Part 2

lesson 21

Stepper Motor

Overview

In this lesson, you will learn a fun and easy way to drive a stepper motor.

The stepper we are using comes with its own driver board making it easy to connect to our ESP32.

Component Required:

(1) x Elegoo ESP32

(1) x 400 tie-points breadboard

(1) x ULN2003 stepper motor driver module

(1) x Stepper motor

(1) x 9V1A Adapter

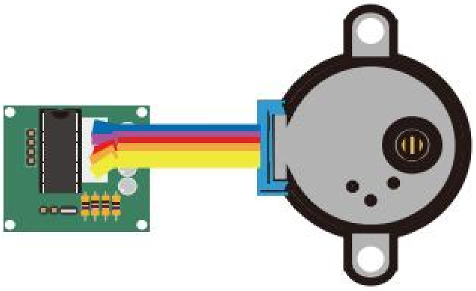
(1) x Power supply module

(6) x F-M wires (Female to Male DuPont wires)

(1) x M-M wire (Male to Male jumper wire)

Component Introduction

Stepper Motor



A stepper motor is an electromechanical device which converts electrical pulses into discrete mechanical movements. The shaft or spindle of a stepper motor rotates in discrete step increments when electrical command pulses are applied to it in the proper sequence The motors rotation has several direct relationships to these applied input pulses. The sequence of the applied pulses is directly related to the direction of motor shafts rotation. The speed of the motor shafts rotation is directly related to the frequency of the input pulses and the length of rotation is directly related to the number of input pulses applied. One of the most significant advantages of a stepper motor is its ability to be accurately controlled in an open loop system. Open loop control means no feedback information about position is needed. This type of control eliminates the need for expensive sensing and feedback devices such as optical encoders. Your position is known simply by keeping track of the input step pulses.

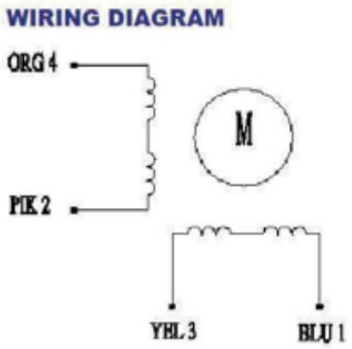
Stepper motor 28BYJ-48 Parameters

Model: 28BYJ-48
Rated voltage: 5VDC
Number of Phase: 4
Speed Variation Ratio: 1/64
Stride Angle: 5.625° /64
Frequency: 100Hz
DC resistance: 50Ω±7%(25℃)
Idle In-traction Frequency: > 600Hz
Idle Out-traction Frequency: > 1000Hz

In-traction Torque >34.3mN.m(120Hz)
Self-positioning Torque >34.3mN.m
Friction torque: 600-1200 gf.cm
Pull in torque: 300 gf.cm
Insulated resistance >10MΩ(500V)
Insulated electricity power:600VAC/1mA/1s
Insulation grade:A
Rise in Temperature <40K(120Hz)
Noise <35dB(120Hz,No load,10cm)

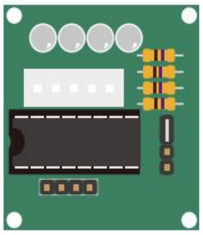
Interfacing circuits

The bipolar stepper motor usually has four wires coming out of it. Unlike unipolar steppers, bipolar steppers have no common center connection. They have two independent sets of coils instead. You can distinguish them from unipolar steppers by measuring the resistance between the wires. You should find two pairs of wires with equal resistance. If you’ ve got the leads of your meter connected to two wires that are not connected (i.e. not attached to the same coil), you should see infinite resistance (or no continuity).



Product Description

- Size: 42mmx30mm
- Use ULN2003 driver chip, 500mA
- A. B. C. D LED indicating the four phase stepper motor working condition.
- White jack is the four phase stepper motor standard jack.
- Power pins are separated
- We kept the rest pins of the ULN2003 chip for your further prototyping.



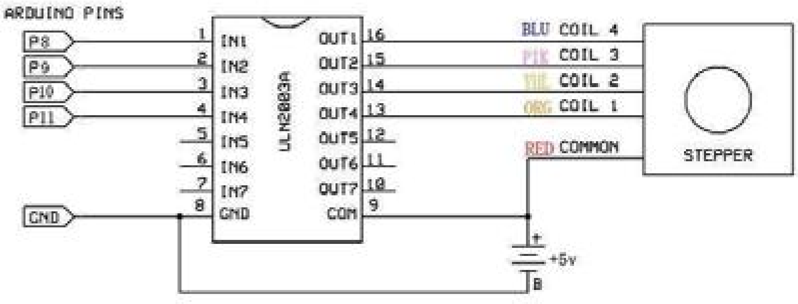
ULN2003 Driver Board

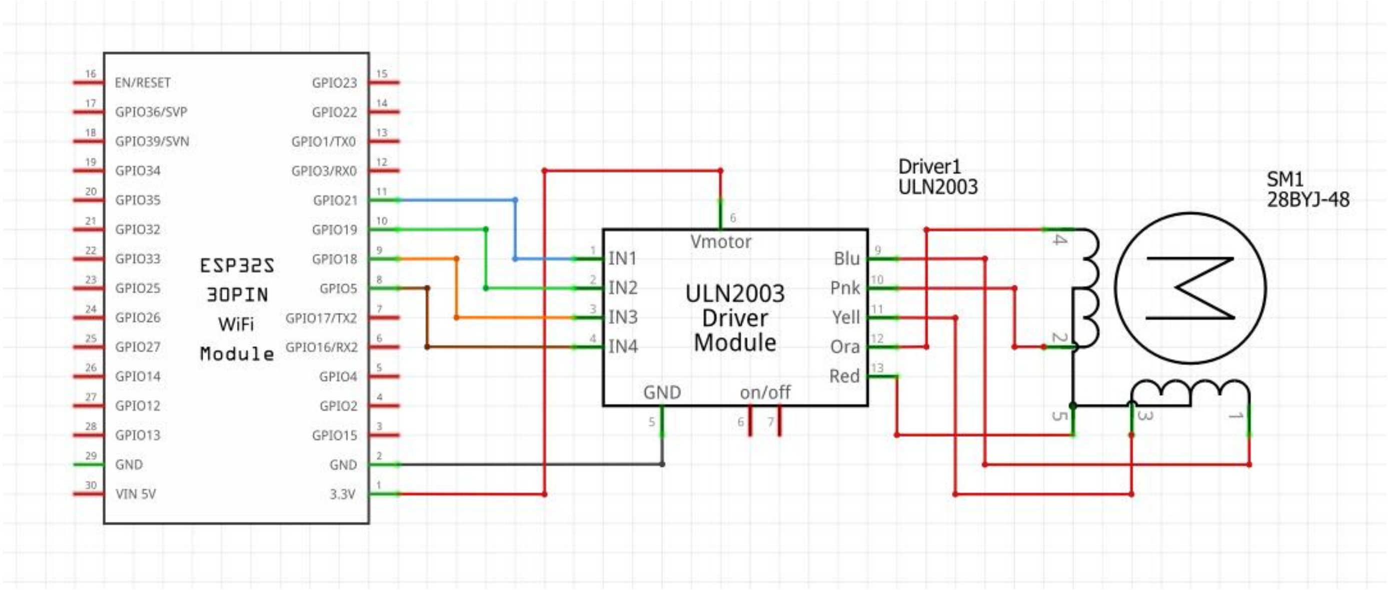
The simplest way of interfacing a unipolar stepper to Arduino is to use a breakout for ULN2003A transistor array chip. The ULN2003A contains seven Darlington transistor drivers and is somewhat like having seven TIP120 transistors all in one package. The ULN2003A can pass up to 500 mA per channel and has an internal voltage drop of about 1V when on. It also contains internal clamp diodes to dissipate voltage spikes when driving inductive loads. To control the stepper, apply voltage to each of the coils in a specific sequence.

The sequence would go like this:

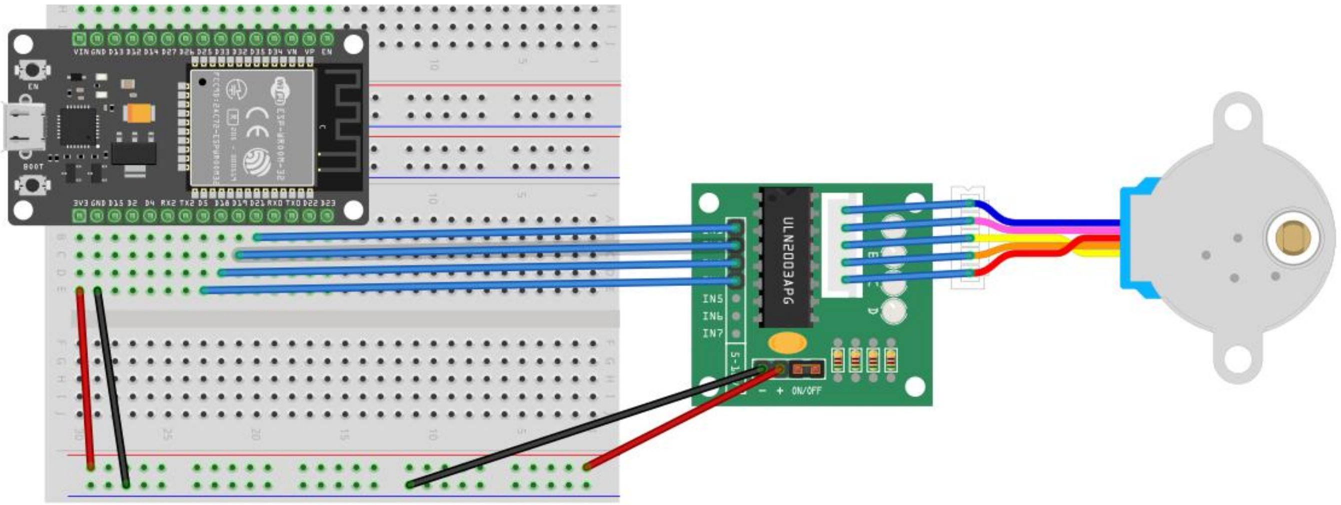
Lead Wire Color	--- > CW Direction (1-2 Phase)							
	1	2	3	4	5	6	7	8
4 ORG	-	-						
3 YEL			-	-				
2 PIK				-	-	-		
1 BLU						-	-	-

Here are schematics showing how to interface a unipolar stepper motor to four controller pins using a ULN2003A, and showing how to interface using four com.





Connection Schematic



we are using 4 pins to control the Stepper. GPIO 5,18,19 ,21(In4-In1) are controlling the Stepper motor. We connect the Ground from the ESP32 to the Stepper motor.

Wiring diagram

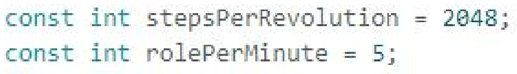
Code

After wiring, please open the program in the code folder- **stepper_Example** and click UPLOAD to upload the program. See Lesson 5 of part 1 for details about program uploading if there are any errors.

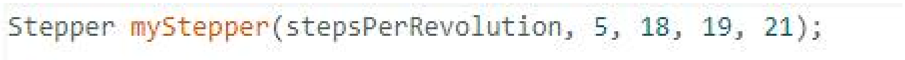
Before you can run this, make sure that you have installed the **< Stepper >** library or re-install it, if necessary. Otherwise, your code won't work.
For details about loading the library file, see Lesson 5 of part 1.

const int stepsPerRevolution = 2048: Defines the number of steps the motor needs to complete one full revolution (360°). For the 28BYJ-48 stepper motor, this value is 2048 (due to its 64:1 gear ratio).

const int rolePerMinute = 5: Sets the motor's rotation speed (in revolutions per minute, rpm). The 28BYJ-48 has a speed range of 0~17 rpm here it’s set to 5 rpm.If the stepper motor vibrates but does not rotate, please adjust the step delay / pulse interval (i.e. increase or decrease the delay between steps).



```
const int stepsPerRevolution = 2048;
const int rolePerMinute = 5;
```



```
Stepper myStepper(stepsPerRevolution, 5, 18, 19, 21);
```

Stepper myStepper(stepsPerRevolution, 5, 18, 19, 21): Creates a Stepper object named myStepper to control the motor:

First parameter: stepsPerRevolution (steps per full rotation).

Remaining parameters: ESP32 GPIO pins connected to the stepper motor's 4 control pins (IN1~IN4 of the 28BYJ-48 driver board).

If you need to reassign the pins connected to the ESP32, simply update the pin mapping parameter