Stepper Motor

Overview

In this project, we will learn how to drive stepping motor, and understand its working principle.

Experimental Materials:

Raspberry Pi *1

T-type expansion board *1

Breadboard*1

ULN2003 stepper motor driver module *1

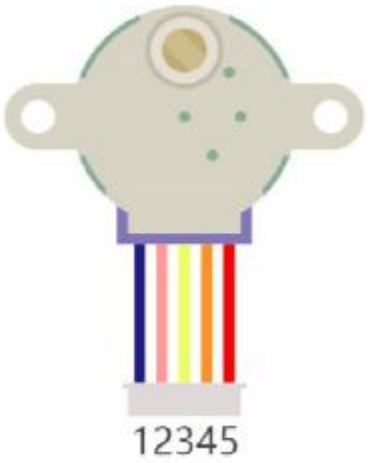
Stepper motor *1

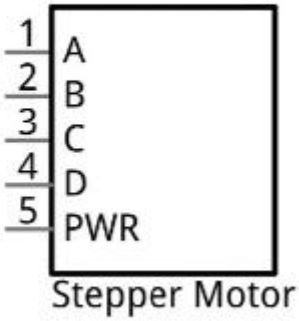
Some DuPont lines

Product description：

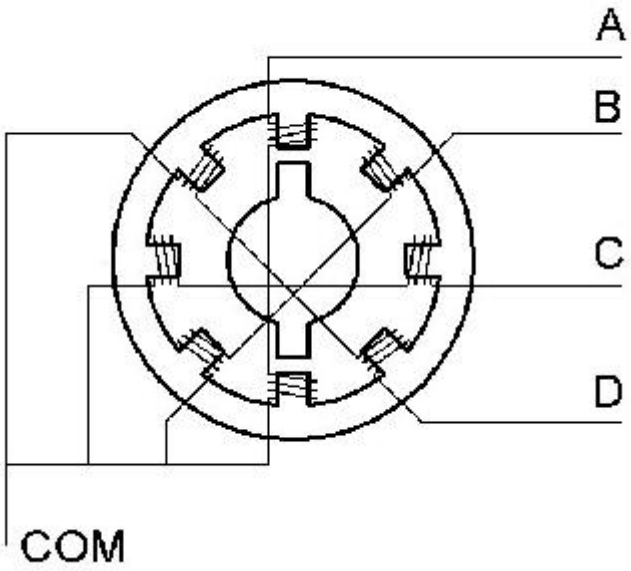
Stepping motor is an open-loop control device which converts the electric pulse signal into angular displacement or linear displacement. In

non-overload condition, the speed of the motor and the location of the stop depends only on the pulse signal frequency and pulse number, and not affected by the load changes. A small four-phase deceleration stepping motor is shown as follows:



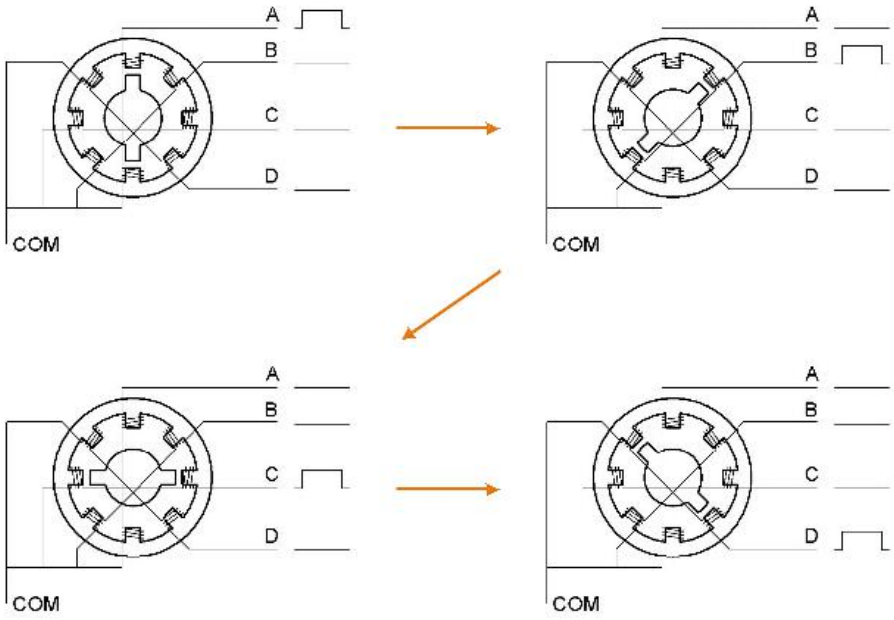


The schematic diagram of four-phase stepping motor is shown below:



The outside piece is the stator and the inside is the rotor of the motor. There are a certain number of coils, usually integer multiple of phases

number, in the stator and when powered on, an electromagnet will be formed to attract a convex part (usually iron or permanent magnet) of the rotor. Therefore, the electric motor can be driven by conducting the coils on stator orderly.



In the course above, the stepping motor rotates a certain angle once, which is called a step. By controlling the number of rotation steps, you can control the stepping motor rotation angle. By controlling the time between two steps, you can control the stepping motor rotation speed.

When rotating clockwise, the order of coil powered on is:

A B C D A…… . And the rotor will rotate in accordance with the order, step by step down, called four steps four pats. If the coils is powered on in the reverse order, D C B A D… , the rotor will rotate in anti-clockwise direction.

Stepping motor has other control methods, such as connect A phase, then connect A B phase, the stator will be located in the middle of the A B, only a half-step. This way can improve the stability of stepping motor, and reduce noise, the sequence of coil powered on is:

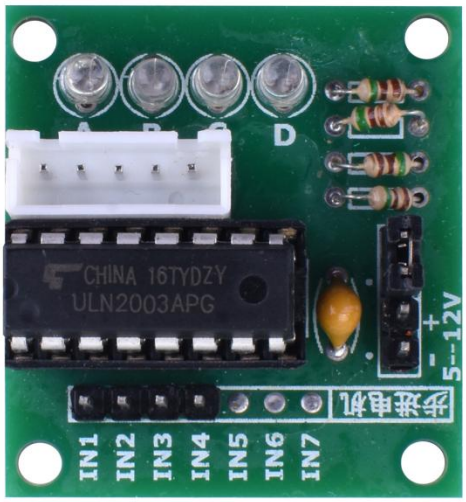
A AB B BC C CD D DA A…… , the rotor will rotate in accordance with the order, a half step by a half step, called four step eight pat. Equally, if the coil is powered on in reverse order, the stepping motor will rotate in reverse rotation.

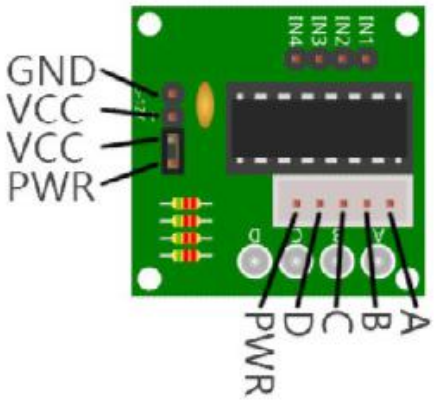
The stator of stepping motor we use has 32 magnetic poles, so a circle needs 32 steps. The output shaft of the stepping motor is connected with a reduction gear set, and the reduction ratio is 1/64. So the final output shaft rotates a circle requiring a 32*64=2048 step.

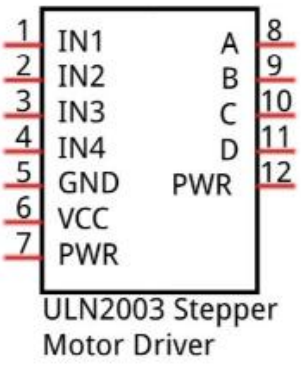
Stepper motor 28BYJ-48 Parameters :

Rated voltage ：	5VDC
Number of Phase	4
Speed Variation Ratio	1/64
Stride Angle	5.625°/64
Frequency	100Hz
DC resistance	50Ω±7%(25℃)
Idle In-traction Frequency	> 600Hz
Idle Out-traction Frequency	> 1000Hz
In-traction Torque	>34.3mN.m(120Hz)
Self-positioning Torque	>34.3mN.m
Friction torque	600-1200 gf.cm
Pull in torque	300 gf.cm
Insulated resistance	>10MΩ(500V)
Insulated electricity power	600VAC/1mA/1s
Insulation grade	A
Rise in Temperature	<40K(120Hz)
Noise	<35dB(120Hz,No load,10cm)
Model	28BYJ-48 – 5V

ULN2003 Driver Board:

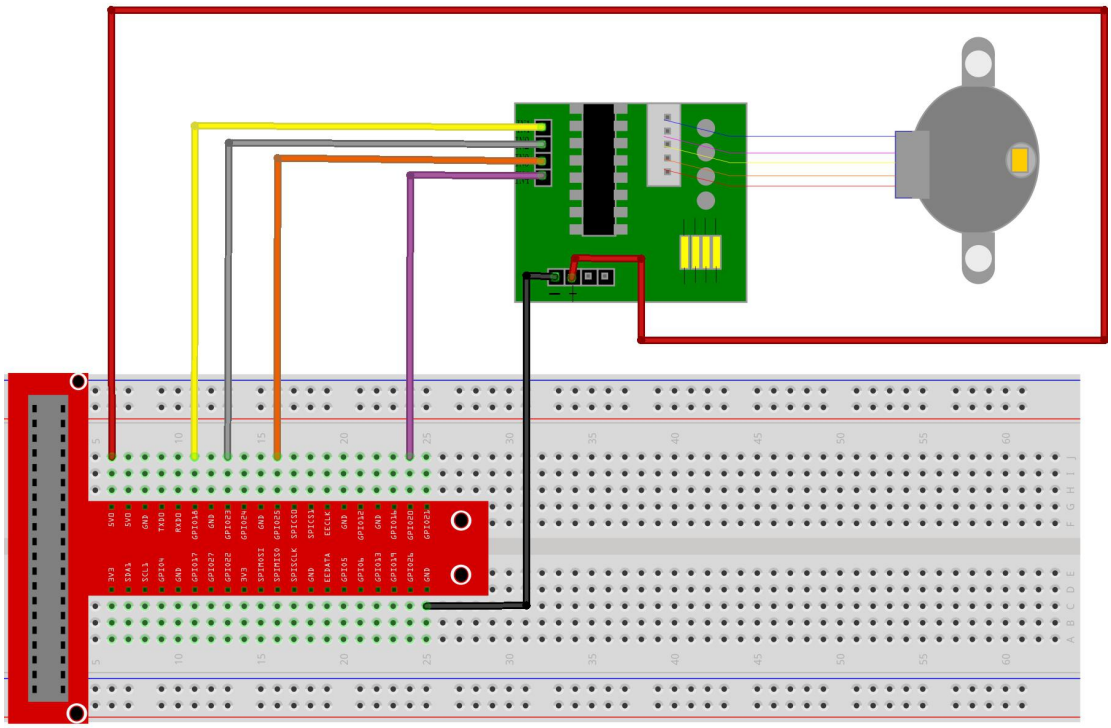






ULN2003 stepping motor driver is used to convert the weak signal into powerful control signal to drive the stepping motor. The input signal IN1-IN4 corresponds to the output signal A-D, and4 LED is integrated in the board to indicate the state of signals. The PWR interface can be used as a power supply for stepping motor. By default, PWR and VCC are connected by a short circuit.

Wiring diagram：



C code:

```
#include <stdio.h>
#include <wiringPi.h>

const int motorPins[]={1,4,6,28};     //define pins connected to four phase ABCD of stepper motor
const int CCWStep[]={0x01,0x02,0x04,0x08};  //define power supply order for coil for rotating anticlockwise
const int CWStep[]={0x08,0x04,0x02,0x01};   //define power supply order for coil for rotating clockwise
//as for four phase stepping motor, four steps is a cycle. the function is used to drive the stepping motor clockwise or anticlockwise to take four steps
void moveOnePeriod(int dir,int ms){
    int i=0,j=0;
    for (j=0;j<4;j++){  //cycle according to power supply order
        for (i=0;i<4;i++){  //assign to each pin, a total of 4 pins
            if(dir == 1)    //power supply order clockwise
                digitalWrite(motorPins[i],(CCWStep[j] == (1<<i)) ? HIGH : LOW);
            else        //power supply order anticlockwise
                digitalWrite(motorPins[i],(CWStep[j] == (1<<i)) ? HIGH : LOW);
            printf("motorPin %d, %d \n",motorPins[i],digitalRead(motorPins[i]));
        }
```

```
            printf("Step cycle!\n");
            if(ms<3)        //the delay can not be less than 3ms, otherwise it will
exceed speed limit of the motor
                ms=3;
            delay(ms);
        }
    }
    //continuous rotation function, the parameter steps specifies the rotation
cycles, every four steps is a cycle
    void moveSteps(int dir, int ms, int steps){
        int i;
        for(i=0;i<steps;i++){
            moveOnePeriod(dir,ms);
        }
    }
    void motorStop(){   //function used to stop rotating
        int i;
        for(i=0;i<4;i++){
            digitalWrite(motorPins[i],LOW);
        }
    }
    int main(void){
        int i;

        if(wiringPiSetup() == -1){ //when initialize wiring failed,print messageto
screen
            printf("setup wiringPi failed !");
            return 1;
        }
        for(i=0;i<4;i++){
            pinMode(motorPins[i],OUTPUT);
        }

        while(1){
            moveSteps(1,3,512);     //rotating   360°  clockwise, a total of 2048
steps in a circle, namely, 512 cycles.
            delay(500);
            moveSteps(0,3,512);     //rotating 360°  anticlockwise
            delay(500);
        }
        return 0;
    }
```

Python code：

```
#!/usr/bin/env python3
import RPi.GPIO as GPIO
import time

motorPins = (12, 16, 22, 38)    #define pins connected to four phase ABCD of
stepper motor
CCWStep = (0x01,0x02,0x04,0x08) #define power supply order for coil for rotating
anticlockwise
CWStep = (0x08,0x04,0x02,0x01)  #define power supply order for coil for rotating
clockwise

def setup():
    print ('Program is starting...')
    GPIO.setmode(GPIO.BOARD)       # Numbers GPIOs by physical location
    for pin in motorPins:
        GPIO.setup(pin,GPIO.OUT)
#as for four phase stepping motor, four steps is a cycle. the function is used
to drive the stepping motor clockwise or anticlockwise to take four steps
def moveOnePeriod(direction,ms):
    for j in range(0,4,1):      #cycle for power supply order
        for i in range(0,4,1):  #assign to each pin, a total of 4 pins
            if (direction == 1):#power supply order clockwise
                GPIO.output(motorPins[i],((CCWStep[j] == 1<<i) and GPIO.HIGH
or GPIO.LOW))
            else :              #power supply order anticlockwise
                GPIO.output(motorPins[i],((CWStep[j] == 1<<i) and GPIO.HIGH or
GPIO.LOW))
        if(ms<3):       #the delay can not be less than 3ms, otherwise it will
exceed speed limit of the motor
            ms = 3
        time.sleep(ms*0.001)
#continuous rotation function, the parameter steps specifies the rotation cycles,
every four steps is a cycle
def moveSteps(direction, ms, steps):
    for i in range(steps):
        moveOnePeriod(direction, ms)
#function used to stop rotating
def motorStop():
    for i in range(0,4,1):
        GPIO.output(motorPins[i],GPIO.LOW)

def loop():
```

```
    while True:
        moveSteps(1,3,512)  #rotating   360 deg clockwise, a total of 2048
steps in a circle, namely, 512 cycles.
        time.sleep(0.5)
        moveSteps(0,3,512)  #rotating 360 deg anticlockwise
        time.sleep(0.5)

def destroy():
    GPIO.cleanup()             # Release resource

if __name__ == '__main__':     # Program start from here
    setup()
    try:
        loop()
    except KeyboardInterrupt:  # When 'Ctrl+C' is pressed, the child program
destroy() will be  executed.
        destroy()
```

Experimental results:

In the directory where the code file is located, execute the following command

C:

```
gcc -Wall -o SteppingMotor SteppingMotor.c -lwiringPi
sudo ./SteppingMotor
```

Python:

```
python SteppingMotor.py
```

After executing the command, the stepper motor rotates one turn and reverses one turn, and it keeps looping.