

# 树莓派智能小车舵机实验

## 目录

一、 实验概述.....	2
二、 实验器材.....	2
三、 知识要点.....	2
四、 实验原理.....	2
五、 代码编写.....	4
六、 硬件连接及运行效果.....	5
七、 思维发散及课后作业.....	6
官 网: <a href="http://www.xiao-r.com">www.xiao-r.com</a> .....	7
论 坛: <a href="http://www.wifi-robots.com">www.wifi-robots.com</a> .....	7
官方商城: <a href="http://wifi-robots.taobao.com">wifi-robots.taobao.com</a> .....	7
微信公众号: .....	7

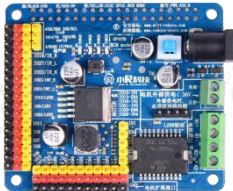
## 一、实验概述

使用树莓派主板、9G舵机以及小R科技配套的PWR.A53系列电源驱动板，实现让舵机转动到指定的角度。

## 二、实验器材



1、树莓派主板



2、PWR 电源板



3、9G 舵机\*1

4、杜邦线若干



5、12V 锂电池



### 三、知识要点

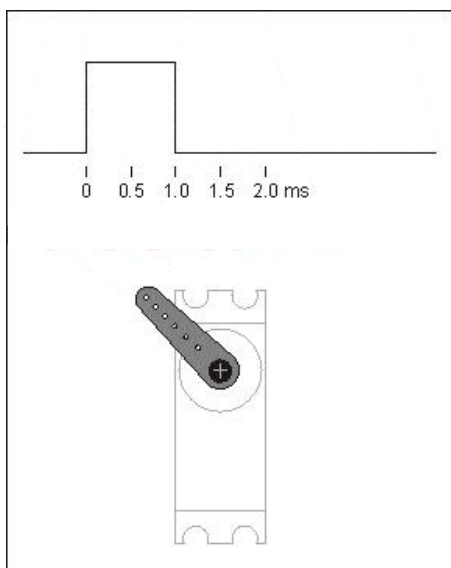
1. PWM 产生
2. 舵机工作原理
3. 树莓派舵机库的使用

### 四、实验原理

#### 舵机工作原理：

控制信号由接收机的通道进入信号调制芯片，获得直流的偏置电压。它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。最后电压差的正负输出到电机驱动芯片决定电机的正反转。当电机转速一定时，通过级联减速此轮带动电位器旋转，使得电压差为 0，电机停止转动。

舵机的控制：一般需要一个 20ms 左右的时基脉冲，该脉冲的高电平部分一般为 0.5ms-2.5ms 范围内的角度控制脉冲部分。本次实验的采用的舵机是 180 度伺服，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

#### 如何产生舵机 PWM 波形

使用 RPI.GPIO 模块的脉宽调制 ( PWM ) 功能

创建一个 PWM 实例：

```
p = GPIO.PWM(channel, frequency)
```

启用 PWM :

```
p.start(dc) # dc 代表占空比 ( 范围 : 0.0 <= dc >= 100.0 )
```

更改频率 :

```
p.ChangeFrequency(freq) # freq 为设置的新频率 , 单位为 Hz
```

更改占空比 :

```
p.ChangeDutyCycle(dc) # 范围 : 0.0 <= dc >= 100.0
```

停止 PWM :

```
p.stop()
```

以下为使 LED 每两秒钟闪烁一次的示例 :

```
import RPi.GPIO as GPIO
```

```
GPIO.setmode(GPIO.BOARD)
```

```
GPIO.setup(12, GPIO.OUT)
```

```
p = GPIO.PWM(12, 0.5)
```

```
p.start(1)
```

```
input('点击回车停止 : ') # 在 Python 2 中需要使用 raw_input
```

```
p.stop()
```

```
GPIO.cleanup()
```

这个方法呢有一个缺点 就是产生的 PWM 波形精度不高 ,因为树莓派没有硬件 PWM ,且里面跑的是系统 ,光靠软件 PWM 实时性不高。

因此 ,我们给树莓派 PWR.A53 电源板配备了硬件 PWM 模块 ,树莓派只需要给模块发号施令 ,而不需要关心 PWM 是如何产生的 ,并且可以扩展 8 路舵机控制 ,精确度足以达到舵机的准度。以下是控制硬件 PWM 模块的指令库

```
XRservo.XiaoRGEEK_SetServo(x,y)
```

说明 : 这个是设置舵机角度的函数,x 代表要设定的舵机标号(1-8) , y 是要设定的角度(0-180)

```
XRservo.XiaoRGEEK_SaveServo()
```

说明 : 将当前所有的舵机角度存储起来

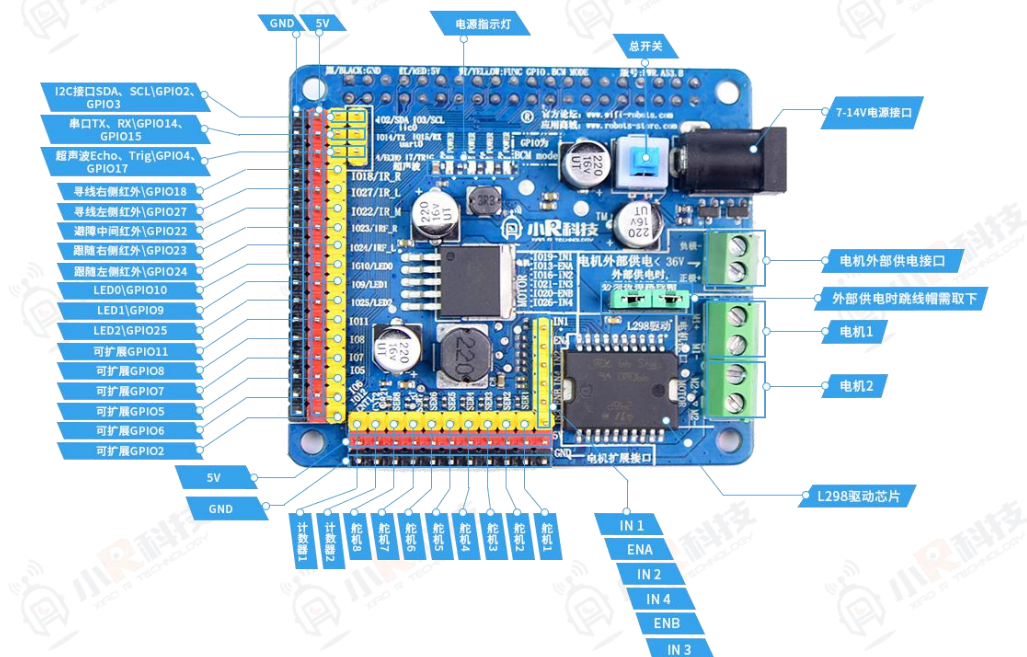
```
XRservo.XiaoRGEEK_ReSetServo()
```

说明：将上次所有存储的舵机角度读取出来，并让舵机恢复到上次存储的角度位置

## 舵机驱动能力

当我们使用多个舵机时候，就会出现另外一个问题，多个舵机同时工作需要的电流比较大，就是树莓派的驱动能力不够，这个时候给舵机单独供电，小R科技的舵机驱动板就起到这个作用。

PWR 电源板的信号接口图



## 五、代码编写

```
#coding:utf-8
```

```
import os
```

```
import RPi.GPIO as GPIO
```

```
import time
```

```
from smbus import SMBus
```

```
XRservo = SMBus(1)
```

```
while True:
```

```
    for i in range(180): #从 0 度转到 180 度
```

```
XRservo.XiaoRGEEK_SetServo(1,i)
time.sleep(0.01)
for i in range(180): #从 180 度转到 0 度
    XRservo.XiaoRGEEK_SetServo(1,180-i)
    time.sleep(0.01)
```



## 六、硬件连接及运行效果

硬件连接如下图所示

注：连接电源板标有 SER1 标志的接口位置。

舵机

树莓派 PWR.A53 电源板

橙色线

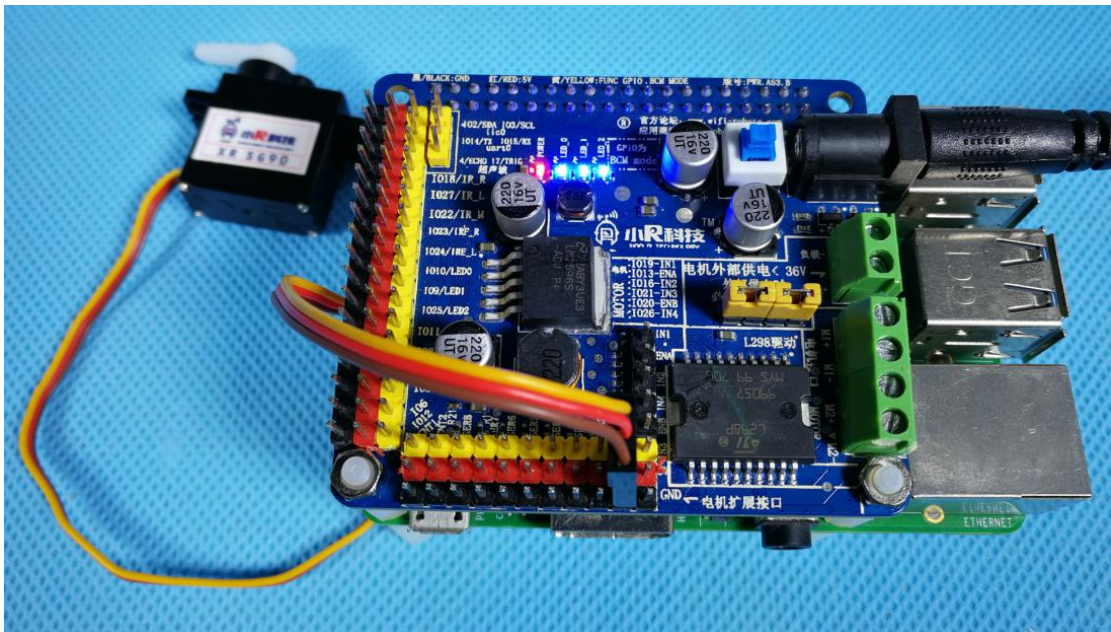
SER1

红色线

5V

灰色线

GND



运行效果:通过 Winscp 将 Servo.py 文件上传到树莓派系统中,使用 `sudo python Servo.py` 即可运行舵机程序。

观察到舵机从 0 度转到 180 度后又从 180 度转到 0 度。

## 七、思维发散及课后作业

现在我们能控制一个舵机转动 180 度，可以在垂直方向添加一个舵机，组合成一个上下转动 180，左右转动 180 度的云台。

官 网：[www.xiao-r.com](http://www.xiao-r.com)

论 坛：[www.wifi-robots.com](http://www.wifi-robots.com)

官方商城：[wifi-robots.taobao.com](http://wifi-robots.taobao.com)

微信公众号：

