Servo

Overview

This course will use the Raspberry Pi to control the servo to rotate 180 degrees clockwise and counterclockwise

Experimental Materials:

Raspberry Pi *1

T-type expansion board *1

Breadboard *1

Steering machine *1

Some DuPont lines

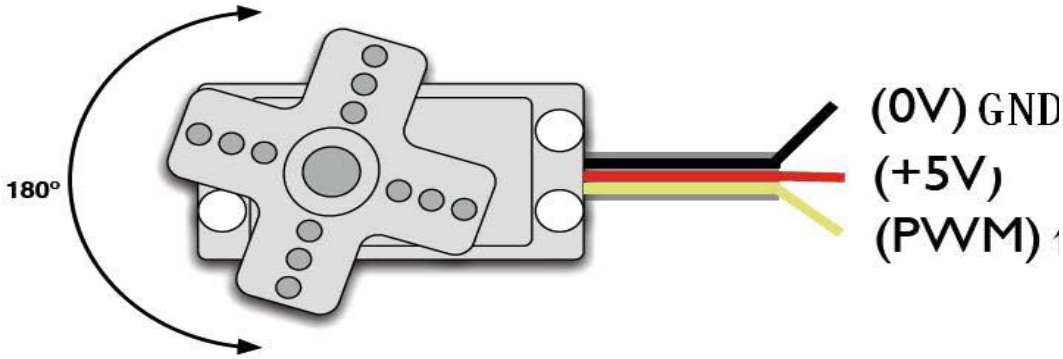
Product description：

● The servo is a position servo driver. It is mainly composed of a housing, a circuit board, a coreless motor, a gear, and a position detector.

● The working principle is that the signal is sent to the servo by the receiver or the single-chip microcomputer. There is a reference circuit inside, which generates a reference signal with a period of 20ms and a width of 1.5ms, and compares the obtained DC bias voltage with the voltage of the potentiometer. Differential voltage output. The direction of rotation is judged by the IC on the circuit board, then the coreless motor is driven to start rotation, the power is transmitted to the swing arm through the reduction gear, and the signal is sent back by the position detector to judge whether the positioning has been reached. It is suitable for control systems that require changing angles and can be maintained. When the motor rotates at a certain speed, the potentiometer is rotated through the cascade reduction gear so that the voltage difference is 0 and the motor stops rotating. The general servo rotation angle range is 0 degrees to 180 degrees.

● He applies to those control systems that require constant change of angle and can be maintained. Currently used in high-end remote control toys, such as aircraft models, including aircraft models, submarine models; remote control robots have been used more commonly. The

servo is a common name and it is actually a servo motor.



● The angle of rotation of the servo is achieved by adjusting the duty cycle of the PWM (pulse width modulation) signal. The period of the standard PWM (pulse width modulation) signal is fixed at 20 ms (50 Hz), and the theoretical pulse width distribution should be at 1 ms. Between 2ms, but in fact, the pulse width can be between 0.5ms and 2.5ms, and the pulse width corresponds to the steering angle of the servo from 0° to 180°. It is worth noting that due to the different servo brands, the angle of rotation of the servos of different brands will be different for the same signal.

Product parameters:

1. Weight: 9g

2. Size: 23x12.2x29mm

3. No-load operation speed: 0.12 seconds/60 degrees (4.8V); 0.10 seconds/60 degrees (6.0V)

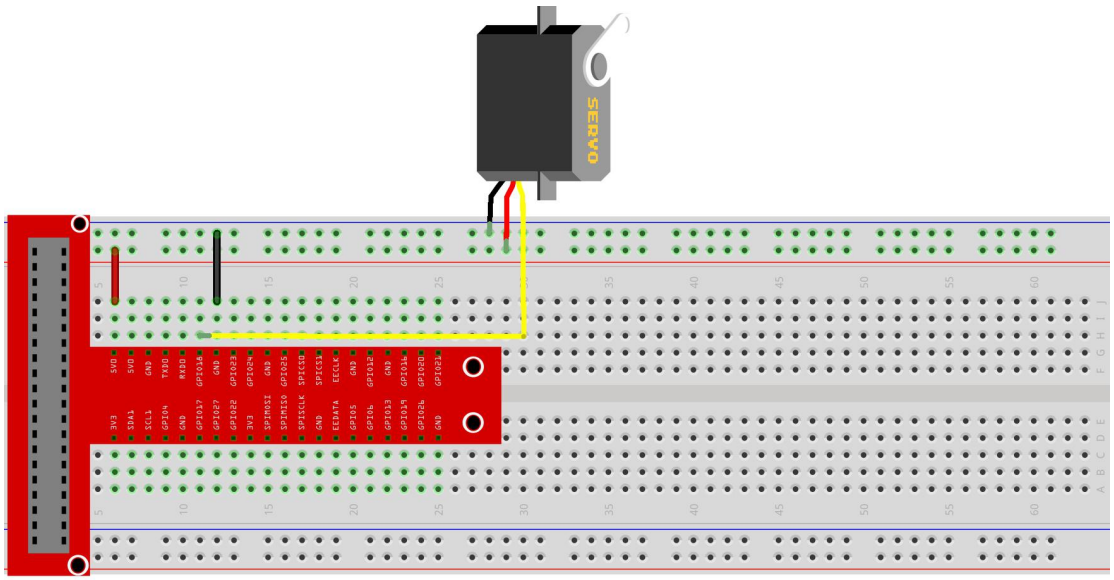
4. Torque: 1.6kg·cm (4.8V)

5 .use temperature: -30 ~ +60 degrees Celsius

6 .Dead zone setting: 5 microseconds

7. Operating voltage: 3.5V~6V

Wiring diagram：



C code：

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

void init();
int main()
{
        init();
        int angle=180;


        int i=0;
        int x=0;
```

```
        int k=180;

while(1)
{
while(k--)
        {
                x=11.11*i;
                digitalWrite(1,HIGH);
                delayMicroseconds(500+x);
                digitalWrite(1,LOW);
                delayMicroseconds(19500-x);
                if(i==angle)
                    break;

                i++;
         }
delay(10);

i=0;
while(k--)
        {
                x=11.11*i;
                digitalWrite(1,HIGH);
                delayMicroseconds(2500-x);
                digitalWrite(1,LOW);
                delayMicroseconds(17500+x);
                if(i==angle)
                    break;

                i++;
         }
delay(10);
i=0;

}

        return 0;
}


void init()
{
        wiringPiSetup();
        pinMode(1,OUTPUT);
```

```
}
```

Python code：

```
#!/usr/bin/env python


import RPi.GPIO as GPIO
import time
import signal
import atexit

atexit.register(GPIO.cleanup)

servopin = 18
GPIO.setmode(GPIO.BCM)
GPIO.setup(servopin, GPIO.OUT, initial=False)
p = GPIO.PWM(servopin,50) #50HZ
p.start(2.5)
time.sleep(1)

while(True):
  for i in range(0,181,10):
    p.ChangeDutyCycle(2.5 + 10 * i / 180)
    time.sleep(0.02)
    p.ChangeDutyCycle(0)
    time.sleep(0.2)

  for i in range(181,0,-10):
    p.ChangeDutyCycle(2.5 + 10 * i / 180)
    time.sleep(0.02)
    p.ChangeDutyCycle(0)
    time.sleep(0.2)
```

Experimental results:

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

C：
gcc -Wall -o servo servo.c -lwiringPi
sudo ./servo

Python:
python servo.py

After the instruction is executed, the rudder will rotate 180 degrees back and forth