1.原理介绍

首先, 让我们了解一下 SG90 伺服:

SG90 舵机

180角舵机 旋转角度 从0到180

棕色线 —— GND

红色线 —— 5V

橙色线 —— 信号(PWM)



分类: 180 伺服

通常舵机有 3 根控制线: 电源线、地线和信号线。

舵机管脚定义: 棕线——GND, 红线——5V, 橙线——信号。

伺服是如何工作的:

舵机中的信号调制芯片接收来自控制板的信号,然后舵机将获得基本的直流电压。伺服内部还有一个参考电路,可以产生标准电压。这两个电压将相互比较,并输出差值。然后电机芯片将接收差值并决定转速、方向和角度。当两个电压没有差异时,伺服将停止。

如何控制伺服:

控制舵机转动,需要使时间脉冲为 20ms 左右,高电平脉冲宽度为 0.5ms~2.5ms 左右,这与舵机的角度限制是一致的。

以 180 角伺服为例,对应的控制关系如下:

0.5ms	0 度
1.0ms	45 度
1.5ms	90 度
2.0ms	135 度
2.5ms	180 度

2.使用案例

双击打开"资料 → 课程 → 课程 1 驱动单个舵机 → Servo-Test"下的 Servo-Test.ino 文件。用 USB 线连接 NodeMcu 开发板和电脑,选择正确的开发板、处理器和端口。将程序下载进单片机、如下图所示。

```
Servo-Test | Arduino 1.8.19
                                                                                 ×
文件 编辑 项目 工具 帮助
 Servo-Test
 13 |#include
14
15 Servo myservo; // create servo object to control a servo
16 // twelve servo objects can be created on most boards
18
19⊟ void setup() {
20
     myservo.attach(9); // attaches the servo on GIO2 to the servo object
22
23∃ void loop() {
24
25
26⊟ for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees
       // in steps of 1 degree
27
       myservo.write(pos);
                                       // tell servo to go to position in variable 'pos'
28
29
       delay(15);
                                       // waits 15ms for the servo to reach the position
30
31 \Box  for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees
                                       // tell servo to go to position in variable 'pos'
32
       myservo.write(pos);
33
        delay(15);
                                       // waits 15ms for the servo to reach the position
35
    }
```

程序下载后, 拔出 USB 线, 连接 SG90 舵机与拓展板, 连接关系如下所示。

给拓展板插上电池, 若舵机是正常的, 应该会从 0 度旋转多 180 度, 然后从 180 度旋转到 0 度, 不断循环。拆开套件中提供的 8 个舵机, 分别按照同样的操作测试一遍, 以保证每个舵机都是正常完好的。