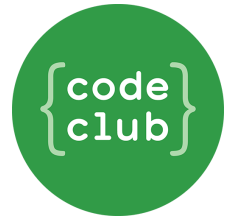


# Motor Control



using

## Python and Raspberry Pi

### Overview

*You are going to learn how to connect motors to your Raspberry Pi. You will then make a game with a motorised wheel which will rotate for a random amount of time, before stopping to select a question / person from a wheel.*

### Introduction

You can't connect a motor directly to the Raspberry Pi as a running motor requires more current than a GPIO pin can provide. You need to connect your Raspberry Pi to an interface called an H-bridge and then the motor to the H-Bridge.

The Pi's GPIO ports can only supply a few mA of current (16mA max). Attempting to draw more than this **will damage the Pi!** Motors typically require at least 400mA to start spinning (although they draw far less after startup). Thus a motor driver chip permits control of a high-current supply (like the Pi's 5V rail, or an external supply) from a low-current control signal (like a Pi's GPIO ports).

Motor drivers are often H-bridge circuits (shown in the diagram to the right), capable of driving a motor forwards or backwards (as well braking or free-running motors in certain cases).

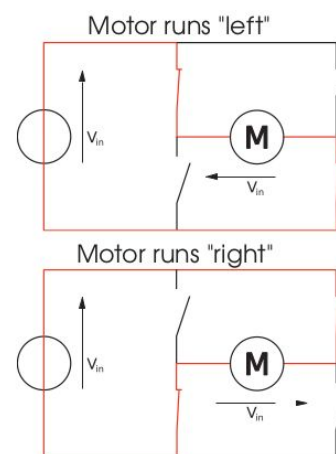
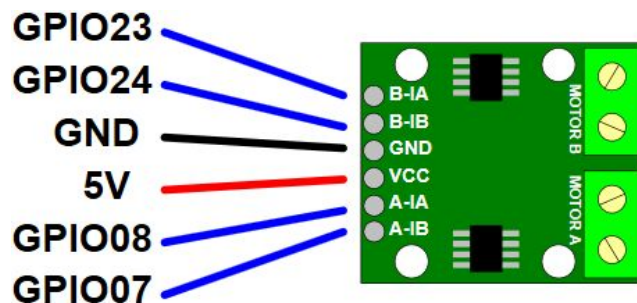
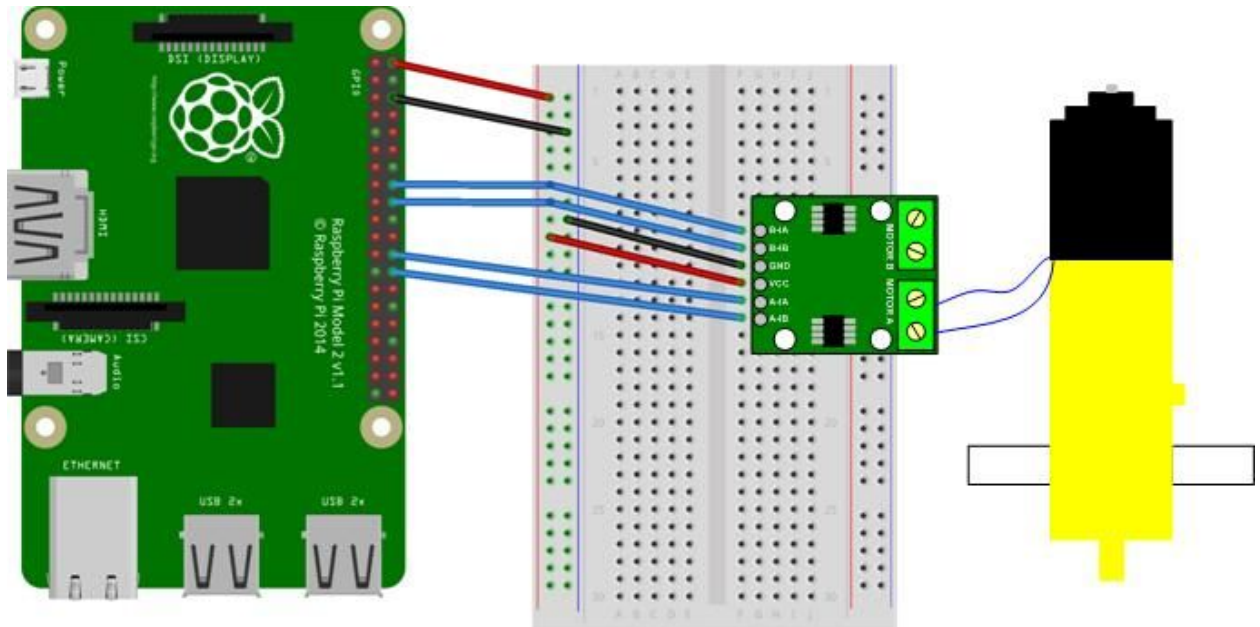


Image © 2006 Cyril Buttray via  
Wikipedia (CC-BY SA 3.0)

## Step 1 - Connecting the Motor and H-bridge to the Pi

1. Connect the H-bridge to the Raspberry Pi and the motor as shown in the diagram below.

**It's really important that you connect the H-bridge to the 5V pin on the Pi.**



## Step 2 - Open the Python IDE

1. On the desktop, click **Menu > Programming > Python 3 (IDLE)**
2. When the Python Shell appears, open a new file. Click **File > New File**

---

## Step 3 - Test the Motor

1. Lets test our wiring. **Add** this code to the new file:

```
from gpiozero import Motor  
from time import sleep
```

```
motorA = Motor(7, 8)
```

```
motorA.forward()
```

```
sleep(5)
```

```
motorA.stop()
```

2. Did the motor run for 5 seconds?
3. Can you explain why we used the command `Motor(7, 8)` ?
4. What command would you use for MotorB?

**Job Done!!!**

## Step 4 - Run the Motor for a random amount of time

1. Modify your code:

```
from gpiozero import Motor
from time import sleep
from random import randint

motorA = Motor(7, 8)

duration = randint(1,10)
print(duration)

motorA.forward()
sleep(duration)
motorA.stop()
```

2. Did the motor run for a random time?

**Job Done!!!**

## Step 5 - Additional Motor Commands

1. You can reverse the motor using:

```
motorA.backward()
```

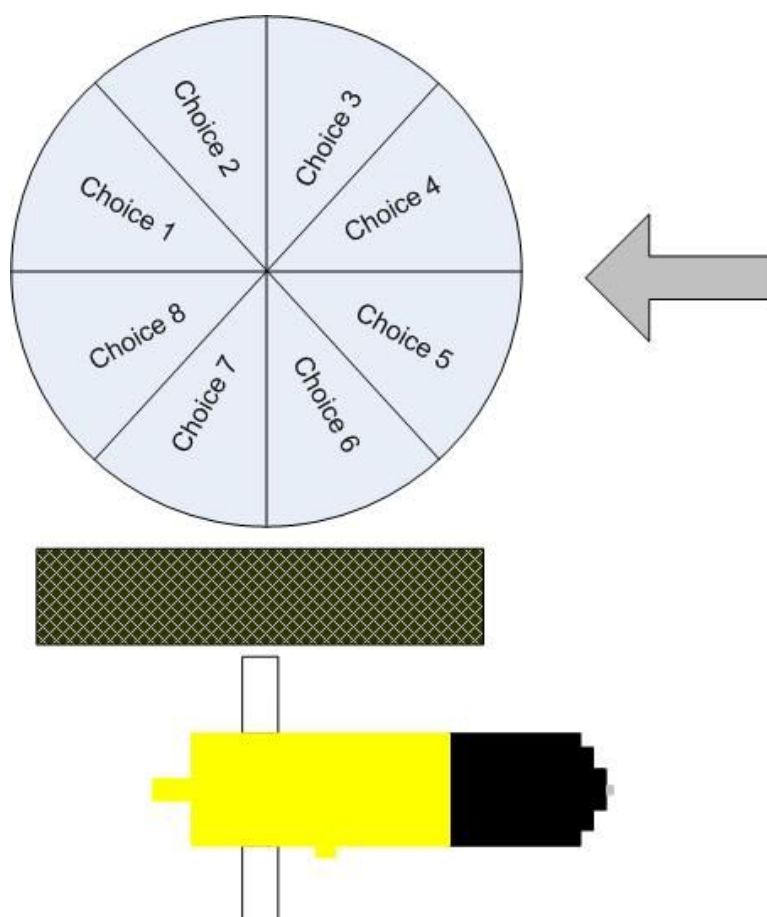
2. The speed at which the motor turns can be set by providing a value between 0 (stopped) and the default 1 (maximum speed) as shown below:

```
motorA.forward(0.7)
```

## Step 6 - Get Creative

Now that you have your motor working, it's time to build something with it. You're going to make a game that randomly chooses a selection from a wheel.

1. Make a selection wheel out of card (perhaps a paper plate).
2. Make a pointer (perhaps a lolly stick, card, pipecleaner or straw).
3. Connect one of the plastic wheels to the motor.
4. Secure the selection wheel on top of the plastic wheel.
5. Secure the motor to the desk (or on an upturned pot) with some blu-tack.
6. Use the code above to spin the motor for a random amount of time.



## Challenge 1

Add a button that spins the motor when you press it.