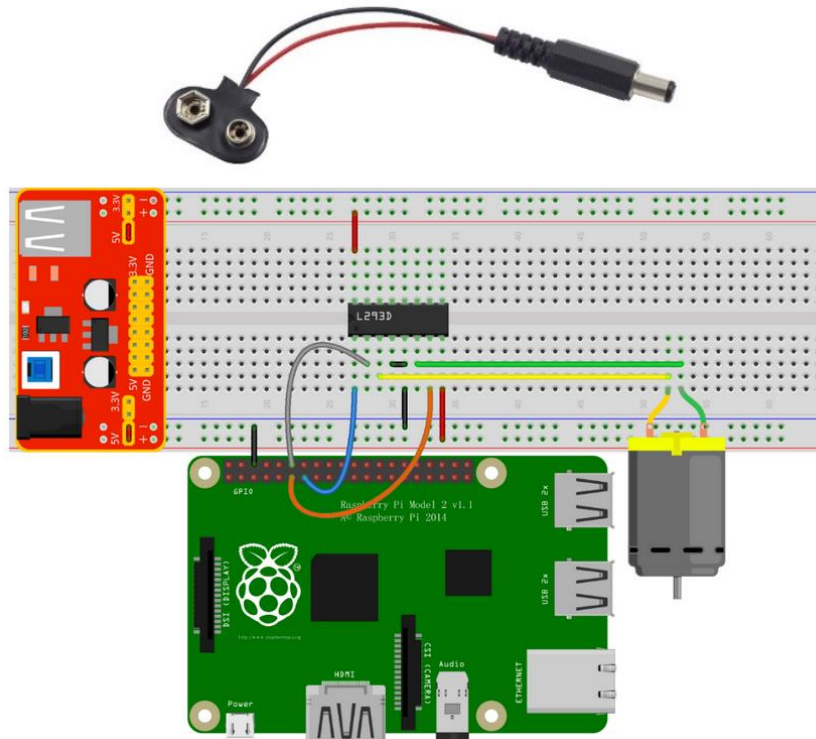
In this experiment ,when Raspberry Pi output is supplied with low (3.3 V) by programming, the power module will be used controlled the supply to the DC motor which will help the DC motor to rotate in clockwise or anticlockwise .





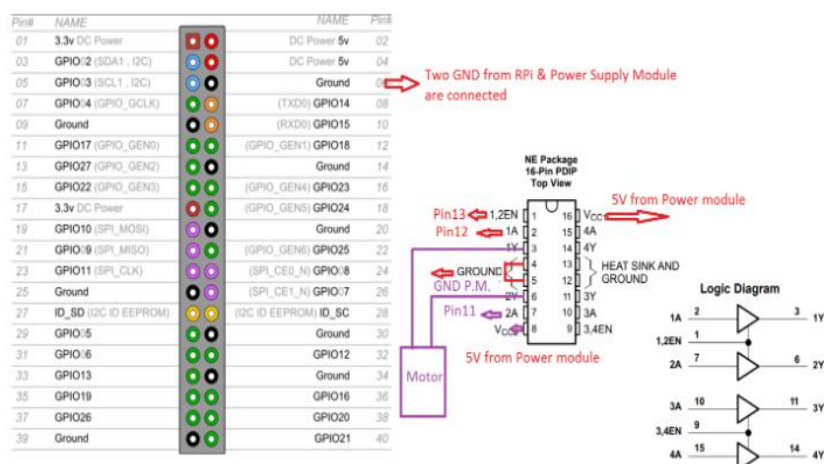
## 2.0 Hardware connections

Use bread board and make connections as shown in visual representation below:



## 3.0 Working

Once the wireless connection is established then open thonny Python IDE which mostly is pre installed in raspberry Pi's.



1. Make sure the Python library is uploaded and running .

2. Type in the program, and check for any errors.
3. Run the program
4. Turn on the power module .
5. Check the DC motor.

### 3.1 Programming

Enter the following program on Thonny Python IDE :

```
import RPi.GPIO as GPIO
import time

MotorPin1 = 11    # pin11
MotorPin2 = 12    # pin12
MotorEnable = 13  # pin13

def setup():
    GPIO.setmode(GPIO.BOARD)          # Numbers GPIOs by physical location
    GPIO.setup(MotorPin1, GPIO.OUT)    # mode --- output
    GPIO.setup(MotorPin2, GPIO.OUT)
    GPIO.setup(MotorEnable, GPIO.OUT)
    GPIO.output(MotorEnable, GPIO.LOW) # motor stop

def loop():
    while True:
        print 'Press Ctrl+C to end the program...'
        GPIO.output(MotorEnable, GPIO.HIGH) # motor driver enable
        GPIO.output(MotorPin1, GPIO.HIGH)   # clockwise
        GPIO.output(MotorPin2, GPIO.LOW)
        time.sleep(5)

        GPIO.output(MotorEnable, GPIO.LOW) # motor stop
        time.sleep(5)

        GPIO.output(MotorEnable, GPIO.HIGH) # motor driver enable
        GPIO.output(MotorPin1, GPIO.LOW)   # anticlockwise
        GPIO.output(MotorPin2, GPIO.HIGH)
        time.sleep(5)

        GPIO.output(MotorEnable, GPIO.LOW) # motor stop
        time.sleep(5)

def destroy():
    GPIO.output(MotorEnable, GPIO.LOW) # motor stop
    GPIO.cleanup()                   # Release resource

setup()
try:
    loop()
except KeyboardInterrupt: # When 'Ctrl+C' is pressed, the child program destroy() will be executed.
    destroy()
```