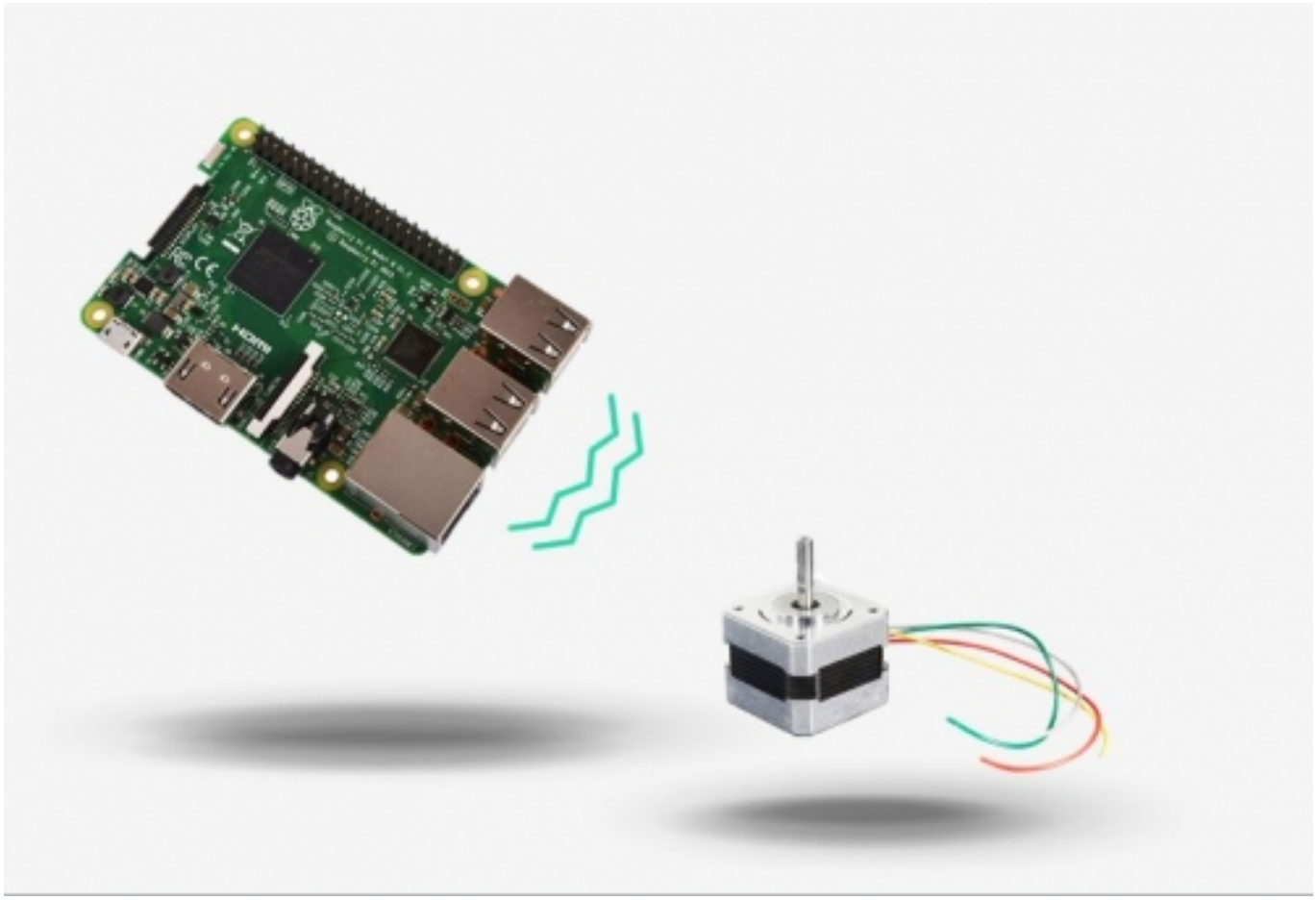
آموزش موتورهای استپر با رسپبری‌پای (قسمت دوم)



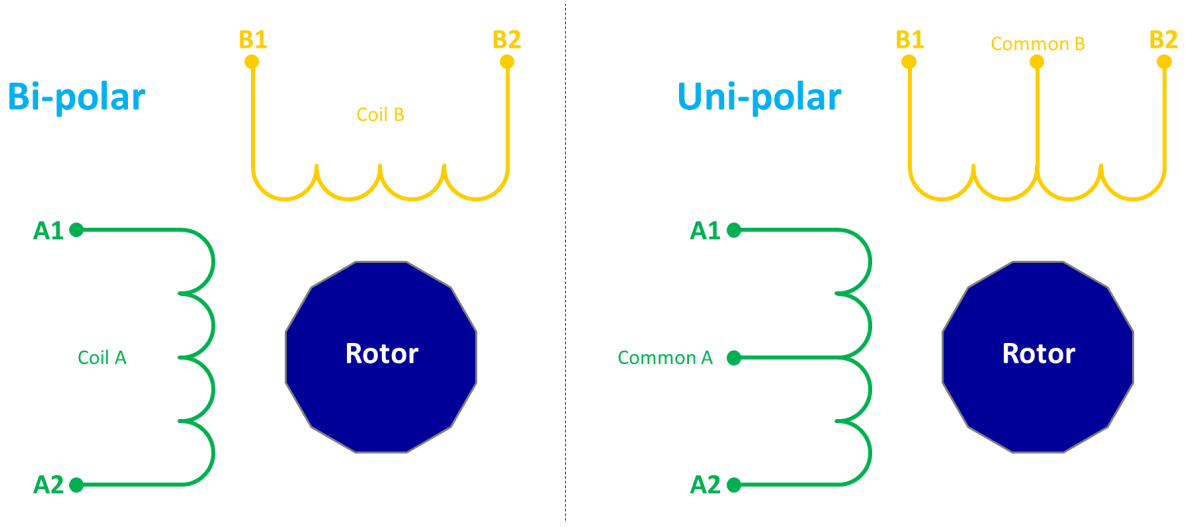
در این آموزش نشان می‌دهم که چگونه موتورهای استپر (پله‌ای) دوقطبی را روی یک رسپبری‌پای در پایتون با استفاده از یک درایور استپ موتور DRV-8825 کنترل کنیم.

در آموزش " آموزش موتورهای استپر با رسپبری‌پای (قسمت اول)" با موتور پله‌ای و درایور DRV-8825 آشنا شده و نحوه اتصال آن‌ها به برد رسپبری‌پای نشان داده شد.

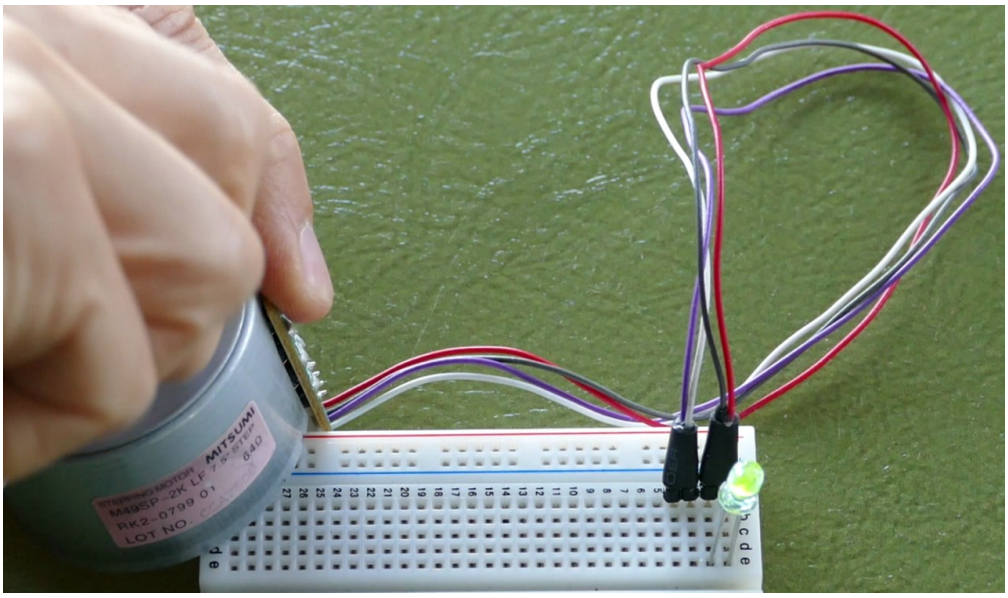
بسیار ایده خوبی است که بر دمای برد درایور و موتور پله‌ای نظارت کنیم. در اینجا از یک قانون 5 ثانیه‌ای استفاده می‌کنیم. اگر بتوانیم موتور و برد درایور را برای مدت‌زمان حداقل 5 ثانیه لمس کنم و انگشتانمان نسوزد، بنابراین جریان احتمالاً مناسب است. لطفاً توجه داشته باشید که جریان هم‌چنین می‌تواند به خاطر قانون اهم با استفاده از مقاومت سیم‌پیچ موتور محدود شود (V=I*R). به‌عنوان‌مثال، اگر مقاومت سیم‌پیچ شما 30 اهم باشد و منبع تغذیه شما 12 ولت باشد بنابراین جریان به 0.4 آمپر محدود خواهد شد.

$$12V = 0.4A \times 30\Omega$$

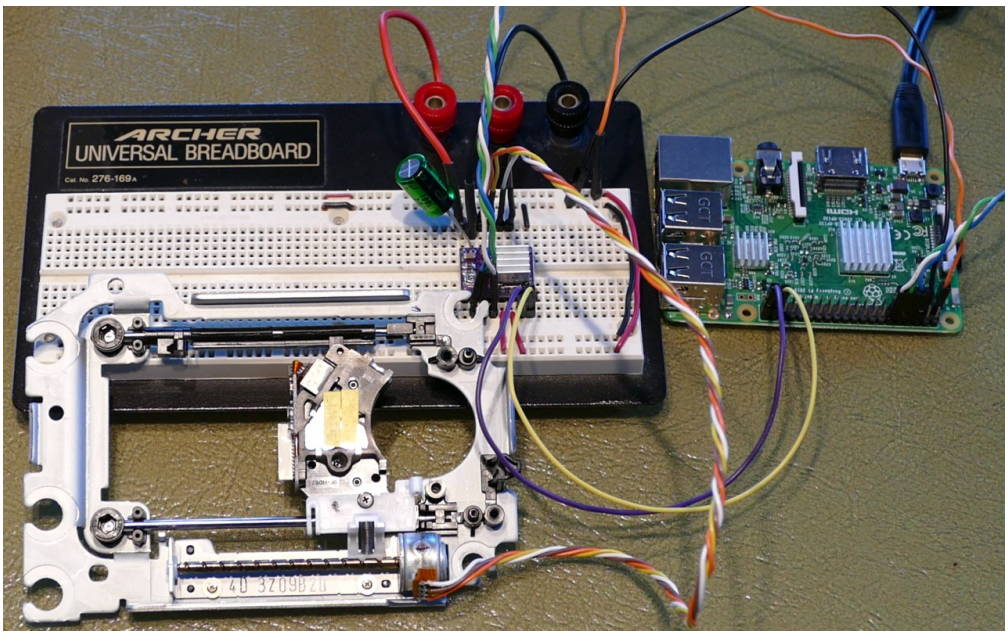
به‌مجرداینکه جریان بیشینه تنظیم شد موتور پله‌ای متواند متصل شود. یک موتور پله‌ای متواند هرچند عدد سیم‌پیچ داشته باشد؛ اما این سیم‌پیچ‌ها به‌صورت گروهی به هم متصل می‌شوند و "phases" نامیده می‌شوند. همه سیم‌پیچ‌ها در یک‌فاز باهم انرژی می‌گیرند و یا تغذیه می‌شوند. موتورهای پله‌ای دوقطبی 4 سیم‌دارند و 2 فازی هستند و 2 گروه از سیم‌پیچ‌ها را دارند. درایور برای چرخاندن روتور قطبیت سیم‌پیچ را تغییر می‌دهد. نوع دیگر موتور پله‌ای تک‌قطبی است که یک مرکز برای هر سیم‌پیچ دارد. درایور تک‌قطبی می‌تواند هر سیم‌پیچ را نیمه تغذیه کند تا قطب را تغییر دهد. این کار مدار را در مقایسه با دوقطبی آسان‌تر می‌کند اما همچنین گشتاور را کاهش می‌دهد زیرا تنها نصف سیم‌پیچ‌ها انرژی‌دار می‌شوند. موتورهای تک‌قطبی 6 سیمی اغلب می‌توانند با نادیده گرفتن مرکز سیم‌پیچ، به‌عنوان دوقطبی کار کنند. موتور پله‌ای تک‌قطبی 5 سیمی دارای 2 مرکز متصل به یکدیگر است. برای استفاده از آن‌ها به‌عنوان دوقطبی، اتصالات بین مرکزها باید رعایت شود.



سیم‌های موتور پله‌ای اغلب برچسب ندارند. یکی از راحت‌ترین روش‌های برای تشخیص اینکه کدام سیم برای کدام سیم‌پیچ است، استفاده از یک LED است. یک LED در طول هر 2 سیم موتور قرار دهید. جهت اهمیتی ندارد. موتور را با دست بچرخانید و اگر LED چشمک زد 2 سیم یک سیم‌پیچ هستند. اگر شما LED ندارید و یا اینکه موتور پله‌ای این‌قدر کوچک است که نمی‌تواند LED را تغذیه کند، شما می‌توانید از یک مالتی متر برای چک کردن سیم‌پیچ‌ها استفاده کنید و با استفاده از تست دیود که بوق میزند سیم‌های متصل به یک سیم‌پیچ را تشخیص دهید.



در اینجا عکسی از یک موتور پله‌ای CD درایور که با سیم به DRV8825 متصل است را نشان می‌دهد:



اتصال به رسپبری‌پای

قبل از اینکه هرگونه نرم‌افزاری بنویسید، لطفاً مطمئن شوید که پای شما به کمک sudo apt-get update و sudo apt-get upgrade به‌روز باشد.

```
sudo apt-get update && sudo apt-get upgrade
```

مثل همیشه در اینجا پیشنهاد می‌کنم که با استفاده از یک پای تازه پاک‌شده با کمک جدیدترین ورژن Raspbian Jessie شروع کنید تا مطمئن باشید که همه نرم‌افزارهای موردنیاز را دارید.

اولین مثال پایتون یک موتور 48srp یا 48 پله در هر دور را یک‌بار در جهت عقربه‌های ساعت می‌چرخاند و سپس با استفاده از کتابخانه RPi.GPIO به عقب می‌چرخاند. پین‌های DIR و STEP به‌عنوان خروجی تنظیم‌شده‌اند. پین DIR برای حالت ساعت‌گرد یک فرض شده است. پس یک حلقه for تا 48 شمارش می‌کند. در هر حلقه پین STEP برای 0.208 ثانیه یک و برای 0.208 ثانیه صفر می‌شود. سپس پین DIR برای حالت پادساعت‌گرد صفر شده و حلقه for دوباره تکرار می‌شود.

```
from time import sleep
import RPi.GPIO as GPIO

DIR = 20   # Direction GPIO Pin
STEP = 21  # Step GPIO Pin
CW = 1     # Clockwise Rotation
CCW = 0    # Counterclockwise Rotation
SPR = 48   # Steps per Revolution (360 / 7.5)

GPIO.setmode(GPIO.BCM)
GPIO.setup(DIR, GPIO.OUT)
GPIO.setup(STEP, GPIO.OUT)
GPIO.output(DIR, CW)

step_count = SPR
```

 |

```
delay = .0208

for x in range(step_count):
    GPIO.output(STEP, GPIO.HIGH)
    sleep(delay)
    GPIO.output(STEP, GPIO.LOW)
    sleep(delay)

sleep(.5)
GPIO.output(DIR, CCW)
for x in range(step_count):
    GPIO.output(STEP, GPIO.HIGH)
    sleep(delay)
    GPIO.output(STEP, GPIO.LOW)
    sleep(delay)

GPIO.cleanup()
```

این کد ممکن است مخصوصاً در سرعت‌های پایین موجب لرزش موتور و حرکات نامنظم شود. یکی از راه‌ها برای مقابله با این نتایج میکرو استپینگ (microstepping) است. قطعه کد زیر به کد بالا اضافه می‌شود. پین‌های مد GPIO به‌عنوان خروجی تنظیم می‌شود. یک dict مقادیر مناسب را برای هر فرمت مرحله‌ای نگه می‌دارد. GPIO خروجی به مد 1/32 تنظیم می‌شود. توجه کنید که شمارش گام‌ها در 32 ضرب می‌شود زیرا هر چرخش اکنون 32 برابر زمان می‌برد که درنتیجه حرکت موتور نرم‌تر می‌شود. تأخیر به 32 تقسیم می‌شود تا گام‌های اضافی را جبران کند.

```
MODE = (14, 15, 18)   # Microstep Resolution GPIO Pins
GPIO.setup(MODE, GPIO.OUT)
RESOLUTION = {'Full': (0, 0, 0),
              'Half': (1, 0, 0),
              '1/4': (0, 1, 0),
              '1/8': (1, 1, 0),
              '1/16': (0, 0, 1),
              '1/32': (1, 0, 1)}
GPIO.output(MODE, RESOLUTION['1/32'])

step_count = SPR * 32
delay = .0208 / 32
```

لطفاً توجه داشته باشید که جنبه‌های منفی قابل‌توجهی برای میکرو استپینگ وجود دارد. اغلب، کمبود قابل‌ملاحظه‌ای در گشتاور وجود دارد که می‌تواند به فقدان دقت منجر شود.

ادامه این آموزش و ویدئوی راه اندازی موتور با رسپبری پای را در "آموزش موتورهای استپر با رسپبری‌پای (قسمت سوم)" دنبال کنید.

نظرات، پیشنهادات و انتقادات خود را براي بهتر شدن محتواي مطالب با ما در ميان بگذاريد...

ترجمه شده توسط تیم الکترونیک صنعت بازار | منبع: سایت rototron.info

