

Wprawianie rzeczy w ruch za pomocą Pythona

Sprawianie, że rzeczy się poruszają za pomocą Pythona, jest niezaprzeczalnie fajne. Dzięki silnikom obliczenia fizyczne wkraczają na zupełnie nowy poziom. Roboty, kuchenki mikrofalowe, lodówki i samochody elektryczne używają silników elektrycznych poruszają się, dmuchają powietrzem, pompują chłodziwo i zawożą Cię 60 mil na godzinę, gdziekolwiek chcesz. Silniki elektryczne są wszędzie! Mówiąc najprościej, silnik elektryczny to maszyna, która przekształca energię elektryczną w energię mechaniczną. W tym rozdziale mówimy o silnikach prądu stałego (DC). Prąd stały to pojedyncze stałe napięcie, takie jak 9 V, 5 V lub 3,3 V. Z drugiej strony prąd przemienny (AC) jest tym, co dostajesz z gniazdek domowych. Co ciekawe, silniki elektryczne zużywają ponad połowę energii elektrycznej produkowanej w Stanach Zjednoczonych.

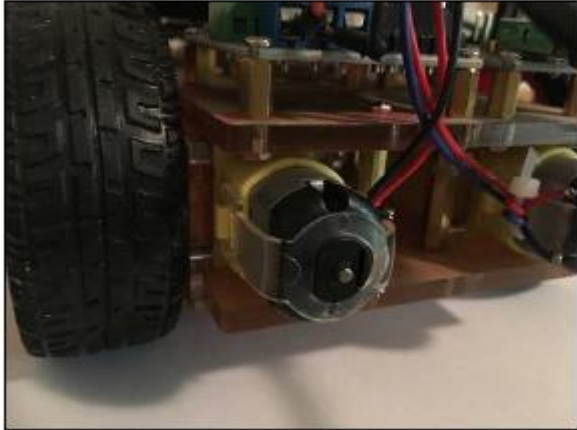
Odkrywanie silników elektrycznych

Silnik elektryczny opiera się na magnetyzmie. Wszystkie silniki wykorzystują magnesy do tworzenia ruchu. Wszystkie magnesy mają biegun północny i południowy. Północ na północ i południe na południe odpychają się, podczas gdy północ i południe się przyciągają. Sprytni ludzie wymyślili, jak wykorzystać ten fakt do stworzenia ruchu. Wszyscy znamy magnesy trwałe, takie jak te, których używasz do wieszania rzeczy z przodu lodówki. Można jednak również tworzyć magnesy, przepuszczając prąd wokół zwiniętego drutu, który wytwarza pole magnetyczne. Poprzez okresowe odwracanie prądu płynącego przez ten elektromagnes można wytworzyć siłę, która następnie zamienia się w ruch. Istnieje wiele sposobów budowania silników, ale jest to podstawowa podstawa wszystkich z nich. W tym rozdziale omówimy trzy popularne typy silników używanych w małych projektach i robotach. Oni są:

- Małe silniki prądu stałego
- Silniki serwo
- Silniki krokowe

Małe silniki prądu stałego

Silnik prądu stałego ma dwa przewody, zasilanie i masę. Po dostarczeniu zasilania (na przykład podając 5 V do linii zasilającej) silnik zacznie się obracać. Odwróć przewody zasilające i uziemiające, a silnik zacznie się obracać w przeciwnym kierunku. Kontrolujesz prędkość silnika prądu stałego za pomocą modulacji szerokości impulsu (PWM), technik służącej do kontrolowania jasności diody LED. Jeśli moc zostanie zmieniona na 50 procent (połowa włączona/połowa wyłączona), silnik będzie się obracał z połową prędkości. Te silniki prądu stałego są niedrogie i doskonale nadają się do napędzania kół. Czasami umieszcza się „enkoder” na wale silnika, więc odczytujesz komputerowi, jak daleko obrócił się wał, przekazując komputerowi informację zwrotną na ten temat może się przydać. Użyj silnika prądu stałego zawsze, gdy chcesz, aby coś obracało się z prędkością RPM (obroty na minutę), na przykład wentylatorem lub kołem samochodu.



Silniki serwo

Serwosilniki są ogólnie kombinacją trzech rzeczy: silnika prądu stałego, prostego obwodu sterującego i zestawu przekładni. Czasami można znaleźć potencjometr (który jest rezystorem zmiennym), który daje sprzężenie zwrotne położenia, podobnie jak „enkoder” w omówionym wcześniej silniku prądu stałego. Serwosilnik otrzymuje polecenie przejścia do określonej pozycji używając ponownie naszego przyjaciela PWM. Jednak w tym przypadku określona fala impulsu utrzyma silnik w określonej pozycji i będzie przeciwstawiać się obciążeniu lub sile próbującej poruszyć silnik. Maksymalna siła, jaką serwosilnik może wywierać na zewnętrzną siłę lub obciążenie, nazywana jest momentem znamionowym tego serwosilnika. Serwa są zasilane w sposób ciągły i generalnie mają tylko około 180-stopniowy zakres ruchu. Klasycznym miejscem, w którym można znaleźć serwomotory, są modele samolotów RC i niektóre rodzaje ramion robotów. Gdy chcesz przesunąć obiekt i przytrzymać go w określonej pozycji, często odpowiedzią jest serwomotor. Użyj serwomotoru, aby uzyskać szybki, wysoki moment obrotowy i dość dokładny obrót do określonej pozycji z ograniczonym zakresem stopni. Dobre zastosowania obejmują sterowanie sterem w łodzi RC lub kłapami w samolocie RC, a także ramiona robotów.

Silniki krokowe

Silnik krokowy jest trochę jak serwosilnik, który wykorzystuje inny sposób poruszania wałem. Podczas gdy serwosilnik wykorzystuje silnik prądu stałego, silnik krokowy wykorzystuje wielozębne elektromagnesy otaczające centralny wał zębaty. Silniki krokowe używają zewnętrznego kontrolera (do tego celu użyjesz Raspberry Pi), który sekwencjonuje elektromagnesy otaczające ten centralny wał, aby centralny wał obracał się „krokowo”, stąd nazwa silnika krokowego. Konstrukcja silnika krokowego zapewnia stały moment trzymania, nawet gdy nie jest zasilany. Porównaj to z serwomotorem, który musi być zasilany, aby zapewnić moment obrotowy. Dopóki obciążenie mieści się w granicach momentu obrotowego silnika serwo, nie ma błędów pozycjonowania. Silniki krokowe służą do powolnych, precyzyjnych obrotów. W tego typu zastosowaniach są lepsze od serwomotorów. Drukarki 3D są doskonałym przykładem wykorzystania silników krokowych. Silniki krokowe nie wymagają

również systemu sprzężenia zwrotnego do określania, gdzie są ustawione, tak jak robią to serwosilniki.



Sterowanie silnikami za pomocą komputera

Teraz możemy trochę pobawić się z Pythonem w sterowaniu silnikami. Następnie przejdziemy przez wszystkie trzy typy silników i pokażemy, jak sterować nimi za pomocą Pythona. Dowiesz się, jak sterować silnikami zarówno przez piny GPIO (wejście ogólnego przeznaczenia), jak i przez kontroler I2C.

Python i silniki prądu stałego

Istnieje wiele różnych sposobów napędzania silników prądu stałego z Raspberry Pi. Istnieją dziesiątki kontrolerów robotów i płyt kontrolerów silników, które będą działać w tym projekcie. Ta, której będziemy używać, to płytka sterowana I2C (powiązana z naszymi projektami z poprzedniego rozdziału), która daje nam kontrolę nad dwoma silnikami, ich indywidualnym kierunkiem i indywidualną prędkością. Całkiem fajne.

Oto lista części:

- * Płyta interfejsu Pi2Grover Grove: Wypróbuj store.switchdoc.com lub [Amazon.com](https://www.amazon.com).
- * Napęd silnikowy Grove I2C: dostępny na stronie www.seeedstudio.com lub [Amazon.com](https://www.amazon.com) (dostarczany z kablem Grove).
- * Dwa małe silniki prądu stałego: Wypróbuj <https://www.adafruit.com/product/711> lub [Amazon.com](https://www.amazon.com).

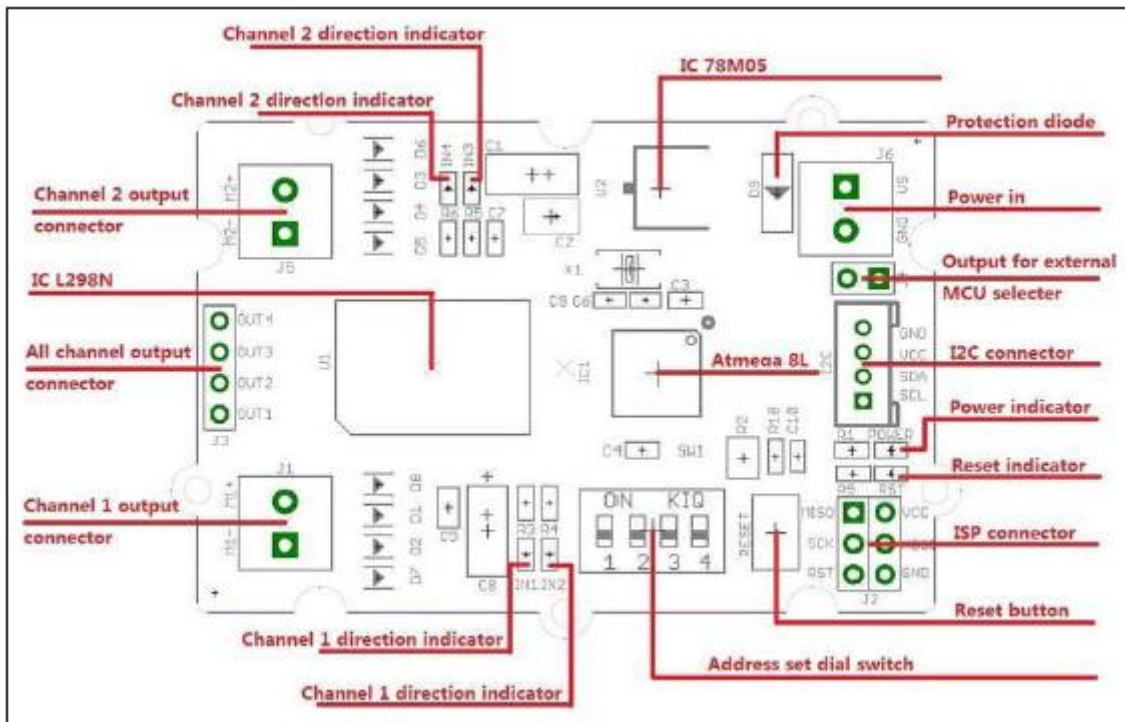
Więcej informacji na temat płytki Pi2Grover można znaleźć w rozdziałach 1–3 tego minibooka. Poświęćmy trochę czasu napędowi silnikowemu Grove I2C, ponieważ ma on dość unikalny sposób działania.

Napęd silnikowy Grove I2C

Napęd silnikowy Grove I2C (patrz rysunek 4.3) jest w stanie napędzać jednocześnie dwa silniki, wszystkie sterowane przez naszego starego znajomego, magistralę I2C z Raspberry Pi.

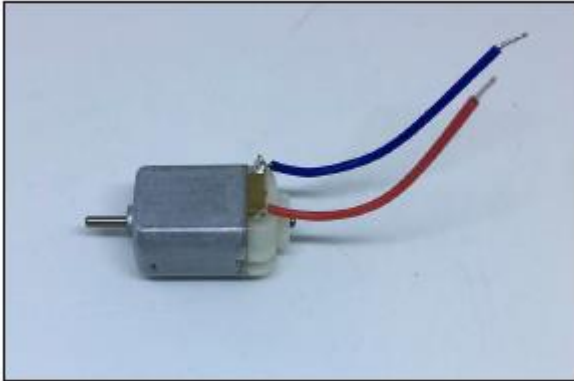


Może napędzać do 2 A dla każdego silnika, ale używamy silników o wiele mniejszych. Opcjonalnie może obsługiwać silniki od 6 V do 15 V, ale znowu używamy małych silników, więc użyjemy tylko zasilacza Raspberry Pi. Jeśli używasz większych silników lub chcesz użyć zewnętrznego źródła zasilania, ta płyta może to zrobić za Ciebie. Powodem, dla którego używamy tej płyty, jest to, że wykorzystuje interfejs I2C do sterowania silnikami, co pozwala nam szybko rozpocząć pracę. Rysunek 4-4 pokazuje, co znajduje się na płycie napędu silnika I2C.

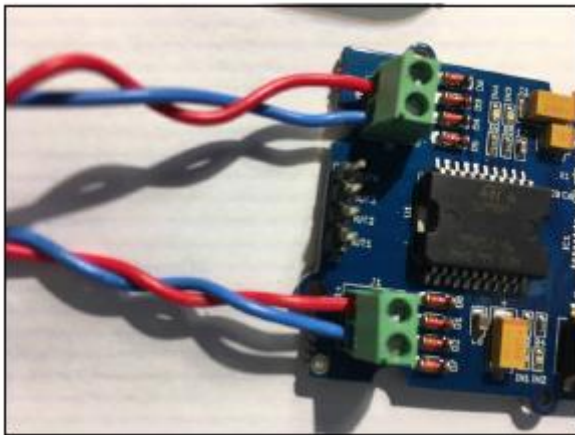


Na tym diagramie jest kilka interesujących rzeczy: Po pierwsze, na tej tablicy jest inny komputer! Jest to Atmega 8L i jest kolejnym małym komputerem, który faktycznie emuluje interfejs I2C, przetwarza polecenia przychodzące z twojego Raspberry Pi, a następnie steruje silnikami. Jeszcze jeden przykład tego, jak nawet płytki do małych komputerów mają na sobie małe komputery. Komputery są wszędzie! Możesz zobaczyć dwa połączenia silnika po lewej stronie płytki, a także, co oznaczają diody LED na środku płytki. Oprogramowanie do komputera pokładowego w napędzie silnikowym Grove I2C można również znaleźć na stronie produktu www.seeedstudio.com. Możesz zmienić programowanie, jeśli chcesz lub przynajmniej zrozumieć, jak sprawić, by mały komputer wyglądał jak urządzenie I2C. Podłączmy go i uruchommy silniki! Wyłącz zasilanie Raspberry Pi przed podłączeniem:

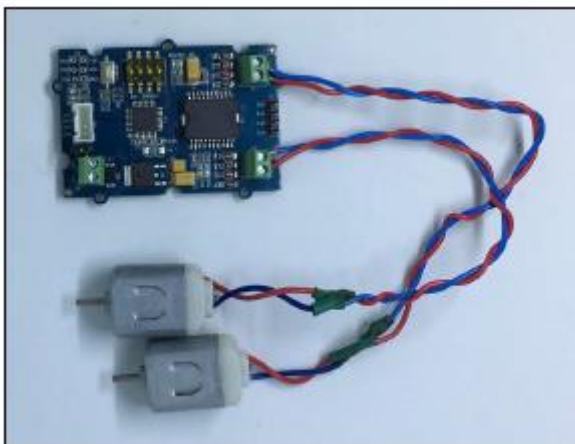
1. Poluzuj zestaw dwóch zacisków śrubowych na końcu i włóż odstłonięte końce przewodów na silnikach



do otworów i dokręć śruby.



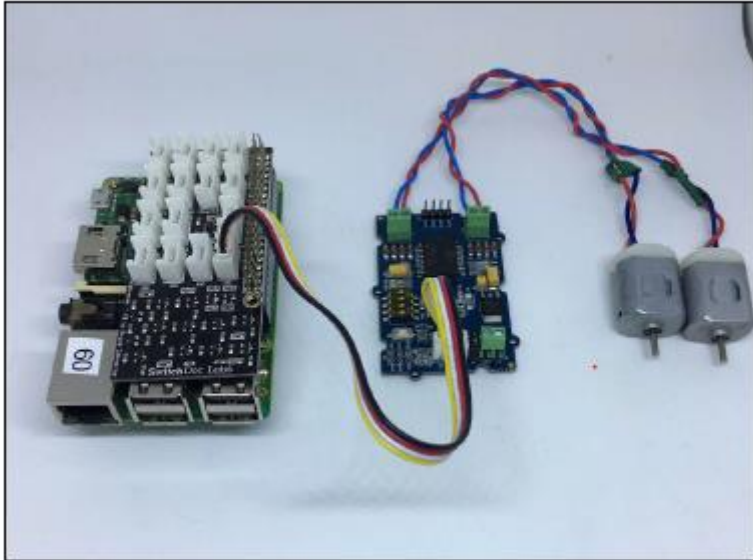
Zwróć uwagę, że tak naprawdę nie ma znaczenia, który kolor pasuje do którego otworu w przypadku silnika prądu stałego. Po prostu będzie się obracać w przeciwnym kierunku. Po prostu dopasuj je oba, jak na rysunku powyżej. Dodaliśmy trochę długości do przewodów, ale jest to opcjonalne. Rysunek przedstawia silnik zainstalowany na płycie napędu silnika Grove I2C.



2. Podłącz kabel Grove do złącza Grove na płycie napędu silnika Grove I2, a następnie do złącza I2C Grove na płycie Pi2Grover. Upewnij się, że zworka jest założona na płytce, (oznaczona jako „Wyjście

dla zewnętrznego selektora MCU” na schemacie). Tak jest w paczce, ale upewnij się, że nadal jest na swoim miejscu. Płyta nie będzie działać jak okablowana, jeśli nie masz zainstalowanej tej zworki.

Jeśli Twoja płyta i silnik nie reagują po podłączeniu, spróbuj nacisnąć przycisk Reset na płycie. Rysunek przedstawia zmontowany silnik i płytkę.



Teraz włączmy Raspberry Pi i zacznijmy pisać trochę w Pythonie! Po włączeniu zasilania powinieneś zobaczyć wszystkie pięć diod LED na płycie napędu silnika Grove I2C. Jeśli nie, wyłącz ponownie Pi i sprawdź okablowanie.

Oprogramowanie silnika prądu stałego w języku Python

Silniki prądu stałego są często używane w kołach robotów, więc słowa do przodu i do tyłu powinny dać ci kilka pomysłów na później w książce, kiedy będziemy budować samochód robota. Oprogramowanie, którego użyjemy do eksperymentu z silnikiem prądu stałego w Pythonie, powinno wydawać się znajome, ponieważ jest bardzo podobne do oprogramowania I2C, o którym mówiliśmy w rozdziałach 2 i 3. -Wykorzystanie bibliotek Pythona jest kluczem do produktywnego pisania- Aplikacje Pythona. Będziemy używać `SDL_Pi_HDC1080_Python3` dostępnego na [Githu-b.com](https://github.com). Aby skonfigurować oprogramowanie, wykonaj następujące kroki:

1. Utwórz katalog w katalogu głównym, wprowadzając:

```
cd
```

```
mkdir dcMotor
```

```
cd dcMotor
```

Teraz jesteś w katalogu `dcMotor`.

2. Zanim spojrzysz na kod Pythona do uruchamiania silników, zainstaluj bibliotekę na Raspberry Pi. Robisz to, „klonując” bibliotekę znajdującą się na github.com, używając następującego polecenia w oknie terminala:

```
klon git https://github.com/switchdoclabs/SDL_Pi_GroveI2CMotorDrive.git
```


Klon git klonuje repozytorium git znajdujące się pod adresem i kopiuje je do twojego Raspberry Pi. Jeśli wpiszesz ls w oknie terminala, zobaczysz następujące dane wyjściowe:

```
pi@RPi3-60:~/dcMotor $ ls
```

```
SDL_Pi_GroveI2CMotorDrive
```

```
pi@RPi3-60:~/I2CTemperature $
```

3. Za pomocą nano (lub ulubionego edytora tekstu) otwórz plik o nazwie dcmotorTest.py i wprowadź następujący kod:

```
import sys

sys.path.append("./SDL_Pi_GroveI2CMotorDrive")

import SDL_Pi_GroveI2CMotorDrive

import time

#"0b1010" defines the output polarity
#"10" means the M is "positive" while the M- is "negative"

MOTOR_FORWARD = 0b1010
MOTOR_BACKWARD = 0b0101

try:
m= SDL_Pi_GroveI2CMotorDrive.motor_drive()

#FORWARD

print("Forward")

#defines the speed of motor 1 and motor 2;)
m.MotorSpeedSetAB(100,100)
m.MotorDirectionSet(MOTOR_FORWARD)

time.sleep(2)

#BACK

print("Back")

m.MotorSpeedSetAB(100,100)

#0b0101 Rotating in the opposite direction
m.MotorDirectionSet(MOTOR_BACKWARD)

time.sleep(2)

#STOP

print("Stop")

m.MotorSpeedSetAB(0,0)
```

```

time.sleep(1)

#Increase speed

for i in range (100):

print("Speed:",i)

m.MotorSpeedSetAB(i,i)

time.sleep(.02)

print("Stop")

m.MotorSpeedSetAB(0,0)

except IOError:

print("Unable to find the I2C motor drive")

print("Hit Reset Button on I2C Motor Drive and Try Again")

```

Ten program uruchamia oba silniki do przodu z pełną prędkością (100), następnie do tyłu z pełną prędkością, zatrzymuje silniki, a następnie uruchamia je do tyłu z małą prędkością, zwiększając do pełnej prędkości, a następnie całkowicie zatrzymuje silniki. Kluczowymi aspektami tego oprogramowania są wywołania biblioteki `SDL_Pi_GroveI2CMotorDrive`. Biblioteka obsługuje następujące funkcje:

`MotorSpeedSetAB(MotorSpeedA, MotorSpeedB)`: Prędkość silnika dla silnika A (M1) i silnika B (M2). Zakres 0–100.

`MotorDirectionSet(Direction)`: Do przodu lub do tyłu — stałe ustawione w programie. `SILNIK_DO PRZODU = 0b1010`, `SILNIK_WSTECZ = 0b0101`

Interesującym elementem biblioteki `SDL_Pi_GroveI2CMotorDrive` jest to, że wykorzystuje ona jedną z wielu dostępnych bibliotek I2C, `smbus`. W bibliotece wysyłasz polecenie do płyty I2C jako zapis bloku składający się z adresu I2C, bajtu polecenia, a następnie argumentów. Oto wezwanie do ustawienia kierunku silnika:

```

#Set motor direction

def MotorDirectionSet(self,Direction):

bus.write_i2c_block_data(self.I2CMotorDriveAdd, self.DirectionSet,

[Direction,0])

time.sleep(.02)

```

Czas uruchomić silniki prądu stałego. Wpisz to w oknie terminala:

```
sudo python3 testMotor.py
```

Zostaniesz nagrodzony widząc zmieniające się diody LED i widząc, jak twoje silniki przechodzą przez sekwencję zaprogramowaną w Pythonie. Powinieneś być w stanie bardzo łatwo tworzyć własne sekwencje z tego przykładu. Teraz przejdź do silnika Servo. Wszystkie te silniki pobierają zasilanie z Raspberry Pi podczas pracy, więc odłącz silniki prądu stałego, gdy będziesz gotowy, aby przejść do następnej sekcji. Najpierw wyłącz Pi!

Python i uruchamianie serwomotoru

Silniki serwo to inna bestia niż silnik prądu stałego. Serwosilnik ma wewnątrz silnik prądu stałego, ale ma również obwód kontrolny, który pozwala nam ustawić silnik prądu stałego w określonej pozycji, a następnie przytrzymać go tam. Świetne zastosowanie serwosilnika znajduje się na ramieniu robota, w którym chcesz się uzbroić, aby przejść do określonej pozycji i tam czekać na dalsze rozkazy. Sterujesz serwomotorami za pomocą PWM (modulacja szerokości impulsu). Chociaż możesz kupić płytki, które będą wykonywać PWM (i obsługiwać większe serwomotory!) RaspberryPi. Oto lista części:

* Płyta interfejsu Pi2Grover Grove: poszukaj jej na stronie store.switchdoc.com lub amazon.com.

* Mikrosilnik serwo SG90: Spróbuj znaleźć go na ebay.com lub amazon.com. Są niedrogie, więc możesz kupić dwa lub więcej za mniej niż 10 USD.

* Pakiet męskich kabli połączeniowych Grove: Przewody łączące złącze Grove z męskim stykiem: dostępne na stronie store.switchdoc.com lub amazon.com.

Pi2Grover został opisany wcześniej w rozdziałach 1, 2 i 3 tego minibooka. Mikrosilnik serwo SG90



jest małym, niedrogim serwosilnikiem dostępnym z wielu źródeł. Ma napięcie robocze 3,0 V - 7,0 V przy poborze prądu około 40 mA (40 miliamperów - miliamper to 1/1000 ampera prądu) maksymalnie, więc 5 V na Raspberry Pi jest dobre do obsługi tego serwomechanizmu . Może obracać się o około 90 stopni w każdym kierunku, w sumie o 180 stopni. W większości serwomotorów są trzy przewody sterujące i SG90 nie jest wyjątkiem. Te trzy przewody to:

Żółty – sygnał sterujący PWM

Czerwony – Zasilanie (5 V, w naszym przypadku)

Brązowy – ziemia

Kable krosowe Grove-męski-pin-to-Grove-connector (patrz rysunek 4-10) służą do połączenia trzech pinów serwomechanizmu ze złączem Grove, a następnie można je podłączyć do Raspberry Pi.



ILE PRĄDU MOŻE DOSTARCZYĆ RASPBERRY PI DO KOŁKÓW 5V?

Niestety, nie ma prostej odpowiedzi na to pytanie, ponieważ zależy to od tego, co podłączyłeś do Raspberry Pi 3 i jakiego rodzaju zasilacza USB używasz. 250 mA to całkiem niezła liczba dla ogólnej zasady, ale jeśli masz mocny zasilacz USB 5 V (powiedzmy 2,5 A), możesz pójść znacznie wyżej, do 1000 mA lub więcej. Raspberry Pi 3 ma bezpiecznik 2,5 A na zasilaczu 5 V. Jest to resetowalny bezpiecznik, więc jeśli go wyskoczysz, po prostu pozwól mu ostygnąć i znów zadziała. Bardzo dobrze, jeśli popełnisz błąd.

A teraz podłączmy przewody i rozbujamy serwomotor!

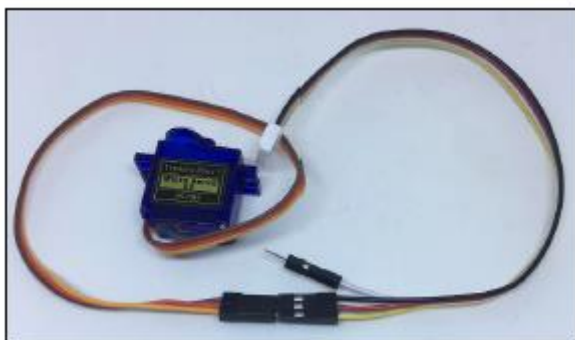
1. Wyłącz Raspberry Pi i odłącz zasilanie.
2. Podłącz kabel połączeniowy Grove do serwosilnika SB90, postępując zgodnie ze schematem połączeń w tabeli 1. (Patrz rysunek 4-11.)

Serwo SG90: Patch Grove: Funkcja kabla

Żółty: przewód Żółty przewód: sygnał PWM

Czerwony: przewód Czerwony przewód: zasilanie

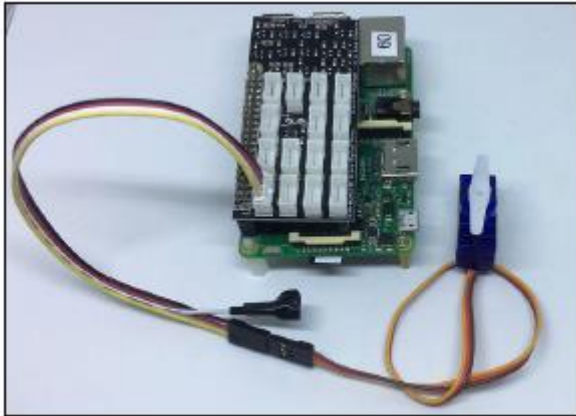
Brązowy: przewód Czarny przewód: uziemienie



Sprawdź dokładnie okablowanie. Możesz uszkodzić swoje Pi i silnik, jeśli odwrócisz te przewody.

3. Podłącz koniec kabla Grove do złącza Pi2Grover Grove oznaczonego jako D4/D5.
4. Umieść kawałek taśmy elektrycznej lub niebieskiej taśmy na białym odsoniętym styku kabla krosowego Grove, aby zapobiec zwarciu. Umieść również jeden z dostarczonych wahaczy na przekładni

serwomotoru, abyśmy mogli łatwiej zobaczyć jego zakres ruchu. Przyjrzyjmy się teraz oprogramowaniu Pythona.



Oprogramowanie serwomechanizmu Python

Nie zamierzamy używać biblioteki serwo wyższego poziomu (a jest ich wiele dostępnych dla Raspberry Pi w Pythonie); zamiast tego pokażemy, jak bezpośrednio sterować serwomotorem za pomocą GPIO i funkcji PWM wbudowanej biblioteki RPi GPIO. Dobrze, dobrze. Używamy biblioteki (RPi.GPIO), ale nie dodajemy warstw wywołań API (interfejsu programowania aplikacji), jak zwykle. Schodzimy i brudzimy się pinami GPIO. Aby to zrobić, wykonaj następujące kroki:

1. Utwórz katalog w katalogu głównym, wprowadzając:

```
cd
mkdir Servo
cd Servo
```

2. Używając nano (lub swojego ulubionego edytora tekstu), otwórz plik o nazwie servoTest.py i wprowadź następujący kod Pythona:

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
ServoPin = 4
GPIO.setup(ServoPin, GPIO.OUT)
p = GPIO.PWM(ServoPin, 50)
p.start(7.5)
try:
while True:
p.ChangeDutyCycle(7.5) # turn towards 90 degree
print ("90 degrees")
```

```

time.sleep(1) # sleep 1 second
print ("0 degrees")
p.ChangeDutyCycle(2.5) # turn towards 0 degree
time.sleep(1) # sleep 1 second
print ("180 degrees")
p.ChangeDutyCycle(12.5) # turn towards 180 degree
time.sleep(1) # sleep 1 second
except KeyboardInterrupt:
p.stop()
GPIO.cleanup()

```

Łamanie kodu

W tym miejscu ustawiamy numer pinu do złącza D4/5 Grove na płycie Pi2Grover:

```

import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
ServoPin = 4

```

Następnie ustawiamy pin GPIO na „Output”:

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

To polecenie ustawia obiekt p na ServoPin (4) i na częstotliwość 50 Hz. 50 Hz to dobra liczba dla tego typu serwowymotoru:

```
p = GPIO.PWM(ServoPin, 50)
```

Poniższy wiersz uruchamia serwowymotor przy 7,5% cyklu pracy. Pamiętaj, jak PWM działa na serwomechanizmie? Przechodzi od jednego końca obrotu serwomechanizmu do drugiego, przechodząc od zgodnie z cyklem roboczym. 7,5 oraz liczby 2,5 i 12,5 są określone przez typ serwa. Po prostu spojrzeliśmy na liczby i empirycznie ustaliliśmy te liczby dla serwomechanizmu SG90. Są one podobne dla większości małych serwomechanizmów:

```
p.start(7.5)
```

Teraz przesuwamy serwo w całym jego zakresie:

```
try:
```

```
while True:
```

```
p.ChangeDutyCycle(7.5) # turn towards 90 degree
```

```
print ("90 degrees")
```

```
time.sleep(1) # sleep 1 second
```

```
print ("0 degrees")
p.ChangeDutyCycle(2.5) # turn towards 0 degree
time.sleep(1) # sleep 1 second
print ("180 degrees")
p.ChangeDutyCycle(12.5) # turn towards 180 degree
time.sleep(1) # sleep 1 second
```

Zwróć uwagę, jak umieściliśmy całe ciało tego programu w try: oprócz: klauzula? Zrobiliśmy to tak, że kiedy twoje sterowanie-c wyjdzie z programu, program wyłączy serwomotor i wykona pewne czyszczenie GPIO (zwolnij wszystkie piny z powrotem do systemu operacyjnego). Jest to ogólnie dobra rzecz do zrobienia, gdy masz do czynienia bezpośrednio z pinami GPIO.

except KeyboardInterrupt:

```
p.stop()
GPIO.cleanup()
```

Teraz czas na uruchomienie programu. Wpisz następujące polecenie w oknie terminala

```
sudo python3 servoTest.py
```

Powinieneś zostać nagrodzony przez wydrukowanie na ekranie następujących zastawów i serwomechanizm radośnie wykonujący zaprogramowane polecenia. Zmieniaj rzeczy i wypróbuj różne kąty i sekwencje. Nie możesz uszkodzić serwomechanizmu, próbując różnych rzeczy.

90 degrees

0 degrees

180 degrees

90 degrees

0 degrees

180 degrees

90 degrees

0 degrees

Teraz mamy serwsilnik działający na naszym Raspberry Pi. Pamiętaj, jak na początku mówiliśmy, że serwsilnik może poruszać ramieniem robota, klapą w samolocie RC lub sterem w zdalnie sterowanej łodzi? Obserwowanie, jak serwsilnik przechodzi przez zaprogramowaną sekwencję, powinno zainspirować Cię do zastanowienia się, co zrobić z serwomotorem. Możesz zobaczyć, dlaczego używasz silnika prądu stałego do kół i serwomechanizmu do robienia rzeczy w sposób nieobrotowy. Eksperymentuj i buduj własne magiczne projekty! Teraz przejdziemy od razu do naszego ostatniego głównego silnika, silnika krokowego. Python i krok krokowy silnika krokowego Silniki krokowe to jednak inna bestia niż silniki prądu stałego czy serwomotory. Służą do dokładnego pozycjonowania przedmiotów z interfejsem cyfrowym. Możesz dokładnie pozycjonować rzeczy również za pomocą serwomotoru, ale wymaga to więcej elektroniki i zdecydowanie potrzebuje tak zwanego sprzężenia

zwrotnego pozycji. Silnik krokowy radzi sobie z tym, dokładnie przechodząc z jednego „kroku” do drugiego pod kontrolą oprogramowania. Silnik jest skonstruowany tak, aby wykorzystywać dwie cewki silnika do przesuwania silnika o jeden krok poprzez wysyłanie określonej sekwencji do silników. Będziesz wdrażał tę sekwencję „krokową” w oprogramowaniu Pythona sterującym silnikiem krokowym. Silnik krokowy ma zwykle dwie cewki używane do przesuwania silnika z jednego kroku do drugiego .

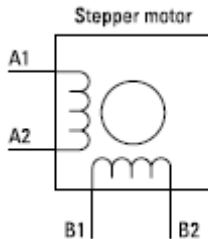


Tabela 2 pokazuje sekwencję krokową silnika krokowego o jeden krok do przodu, a Tabela 3 pokazuje sekwencję krokową do przesunięcia silnika krokowego o jeden krok do tyłu. Ten wzorzec kroków będzie bardzo oczywisty w naszym oprogramowaniu Python.

Stepper do przodu

Cewka_A_1 (pin 12) : Cewka_A_2 (pin 20) : Cewka_B_1 (pin 13) : Cewka_B_2 (pin 21)

1 : 0 : 1 : 0

0 : 1 : 1 : 0

0 : 1 : 0 : 1

Stepper do tyłu

Cewka_A_1 (pin 12) : Cewka_A_2 (pin 20) : Cewka_B_1 (pin 13) : Cewka_B_2 (pin 21)

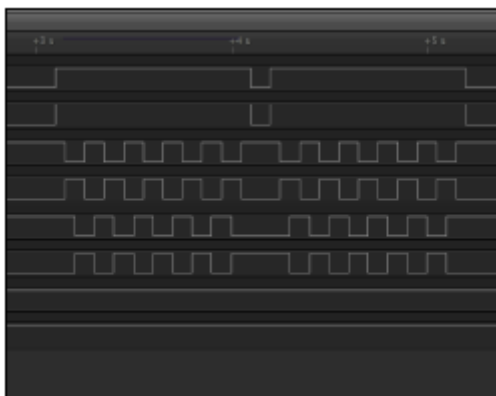
1 : 0 : 0 : 1

0 : 1 : 0 : 1

0 : 1 : 1 : 0

1 : 0 : 1 : 0

Na rysunku ta cyfrowa sekwencja jest przedstawiona graficznie z analizatora logicznego podłączonego do pinów Raspberry Pi GPIO używanych do napędzania silnika krokowego.



Cóż, teraz wiesz wszystko, co musisz wiedzieć o silnikach krokowych, aby zbudować swój pierwszy projekt.

INFORMACJA ZWROTNA: JAKA PRZYDATNA RZECZ!

Sprężenie zwrotne występuje, gdy kierujesz wyjście systemu z powrotem do wejść systemu jako pętlę. Brzmi skomplikowanie? Może być, ale podstawy są proste. Załóżmy, że publikujesz i publikujesz artykuły i prosisz o komentarze. Ludzie dostarczają komentarzy (miejmy nadzieję, że miłych), a ty zmieniasz artykuł na podstawie niektórych z tych komentarzy. To jest informacja zwrotna! Wykorzystujesz sprzężenie zwrotne w obwodach elektrycznych, aby na przykład uzyskać lepsze i dokładniejsze pozycjonowanie serwomotoru. Odczytując enkoder na wale serwomechanizmu (enkoder daje sygnał elektryczny proporcjonalny do położenia wału), oprogramowanie może dostosować cykl pracy PWM, aby uzyskać dokładniejszą pozycję w zależności od sprzężenia zwrotnego. Istnieją dwa rodzaje informacji zwrotnej: negatywna i pozytywna informacja zwrotna. W naszym powyższym przykładzie pozytywne opinie to dobre komentarze lub przynajmniej konstruktywna krytyka naszego artykułu, podczas gdy negatywne opinie to tylko komentarze mówiące nam, jak zły jest artykuł. Jednak w elektronice zwykle lubisz negatywne opinie i nie lubisz pozytywnych opinii, przynajmniej ogólnie. Negatywne sprzężenie zwrotne jest tym, czego używamy, aby na przykład zbliżyć się do pozycji na wale, w której chcemy się znaleźć. Pozytywne opinie zwykle powodują, że różnice stają się większe. Słyszałeś kiedyś, jak głośnik zawodzi, gdy mikrofon znajduje się zbyt blisko głośnika? To pozytywne opinie.

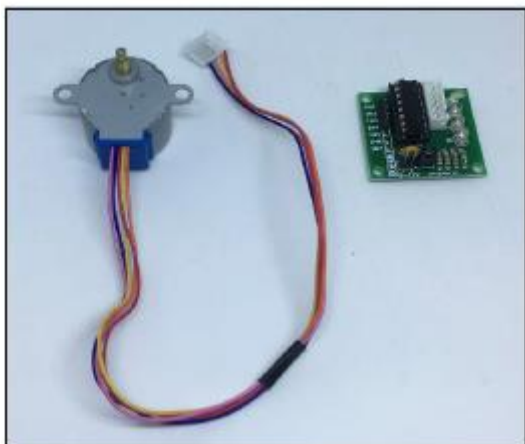
Oto lista części:

* Płyta interfejsu Pi2Grover Grove: Wypróbuj store.switchdoc.com lub amazon.com.

* Silnik krokowy 28BYJ-48 ULN2003 5 V: szukaj ich na eBay.com lub <https://amzn.to/2BuNDVI>. Są niedrogie, więc może się okazać, że kupisz je za 12 USD. Upewnij się, że masz te z płytami sterowników (takie jak te z poprzedniego łącza Amazon.com).

* Pakiet żeńskich kabli krosowych Grove, Grove-connector-to-femalepins: Dostępne na stronie store.switchdoc.com lub amazon.com.

Pi2Grover został opisany wcześniej w rozdziałach 1, 2 i 3 tego minibooka. Silnik krokowy 28BYJ-48 to silnik krokowy 5 V, który ma $5,625 \times 1/64$ stopnia na krok (około 0,822 stopnia na krok) i jest dostarczany z płytą sterownika z układem napędowym silnika ULN2003, a co najlepsze, cztery diody LED, które pokażą, co robisz z silnikiem ze swojego oprogramowania.



Kable połączeniowe żeńskie Grove służą do podłączenia płyty napędu silnika krokowego do Raspberry Pi.



Czas na zbudowanie projektu silnika krokowego! Wykonaj następujące kroki:

1. Wyłącz Raspberry Pi i odłącz zasilanie.
2. Weź żeński kabel krosowy Grove i podłącz go do płyty sterownika UNL2003, jak pokazano na schemacie połączeń w Tabeli 4-4.

Pierwszy żeński kabel krosowy Grove do płyty sterownika UNL2003

Kabel krosowy Grove: UNL2003 Płyta sterownicza: Funkcja

Żółty: Przewód IN1: A_1

Biały : Przewód IN2 : Cewka B_1

Czerwony : Przewód : Zasilanie

Czarny : przewód : uziemienie

Zwróć uwagę, że na kablu umieściliśmy opaskę drucianą, aby zachować porządek.

Przyjrzyj się uważnie czerwono-czarnemu przewodowi na kablu krosowym Grove, aby upewnić się, że jest podłączony.



3. Weź drugi żeński kabel krosowy Grove i podłącz go zgodnie ze schematem połączeń w Tabeli 4-5.

Drugi żeński kabel krosowy Grove do płyty sterownika UNL2003

Kabel krosowy Grove: UNL2003 Płyta sterownicza: Funkcja

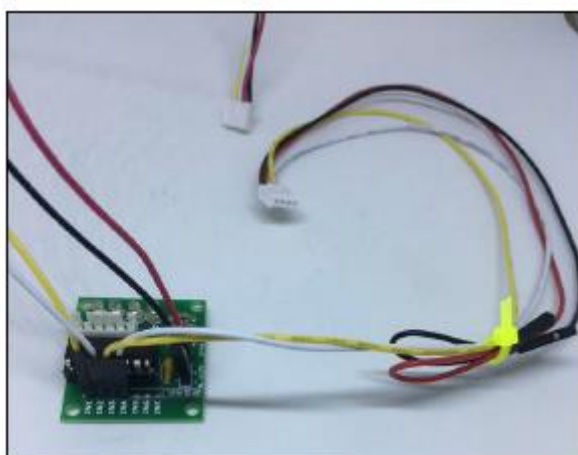
Żółty: Przewód IN3: Cewka A_2

Biały : Przewód IN4 : Cewka B_3

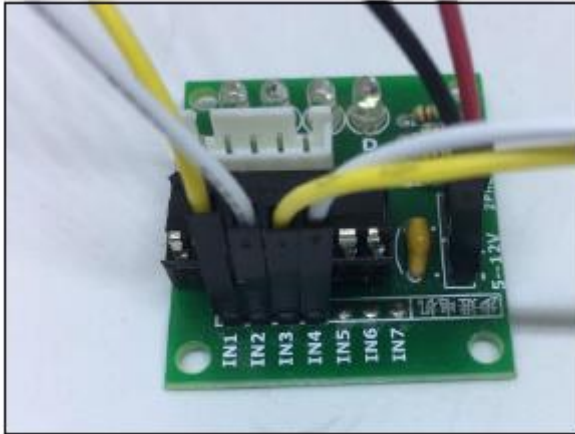
Czerwony: przewód: brak połączenia

Czarny: przewód: brak połączenia

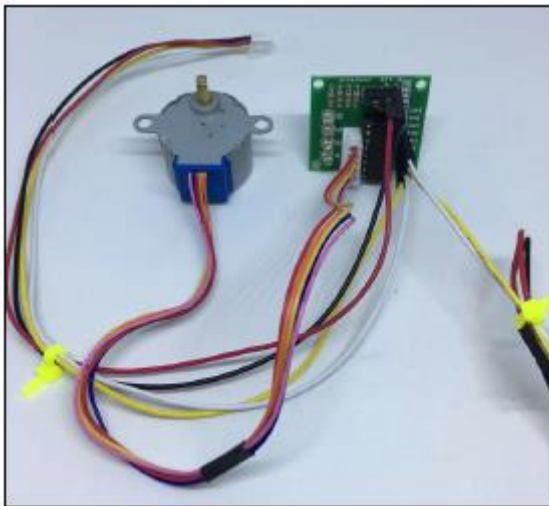
Użyj opaski zaciskowej lub kawałka taśmy, aby trzymać nieużywane czerwone i czarne przewody w górze i nie przeszkadzać.



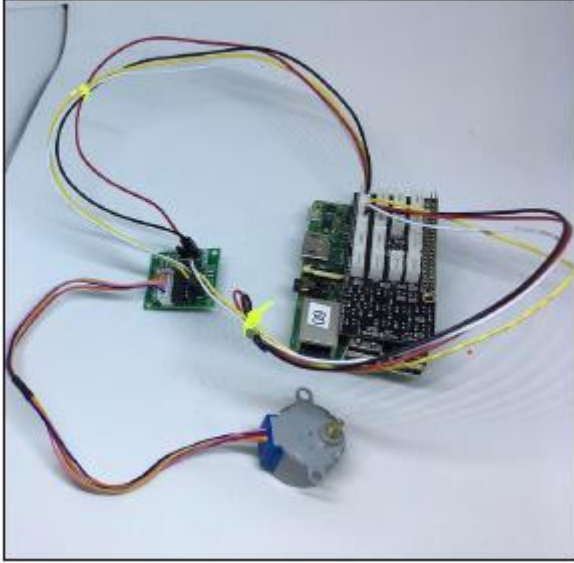
Teraz ponownie sprawdź wszystkie połączenia i upewnij się, że wyglądają tak.



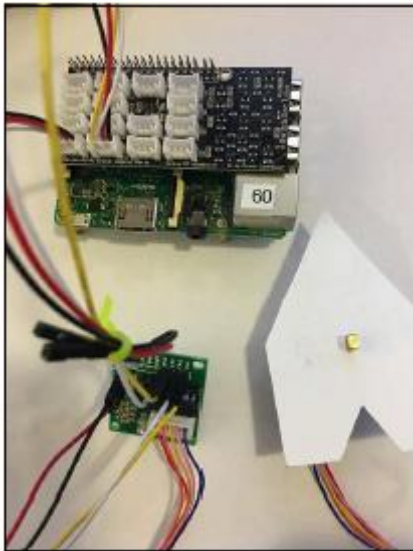
4. Podłącz kabel silnika krokowego 28BYJ-4 do złącza płyty sterownika UNL2003. Jest zamykany na klucz i działa tylko w jedną stronę.



5. Podłącz pierwszy kabel krosowy Grove (ten, który ma wszystkie cztery przewody podłączone do płyty sterownika UNL2003) do złącza Pi2Grove Grove oznaczonego D12/13 i drugi kabel krosowy Grove (ten, który ma tylko żółte i białe przewody podłączone) do złącza Pi2Grove Grove oznaczonego jako D20/21. Okablowanie zostało zakończone.



6. Umieść tekturową strzałkę na wale silnika krokowego, tak aby naprawdę można było zobaczyć, jak się porusza.



Oprogramowanie krokowe Pythona

Podobnie do tego, co zrobiliśmy z serwo silnikiem, nie będziemy używać biblioteki krokowej wyższego poziomu (a jest ich wiele dostępnych dla Raspberry Pi w Pythonie — nie wszystkie działają ze wszystkimi silnikami krokowymi!), Zamiast tego zamierzamy pokazać, jak bezpośrednio sterować silnikiem krokowym za pomocą pinów GPIO. Wykonaj następujące kroki:

1. Utwórz katalog w katalogu głównym, wprowadzając:

```
cd
```

```
mkdir Stepper
```

```
cd Servo
```

Używając nano (lub swojego ulubionego edytora tekstu), otwórz plik o nazwie stepperTest.py i wprowadź następujący kod Pythona:

```
import sys

import RPi.GPIO as GPIO

import time

GPIO.setmode(GPIO.BCM)

GPIO.setwarnings(False)

coil_A_1_pin = 12

coil_B_1_pin = 13

coil_A_2_pin = 20

coil_B_2_pin = 21

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)

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)
```

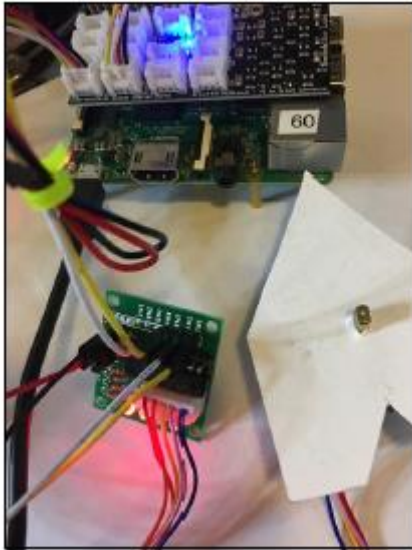
```

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:
try:
# Delay between steps (milliseconds)
delay = 10
# How many Steps forward
steps = 50
forward(int(delay) / 1000.0, int(steps))
# How many Steps backwards
steps = 50
backwards(int(delay) / 1000.0, int(steps))
except KeyboardInterrupt:
# shut off all coils
setStep(0,0,0,0)
sys.exit()

```

Łamanie kodu

Kod stepperTest.py jest dość prosty. Wyjścia GPIO zamieniamy na 1 i 0, zgodnie z sekwencjami silnika krokowego pokazanymi w tabelach 2 i 3. Dokładne sekwencje można zobaczyć w kodzie w funkcjach forward i reverse. Teraz uruchommy kod i zaczniemy odchodzić. Upewniamy się, że wyłączamy wszystkie wyjścia po naciśnięciu Ctrl-C w celu zatrzymania programu. Czas biegać! Włącz swoje Pi i otwórz okno terminala. Zwróć uwagę, że wszystkie cztery diody LED włączają się po włączeniu zasilania, ale zostaną wyłączone po uruchomieniu programu w języku Python.



Zobaczysz, jak silnik krokowy obraca się 50 kroków w lewo, a następnie 50 kroków w prawo. Spróbuj zmienić te zmienne w powyższym programie, aby przesunąć silnik krokowy w inne pozycje. Czy widzisz, jak te silniki można wykorzystać w drukarkach 3D i robotach do dokładnego pozycjonowania głowic drukujących, wysokości stołu i ramion robota?