Experiment No: 7

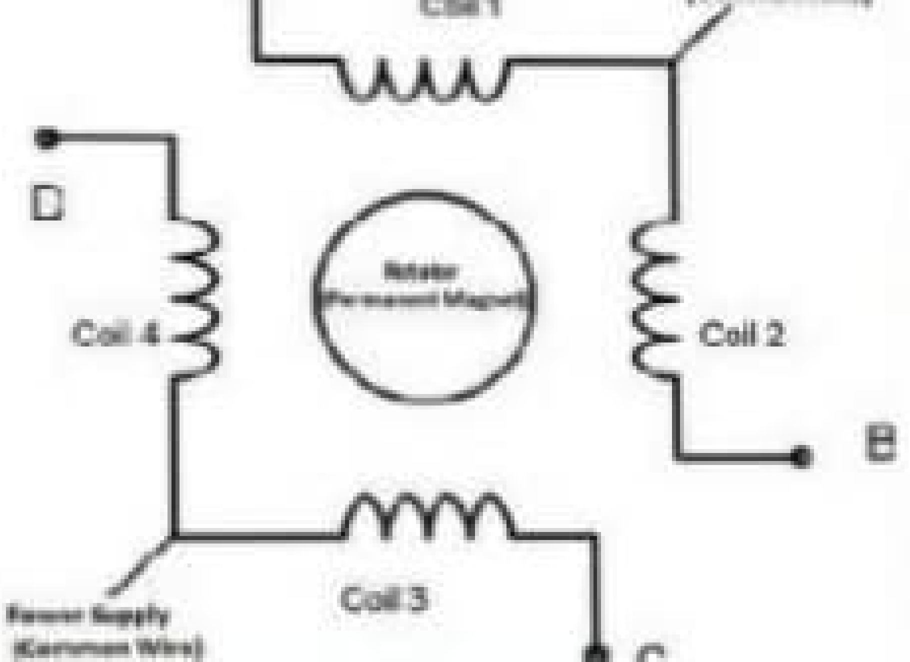
Title : Write an application using Raspberry Pi / Beagle Board to control the opeiation of Stepper Motor.

Theory :

Stepper Motor

In stepper Motar, as the name itself says, the rotation of shift 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 Partilular angle by giving it proper instructions.

unipolar Stepper Motor fig.1


Coil 1
(Common Wire)
D
Rotor
(Permanent Magnet)
Coil 4
Coil 2
B
Coil 3
Power Supply
(Common Wire)
C

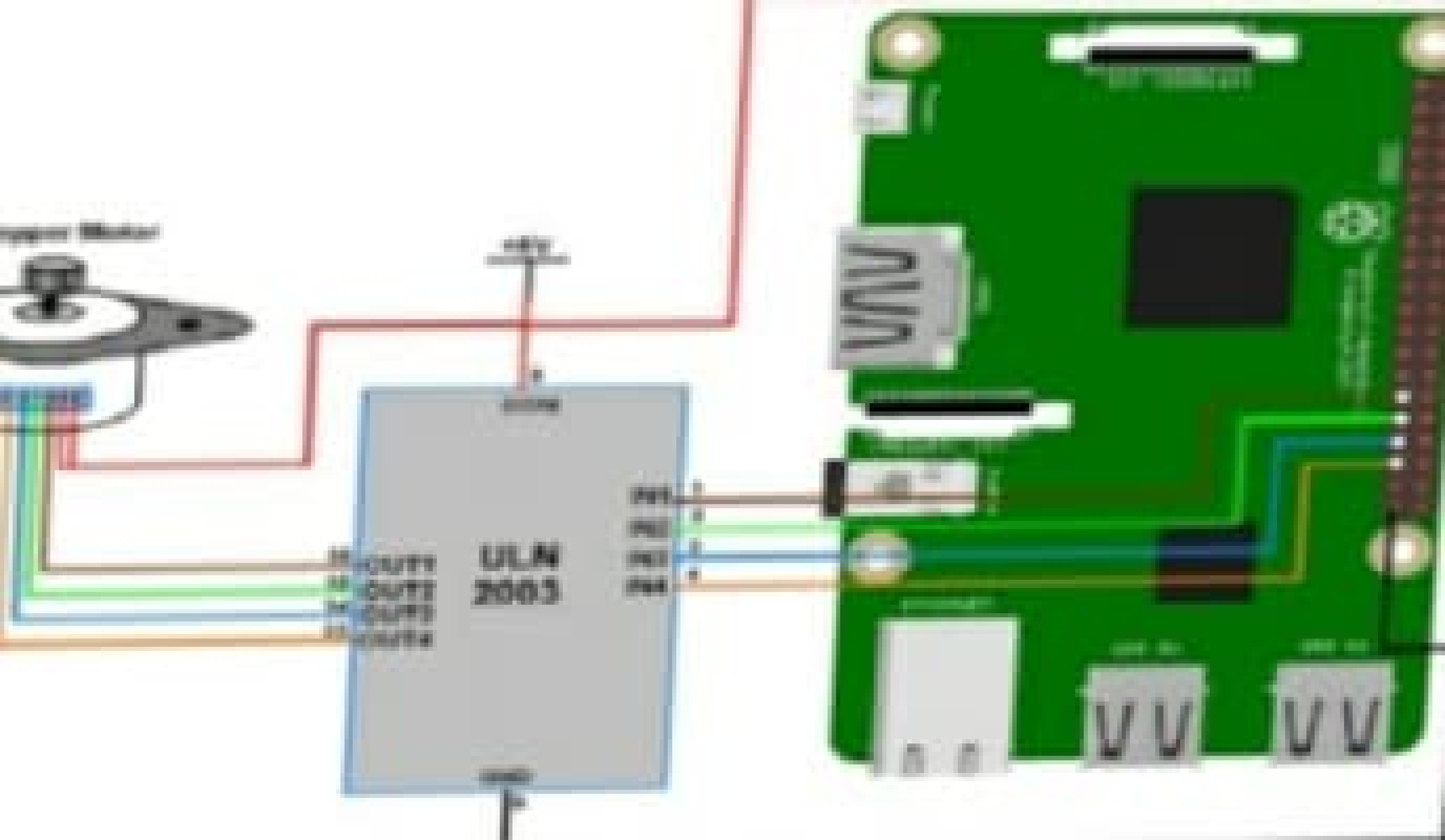
unipolar Stepper Motor fig. 1.

To rotate this four stage stepper Motor, we will deliver power pulses by using Stepper motor Driver Circuit. The deliver Circuit takes logic triggers from PI. If we Control the logic triggers, we Control the power pulses & hence the Speed of Stepper Motor.

There are 40 GIPO output pins in Raspberry Pi 2. But out of 40 only 26 GPIO pins can be programmed. Some of these pins perform some Special function. With Special GPIO put aside, we have only 17 GPIO remaining. Each of these 17 GPIO pin can deliver a Maximum of 15mA Current, & the sum of Currents from all GPIO pins cannot 50

There are +5v & 3.3v Power o/p pins. on the Board four Connecting other Modules & 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. Search your Stepper Motor Model number to known voltage rating. Depending on the rating choose the secondary source appropriately.

fig


Stepper Motor
+5V
ULN
2003
OUT1
OUT2
OUT3
OUT4
IN1
IN2
IN3
IN4
GND

fig

Sample Program.

Python 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 in/op 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 anti-clockwise\n')
  GPIO.output(motor-channel (GPIO.HIGH,
   GPIO.low, GPIO.low, GPIO.HIGH))
  slep(0.02)
  GPIO. output (motor-chaneel (GPIO.low,
  GPIO.low, GPIO. HIGH, GPIO. HIGH))
  sleep (0.02)
  GPIO. output (motor-chaneel (GPIO.low
     GPIO.HIGH GPIO.HIGH, GPIO.low))
  sleep(0.02)
  GPIO.output (motor-channel (GPIO.HIGH,
   GPIO.HIGH, ~~HIGH~~.GPIO.low, GPIO
                    low))
  Sleep (0.02.

  expect keyboardInterrupt:
  Motor-direction= input('select motor
      direction a= anticlockwise,
   c= clockwise or q=exit:')
```

```
if (motor_direction =='q'):
Print ('Motor Stopped')
 sys. exit (0)
```

Conclusion: Thus we have implemented application of Stepper Motors using Python with Raspberry Pi