keyestudio

Project 3: Servo

1. Description

For those DIY smart cars, they often have the function of automatic obstacle avoidance. In the DIY process, we need a servo to control the ultrasonic module to rotate left and right, and then detect the distance between the car and the obstacle, so as to control the car to avoid the obstacle.

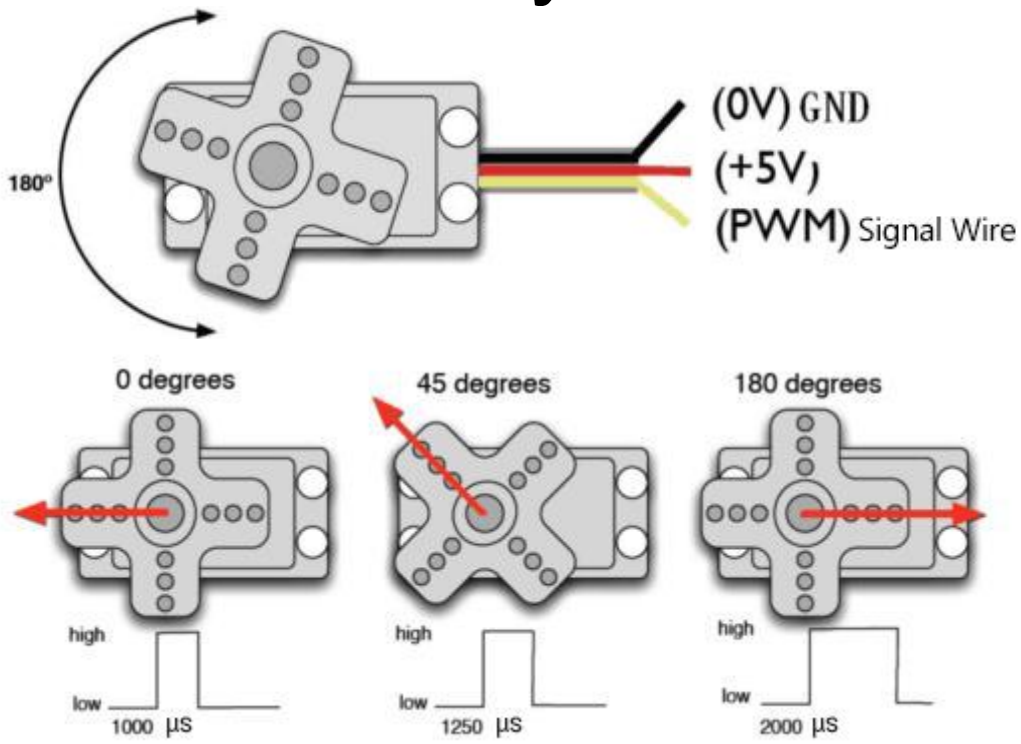
If other microcontrollers are used to control the rotation of the servo, we need to set a certain frequency and a certain width of pulse to control the servo angle. But if arduino is used to control the servo angle, we only need to set the control angle in the development environment configuration where the corresponding pulse will be automatically set to control the servo rotation. In this project, you will learn how to control the servo to rotate back and forth between 0° and 180°.

2. Component Knowledge

Angle range: 0°~180° (There are 0°~360°servo, 0°~180°servo, 0°~90° servo)

Drive voltage: 3.3V or 5V

The pins are usually three wires



GND: This is a grounded pin, which is brown.

VCC: This is a pin connected to +5v (3.3V) power, which is red.

S: This is a pin controlled by PWM signal, which is orange.(Here we connect it to **D9)**

Control Principle:

The rotation angle of servo is controlled by regulating the duty cycle of PWM (Pulse-Width Modulation) signal. The standard cycle of PWM signal is 20ms (50Hz). Theoretically, the width is distributed between 1ms-2ms, but in fact, it's between 0.5ms-2.5ms. The width corresponds to the rotation angle from 0° to 180°. But note that for different brand servo, the same signal may have different rotation angle.

The corresponding servo angle is shown in the figure below:

High level time	Servo angle
0.5ms	0 degree
1ms	45 degree
1.5ms	90 degree
2ms	135 degree
2.5ms	180 degree

3. Test Code

```
//*****************************************************************************
/*
  Keyestudio 4WD Mecanum Robot for Arduino
  lesson 3.1
  Servo
  http://www.keyestudio.com
*/
#define  servo_pin  9      //The servo is connected to D9

void setup() {
  pinMode(servo_pin, OUTPUT);    //Set the servo pin mode to output
}
void loop() {
  for (uint8_t angle = 0; angle < 180; angle++)
  {
    servopulse(servo_pin, angle);
  }
  for (uint8_t angle = 180; angle > 0; angle--)
  {
    servopulse(servo_pin, angle);
  }
}

void servopulse(int pin, int myangle) {   //Impulse function
  int pulsewidth = map(myangle, 0, 180, 500, 2500); //Map angle to pulse width
  //Output pulse
  digitalWrite(pin, HIGH);         //Set the servo interface to high level
  delayMicroseconds(pulsewidth);   //Delay the number of ms of pulse width value
  digitalWrite(pin, LOW);          //Set the servo interface to low level
  delay(20 - pulsewidth / 1000);   //Cycle is 20 ms
}
```

```
//*****************************************************************************
```

4.Test Result

After compiling and uploading the code, then turn the DIP switch to the ON end and power on, we will see that the servo will turn back and forth from 0 degrees to 180 degrees.

5. Code Explanation

#define servo_pin 9	Define the pin number of the servo to D9
pinMode(servo_pin, OUTPUT);	Set the pin connecting the servo to output mode, after setting, high/low level can be output.
servopulse(servo_pin, angle);	Impulse function that causes the servo connected to the servo_pin to be rotated to the angle position.
map(myangle, 0, 180, 500, 2500);	A mapping function that maps myangle from 0 to 180 to 500 to 2500, such as 1500 when myangle is 90.
digitalWrite(pin, HIGH); **digitalWrite(pin, LOW);**	The first parameter pin is the output pin. When the second parameter is HIGH, then output high level (3.3V). When it is LOW, output low level (0V)
delayMicroseconds(pulsewidth);	Delay **pulsewidth in ms**

6. Expanded Project: Use servo library to drive

```
//*****************************************************************************
/*
  Keyestudio 4WD Mecanum Robot for Arduino
  lesson 3.2
  Servo
  http://www.keyestudio.com
*/
#include <Servo.h>
Servo myservo;    //Define a servo instance

void setup() {
  myservo.attach(9);    //The servo pin is connected to D9
}

void loop() {
  for (uint8_t angle = 0; angle < 180; angle++)//From 0 to 180 degrees
  {
    myservo.write(angle); //Rotate to angle
    delay(15);  //Wait for a while
  }
  for (uint8_t angle = 180; angle > 0; angle--)//From 180 to 0 degrees
  {
    myservo.write(angle); //Rotate to angle
    delay(15);
  }
}
//*****************************************************************************
```

First, make sure the library files are installed, otherwise the code will fail to compile. Here we use the library file "MecanumCar_v2", please refer to the development environment configuration for installation.After burning the code, the servo rotates back and forth between 0 and 180 degrees.