# 舵机应用

舵机是用来控制舵的，比如轮船的方向舵，飞机的方向舵、升降舵等，这些都需要控制一定的角度，但并非需要连续旋转。所以一般舵机都是只能转动一定的角度，这里说的舵机主要应用于航模、车模和监控等多种领域。之前提及的舵机一般是正负90度之间转动（连续旋转舵机除外），舵机内部是有直流电机，位置电位器和驱动反馈电路板组成，当需要舵机转到一定角度时，输入信号会与标准信号比较，如果反馈位置不是所需要的位置，电机则会朝向需要的方向转动，直到转到指定位置，电位器反馈信息促使电机停止转动。

舵机是一种位置伺服的驱动器，主要是由外壳、电路板、无核心马达、齿轮与位置检测器所构成。其工作原理是由接收机或者单片机发出信号给舵机，其内部有一个基准电路，产生周期为20ms，宽度为1.5ms的基准信号，将获得的直流偏置电压与电位器的电压比较，获得电压差输出。经由电路板上的IC判断转动方向，再驱动无核心马达开始转动，透过减速齿轮将动力传至摆臂，同时由位置检测器送回信号，判断是否已经到达定位。适用于那些需要角度不断变化并可以保持的控制系统。当电机转速一定时，通过级联减速齿轮带动电位器旋转，使得电压差为0，电机停止转动。一般舵机旋转的角度范围是0度到180度。

舵机有很多规格，但所有的舵机都有外接三根线，分别用棕、红、橙三种颜色进行区分，由于舵机品牌不同，颜色也会有所差异，棕色为接地线，红色为电源正极线，橙色为信号线。

舵机的转动的角度是通过调节PWM（脉冲宽度调制）信号的占空比来实现的，标准PWM（脉冲宽度调制）信号的周期固定为20ms（50Hz），理论上脉宽分布应在1 ms到2 ms之间，但是，事实上脉宽可由0.5 ms到2.5 ms之间，脉宽和舵机的转角0°～180°相对应。有一点值得注意的地方，由于舵机牌子不同，对于同一信号，不同牌子的舵机旋转的角度也会有所不同。

图片包含 文字

描述已自动生成

## 原理图

图片包含 物体

描述已自动生成

## 实物连接图

图片包含 电子产品

描述已自动生成

## 程序

#include <Servo.h>

Servo myservo; // 最多可以控制 8 路舵机

int pos = 0; // 用于存储舵机位置的变量

//初始化

void setup()

{

myservo.attach(9); // 舵机控制信号引脚

}

//主循环

void loop()

{

for (pos = 0; pos < 180; pos += 1) // 从 0 度-180 度

{ // 步进角度 1 度

myservo.write(pos); // 输入对应的角度值，舵机会转到此位置

delay(15); // 15ms 后进入下一个位置

}

for (pos = 180; pos >= 1; pos -= 1) // 从 180 度-0 度

{

myservo.write(pos); // 输入对应的角度值，舵机会转到此位置

delay(15); // 15ms 后进入下一个位置

}

}

Arduino自带的Servo函数及其语句中舵机函数的几个常用语句：

1、attach(接口)——设定舵机的接口，只有数字9或10接口可利用。

2、write(角度)——用于设定舵机旋转角度的语句，可设定的角度范围是0°到180°。

3、read()——用于读取舵机角度的语句，可理解为读取最后一条 write()命令中的值。

4、attached()——判断舵机参数是否已发送到舵机所在接口。

5、detach()——使舵机与其接口分离，该接口（数字 9 或 10 接口）可继续被用作PWM接口。