**Conectando la Raspberry pi 4 al módulo L298N y motor de Corriente directa.**

El objetivo es poder interactuar desde la raspberry pi con un motor de corriente directa por medio del módulo L298N, usando un script de python.

Primero que nada, cuando se dice interactuar es en el sentido de poder controlar algunos aspectos del motor de corriente directa, como son: avanzar hacia una dirección o la contraria, detenerlo o cambiar su velocidad por medio del voltaje suministrado.

**Materiales necesarios.**

1. Raspberry Pi 4

2. Conector tipo T

3. Flex para conector tipo T

4. Protoboard.

5. Modulo puente H, L298N

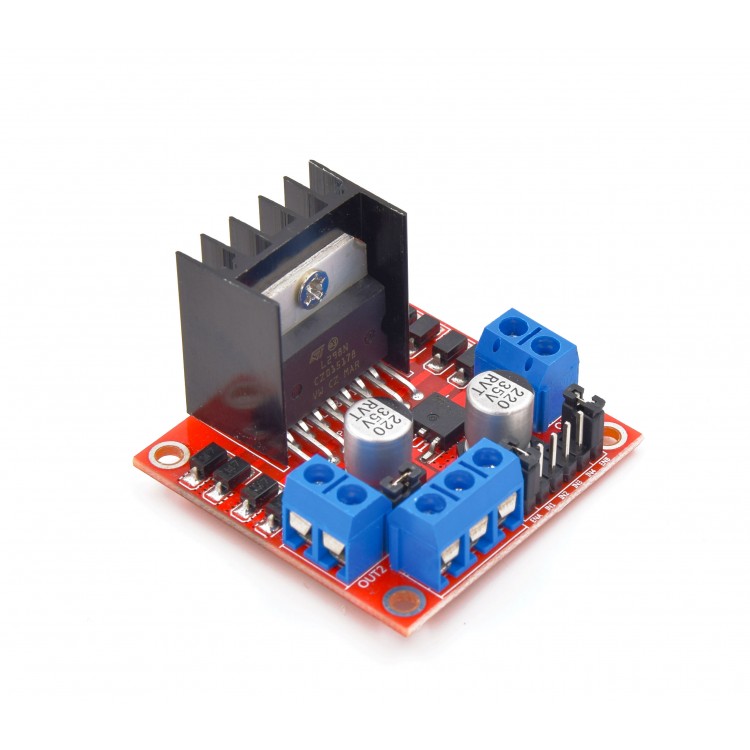
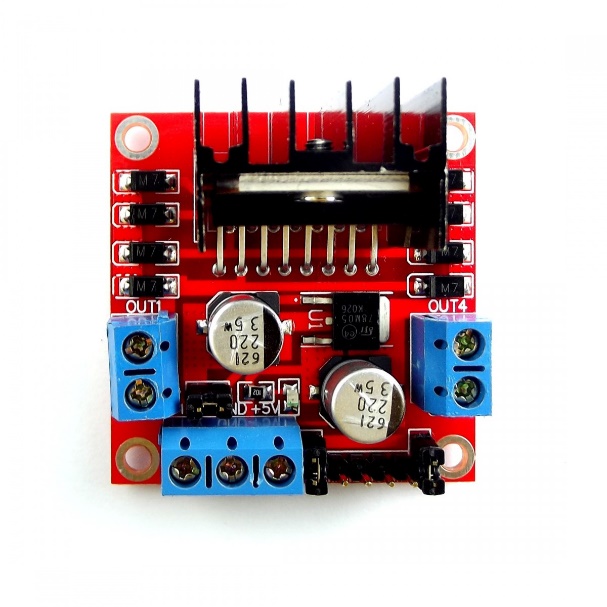
6. Motor de corriente directa de 12 v con encoder,

7. Adaptador fuente de 12 v.

8. Fuente de voltaje para Raspberry pi

**Módulo L298N**

El módulo de controlador de motor L298N está basado en el potente IC del controlador de motor L298N. Este IC, junto con algunos otros componentes conforman el módulo de controlador de motor L298N.



Este controlador tiene la capacidad para manejar dos motores al mismo tiempo,

La imagen de arriba muestra el módulo de controlador de motor L298N con los nombres de todos los componentes que lo conforman. Por nombrar algunos, contiene terminales de tornillo para conectar motores y fuente de alimentación, cabezales macho para conectar diferentes pines como habilitar, IN1, IN2, etc.

Nota:

- Hay un puente cerca de las terminales de la fuente de alimentación del módulo de controlador de motor L298N. Si está conectado, habilitará el regulador de 5 V incorporado, el cual proporciona un suministro lógico de +5 V para L298N IC.

- Esto debe ser usado solo si la alimentación de voltaje es menor o igual a +12 V. Si es más de +12 V, se desconecta el puente para desactivar el regulador.

Diagrama del circuito entre Raspberry, L298N y Motor de corriente directa

Circuito propuesto para la conexión del puente H L298N con la fuente de voltaje de 12 v, el motor de corriente directa y el conector tipo T de la raspberry

Diagrama, Esquemático

Descripción generada automáticamente

Codigo en Python.

Primero conecta la fuente de alimentación de 12 V al módulo de controlador de motor L298N, después, haga que las terminales GND del Módulo de controlador de motor L298N sean comunes (conéctelos).

Ahora, dado que estamos controlando un solo motor DC, necesitamos usar un solo canal de L298N. Para hacerlo, conecte el pin ENA de L298N al pin físico 22 (GPIO25) de Raspberry Pi.

Viniendo a las entradas del motor, conecte el IN1 y el IN2 del Módulo L298N al pin físico 16 y 18 (GPIO23 y GPIO24). Eso es todo. El resto del trabajo es hecho por el script Python.

Antes de escribir el código debemos de instalar la librearia L298N en Python, lo cual es muy fácil de hacer desde Thonny, el cuál es el editor que usaremos para todos los programas en Python. Nos vamos a Thonny y desde ahí abrimos el menú **TOOLS** y luego **Manage Packages** y se abre una ventana como la de abajo

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Escogemos la librería y la instalamos. Con lo anterior estamos listos para el código.

import RPi.GPIO as GPIO

from time import sleep

# Pines de conexión con el shield L298N

in1 = 24

in2 = 23

en = 25

GPIO.setmode(GPIO.BCM) # Para fijar el modo para los pines

GPIO.setup(in1,GPIO.OUT) # Definimos a in1 como de salida en raspberry

GPIO.setup(in2,GPIO.OUT) # Definimos a in2 como de salida en raspberry

GPIO.setup(en,GPIO.OUT) # Definimos a en como de salida en raspberry

GPIO.output(in1,GPIO.LOW)# Para estar seguros que el motor se detiene antes de comenzar

GPIO.output(in2,GPIO.LOW)# Para estar seguros que el motor se detiene antes de comenzar

p=GPIO.PWM(en,1000) # La frecuencia para el pulso PWM en el pin en, de la variable p

p.start(25) # Para comenzar con el 25% con el movimiento del motor

print("\n")

print("Recuerda que debes de suministrar un porcentaje del voltaje (entre 0 y 100)")

print("Esperando...")

print("\n")

#Iniciamos el motor de dc con el 0% del voltaje total (12 v)

GPIO.output(in1,GPIO.HIGH)

GPIO.output(in2,GPIO.LOW)

p.ChangeDutyCycle(0)

while True:

x=input("% del voltaje\n")

z=int(x) # Convertimos el valor de str de x a int (entero)

print(x)

if z == -1: # Para salirnos del ciclo

p.ChangeDutyCycle(0)

break

p.ChangeDutyCycle(z) # Cambiamos el porcentaje de voltaje (0-100)

print("Fin del programa, ¡¡¡Saludos!!!")

#Para asegurarnos de apagar el motor

GPIO.output(in1,GPIO.LOW)

GPIO.output(in2,GPIO.LOW)

GPIO.cleanup() # Limpiamos todos los pines