TEA 17

Assignment NO: 7

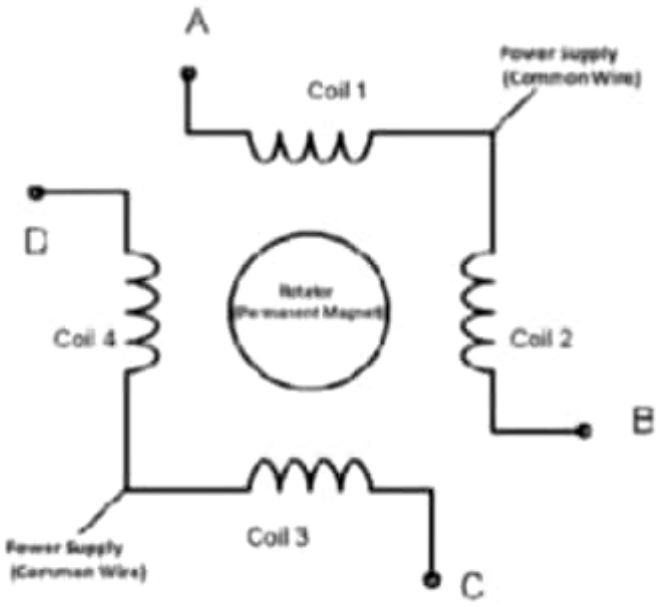
Nikita Bhosale

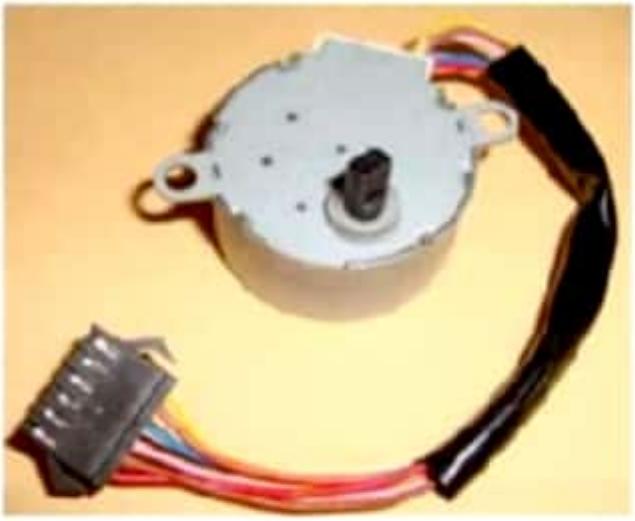
Title: Write an application using Raspberry-Pi / Beagle board to control the operation of stepper motor

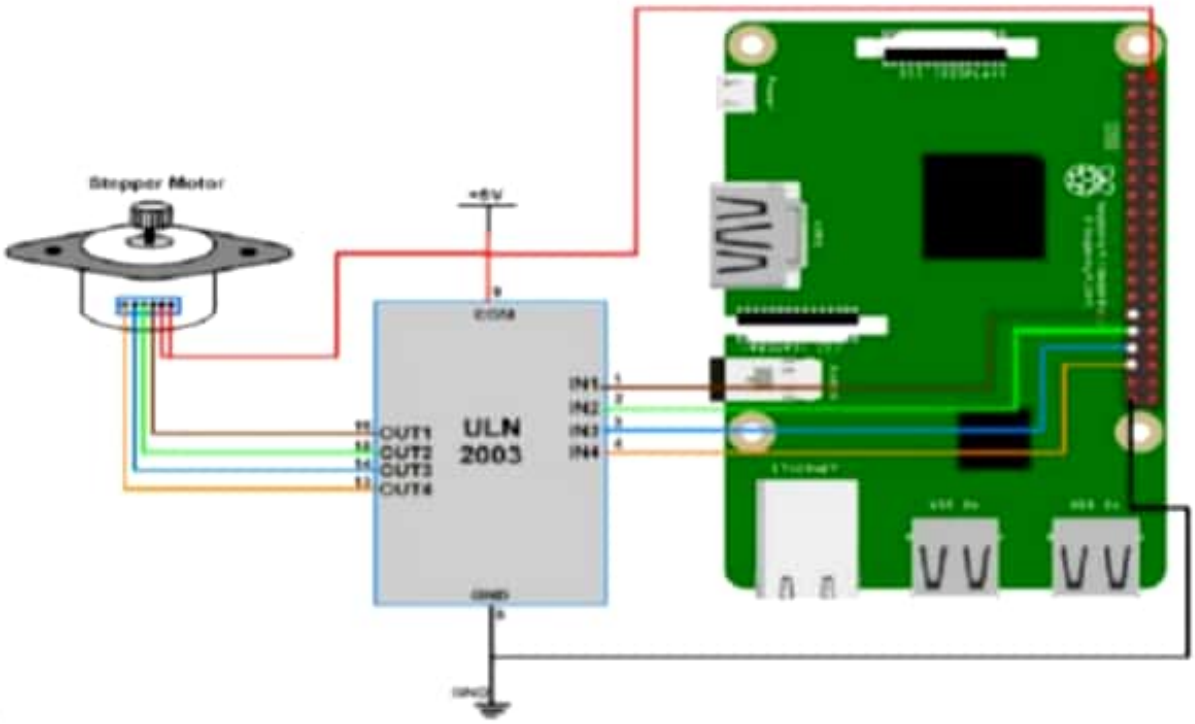
Theory:

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 motor.
- To rotate this 4 stage stepper motor we will deliver power pulses by using stepper motor ~~De~~ Driver circuit
- There are 40 GPIO o/p pins in Raspberry Pi 2 But out of 40, only 26 GPIO pins (GP102 to GP1027) can be programmed some of these pins perform some special functions, then we have only 17 GPIO remaining. each of these pins can deliver a maximum of 15mA current
- There are +5V (Pin 2&4) and +3.3V (Pin 1&17) power o/p pins on the board for connecting other modules and sensors - these power rails can't be







GURUKUL Page No.
Date / /

used to drive the stepper motor because we need more power to rotate it

Python Program for 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_direction == '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_direction = input('select motor
            direction a=anticlockwise, c
            =clockwise or q=exit: ')
        #check for exit
        if(motor_direction == 'q'):
            print('motor stopped')
            sys.exit(0)
```