Name :- Sakshi Anil Fokane
class :- TE-A
Roll No :- 42

Assignment No : 7

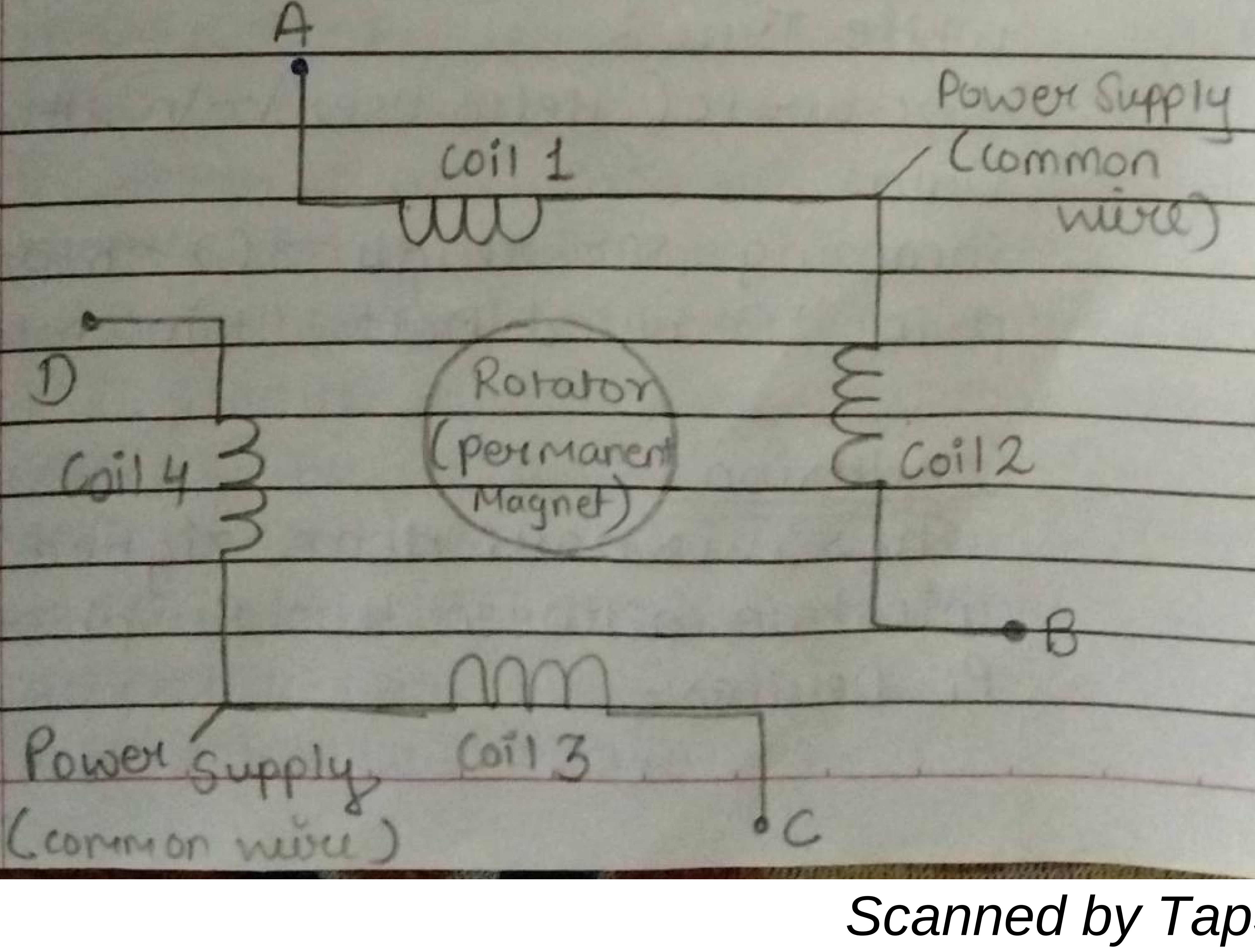
Aim :-

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

Theory :-

Stepper 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 Motar unlike DC Motor, we can rotate Stepper motor to any particular angle by giving it proper instructions.


A
coil 1
Power Supply (common wire)
D
Coil 4
Rotator (permanent Magnet)
Coil 2
B
Coil 3
Power supply (common wire)
C

To rotate this Four Stage Stepper Motor we will deliver power pulses by Using Stepper Motor Driver Circuit. The driver circuit takes logic triggers from PI. If will control the logic triggers, we control the power pulses & 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) & +3.3V (Pin 1 & 17) Power output pins on the board for Connecting other modules & sensors. These power rails Cannot be used to drive the Stepper Motor, bcz 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 am using a 9v battery as my second power Source. Search your stepper motor Model number to know voltage rating.

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 input/output 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_diagram == 'a'):
print('motor running anti-clockwise\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)

#press Ctrl+C for keyboard interrupt
except KeyboardInterrupt:
#query for setting motor direction or exit
motor-direct^n = i/p('select Motor dir^n a=anticlock-
kwise, c=clockwise or q=exit:')
# check for exit
if(Motor-direction == 'q'):
Print('motor stopped')
sys·exit(0)
```

Conclusion :-

Thus, we have implemented application of stepper motors using python with Raspberrey Pi.