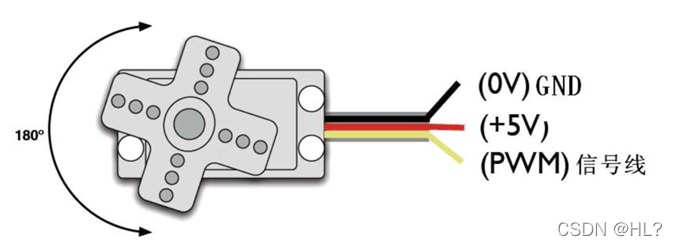
舵机简介及通过Arduino控制舵机180度/360度

## 舵机简介

常见的舵机有很多种分类，我以一种不成文的分类方式有，180°舵机，360°舵机，总线舵机。这些舵机在真正使用的时候也是存在区别的。180度舵机能控制角度，**360度舵机只能控制速度和方向**，总线舵机为可以进行逐级串联的（当前舵机输出端为下一舵机输入端），所以串级连接中，当前舵机的供电来源于串级系统中上一个的舵机的输出端。有的数字舵机只带了通信功能在价格上也比较昂贵，达到单个数千元。



## （1）180度舵机控制方式

180度是在普通场合时最常用的，一半包含三根线（VCC --> 电源正,GND --> 电源负，SINGAL --> PWM输出引脚），控制原理如下。简单地说，舵机的控制需要通过周期为20ms的PWM波来实现。对于180度舵机的20ms周期的PWM波，在0.5ms时对应左极限位置，2.5ms时对应右极限位置。1.5ms对应中间位置。

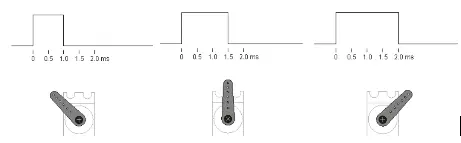


图 1不同角度PWM波下的舵机角度

180度角度伺服对应的控制关系是：

0.5ms--------------0度；

1.0ms------------45度；

1.5ms------------90度；

2.0ms-----------135度；

2.5ms-----------180度；

## （2）360度舵机控制方式

360度舵机控制方式与180度舵机不同在于，**360舵机只能控制方向和速度，**如上图，把角度理解成方向，如20msPWM方波中0.5ms时刻为左转（反转）最大速度，1.5ms为停止转动，2.5ms时刻为右转（正转）最大速度。

## （3）总线舵机

总线舵机通常支持通信控制，如485，232等。某些昂贵的舵机自带位置反馈，但并不回给控制带来难度。所以，简单说这种舵机的内部实现逻辑我们并不需要知道，直接通过通信来完成就行了。

附录

**附录A：180度舵机**

#include <Arduino.h>

int freq = 50; // 1/50秒，50Hz的频率，20ms的周期，这个变量用来存储时钟基准。

int channel = 8; // 通道(高速通道（0 ~ 7）由80MHz时钟驱动，低速通道（8 ~ 15）由 1MHz 时钟驱动。)

int resolution = 8; // 分辨率设置为8，就是2的8次方，用256的数值来映射角度，看下方的占空比函数。

const int led = 13;

int calculatePWM(int degree) //定义函数用于输出PWM的占空比

{ //0-180度

//20ms周期内，高电平持续时长0.5-2.5ms，对应0-180度舵机角度，参考上面的180度对应图。

const float deadZone = 6.4;//对应0.5ms（0.5ms/(20ms/256）)

const float max = 32;//对应2.5ms（2.5ms/（20ms/256））

if (degree < 0)

degree = 0;

if (degree > 180)

degree = 180;

return (int)(((max - deadZone) / 180) \* degree + deadZone); //返回度数对应的高电平的数值

}

void setup()

{

Serial.begin(9600);

ledcSetup(channel, freq, resolution); // 用于设置 LEDC 通道的频率和分辨率。

ledcAttachPin(led, channel); // 将通道与对应的引脚连接

}

void loop()

{

for (int d = 0; d <= 180; d += 10)

{

ledcWrite(channel, calculatePWM(d)); // 输出PWM，设置 LEDC 通道的占空比。

Serial.printf("value=%d,calcu=%d\n", d, calculatePWM(d));

delay(300);

}

}

**附录B：360度舵机(也可用于180度舵机)**

#include <Arduino.h>

int channel\_PWM = 3; //使用3号通道 定时器1 总共16个通道

// 舵机频率，那么周期也就是1/50，也就是20ms ，PWM⼀共有16个通道，0-7位⾼速通道由80Mhz时钟驱动，后⾯8个为低速通道由1Mhz

int freq\_PWM = 50; //50HZ pwm波

// PWM分辨率，取值为 0-20 之间 ，这⾥填写为10，那么后⾯的ledcWrite 这个⾥⾯填写的pwm值就在 0 - 2的10次⽅ 之间 也就是 0-1024，如果是要求不⾼的东西你可以直接拿1000去算了

int res\_PWM = 10; //分辨率 0-1024 共1025

const int PWM\_PIN = 13; //使用13号引脚

void setup() {

ledcSetup(channel\_PWM,freq\_PWM,res\_PWM); //设置通道

ledcAttachPin(PWM\_PIN,channel\_PWM); //将引脚绑定到通道上

}

//正传t秒 停止t秒 反转t秒 停止t秒 循环...

void loop() {

ledcWrite(channel\_PWM,52); //20ms⾼电平为1ms左右 ，也就是1/20\*1024 = 52，此时360度舵机正传5秒

delay(1000);

ledcWrite(channel\_PWM, 77); //20ms⾼电平为1.5ms左右 ，也就是1.5/20\*1024 = 77，此时360度舵机⼏乎停转

delay(1000);

ledcWrite(channel\_PWM, 102); //20ms⾼电平为2ms左右 ，也就是2/20\*1024 = 102，此时360度舵机反转

delay(1000);

ledcWrite(channel\_PWM, 77); //20ms⾼电平为1.5ms左右 ，也就是1.5/20\*1024 = 77，此时360度舵机⼏乎停转

delay(1000);

//ledcDetachPin(PWM\_Pin); //这个是解除IO⼝的pwm输出功能模式

}

**附录C：360度舵机代码改180度舵机**

#include <Arduino.h>

int channel\_PWM = 8; //使用3号通道 定时器1 总共16个通道

// 舵机频率，那么周期也就是1/50，也就是20ms ，PWM⼀共有16个通道，0-7位⾼速通道由80Mhz时钟驱动，后⾯8个为低速通道由1Mhz

int freq\_PWM = 50; //50HZ pwm波

// PWM分辨率，取值为 0-20 之间 ，这⾥填写为10，那么后⾯的ledcWrite 这个⾥⾯填写的pwm值就在 0 - 2的10次⽅ 之间 也就是 0-1024，如果是要求不⾼的东西你可以直接拿1000去算了

int res\_PWM = 10; //分辨率 0-1024 共1025

const int PWM\_PIN = 13; //使用13号引脚

void setup() {

ledcSetup(channel\_PWM,freq\_PWM,res\_PWM); //设置通道

ledcAttachPin(PWM\_PIN,channel\_PWM); //将引脚绑定到通道上

}

//正传t秒 停止t秒 反转t秒 停止t秒 循环...

void loop() {

ledcWrite(channel\_PWM,26); //20ms⾼电平为0.5ms左右 ，也就是0.5/20\*1024 = 26，此时180度舵机角度为0°

delay(1000);

ledcWrite(channel\_PWM, 52); //20ms⾼电平为1ms左右 ，也就是1/20\*1024 = 52，此时180度舵机角度为45°

delay(1000);

ledcWrite(channel\_PWM, 77); //20ms⾼电平为1.5ms左右 ，也就是1.5/20\*1024 = 77，此时180度舵机角度为90°

delay(1000);

ledcWrite(channel\_PWM, 102); //20ms⾼电平为2ms左右 ，也就是2/20\*1024 = 102，此时180度舵机角度为135°

delay(1000);

ledcWrite(channel\_PWM, 128); //20ms⾼电平为2.5ms左右 ，也就是2.5/20\*1024 = 128，此时180度舵机角度为180°

delay(1000);

//ledcDetachPin(PWM\_Pin); //这个是解除IO⼝的pwm输出功能模式

}