

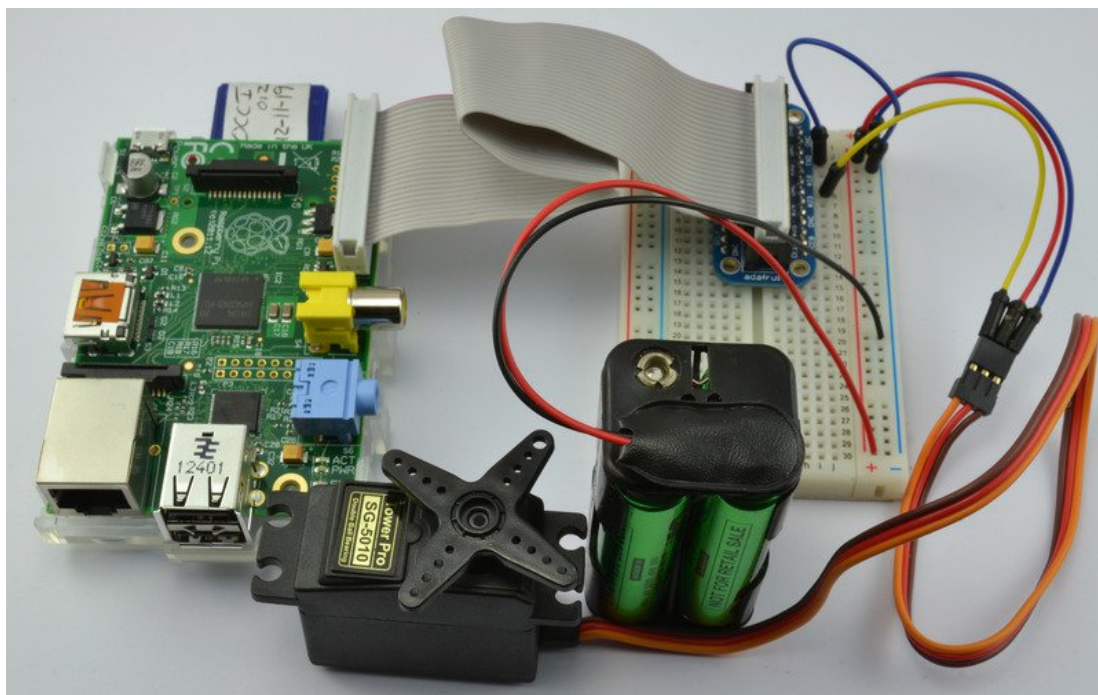
کنترل سروو موتور با برد رزبری | درس نهم



سری آموزش رزبری پای: درس نهم

در این درس نحوه کنترل سروو موتور را با استفاده از برنامه نویسی پایتون و برد رزبری پای یاد خواهید گرفت.

[کنترل برد رزبری پای با کامپیوتر از راه دور | درس هشتم](#)



کنترل سروو موتورها توسط پالس هایی با طول های متفاوت انجام می گیرد که نیاز به زمان نسبتا دقیقی دارند. برد رسیپری پای توانایی تولید پالس را توسط یک پین سخت افزاری دارد، بدون اینکه به برنامه نویسی خاصی احتیاج داشته باشد.

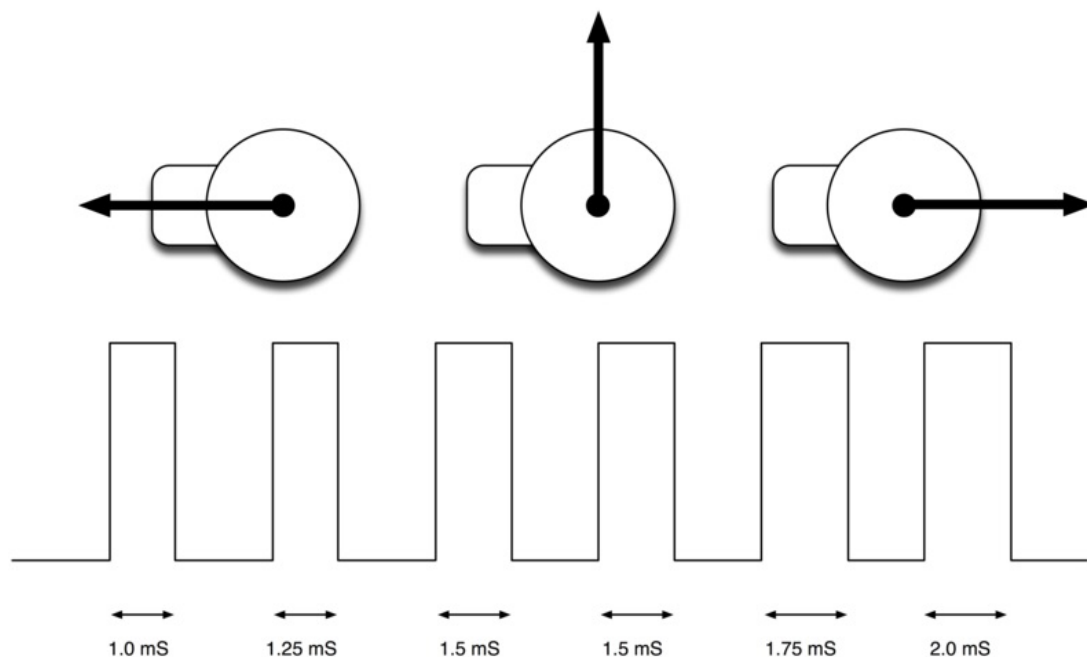
قطعات مورد نیاز:

برای این آموزش شما به قطعات زیر احتیاج دارید، می توانید تمامی قطعات را از بخش فروشگاه صنعت بازار تهیه کنید:

برد رسیپری پای	1 عدد	
برد GPIO تی شکل + کابل	1 عدد	
سیم اتصال برد برد	1 بسته	
برد برد	1 عدد	
سروو موتور	1 عدد	
جعبه باتری 4 تایی	1 عدد	

سروو موتور:

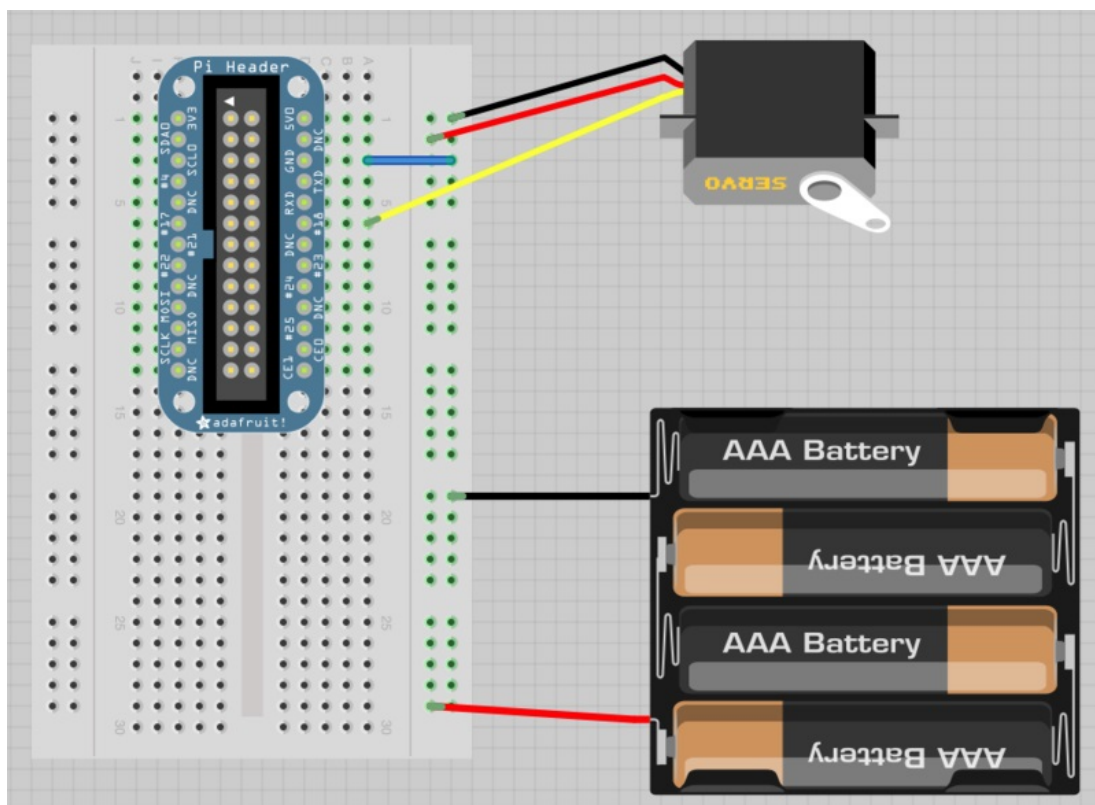
مکان و محل قرار گیری سروو موتور با توجه به طول پالس تنظیم می شود. سروو موتور باید یک پالس را تقریبا هر 20 میلی ثانیه دریافت کند. در صورتی که این پالس برای مدت 1 میلی ثانیه در وضعیت high قرار داشته باشد، سروو موتور در زاویه صفر قرار خواهد گرفت، در صورتی که این زمان 1.5 میلی ثانیه باشد، در موقعیت میانی خود قرار می گیرد و در صورتی که 2 میلی ثانیه باشد، در زاویه 180 درجه قرار خواهد گرفت.



نقطه پایانی در مدل های مختلف، متفاوت است و در بیشتر مدل ها مقدار آن 170 درجه می باشد. در صورتی که مدل های continuous را خریداری کنید می توانید تا 360 درجه نیز چرخش انجام دهید.

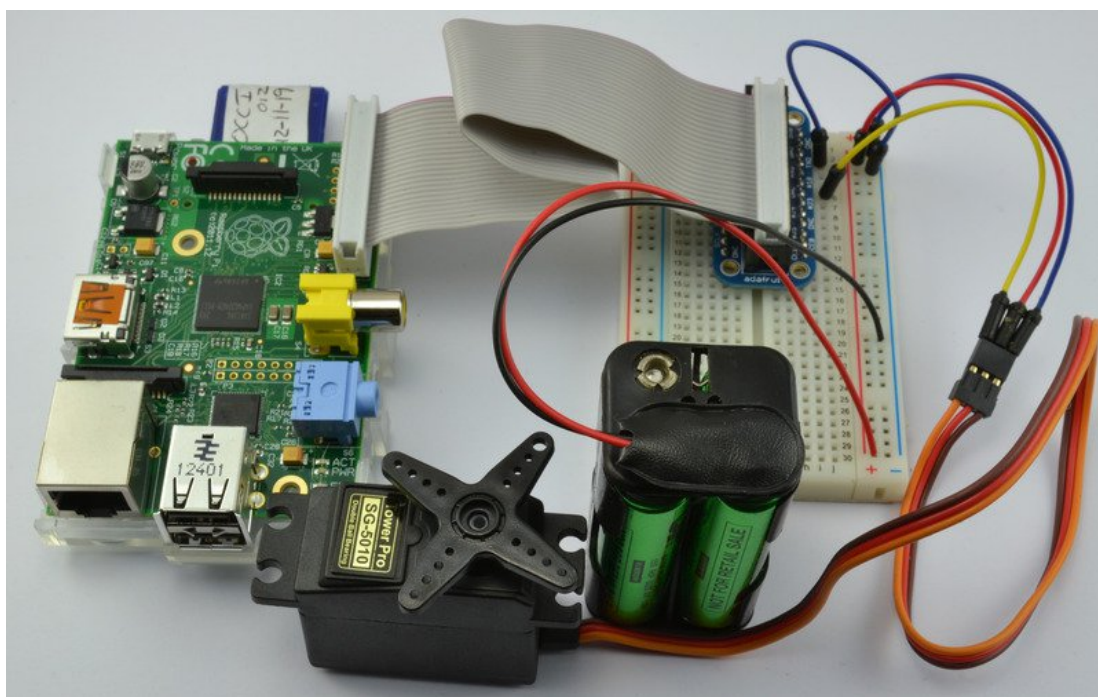
سخت افزارها:

فقط یک پین بر روی برد رسیبری پای قادر به تولید پالس از این راه می باشد (پین 18 پورت GPIO)، که به پین کنترل سروو موتور متصل می شود. توان مورد نیاز برای راه اندازی سروو از طریق باتری های خارجی تامین می شود، چون در صورت به کارگیری برد رسیبری پای برای تامین توان، به دلیل جریان کشی بالای موتور در هنگام شروع کار، امکان آسیب دیدن برد رسیبری پای وجود دارد. سروو موتورها به ولتاژ 4.8-6 ولت برای راه اندازی احتیاج دارند، در صورتی که ولتاژ سیگنال کنترلی (پالس خروجی) می تواند 3.3 ولت باشد، لذا اتصال مستقیم آن به پین های GPIO مشکلی ایجاد نمی کند.



برد cobbler برای اتصال برد رسیبری پای به بردبورد مورد استفاده قرر می گیرد. در صورتی که قبلا از آن استفاده نکرده اید، درس چهارم از این مجموعه آموزش را مطالعه کنید.

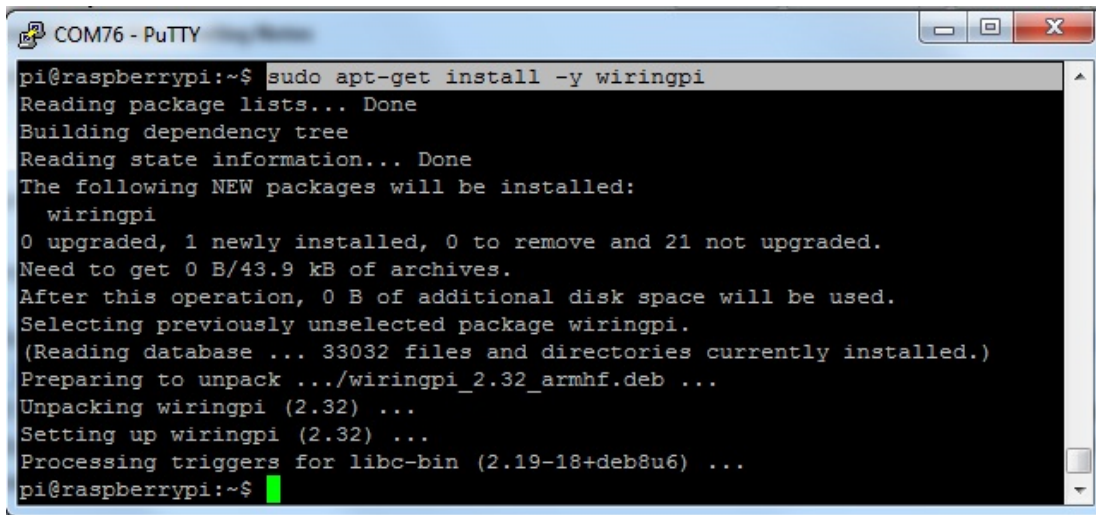
سروو موتور ها دارای سه سیم خروجی می باشند که به یک سوکت متصل شده اند. سیم قرمز و قهوه ای مربوط به تامین توان(مثبت به قرمز متصل می شود) و سیم زرد و یا نارنجی مربوط به سیگنال کنترلی است. برای اتصال آن ها به بردبورد از سیم های جامپر نری به نری استفاده کنید.



نرم افزار:

برای کنترل سروو موتور از ابزار قدرتمند wiringPi مربوط به GPIO استفاده می شود. در ابتدا gpio را نصب کنید:

```
sudo apt-get install -y wiringpi
```



پین #18 را به عنوان خروجی PWM تنظیم کنید.

```
gpio -g mode 18 pwm
```

با این دستور پین #18 به عنوان خروجی PWM تنظیم می شود، ولی باید فرکانس آن تنظیم گردد. فرکانس مورد نیاز 50Hz می باشد.

نقشه PWM برای برد رسیبری پای به این صورت است:

$$PWM \text{ Frequency in Hz} = 19,200,000 \text{ Hz} / pwmClock / pwmRange$$

در صورتی که pwmClock برابر با 192 و pwmRang برابر با 2000 باشد، در این صورت فرکانس PWM برابر با 50Hz خواهد بود.

این اعداد را در gpio تنظیم کنید:

```
gpio pwm-ms
```

```
gpio pwmc 192
```

```
gpio pwmr 2000
```

حال می توانید وضعیت سروو موتور را تنظیم کنید. برای قرارگیری در سمت چپ (1 میلی ثانیه) دستور زیر را وارد کنید:

```
gpio -g pwm 18 100
```

جهت قرارگیری در حالت میانی (1.5 میلی ثانیه):

```
gpio -g pwm 18 150
```

و برای قرارگیری در سمت راست (2 میلی ثانیه):

```
gpio -g pwm 18 200
```

سروو موتورها معمولاً به محدوده بیشتر از 1-2 میلی ثانیه نیز پاسخ می دهند، بنابراین سعی کنید اعداد 50 (0.5 میلی ثانیه) و 250 (2.5 میلی ثانیه) را نیز تست کنید.

شما می توانید هر عددی بین 50 تا 250 را استفاده کنید، لذا امکان قرارگیری در 200 موقعیت وجود دارد.

همچنین در نظر داشته باشید که خروجی PWM برد رزبری پای لزوماً پایدار نیست و امکان مقداری خطا در آن وجود دارد.

مثال پایتون:

شما همچنین می توانید از wiringPi در پایتون استفاده کنید! دستور زیر را اجرا کنید:

```
sudo apt-get install -y python-pip
sudo pip install wiringpi
```

برنامه پایتون برای تنظیم حرکت رفت و برگشتی به صورت زیر می باشد:

```
Servo Control #
import time
import wiringpi

'use 'GPIO naming #
()wiringpi.wiringPiSetupGpio

set #18 to be a PWM output #
(wiringpi.pinMode(18, wiringpi.GPIO.PWM_OUTPUT

set the PWM mode to milliseconds stype #
(wiringpi.pwmSetMode(wiringpi.GPIO.PWM_MODE_MS

divide down clock #
(wiringpi.pwmSetClock(192
(wiringpi.pwmSetRange(2000
```

```

delay_period = 0.01

while True:
    for pulse in range(50, 250, 1):
        wiringpi.pwmWrite(18, pulse)
        time.sleep(delay_period)
    for pulse in range(250, 50, -1):
        wiringpi.pwmWrite(18, pulse)
        time.sleep(delay_period)

```

پایه و اساس برنامه همان به کارگیری تابع wiringpi می باشد که در اینجا در پایتون به کار رفته است، لذا به راحتی می توانید از امکانات gpio استفاده کرده دستورات را در پایتون وارد کنید.

متغیر delay_period شامل زمان هایی برحسب ثانیه در بین هر پله از حرکت سروو می باشد.

حلقه while همواره ادامه خواهد داشت مگر اینکه برنامه توسط فشردن کلیدهای CTRL-C متوقف گردد. داخل حلقه while دو حلقه یکسان for وجود دارد. حلقه اول طول پالس ها را از 0.5 تا 2.5 میلی ثانیه افزایش داده و حلقه دوم برعکس عمل کرده و طول پالس ها را از 2.5 تا 0.5 میلی ثانیه کاهش می دهد و موقعیت را به حالت اول باز می گرداند.

برای نصب نرم افزار توسط SSH به برد رسیبری پای متصل شده و دستور زیر را وارد کنید:

```
nano servo.py $
```

کد بالا را در ویرایشگر وارد کرده و سپس با استفاده از CTRL-X و Y فایل را ذخیره کنید.

```

pi@raspberrypi: ~ — ssh — 80x24
GNU nano 2.2.6 File: servo.py Modified

# Servo Control
import time
def set(property, value):
    try:
        f = open("/sys/class/rpi-pwm/pwm0/" + property, 'w')
        f.write(value)
        f.close()
    except:
        print("Error writing to: " + property + " value: " + value)

def setServo(angle):
    set("servo", str(angle))

set("delayed", "0")
set("mode", "pwm [servo] audio")
set("servo_max", "180")
set("active", "1")

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell

```

برای اجرای برنامه کنترل سروو موتور فقط دستور زیر را در پنجره SSH وارد کنید:

```
sudo python servo.py $
```

سروو شروع به حرکت خواهد کرد.

نکات نهایی:

در صورتی که می خواهید سروو موتور سریع تر حرکت کند، زمان `delay_period` را کوچک تر کنید، به طور مثال 0.001 ثانیه. و در صورتی که می خواهید سرعت آن را کاهش دهید این زمان را به 0.1 افزایش دهید.

همچنین امکان کنترل چند سروو موتور توسط برد رزبری پای وجود دارد که نیاز به استفاده از چند تجهیزات جانبی است. به زودی آموزش مربوط در سایت قرار خواهد گرفت.

کنترل موتور DC با برد رزبری پای | درس دهم

برای بهتر شدن محتوای مطالب لطف کنید نظرات، انتقادات و پیشنهادات خود را با ما در میان بگذارید...

ترجمه شده و تکمیل توسط تیم الکترونیک صنعت بازار | منبع: سایت `adafruit`