**Πείραμα με DC Motor σε συνδυασμό με Chip L293D**

1. **Περιγραφή:**

Στο παρακάτω πείραμα χρησιμοποιώ module με DC Motor το οποίο περιέχει και chip L293D με το οποίο μπορώ να ελέγχω την κατεύθυνση και το σήμα (ON-OFF) του κινητήρα.

Ο κινητήρας συνεχούς ρεύματος είναι μια συσκευή που μετατρέπει την ηλεκτρική ενέργεια σε μηχανική. Λόγω της ευκολίας ελέγχου, χρησιμοποιείται συνήθως σε ανεμιστήρες, ηλεκτρονικά παιχνίδια, ξυριστικές μηχανές κ.λπ.

1. **Υλικά:**

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **1\* Raspberry pi** | **1\* GPIO Extension board** | **1\* 40 pin colorful jumper wires** |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **1\* Breadboard** | **Jumper Wires** | 1\* DC motor with voltage up to 5V and speed >5000rpm with chip L293D |

1. **Γνώση υλικού:**

**DC-Motor:**

Ο κινητήρας συνεχούς ρεύματος (DC Motor) είναι ο πιο συνηθισμένος τύπος κινητήρα. Οι κινητήρες συνεχούς ρεύματος έχουν συνήθως μόνο δύο καλώδια, ένα θετικό και ένα αρνητικό. Εάν συνδέσετε αυτούς τους δύο αγωγούς απευθείας σε μια μπαταρία, ο κινητήρας θα περιστραφεί. Εάν αλλάξετε τους αγωγούς, ο κινητήρας θα περιστραφεί προς την αντίθετη κατεύθυνση.



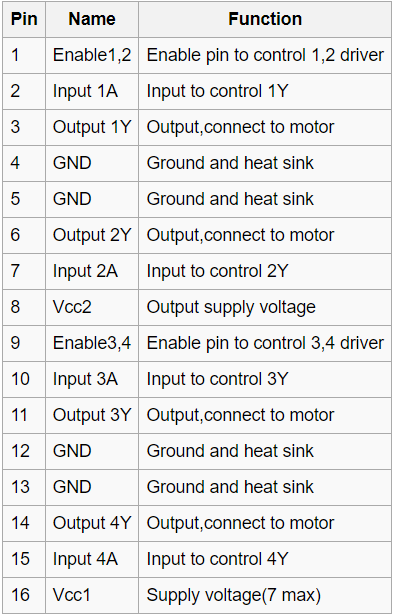
Δεν οδηγούμε τον κινητήρα απευθείας από τις ακίδες της πλακέτας Raspberry. Αυτό μπορεί να προκαλέσει ζημιά στην πλακέτα. Χρησιμοποιήστε ένα κύκλωμα οδήγησης ή ένα ολοκληρωμένο κύκλωμα.Στην προκειμένη περίπτωση χρησιμοποιούμε το chip L293D.

**L293D:**

Ο L293D είναι τετραπλός οδηγός υψηλής έντασης ρεύματος . Έχει σχεδιαστεί για να παρέχει αμφίδρομα ρεύματα οδήγησης έως και 600-mA σε τάσεις από 4,5 V έως 36 V. Και οι δύο διατάξεις έχουν σχεδιαστεί για να οδηγούν επαγωγικά φορτία όπως ρελέ, κινητήρες συνεχούς και διπολικού ρεύματος, καθώς και άλλα φορτία υψηλής έντασης/υψηλής τάσης σε εφαρμογές θετικής τροφοδοσίας.

A circuit board with numbers and symbols

Description automatically generated



**Πειραματική συσκευή:**

■ Πίνακας ελέγχου Raspberry Pi

■ Μονάδα DC-Motor

■ Γραμμές σύνδεσης

**Πειραματική σύνδεση:**

|  |  |
| --- | --- |
| The DC-Motor module | Raspberry Pi |
| Vcc | 5V |
| Gd | GND |
| DIR | 17 |
| SIG | 18 |

1. **Πειραματικό συμπέρασμα:**

Με τον παρακάτω κώδικα ρυθμίζουμε τις ακίδες GPIO για DIR (κατεύθυνση)και SIG(σήμα) και τις χρησιμοποιεί για τον έλεγχο της κατεύθυνσης και της ταχύτητας του κινητήρα συνεχούς ρεύματος που είναι συνδεδεμένος στο ολοκληρωμένο κύκλωμα οδήγησης κινητήρα L293D. Προσαρμόστε τους αριθμούς των ακροδεκτών GPIO στον κώδικα ώστε να ταιριάζουν με την πραγματική σας καλωδίωση. Βεβαιωθείτε ότι παρέχετε μια εξωτερική παροχή ρεύματος για την τροφοδοσία του κινητήρα μέσω του οδηγού κινητήρα L293D.

Ο κινητήρας ξεκινά προς μια κατεύθυνση ,σταματά για 2 δευτερόλεπτα ,κάνει περιστροφή προς την άλλη κατεύθυνση και ξανασταματά συνεχώς.

**Βεβαιωθείτε ότι έχετε εγκαταστήσει τη βιβλιοθήκη RPi.GPIO στο Raspberry Pi σας, αν δεν είναι ήδη εγκατεστημένη. Μπορείτε να το κάνετε αυτό με την ακόλουθη εντολή στη γραμμή εντολών(command line):**

**pip install RPi.GPIO**

1. **Κώδικας Python:**

import RPi.GPIO as GPIO

import time

# Set the GPIO mode

GPIO.setmode(GPIO.BCM)

# Define the GPIO pins for DIR and SIG

dir\_pin = 17  # Replace with the actual GPIO pin number for DIR

sig\_pin = 18  # Replace with the actual GPIO pin number for SIG

# Initialize the GPIO pins

GPIO.setup(dir\_pin, GPIO.OUT)

GPIO.setup(sig\_pin, GPIO.OUT)

try:

    while True:

        # Motor forward

        GPIO.output(dir\_pin, GPIO.HIGH)

        GPIO.output(sig\_pin, GPIO.HIGH)

        time.sleep(2)

        # Motor backward

        GPIO.output(dir\_pin, GPIO.LOW)

        GPIO.output(sig\_pin, GPIO.HIGH)

        time.sleep(2)

except KeyboardInterrupt:

    pass

# Stop the motor

GPIO.output(sig\_pin, GPIO.LOW)

# Clean up GPIO on script exit

GPIO.cleanup()

**Παραλλαγή πειράματος:**

**Πειραματική σύνδεση:**

|  |  |
| --- | --- |
| The DC-Motor module | Raspberry Pi |
| Vcc | 5V |
| Gd | GND |
| DIR | 17 |
| SIG | 18 |

Στον παρακάτω κώδικα Python ζητάμε συνεχώς από τον χρήστη να εισάγει μια ενέργεια ('f' για εμπρός, 'b' για πίσω, 's' για stop ή 'q' για quit) και εκτελεί την επιλεγμένη ενέργεια για διάρκεια 10 δευτερολέπτων.

**Κώδικας Python:**

import RPi.GPIO as GPIO

import time

# Set the GPIO mode

GPIO.setmode(GPIO.BCM)

# Define the GPIO pins for DIR and SIG

dir\_pin = 17  # Replace with the actual GPIO pin number for DIR

sig\_pin = 18  # Replace with the actual GPIO pin number for SIG

# Initialize the GPIO pins

GPIO.setup(dir\_pin, GPIO.OUT)

GPIO.setup(sig\_pin, GPIO.OUT)

try:

    while True:

        user\_input = input("Enter action (f: forward, b: backward, s: stop, q: quit): ").lower()

        if user\_input == 'f':

            # Move motor forward

            GPIO.output(dir\_pin, GPIO.HIGH)

            GPIO.output(sig\_pin, GPIO.HIGH)

            time.sleep(10)

        elif user\_input == 'b':

            # Move motor backward

            GPIO.output(dir\_pin, GPIO.LOW)

            GPIO.output(sig\_pin, GPIO.HIGH)

            time.sleep(10)

        elif user\_input == 's':

            # Stop the motor

            GPIO.output(sig\_pin, GPIO.LOW)

        elif user\_input == 'q':

            break

except KeyboardInterrupt:

    pass

# Stop the motor

GPIO.output(sig\_pin, GPIO.LOW)

# Clean up GPIO on script exit

GPIO.cleanup()

Το ίδιο αποτέλεσμα έχω και με τα παρακάτω,απλά εδώ χρησιμοποιώ συναρτήσεις για κάθε μια από τις επιλογές του χρήστη.

import RPi.GPIO as GPIO

import time

# Set the GPIO mode

GPIO.setmode(GPIO.BCM)

# Define the GPIO pins for DIR and SIG

dir\_pin = 17  # Replace with the actual GPIO pin number for DIR

sig\_pin = 18  # Replace with the actual GPIO pin number for SIG

# Initialize the GPIO pins

GPIO.setup(dir\_pin, GPIO.OUT)

GPIO.setup(sig\_pin, GPIO.OUT)

# Function to move the motor forward

def move\_forward():

    GPIO.output(dir\_pin, GPIO.HIGH)

    GPIO.output(sig\_pin, GPIO.HIGH)

# Function to move the motor backward

def move\_backward():

    GPIO.output(dir\_pin, GPIO.LOW)

    GPIO.output(sig\_pin, GPIO.HIGH)

# Function to stop the motor

def stop\_motor():

    GPIO.output(sig\_pin, GPIO.LOW)

try:

    while True:

        user\_input = input("Enter action (f: forward, b: backward, s: stop, q: quit): ").lower()

        if user\_input == 'f':

            move\_forward()

            time.sleep(10)  # Move forward for 10 seconds

        elif user\_input == 'b':

            move\_backward()

            time.sleep(10)  # Move backward for 10 seconds

        elif user\_input == 's':

            stop\_motor()

        elif user\_input == 'q':

            break

except KeyboardInterrupt:

    pass

# Stop the motor

stop\_motor()

# Clean up GPIO on script exit

GPIO.cleanup()