HTL Leonding
next level

WLC

Protokoll

17.01.2025

Markus Harnúšek

Markus Harnúšek 1CHIF 17.01.2025

Motor controller with ESP32

This protocol gives a tutorial on how to control a motor using an ESP32, giving input to where the motor should rotate to and at what speed and showing the current motor speed and pos on the OLED Display.

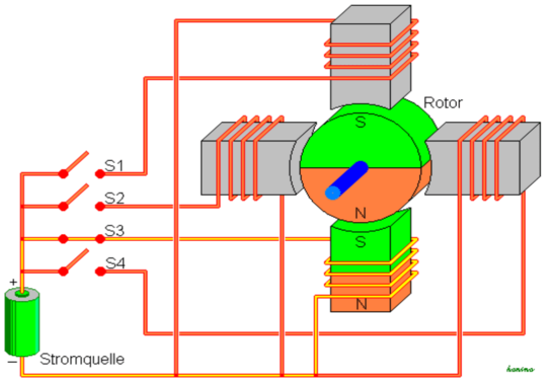
Contents:

Hardware

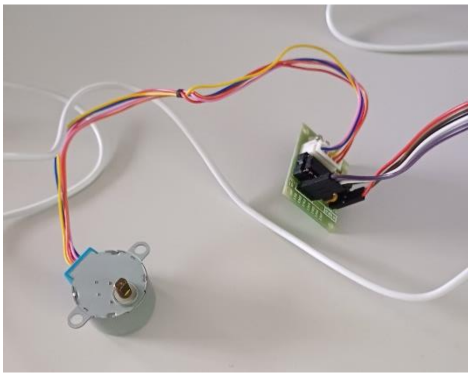
Due to the fact that Arduinos can’t produce enough voltage to power motors, there needs to be a motor driver to power the motor. And since there is no controller to just straight up power the motor, there is a overall power input, GND as well as cables to control each of the motors magnets.

1. How a motor stepper works

To power a motor controlling each electromagnet manually, there is a certain procedure called motor stepping. Here’s how it works:



1 Anatomy of a stepper motor https://res.cloudinary.com/rspoc/image/upload/w_620,h_413,c_crop,c_pad,b_white,f_auto,q_auto/dpr_auto/v1666096543/RS%20CONTENTFUL/Discovery/Guides/Schrittmotor.jpg



2 The setup with the motor and the motor driver

A. Full steps

When doing full steps, one motor is turned on while the one before is turned off. This ensures that the rotor is turning from one electromagnet to another.

B. Half Steps

Half stepping works similar to full stepping, however there are times when two neighboring magnets are turned on at the same time. Doing this a better rotation is achieved, when turning one magnet then the next, and only then turning the previous off.

C. Ramping

Ramping is crucial when starting the motor because, especially with motors with higher rotation per minute, it can't accelerate from zero to full load instantaneously. Therefore the same motor stepping procedure is used, but the delay between the steps is larger at the beginning and then continuously reduced to get up to the wanted speed.

Software

1. Modulo Operators

In this case every motor has each position where it is turned on. Therefore a pos value is increased or decreased with each motor magnet change. Then it will be checked if the current pos value is dividable by the assigned pos value of each motor magnet without it being a floating-point number. In this case modulo operations are important. For example 165 % 8 is 5. Modulo divides the number and takes the nearest multiple of the number being divided through.

2. The code

The code is to be found in the Github Repository linked below:

https://github.com/MarkusHarnusek/ESP32MoveMotor