

Arduino 舵机控制程序详解

本实验目标：使用主板或者传感器，控制舵机的运行

目标分析：为了完成这个实验，我们首先对于它进行架构的分析

传感器：按钮或是电位器（以电位器为例）

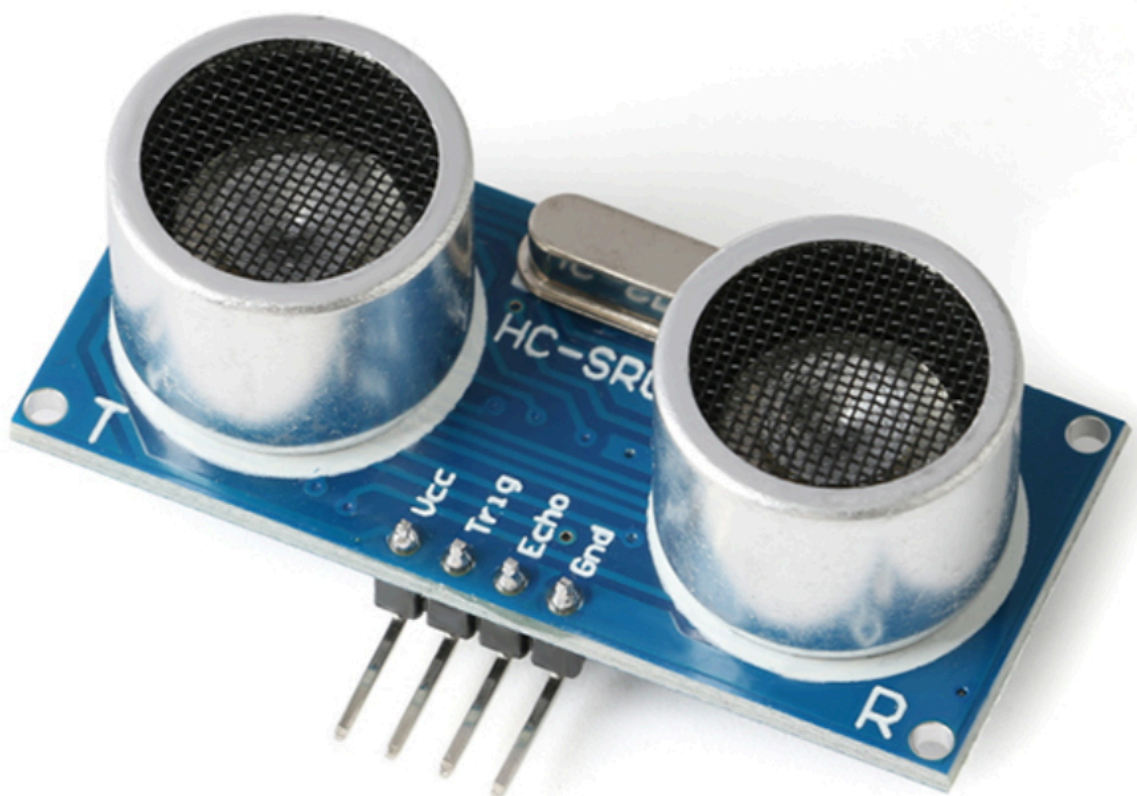
控制器：UNO/ESP32

执行器：180舵机

接着是效果分析：

使舵机以我们设计的方式进行转动。

明确目标系统与效果，开始正式学习---



程序架构说明

以下两个程序需分别烧录至Arduino开发板运行，每个程序实现不同控制方式：

程序一：舵机基础转动控制

功能描述

控制舵机周期性在0°、90°、180°之间转动，每个角度保持1秒。

完整代码

```
#include <Servo.h> // 引入舵机控制库

Servo servo_7;      // 创建舵机对象

void setup(){
    servo_7.attach(7); // 绑定舵机到数字引脚7
}

void loop(){
    servo_7.write(0); // 转动到0度位置
    delay(1000);      // 保持1秒
    servo_7.write(90); // 转动到90度位置
    delay(1000);
    servo_7.write(180); // 转动到180度位置
    delay(1000);
}
```

逐行解析

- `#include <Servo.h>`
 - 作用：导入舵机控制库文件
 - 说明：提供Servo类及相关控制方法，简化底层PWM信号生成
- `Servo servo_7;`
 - 创建Servo类实例
 - 命名规则：`servo_引脚号`，提高代码可读性
- `servo_7.attach(7);`
 - 参数：数字引脚编号（需支持PWM，UNO的3,5,6,9,10,11）
 - 功能：初始化舵机控制信号线连接
- `servo_7.write(角度)`
 - 参数范围：0~180（对应舵机转动角度）
 - 内部转换：将角度值映射为500~2500μs脉冲宽度
- `delay(1000)`
 - 维持当前角度状态1秒
 - 防止舵机因指令变化过快产生抖动

程序二：电位器控制舵机

功能描述

通过旋转电位器实时控制舵机角度，实现精确位置调节。

完整代码

```
#include <Servo.h>

Servo servo_7;

void setup(){
    servo_7.attach(7); // 初始化舵机连接
}

void loop(){
    int potValue = analogRead(A0); // 读取电位器值
    int angle = map(potValue, 0, 1023, 0, 180); // 数值映射
    servo_7.write(angle); // 更新舵机角度
    delay(10); // 微小延时稳定信号
}
```

关键代码解析

1. `analogRead(A0)`

- 功能：读取模拟输入电压
- 返回值：0 (0V) ~1023 (5V) 的整数
- 对应电位器位置：完全逆时针→完全顺时针

2. `map()` 函数

`map(value, fromLow, fromHigh, toLow, toHigh)`

- 数学公式：

$$\text{输出值} = (\text{value} - \text{fromLow}) * (\text{toHigh} - \text{toLow}) / (\text{fromHigh} - \text{fromLow}) + \text{toLow}$$

- 本程序应用：将0-1023映射为0-180度

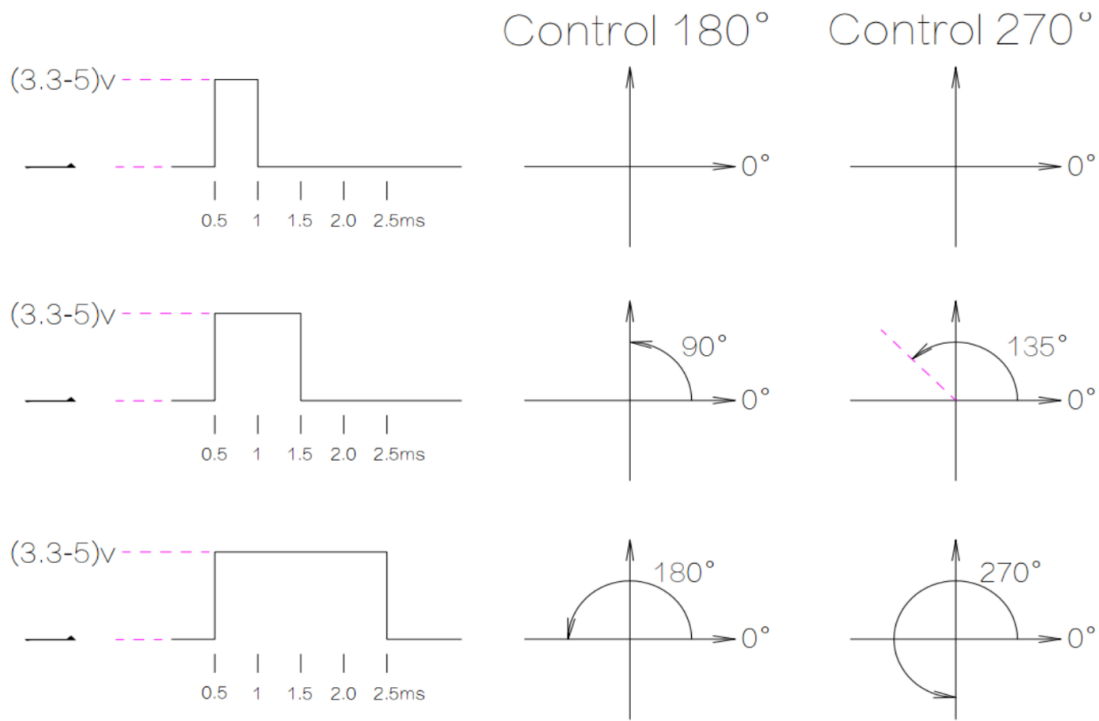
3. `delay(10)` 优化作用

- 防止频繁写入导致舵机响应滞后
- 降低处理器负载，提高系统稳定性

核心概念详解

1. 舵机工作原理

6. PWM - About PWM Control



参数	说明
控制信号	50Hz PWM方波（周期20ms）
脉冲宽度范围	500μs（0°）~2500μs（180°）
扭矩	9g·cm ~ 25kg·cm（型号决定）
供电要求	5-6V（可直接使用开发板供电，但易电压不稳）

2. PWM信号详解

PWM (Pulse width Modulation)	
高电平时间	控制舵机角度
0.5ms	0度
1.5ms	90度
2.5ms	180度

固定周期：20ms（50Hz）

3. 模拟信号采集

电位器 → 分压电路 → Arduino模拟输入
▲
电压范围：0V（完全左旋）~5V（完全右旋）
采样精度：10位（2^10=1024级）

硬件连接规范

舵机接线

舵机信号线（黄/橙） → D7（需PWM支持）
舵机电源正极（红） → 外部5V电源+
舵机电源负极（棕） → 外部电源- & Arduino GND

电位器接线

电位器左引脚 → Arduino GND
电位器中引脚 → A0
电位器右引脚 → Arduino 5V

调试与优化

常见问题排查

现象	解决方案
舵机抖动不转	检查供电是否充足
角度偏移	使用 writeMicroseconds() 校准
代码上传失败	断开舵机电源防止电流干扰

校准程序示例

```
void loop(){  
  servo_7.writeMicroseconds(1500); // 精确设置脉冲宽度  
  delay(1000);  
}
```

进阶应用扩展

多舵机同步控制

```
#include <Servo.h>  
Servo servo[3]; // 创建舵机数组  
  
void setup(){  
  for(int i=0; i<3; i++){  
    servo[i].attach(7+i); // 分别连接7,8,9引脚  
  }  
}
```

速度控制算法

```
int currentAngle = 0;
int targetAngle = 90;

void loop(){
    while(currentAngle != targetAngle){
        currentAngle += (targetAngle > currentAngle) ? 1 : -1;
        servo_7.write(currentAngle);
        delay(20); // 控制转动速度
    }
}
```

总结

本次涵盖舵机控制的核心知识点：

1. **PWM信号生成**：理解脉冲宽度与角度的对应关系
2. **模拟信号处理**：掌握电位器数值的采集与映射
3. **库函数应用**：熟练使用Servo类控制方法
4. **硬件电路设计**：正确搭建外设供电系统

通过实践这两个程序，可掌握基础机电控制原理，有了这个基础便已经可以搓一些简单的可动玩家。可使用示波器观察PWM波形变化，加深对舵机控制机制的理解。

有问题找柯萌ㄥㄥ ㄣ`｡｡｡
邮箱mingyoufhh@outlook.com