ME:4145 Industrial Internet of Things

Lab #5
Stepper Motor Control & WiFi

Introduction

The objective of this lab is control a stepper motor using a RPi. This includes specifying the state (On/Off), direction (CW/CCW) and the number of steps (steps). To execute these parameters, the RPi will be setup as a local server and your computer will be used as a client to request execution. Thus, for this lab you will only be using your RPi to start and stop the server side script. All motor control requests will come from your computer (client).

Getting Started

What you will need

To complete this lab, you will need the following components:

- 5-wire unipolar stepper motor (28BYJ-48 or similar)
- L293D dual h-bridge motor driver
- breadboard
- resistors
- jumper wires
- laptop (or other client device)
- 1 × GREEN LED
- 1 × RED LED
- 1 × BLUE LED
- 1 × ORANGE LED (or substitute)

Raspberry Pi connections

L293D and motor

The L293D h-bridge motor driver will be used to control the stepper motor. If you are using the recommended kit, the motor has 4 phases and 32 steps (11.25°) per revolution. In addition, there is a 1:16 reduction gearbox, meaning it needs 32 x 16 = 512 steps to complete a rotation in full stepping mode. The internal coil wiring diagram is provided in Fig. 1. To complete a full step, the coils must be powered in the sequence provided in Table 1, where

Table 1: Step coil sequence.

Step	Blue	Pink	Yellow	Orange
1	1	1	0	0
2	0	1	1	0
3	0	0	1	1
4	1	0	0	1

0 and 1 represent low and high, respectively. Similar to the DC motor lab, care must be

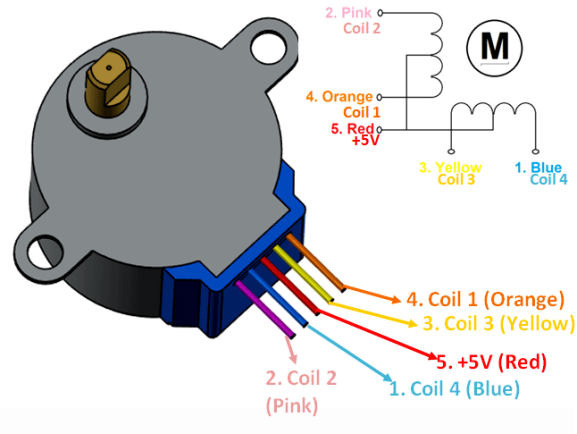

Figure 1: Wiring diagram for 28BYJ-48 unipolar stepper motor.

taken to not damage the chip or the RPi by exceeding the 60 mA current limit. Therefore, always check the power requirements of your motor before making electrical connections. A general wiring diagram for the L293D and stepper motor is provided in Fig. 2. To identify the orientation of the L293D, look for the small notch on the top of the chip.

In this lab, we do not use the common red wire on the stepper motor. This connection is for different drive circuits that do not allow the current in each coil to be reversed. Having a center connection to each coil means that you can either energize the left or right side of the coil, and get the effect of reversing the current flow without having to use a circuit that can reverse the current. However, since we are using the L293D which can easily reverse the current, we do not need this common connection.

Example Python scripts for controlling the stepper motor, and server/client communication over WiFi are provided on the ICON course web page in the Lab #6 folder. The example stepper motor script uses GPIO pin descriptions according to the **BCM** and pin connections shown in Fig. 2a. Do **NOT** use this script until you have decide which description you are using and verified your own pin assignments. Failure to do so can result in damage to the RPi, L293D and stepper motor.

Referring back to Fig. 2a, the stepper motor is powered directly from the RPi. Since the stepper motor in the kit draws little current this is fine. However, for higher powered steppers the power source would need to come from an external power supply. While not necessary for this lab, this type of configuration is shown in Fig. 3 for your reference.

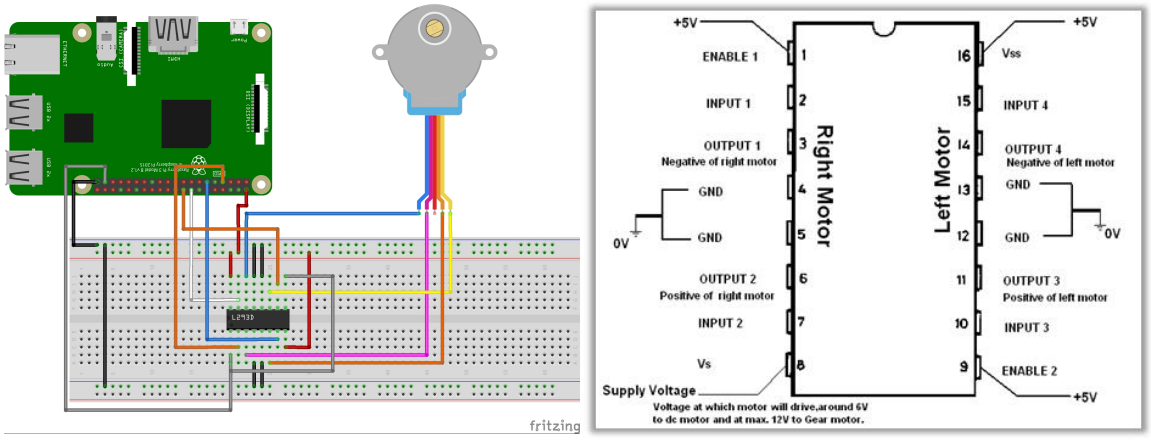

Figure 2: (a) L293D and stepper motor wiring example (b) L293D diagram.

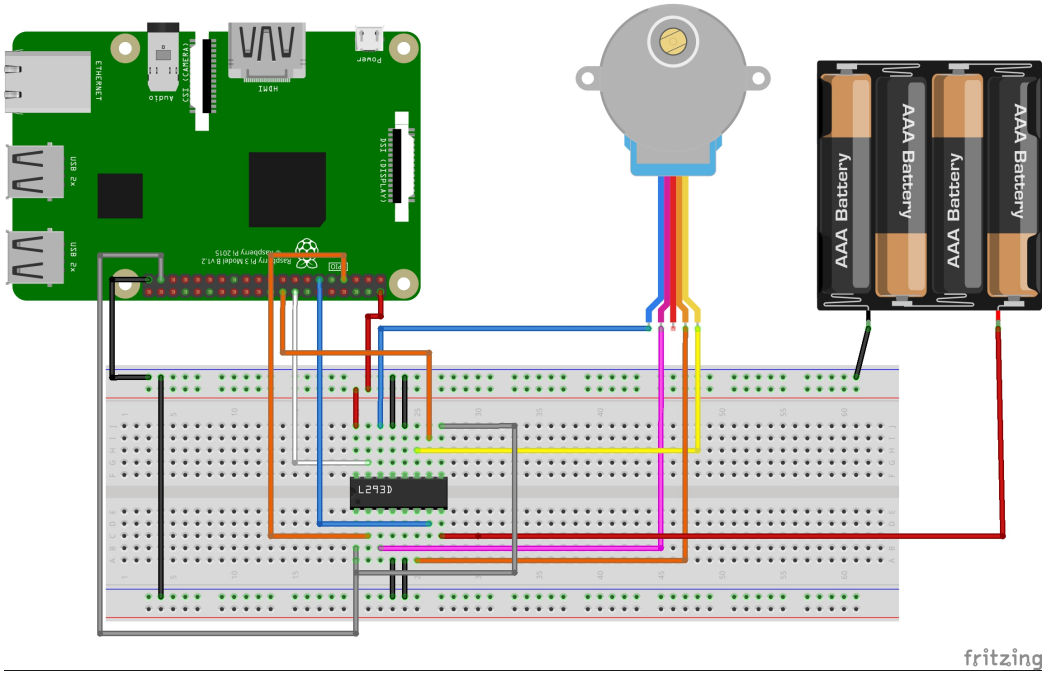

Figure 3: L293D and DC motor wiring example with external power source.

Task

Write a client (program that runs on your laptop) and server (program that runs on your RPi) Python scripts. To help keep your code from becoming too complicated, considering using separate scripts for your server and motor control functions. Your program should be able to perform the following:

1. define a home position for your motor
    - this can be where the shaft is located when you start you program
2. rotate the stepper to a specific position (degrees), direction (CW or CCW) and speed (0-100%).
3. go back home taking the shortest route.

To make identify the motor position easier, attach a piece of tape, cardboard or ziptie to the motor shaft. Furthermore, your program should also control LEDs according to the following conditions:

- turn a green LED on when your motor is running (CW or CCW).
- turn a blue LED on when your motor is rotating CW.
- turn a orange LED on when your motor is rotating CCW (or substitute a different color).
- turn a red LED on when your motor is off.

Submission Materials

Upload the following to ICON no later than the due date:

1. Python script/s
2. A video of your motor doing the following:
    - starting from home, move CW 135° at full speed (you define full speed)
    - go back home
    - move CCW 135° at 75% speed
    - go back home

    Note: CW and CCW are referenced when viewing the motor from the shaft side.

Additionally, bring your assembled circuit and laptop to the next lab and:

4. be prepared to show your functioning circuit and code.
5. move your motor according to requests from the instructor based on the items listed in the Task section.
6. Extra Credit (25%)

    - **5%:** In addition to a laptop, use a cell phone to execute all of items in the Task section. You will show this to the instructor during the lab.
    - **20%:** Allow the user to define a new home position based on the current positon and be able to reset back to the default home position upon request.