

Practical no: 7

Title: Write an application using Raspberry-Pi /Beagle board to control the operation of stepper motor.

Name: Aditi Dinesh Mulay

Class: T.E. Computer

Subject: ES&IOT

Div: A

Roll no: 02

PRN No. 71918146B

Title

Write an application using Raspberry-Pi/Beagle board to control the operation of stepper motor.

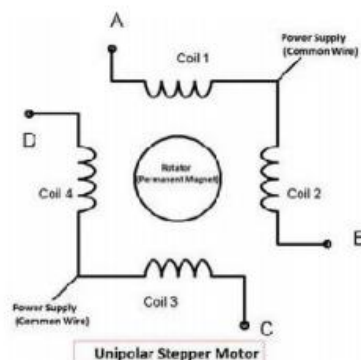
TheoryStepper Motor

In Stepper Motor, as the name itself says, the rotation of shaft is in step form. There are different types of Stepper Motor, in here we will be using the most popular one that is Unipolar Stepper Motor. Unlike DC Motor, we can rotate stepper motor to any particular angle by giving it proper instructions.

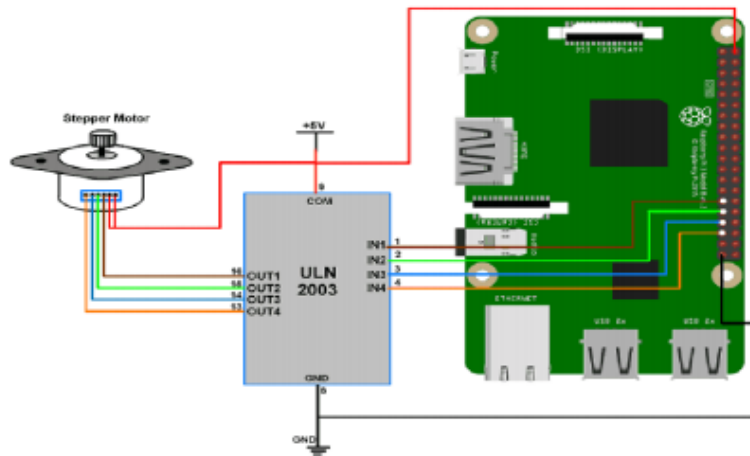
To rotate this 4 stage Stepper Motor, we will deliver power pulses by using Stepper Motor Driver Circuit. The Driver circuit takes logic triggers from PI. If we control the logic triggers, we control the power pulses and hence the speed of stepper motor.

There are 40 GPIO output pins in Raspberry Pi 2. But out of 40, only 26 GPIO pins can be programmed. Some of these pins perform some special functions. With special GPIO put aside, we have only 17 GPIO remaining. Each of these 17 GPIO pin can deliver a maximum of 15mA current. And the sum of currents from all GPIO pins cannot exceed 50mA.

There are +5V (Pin 2 & 4) and +3.3V (Pin 1 & 17) power output pins on the board for connecting other modules.



Unipolar Stepper motor Fig.1



and sensors. These power rails cannot be used to drive the stepper motor, because we need more power to rotate it. So we have to deliver the power to stepper motor from another power source. My stepper motor has a voltage rating of 9V so I'm using a 9V battery as my second power source. Search your stepper motor model no. to know voltage using rating. Depending on rating choose the secondary source approx.

Sample program.

Stepper Motor Interfacing with Raspberry pi.

```
import RPi.GPIO as GPIO
from time import sleep
import sys
```

```
# assign GPIO pins for motor
motor_channel = (29, 31, 33, 35)
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)
```

```
# for defining more than 1 GPIO channel as i/p o/p use
GPIO.setup(motor_channel, GPIO.OUT)
```

```
motor_direction = input('select motor direction a =  
anticlockwise , c = clockwise :')
```

```
while True:
```

```
    try:
```

```
        if (motor_direction == 'c');
```

```

print('motor running clockwise\n')
GPIO.output(motor-channel, (GPIO.HIGH, GPIO.Low,
                             GPIO.Low, GPIO.HIGH))

sleep(0.02)
GPIO.output(motor-channel, (GPIO.HIGH, GPIO.HIGH,
                             GPIO.Low, GPIO.Low))

sleep(0.02)
GPIO.output(motor-channel, (GPIO.Low, GPIO.HIGH,
                             GPIO.HIGH, GPIO.Low))

sleep(0.02)
GPIO.output(motor-channel, (GPIO.Low, GPIO.Low,
                             GPIO.HIGH, GPIO.HIGH))

sleep(0.02)

elif (motor-direction == 'a'):
print('motor running anticlockwise\n')
GPIO.output(motor-channel, (GPIO.HIGH, GPIO.Low,
                             GPIO.Low, GPIO.HIGH))

sleep(0.02)
GPIO.output(motor-channel, (GPIO.Low, GPIO.Low,
                             GPIO.HIGH, GPIO.HIGH))

sleep(0.02)
GPIO.output(motor-channel, (GPIO.Low, GPIO.HIGH,
                             GPIO.HIGH, GPIO.Low))

sleep(0.02)
GPIO.output(motor-channel, (GPIO.HIGH, GPIO.HIGH,
                             GPIO.Low, GPIO.Low))

sleep(0.02)

```

Conclusion:

Thus, we have implemented application of stepper motors using Python with Raspberry Pi.