

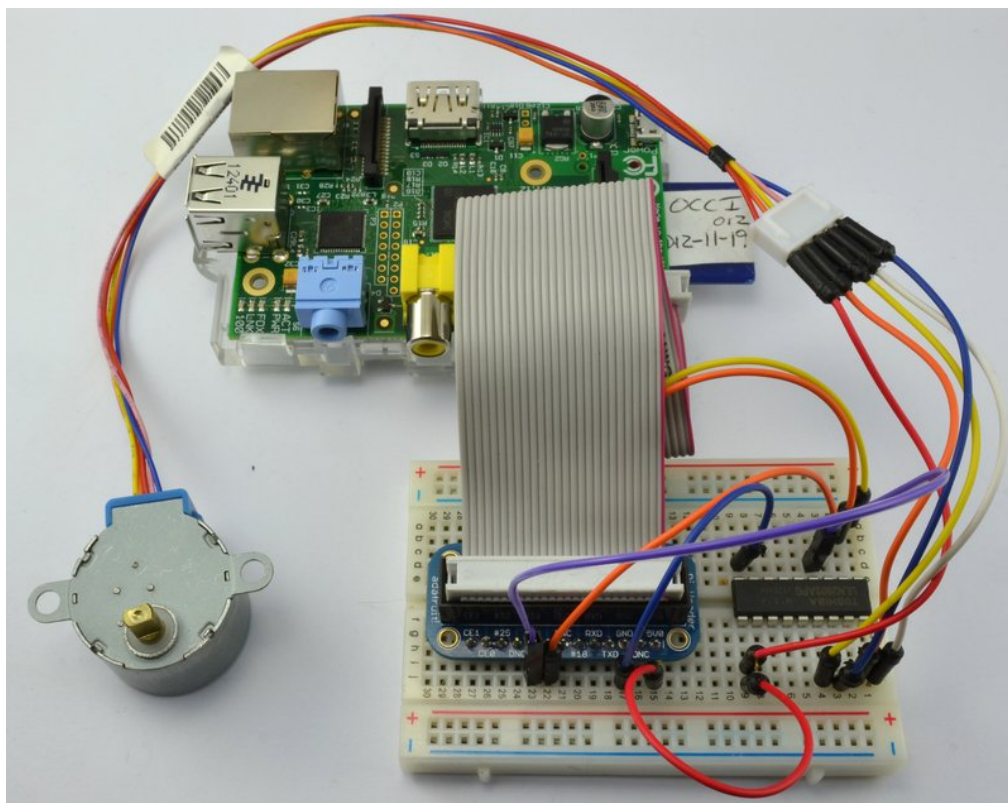
کنترل استپر موتور با برد رزبری پای | درس یازدهم



سری آموزش رزبری پای: درس یازدهم

استپر موتورها در واقع چیزی میان موتورهای DC و سروو موتورها می باشند و این مزیت را دارند که می توان به راحتی آنها را موقعیت یابی کرد، موتور حرکت رو به جلو یا عقب داشته باشد و یا اینکه به طور مداوم بچرخد.

کنترل موتور DC با برد رزبری پای | درس دهم



در این مقاله نحوه کنترل استپر موتور توسط برد رسیبری پای و آی سی کنترل کننده موتور L293D که در درس دهم برای کنترل موتور DC از آن استفاده گردید، مورد بررسی قرار می گیرد.





همچنین چگونگی جایگزینی آی سی ULN2803 به عنوان درایور آموزش داده می شود.



برای این پروژه فرقی نمی کند که از L293D یا ULN2803 استفاده کنید. اگر تراشه دیگری ندارید، استفاده از ULN2803 به خاطر قیمت ارزان و دارا بودن چهار خروجی جداگانه، آن را به بهترین گزینه تبدیل می کند.

استپر موتورها دارای توان مصرفی پایین بوده و نسبت به موتور های DC و سروو به جریان الکتریکی کمتری نیاز دارند. برای کارکرد مناسب این پروژه می توانید از خروجی 5V رسیبری پای استفاده کنید، فقط در نظر داشته باشید که رسیبری پای به یک منبع تغذیه مناسب با حداقل جریان آمپر 1 متصل باشد.

قطعات مورد نیاز:

قطعات مورد نیاز برای انجام این آموزش را می توانید از بخش فروشگاه صنعت بازار تهیه کنید:

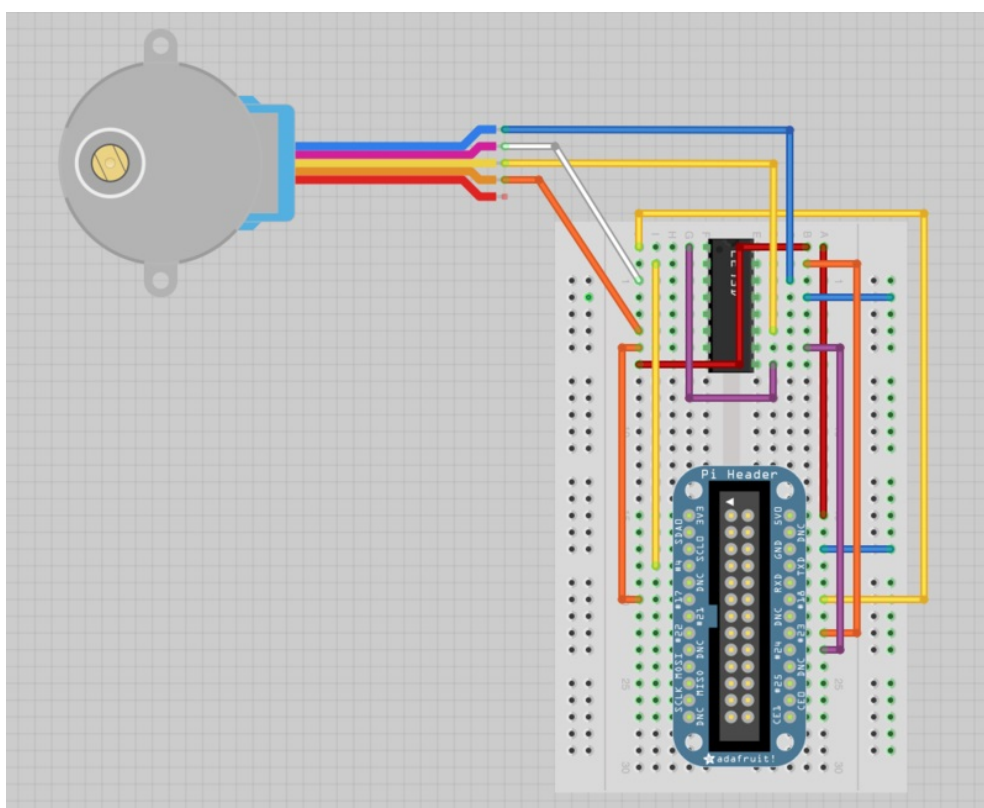
	1 عدد	برد رسیبری پای
	1 عدد	برد GPIO تی شکل + کابل
	1 بسته	سیم اتصال برد برد
	1 عدد	برد برد

	1 عدد	آی سی درایور موتور L293D
	1 عدد	آی سی ULN2803
	1 عدد	استپر موتور

سخت افزار (L293D):

استپر موتور به کار گرفته شده در این آموزش از نوع 5 سیم می باشد، و برای راه اندازی آن از هر دو سمت درایور L293D استفاده می شود، که به معنای وجود اتصالات زیاد در برد مورد می باشد.

در انتهای سیم ها یک سوکت 5 راهه وجود دارد که از طریق سیم جامپر می توان آن ها به برد مورد متصل کرد.



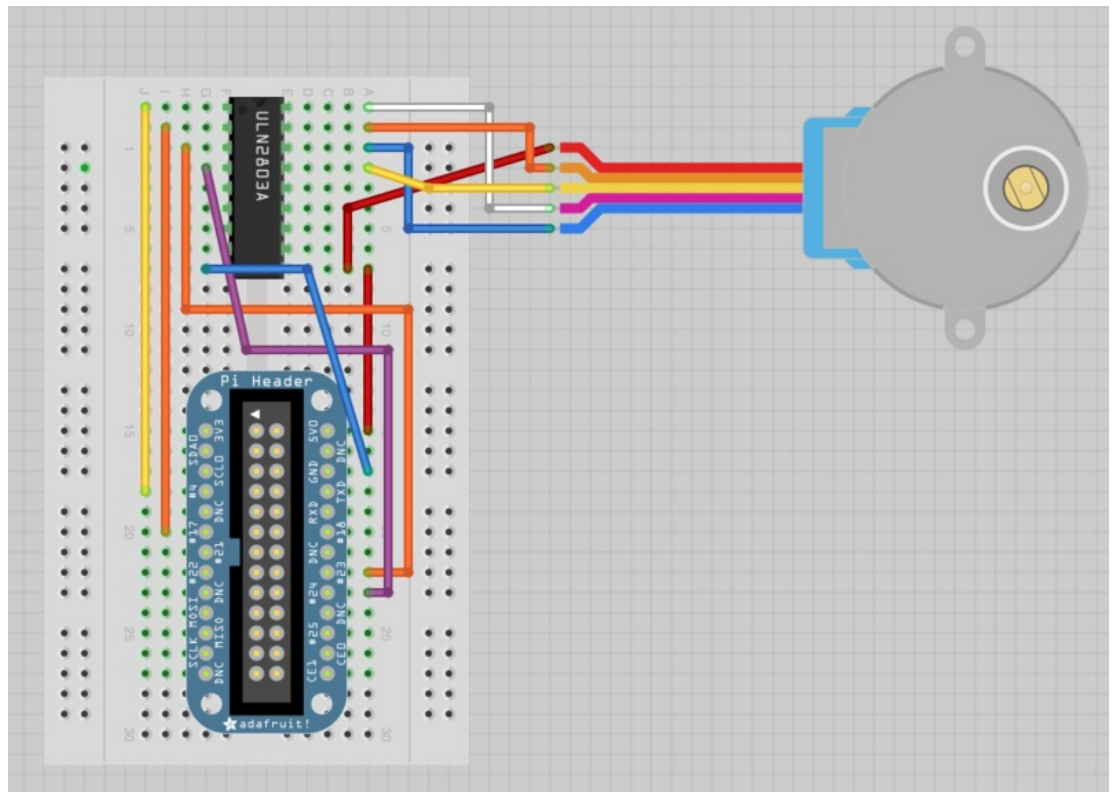
توجه داشته باشید که سیم قرمز رنگ به جایی متصل نمی شود.

نکته: فقط از رنگ سیم ها برای شناسایی آنها استفاده کنید، محل خارج شدن سیم از موتور راه مناسبی برای شناسایی سیم نمی باشد.

سخت افزار (ULN2803):

در صورتی که از ULN2803 استفاده می کنید، هر 5 سیم به کار خواهد رفت. مانند حالت قبل از سیم جامپر ها برای اتصال به برد مورد استفاده کنید.

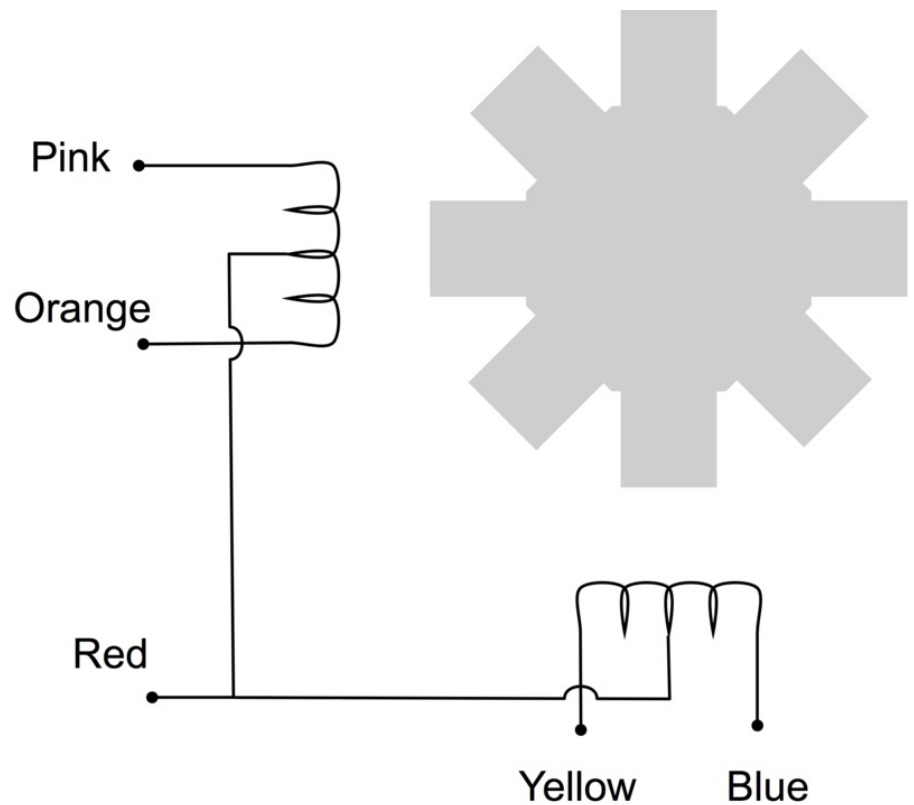
توجه کنید که روش و مباحث بیان شده، فقط برای کنترل استپر موتورهای 5 سیم دو قطبی می باشد.



اگرچه در کد زیر از پایه 18 پورت GPIO به عنوان پایه فعالساز استفاده شده ولی اینکار فقط زمانی که از L293D استفاده می کنید، لازم است.

استپر موتور:

این موتورها از چرخ دنده و آهنربای الکتر مغناطیسی برای چرخش در هر پله استفاده می کنند.



با وارد کردن انرژی به سیم پیچ سمت راست، موتور شروع به چرخش می کند. تعداد پله ها برای چرخش 360 درجه، در واقع همان تعداد دندانه های چرخ دنده می باشد.

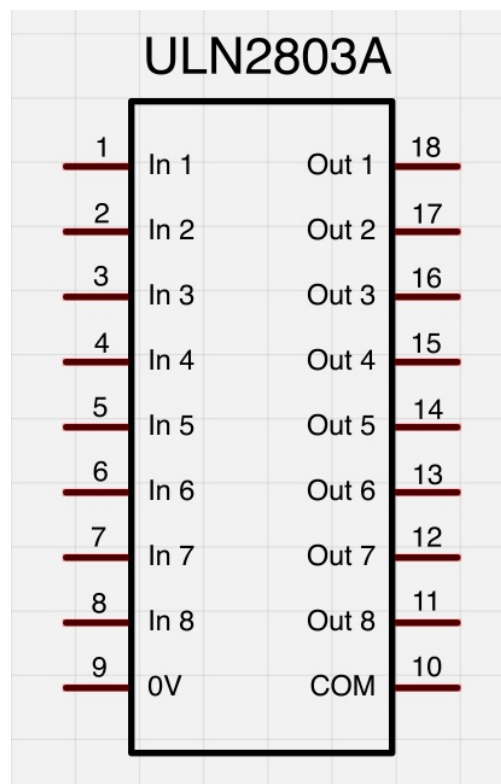
موتور استفاده شده در این درس، 8 پله می باشد. اما با توجه به دارا بودن جعبه دنده 1:64، برای چرخش کامل نیازمند $512 = 64 \times 8$ پله است.

در این درس از سیم قرمز رنگ استفاده نمی شود. به کارگیری آن تنها زمانی توصیه می شود، که نوع متفاوتی از مدار درایور استفاده شود، که اجازه بازگشت هیچ جریانی از سیم پیچ ها را نمی دهد. وجود اتصال مرکزی میان سیم پیچ ها بدین معناست، که بدون استفاده از مدار معکوس کننده جریان می توان به بخش راست یا چپ سیم پیچ نیرو وارد کرده و اثر جریان بازگشتی را دریافت کرد.

در صورتی که از درایور L293D استفاده می کنید، نیازی به این اتصال میانی نیست، چون این درایور توانایی تولید جریان معکوس را دارد. لذا شما می توانید جریان را به هر کدام از سیم پیچ ها وارد کنید.

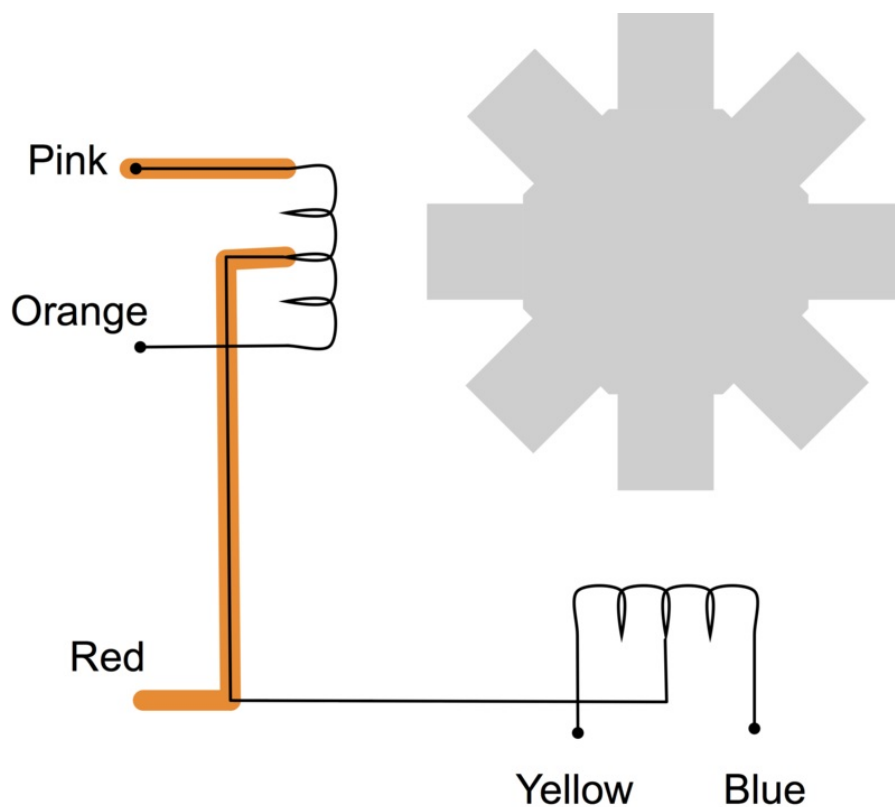
: ULN2803

آی سی درایور موتور L293D در درس دهم مورد بررسی قرار گرفت. حال آی سی پرکاربرد ULN2803 بررسی می گردد.



L293D در حقیقت دارای 4 خروجی می باشد، که می توانند به طور متقارن عمل معکوس سازی را انجام دهند. اما 8 ULN2803 خروجی دارد، که سیگنال های ضعیف تولید شده توسط پین های GPIO برد رزبری پای را تقویت کرده و اجازه اعمال جریان های قوی تر را می دهد.

برخلاف L293D، خروجی ULN2803 فقط در یک جهت جریان تولید می کند، لذا باید از سیم قرمز مشترک استفاده شود. بنابراین به جای به کارگیری تمامی سیم پیچ بین سیم صورتی و نارنجی، فقط به نیمی از آن که بین سیم قرمز مشترک و صورتی قرار دارد، انرژی وارد می شود.



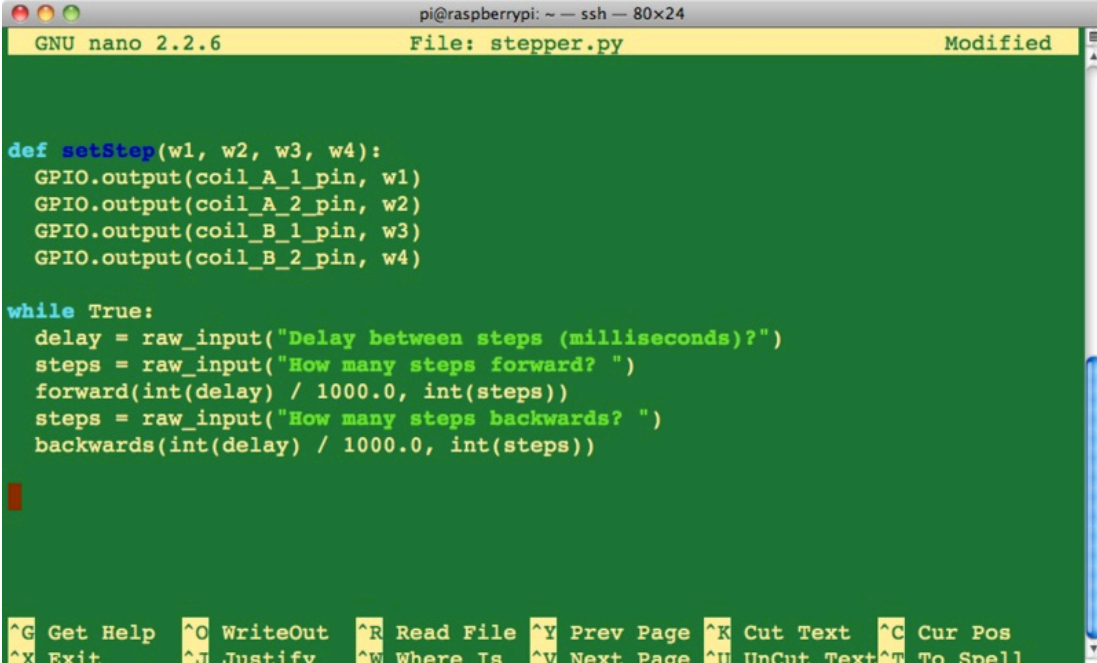
نرم افزار:

نرم افزار مورد استفاده برای هر کدام از تراشه های L293D یا ULN2803 کاملاً مشابه می باشد. در این پروژه از کتابخانه ی Rpi.GPIO استفاده شده است. اگر شما قبلاً از آن استفاده نکرده اید، ابتدا از طریق این لینک آن را نصب کنید.

برای نصب کد، باید به وسیله SSH به برد رزبری پای خود متصل شده با وارد کردن دستور زیر پنجره ی ویرایشگر (editor window) را باز کنید:

```
nano stepper.py$
```

در ادامه کد زیر را در آن وارد کرده و در نهایت با استفاده از CTRL-X و سپس Y آنرا ذخیره نمایید.



```

pi@raspberrypi: ~ -- ssh -- 80x24
GNU nano 2.2.6 File: stepper.py Modified

def setStep(w1, w2, w3, w4):
    GPIO.output(coil_A_1_pin, w1)
    GPIO.output(coil_A_2_pin, w2)
    GPIO.output(coil_B_1_pin, w3)
    GPIO.output(coil_B_2_pin, w4)

while True:
    delay = raw_input("Delay between steps (milliseconds)?")
    steps = raw_input("How many steps forward? ")
    forward(int(delay) / 1000.0, int(steps))
    steps = raw_input("How many steps backwards? ")
    backwards(int(delay) / 1000.0, int(steps))

^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
  
```

```

import RPi.GPIO as GPIO
import time

(GPIO.setmode(GPIO.BCM

enable_pin = 18
coil_A_1_pin = 4
coil_A_2_pin = 17
coil_B_1_pin = 23
coil_B_2_pin = 24
  
```



```

(GPIO.setup(enable_pin, GPIO.OUT)
(GPIO.setup(coil_A_1_pin, GPIO.OUT)
(GPIO.setup(coil_A_2_pin, GPIO.OUT)
(GPIO.setup(coil_B_1_pin, GPIO.OUT)
(GPIO.setup(coil_B_2_pin, GPIO.OUT)

(GPIO.output(enable_pin, 1)

:(def forward(delay, steps
:(for i in range(0, steps
(setStep(1, 0, 1, 0
(time.sleep(delay)
(setStep(0, 1, 1, 0
(time.sleep(delay)
(setStep(0, 1, 0, 1
(time.sleep(delay)
(setStep(1, 0, 0, 1
(time.sleep(delay)

:(def backwards(delay, steps
:(for i in range(0, steps
(setStep(1, 0, 0, 1
(time.sleep(delay)
(setStep(0, 1, 0, 1
(time.sleep(delay)
(setStep(0, 1, 1, 0
(time.sleep(delay)
(setStep(1, 0, 1, 0
(time.sleep(delay)

:(def setStep(w1, w2, w3, w4
(GPIO.output(coil_A_1_pin, w1)
(GPIO.output(coil_A_2_pin, w2)
(GPIO.output(coil_B_1_pin, w3)
(GPIO.output(coil_B_2_pin, w4)

:while True
("?(delay = raw_input("Delay between steps (milliseconds
(" ?steps = raw_input("How many steps forward
((forward(int(delay) / 1000.0, int(steps)

```



```
(" ?steps = raw_input("How many steps backwards\n\n((backwards(int(delay) / 1000.0, int(steps
```

استپر موتورها در زمان سکون هم هنوز فعال بوده و موقعیت خود را حفظ می کنند، که این خود نیاز به مصرف انرژی دارد. در صورتی که نیاز به حفظ موقعیت و وضعیت ندارید، دستور `setStep(0,0,0,0)` را فراخوانی کنید تا سیم پیچ ها آزاد گردند. در این صورت محور موتور آزادانه حرکت می کند و جریان زیادی مصرف نخواهد کرد.

تست و نکات نهایی:

برنامه باید در حالت `super-user` اجرا گردد، بنابراین دستور زیر را در `SSH session` وارد کنید.

```
sudo python stepper.py$
```

در ادامه مقدار تاخیر (5 مقدار خوبی است) و سپس تعداد پله ها (512 برای چرخیدن کامل) را وارد کنید.

```
pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi ~ $ sudo python stepper.py
Delay between steps (milliseconds)?5
How many steps forward? 128
How many steps backwards? 128
Delay between steps (milliseconds)?
```

برای پیدا کردن حداکثر سرعت موتور، زمان تاخیر را کاهش دهید.

[راه اندازی سنسور دمای دیجیتال DS18B20 با برد رزبری پای | درس دوازدهم](#)

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

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