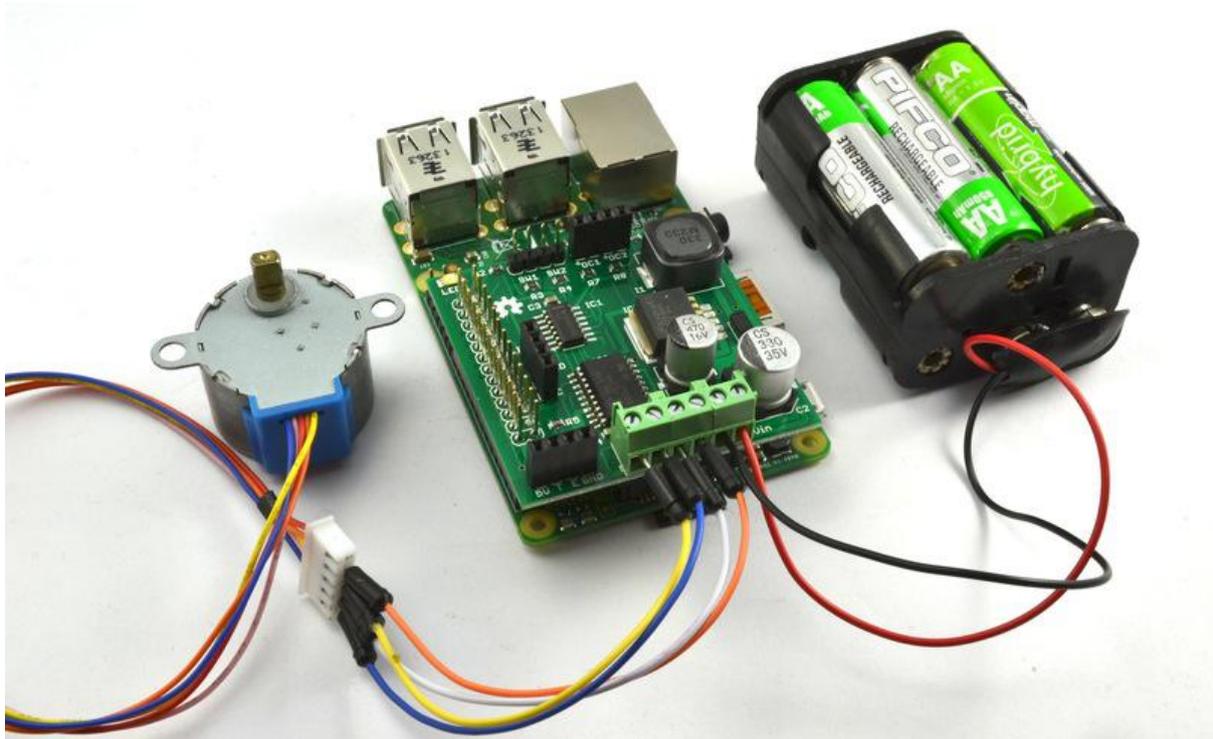




# Using a 5V Stepper Motor with the RasPiRobot Board V2

Created by Simon Monk



<https://learn.adafruit.com/using-a-5v-stepper-motor-with-the-raspirobot-board-v2>

Last updated on 2021-11-15 06:23:18 PM EST

# Table of Contents

Overview	3
Wiring	3
Software	5
Next Steps	7

---

# Overview

In this tutorial, you will learn how to drive a 5V stepper motor using a Raspberry Pi and the RasPiRobot Board V2 (RRBv2) expansion board for the Raspberry Pi.

A simple Python test program will allow you to control the motor. The code can then easily be incorporated into your projects.

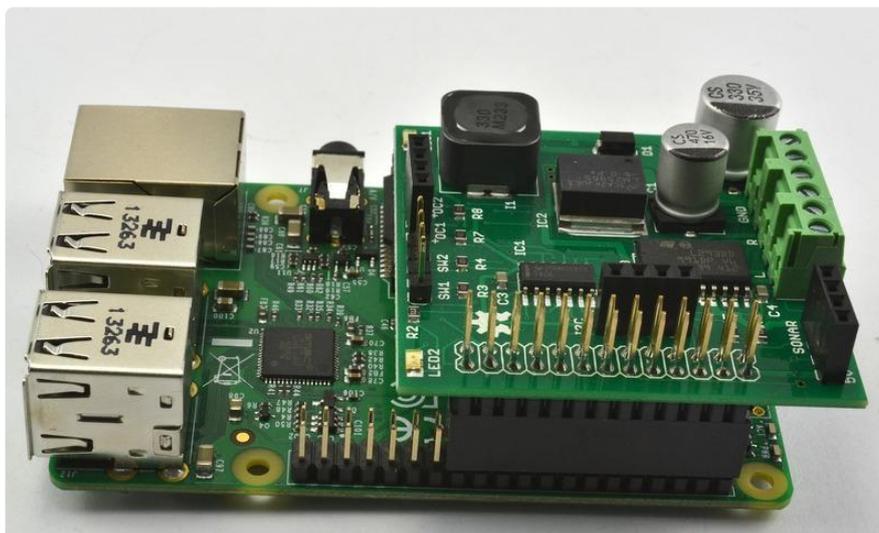


Note that the RRBv2 is not suitable for use with 12V stepper motors.

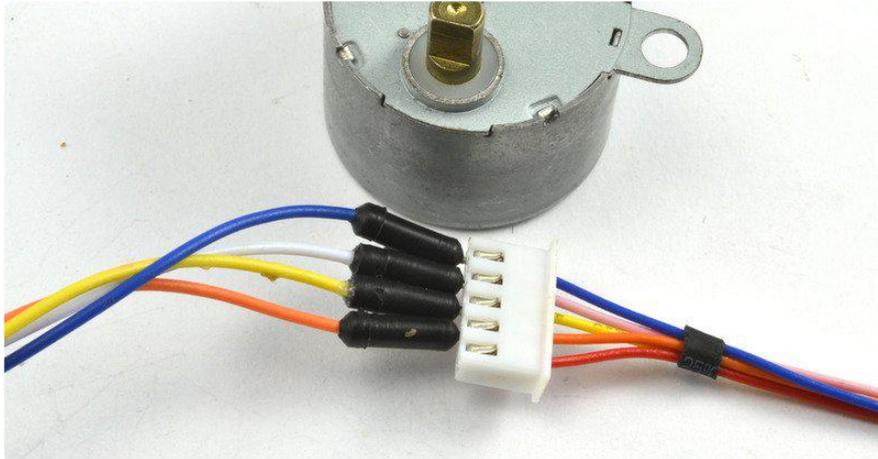
---

# Wiring

Fit the RRBv2 shield over the GPIO pins furthest from the USB ports (see the picture below).

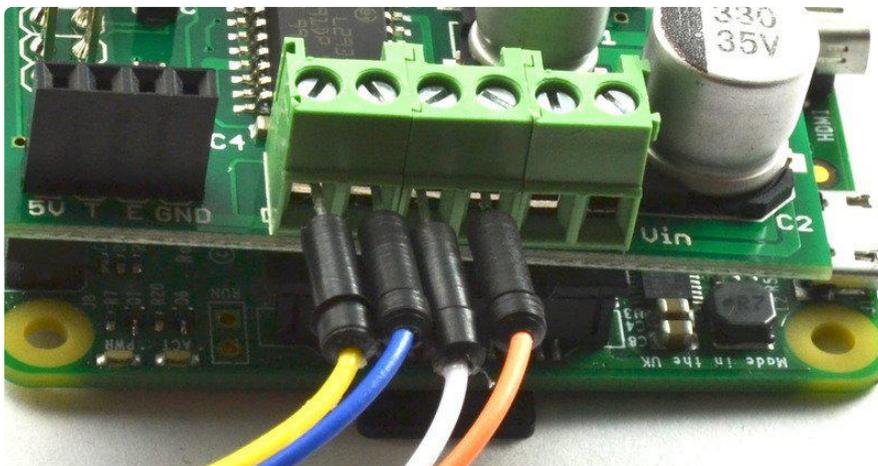


Use male to male jumper wires to connect from the 5 pin socket of the stepper motor to the RRBv2. Note that the red wire from the stepper is not used.



Try to use the same color of jumper lead as the wires from the motor. I didn't have a pink jumper lead, so I used white instead of pink.

Next connect the other end of the jumper wires to the L and R screw terminals of the RRBv2 as shown below. The order of connections from top to bottom is: orange, white (pink), blue and yellow.

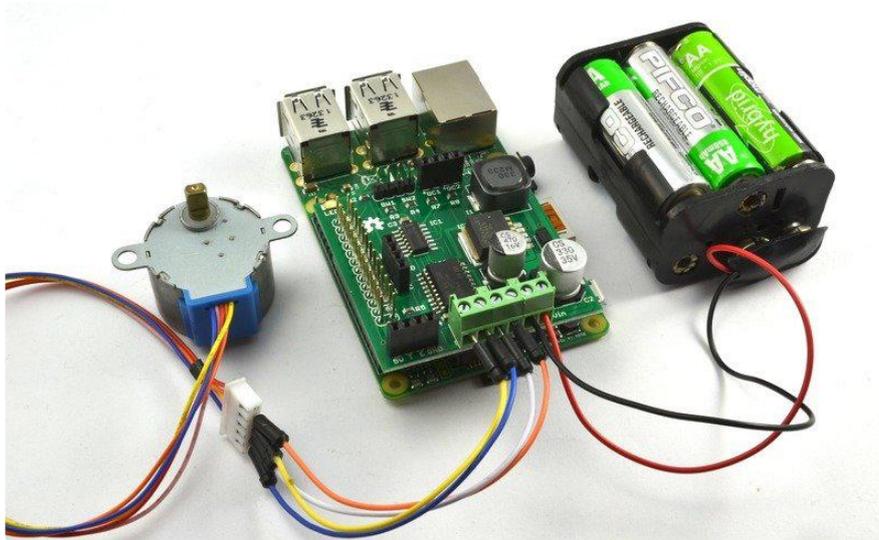


Note that the screw terminals are designed to connect well even to thin leads, so you will need to first unscrew them (counter clockwise) place the lead in the hole and then screw up the connector clockwise.

Connect the battery pack to the screw terminals of the RRBV2. If you are using a battery pack with flying leads then these can be attached into the screw terminals directly. If you have a battery pack with a 2.1mm plug on the end, then either use a screw terminal to 2.1mm socket adaptor like this and some short lengths of wire.

When attaching the battery make sure that the positive (red) lead goes to the right-most screw terminal marked Vin and the black lead goes to the screw terminal marked GND.

With the battery connected, the Raspberry Pi will start to boot, it will be supplied with power from the battery pack.



---

## Software

Your Pi will need to be connected to the Internet so that you can download the RRBV2 Python library. To download the library open LXTerminal (or use SSH) and type in the following commands:

```
$ wget https://github.com/simonmonk/raspirobotboard2/raw/master/python/dist/rrb2-1.1.tar.gz
$ tar -xzf rrb2-1.1.tar.gz
$ cd rrb2-1.1
$ sudo python setup.py install
```

Rather than type in the long URL above it is probably a good idea to browse to this tutorial from your Raspberry Pi so that you can copy and paste the commands.

The example program that follows will prompt you for the delay between steps and the number of steps to take, first forwards and then backwards.

To install the code, you can [connect to your Pi using SSH \(https://adafru.it/aWc\)](https://adafru.it/aWc) and open an editor by typing:

```
$ nano stepper.py
```

Then paste the code below into the editor window and save it using CTRL-X and then Y.

```
from rrb2 import *
import time

rr = RRB2()

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

def backwards(delay, steps):
    for i in range(0, steps):
        rr.set_motors(1, 1, 1, 0)
        time.sleep(delay)
        rr.set_motors(1, 0, 1, 0)
        time.sleep(delay)
        rr.set_motors(1, 0, 1, 1)
        time.sleep(delay)
        rr.set_motors(1, 1, 1, 1)
        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? ")
    backwards(int(delay) / 1000.0, int(steps))
    rr.set_motors(0, 0, 0, 0) # All coils off
```

Run the program by typing the command:

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

You will then see a prompt for the delay between steps. The minimum value here is 3, increase the number to make the stepper turn more slowly. 200 is a good number of steps to enter at the next prompt. When you press enter, you should see the stepper turn.

```
$ sudo python stepper.py
Delay between steps (milliseconds)?3
How many steps forward? 200
How many steps backwards? 200
Delay between steps (milliseconds)?5
How many steps forward? 100
How many steps backwards? 100
Delay between steps (milliseconds)?
```

The program will continue in the loop until you press CTRL-C to quit.

---

## Next Steps

See also [this Adafruit tutorial \(https://adafru.it/CeP\)](https://adafru.it/CeP) on using stepper motors with the Raspberry Pi.

For more information of the RaspiRobot Board V2 and its Python library, see [raspirobot.com \(https://adafru.it/ekN\)](https://adafru.it/ekN).