08b Stepper Motor Introduction

Nicholas Bruzzese

January 6, 2025

What is a Stepper Motor?

A stepper motor is a type of DC motor that moves in discrete steps. Unlike regular motors that spin continuously when powered, stepper motors are designed to:

- Rotate in precise increments.
- Maintain their position when not moving.
- Be controlled for speed, direction, and step count.

Key Features

- **Steps per Revolution**: Defines how many steps the motor takes to complete one full rotation.
- **Direction Control**: Easily reversible rotation.
- **Precision**: Ideal for applications requiring accurate positioning, like 3D printers or camera sliders.

Hardware Requirements

- Raspberry Pi (any model with GPIO pins).
- 28BYJ-48 stepper motor with ULN2003 driver board.
- Jumper wires for connections.

Wiring Diagram

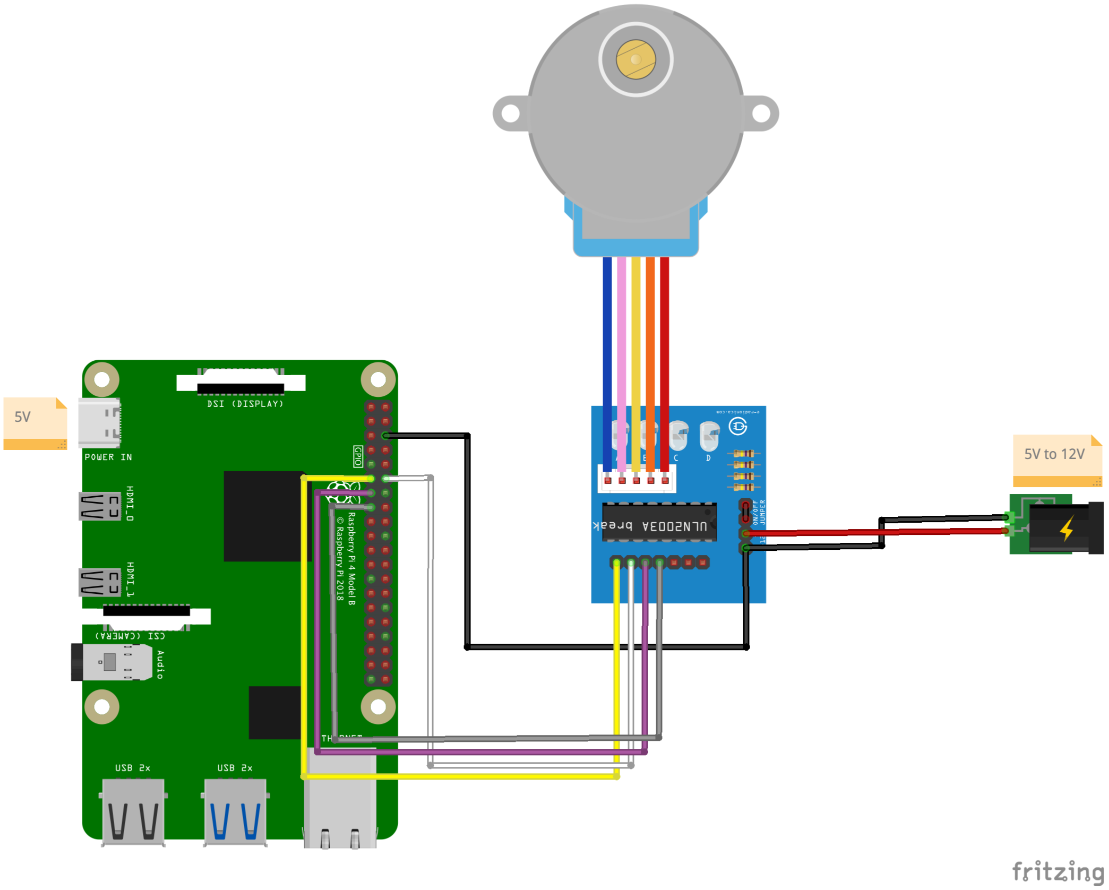

Figure 1: Wiring Diagram

Stepper Motor Setup with Raspberry Pi

Connections

- Connect the `IN1, IN2, IN3, IN4` pins of the ULN2003 driver board to GPIO pins `17, 18, 27, and 22` of the Raspberry Pi.
- Connect `VCC` on the driver board to the 5V pin of the Raspberry Pi.
- Connect `GND` on the driver board to a GND pin of the Raspberry Pi.

Python Code Explanation

The following Python code controls the stepper motor by sending precise signals to its pins. Let's break it down:

Setup

Define GPIO pins to control the motor and initialize the GPIO settings.

```python
import RPi.GPIO as GPIO
import time

# Pin definitions
in1 = 17
in2 = 18
in3 = 27
in4 = 22

# GPIO setup
GPIO.setmode(GPIO.BCM)
GPIO.setup(in1, GPIO.OUT)
GPIO.setup(in2, GPIO.OUT)
GPIO.setup(in3, GPIO.OUT)
GPIO.setup(in4, GPIO.OUT)
GPIO.output(in1, GPIO.LOW)
GPIO.output(in2, GPIO.LOW)
GPIO.output(in3, GPIO.LOW)
GPIO.output(in4, GPIO.LOW)
```

Stepper Motor Sequence

The stepper motor rotates when its coils are energized in a specific sequence.

```python
step_sequence = [[1,0,0,1],
[1,0,0,0],
[1,1,0,0],
[0,1,0,0],
[0,1,1,0],
[0,0,1,0],
[0,0,1,1],
[0,0,0,1]]
```

Motor Rotation Logic

The motor rotates based on:

- **Step Count**: Total steps for one full rotation (e.g., 4096 for the 28BYJ-48 motor).

- **Direction**: Clockwise (CW) or Counterclockwise (CCW).
- **Sleep Time**: Time delay between steps controls motor speed.

```
step_sleep = 0.002 # Delay between steps
step_count = 4096 # Total steps for one full rotation
direction = False # False for CCW, True for CW

motor_pins = [in1, in2, in3, in4]
motor_step_counter = 0
```

Main Loop

The loop iterates through the steps, updating the motor's state based on the step sequence.

```
try:
for i in range(step_count):
for pin in range(0, len(motor_pins)):
GPIO.output(motor_pins[pin],
    step_sequence[motor_step_counter][pin])
motor_step_counter = (motor_step_counter + 1) % 8 if not direction
    else (motor_step_counter - 1) % 8
time.sleep(step_sleep)
except KeyboardInterrupt:
pass
finally:
GPIO.cleanup()
```

Running the Code

Save the code as `stepper_motor.py` and run it using the command:

```
python3 stepper_motor.py
```

Expected Behavior

- The motor will rotate 360° in 4096 steps.
- To reverse the direction, set `direction = True` in the code.

Experiment Ideas

- **Speed Control**: Change `step_sleep` to see how it affects speed.

- **Partial Rotations**: Adjust `step_count` for specific angles (e.g., 90° = 1024 steps).
- **Interactive Control**: Use user input to set the direction or steps dynamically.