Introduction to Motion Control

ROCHESTER MAKERSPACE

2021

Class Objectives

1. Become familiar common types of motors
2. Focus on motors typically used by hobbyists
3. Understand how to control motors
4. Understand cautions when using motors with microcontrollers/microcomputers
5. Provide projects to demonstrate motor control on Raspberry Pi and Arduino
6. Provide examples of typical motor applications
7. Provide list of resources for learning more

Class notes and code

- All the class notes and code can be downloaded from GitHub:
  - https://github.com/RochesterMakerSpace/MotionControlClass
- Pull down the green “Code” button and select “Download ZIP”

Types of Common Electric Motors

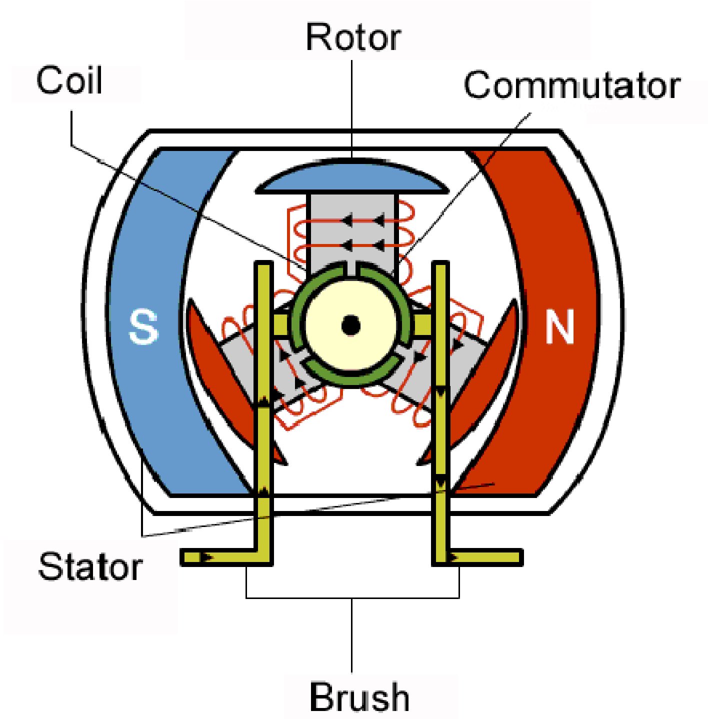
- There are two main motor classifications, ac and dc, with three application areas, constant speed, variable speed and position (or torque) control.
- AC Motors

  AC motors are typically associated with industrial applications and will not be address further in this class.
- DC Motors

  Although large DC motor are common in industrial application smaller DC motor are favored by the hobbyist and include:
  - DC Brushed Motors – simple, cost effective, lightweight and easy to control. Excellent torque at low speed but produce electrical noise. Used as drives for hobby vehicles.
  - DC Brushless Motors -- mechanically simpler than brushed motors but require more complex controllers and gearboxes and can be found in multicoptors and drones.
  - Servo Motors – good for projects needing precise control of angular or linear position such as robots.
  - Stepper Motors – move in discrete, precise steps and are found in CNC machines and 3D printers.

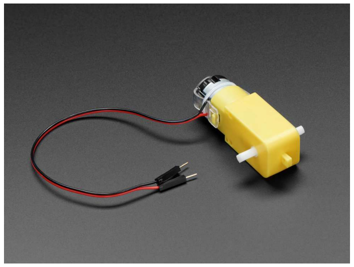
Brushed DC Motor Description

- Components:
  - Stator: Generates a stationary magnetic field that surrounds the rotor
  - Rotor (armature): Made up of one or more windings. Produce a magnetic field when energized
  - Brushes: As the motor turns, carbon brushes slide over the commutator, coming in contact with different segments of the commutator
  - Commutator: Segments are attached to different rotor windings, generating a dynamic magnetic field inside the motor.



Brushed DC Motor Example

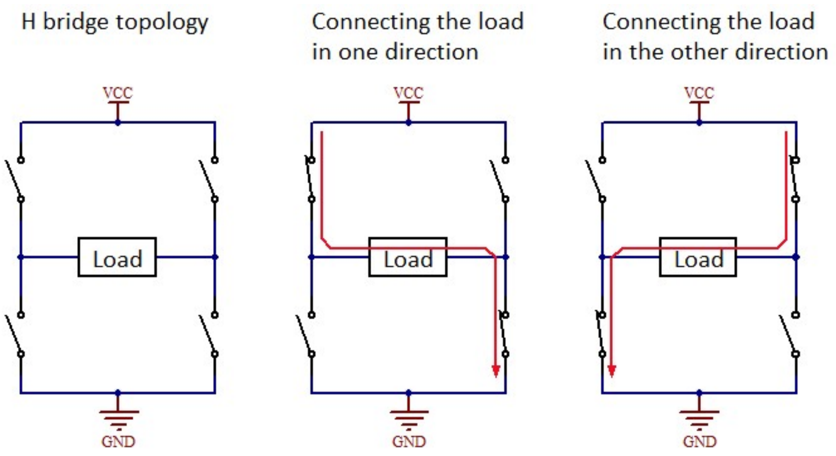
- DC Gearbox Motor – "TT Motor" – 200 RPM – 3 to 6 VDC
  - Operating Voltage Range: 3~7.5V
  - Rated Voltage: 6V
  - Max No-load Current (3V): 140 mA
  - Max No-load Current (6V): 170 mA
  - No-load Speed (3V): 90 rpm
  - No-load Speed (6V): 160 rpm
  - Max Output Torque: 0.8 kgf cm
  - Max Stall Current: 2.8 A
- Pololu Micro Metal Gearmotor
  - See spec sheet



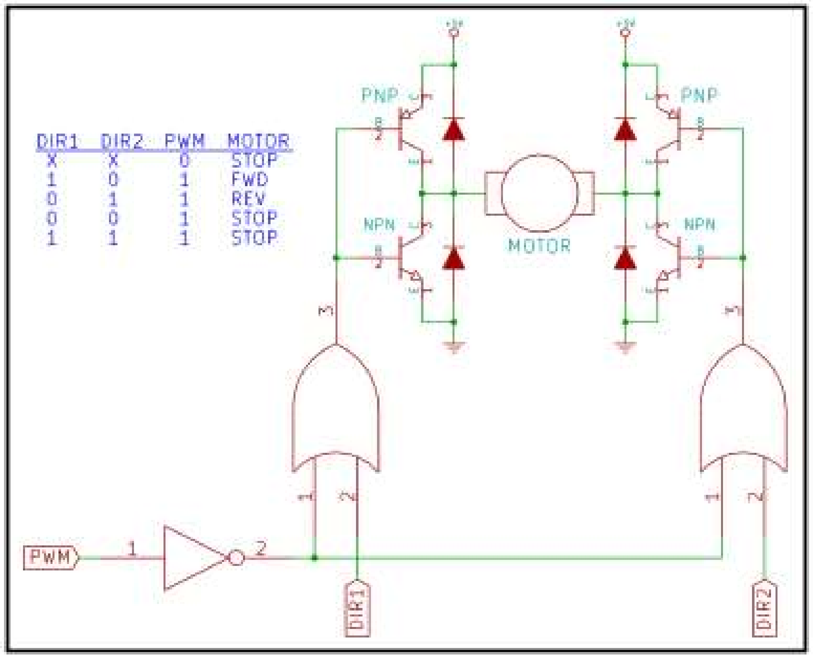
www.pololu.com

Brushed DC Direction Control: H-Bridge

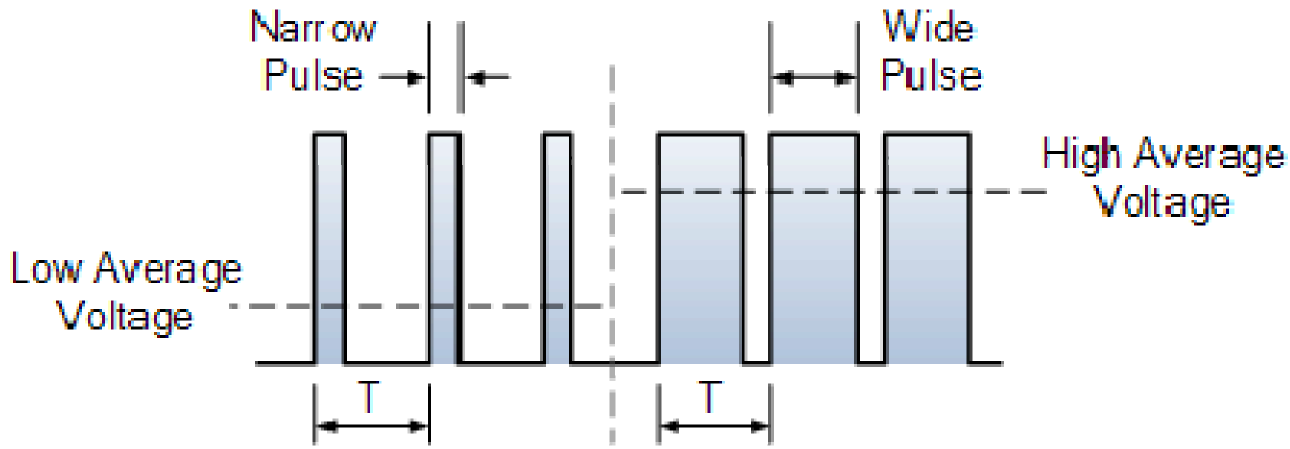
- An H-Bridge circuit contains four switches with the motor at the center forming an H-like arrangement.
- Closing two particular switches at the same time reverses the polarity of the voltage applied to the motor. This causes change in spinning direction of the motor.
- The control circuit prevents closing both switches on the same side of the H-Bridge, thus preventing a short from Vcc to GND.



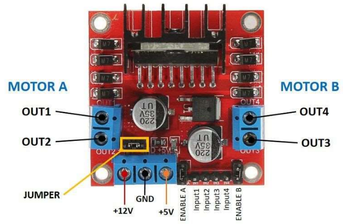
H-Bridge with PWM Control

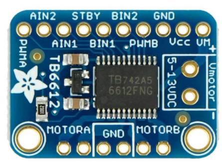


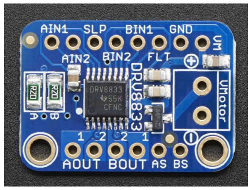
Brushed DC Speed Control: PWM

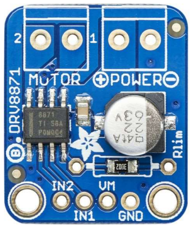


Brushed DC Motor Drivers









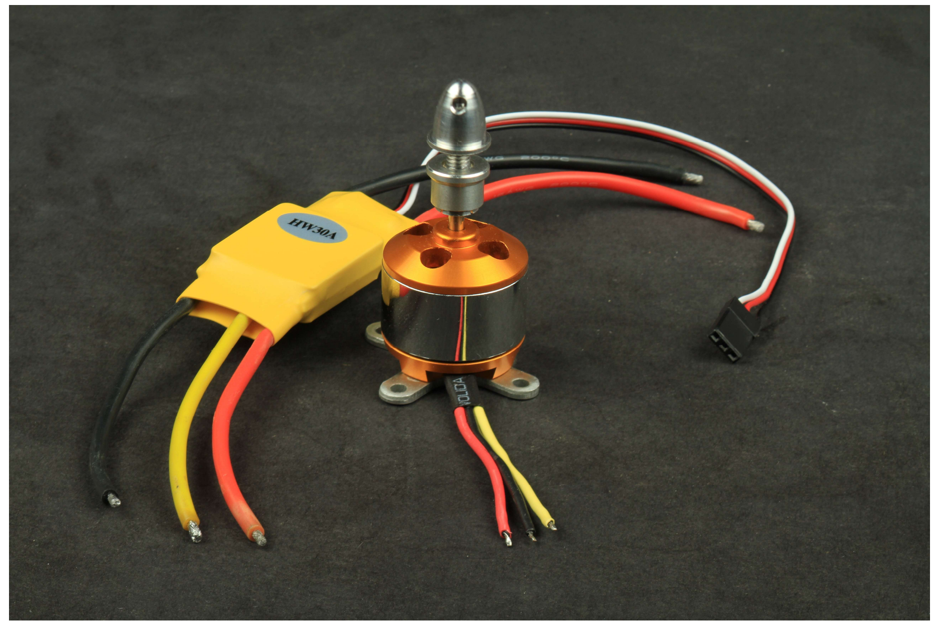
Model	L298N	TB6612	DRV8833	DRV8871
Current	2A	1.2A	1.5A	2A
Peak Current	3A	3.3A	2A	3.6A
Operating Voltage	3.2 – 40 V	4.5 – 15 V	2.7 – 10.8 V	6.5 – 45 V
Efficiency	40-70%	91-95%	70-90%	
Total MCU pins	6	6	4	2
PWM Pins	2	2	2/4	2
Number of motor outputs	2	2	2	1
Technology	Bipolar	MOSFET	MOSFET	MOSFET
Approximate cost	$2.50	$5.00	$5.00	$7.50

Servo Motor Description

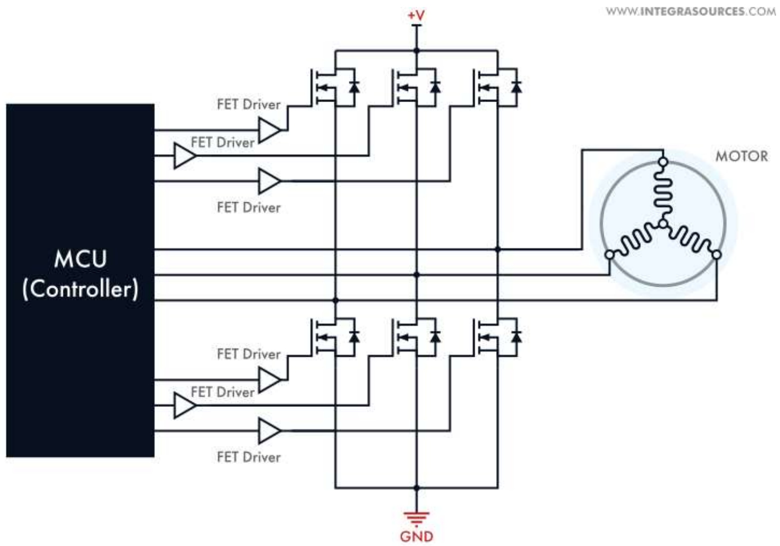
- Components:
  - Motor
  - Potentiometer: Shaft position
  - Gear Assembly: Reduce RPM to increase torque
  - Controlling circuit: Feedback
  - Input signal: PWM pulse width
- An error signal is generated between the potentiometer and the input signal. The error signal causes the motor shaft to turn, changing the voltage on the potentiometer, thus reducing the error until the motor stops when the error is 0.

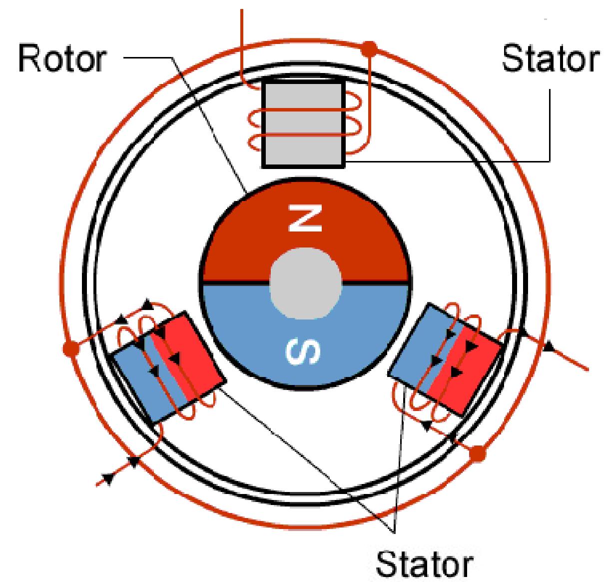
Brushless DC Motor (BLDC) Description

- Brushless DC motor control is not quite as simple as the regular brushed type. Brushless motors require a controller that can sense the proper time to reverse the voltage to the coils.
- Some motors have built-in hall-effect sensors that can detect the orientation of the rotor. Controllers for this type of motor require sensor inputs to read these hall sensors.
- Other 'sensorless' motor controllers sense the back EMF in the motor coils themselves to detect the rotor position. Most ESC (Electronic Speed Controls) for brushless RC motors use this technique.
- Brushless motors designed for autonomous and remote control aircraft and vehicles typically require a separate controller. These are typically of the sensorless type and use standard servo type pulsed signals for speed control.
- Applications include RC controlled planes and drones.

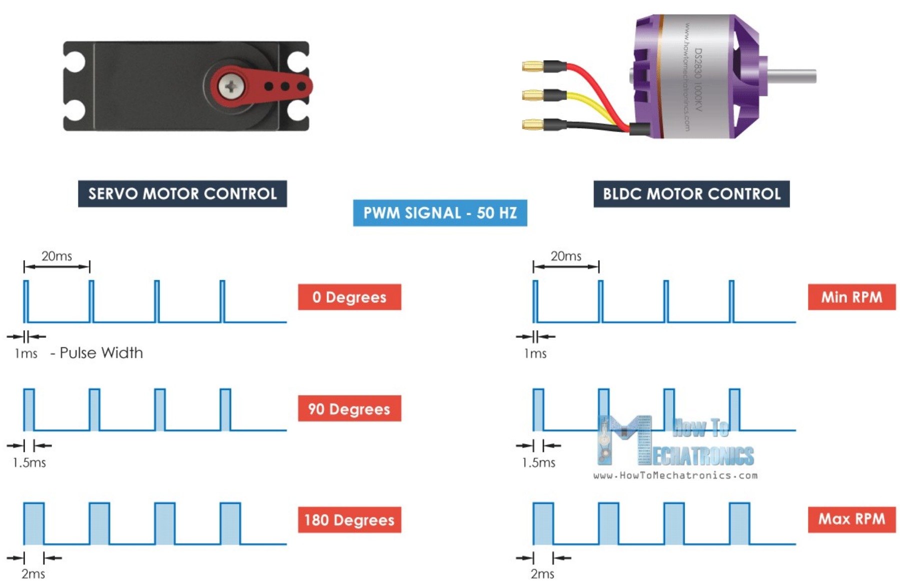


Brushless DC Motor (BLDC) Description





Servo/BLDC Control

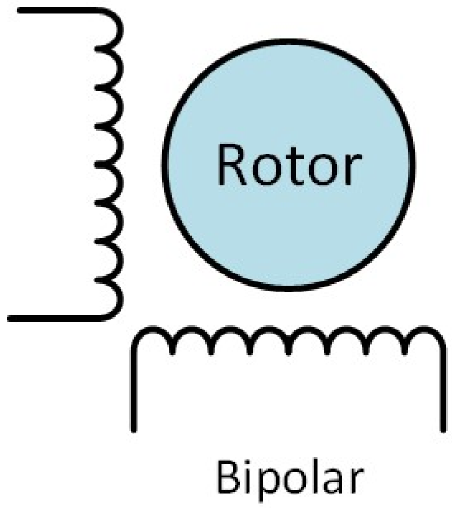


Stepper Motor Description

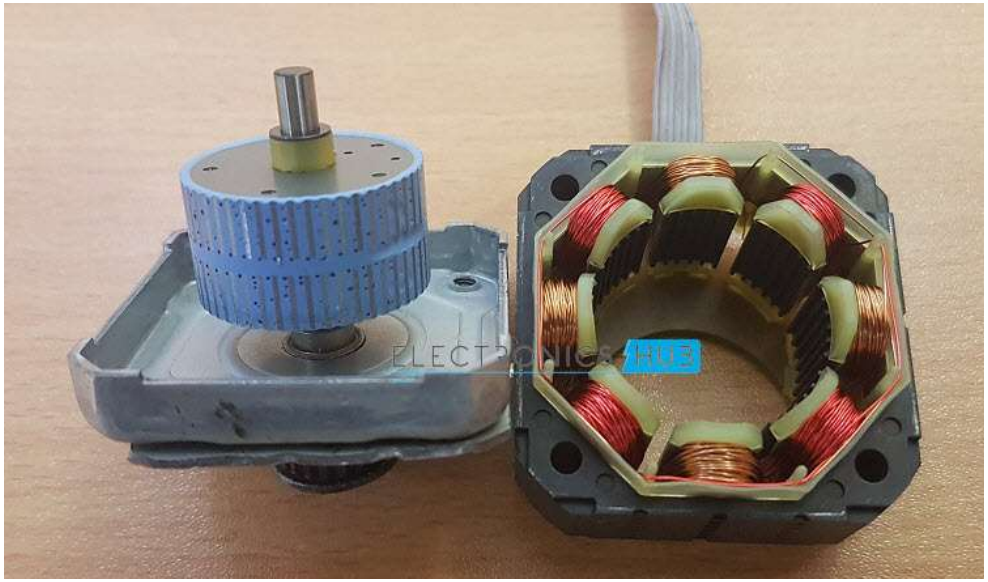
- Advantages
  - Steppers don’t require a sensor to detect the motor position since the motor moves by simply counting steps.
  - Although the motor does need a controller it doesn’t need complex calculation or tuning to work properly.
  - With micro stepping high positional accuracy can be achieved.
  - Good torque at low speed and great for holding position.
- Disadvantages
  - Can miss steps if torque load is too high.
  - Always drain maximum current.
  - Low torque and noisy at high speeds.

Bipolar Stepper Motor Example

- TPP-47-002
  - NEMA 17
  - Step Angle: 1.8 degree (200 steps/rev)
  - Holding Torque: 33Ncm(47oz.in)
  - Rated Current/phase: 1.2 A
  - Phase Resistance: 2.4 ohms
  - Inductance: 4.7 mH
  - Application: 3D Printer

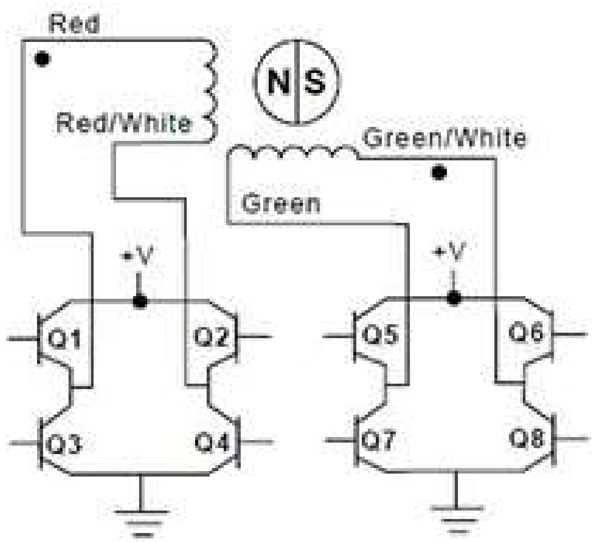


Stepper Motor Windings



Source: https://www.electronicshub.org/stepper-motor-control-using-arduino/

Bipolar Full Step Sequence



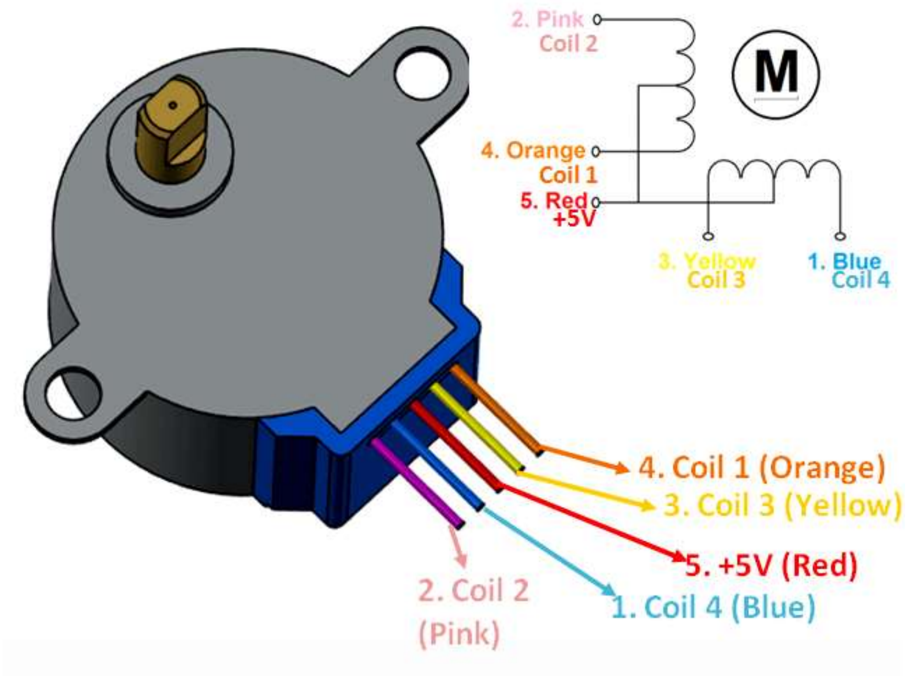
CW Rotation ↓

Bipolar Step	Q2-Q3	Q1-Q4	Q6-Q7	Q5-Q8
1	ON	OFF	ON	OFF
2	OFF	ON	ON	OFF
3	OFF	ON	OFF	ON
4	ON	OFF	OFF	ON
1	ON	OFF	ON	OFF

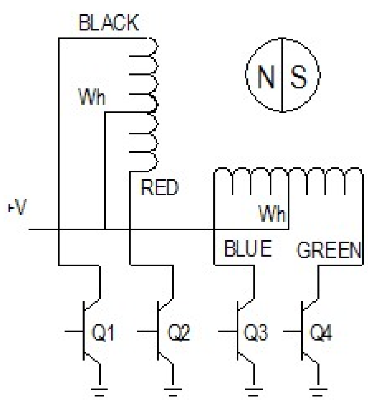
CCW Rotation ↑

Unipolar Stepper Motor Example

- 28BYJ-48
  - 5V Nominal Voltage
  - Number of Phases: 4
  - 64:1 (63.68395:1) gearing
  - Full Step Sequence:
    - 2048 (2038) steps per revolution
    - 11.25°/64 (63.68395) Stride Angle
  - Half Step Sequence:
    - 4096 (4076) steps per revolution
    - 5.625°/64 (63.68395) Stride Angle
  - Pull in torque: 300 gf.cm
  - Insulated Power: 600VAC/1mA/1s
  - Coil: Unipolar 5 lead coil
- More Info: https://lastminuteengineers.com/28byj48-stepper-motor-arduino-tutorial/#gear-reduction-ratio



Unipolar Full Step Sequence

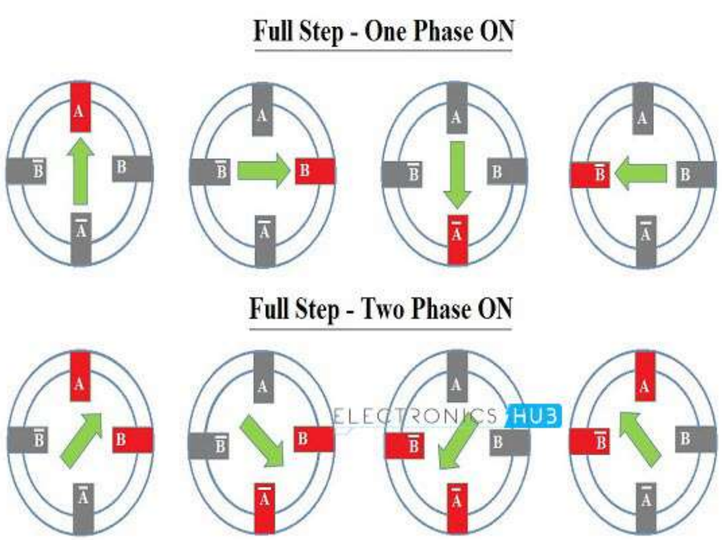


CW Rotation

Unipolar Step	Q1	Q2	Q3	Q4
1	ON	OFF	ON	OFF
2	OFF	ON	ON	OFF
3	OFF	ON	OFF	ON
4	ON	OFF	OFF	ON
1	ON	OFF	ON	OFF

CCW Rotation

Stepper Motor – Full Step Sequence



Full Step - One Phase ON

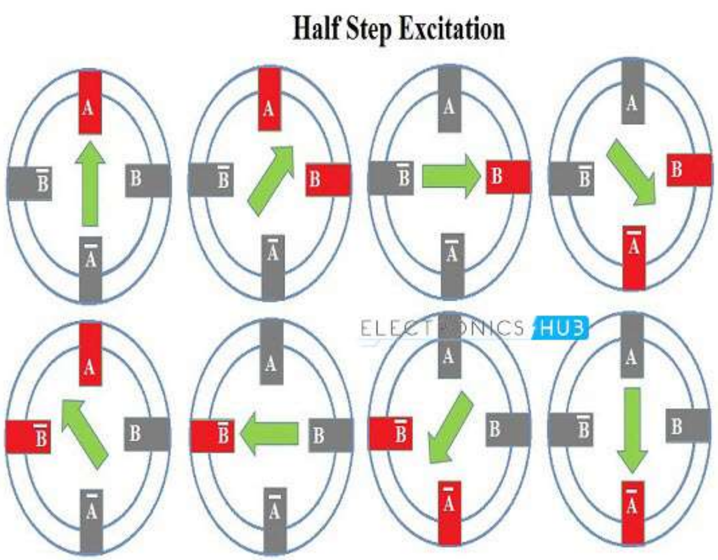
Step	Phase			
	A	B	$\overline{A}$	$\overline{B}$
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

Full Step - Two Phase ON

Step	Phase			
	A	B	$\overline{A}$	$\overline{B}$
1	1	1	0	0
2	0	1	1	0
3	0	0	1	1
4	1	0	0	1

Source: https://www.electronicshub.org/stepper-motor-control-using-arduino/

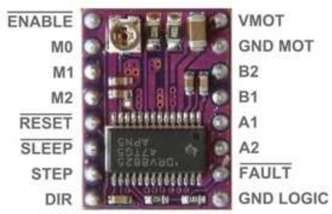
Stepper Motor – Half Step Sequence

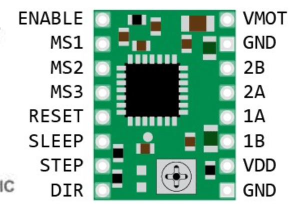


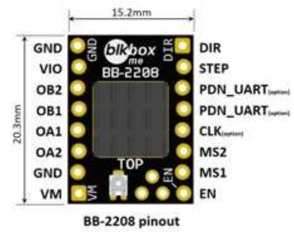
Step	Phase			
	A	B	$\overline{A}$	$\overline{B}$
1	1	0	0	0
2	1	1	0	0
3	0	1	0	0
4	0	1	1	0
5	0	0	1	0
6	0	0	1	1
7	0	0	0	1
8	1	0	0	1

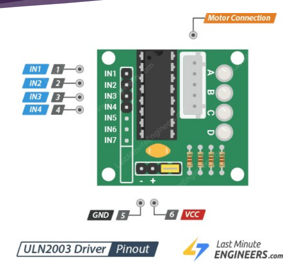
Source: https://www.electronicshub.org/stepper-motor-control-using-arduino/

Stepper Motor Drivers





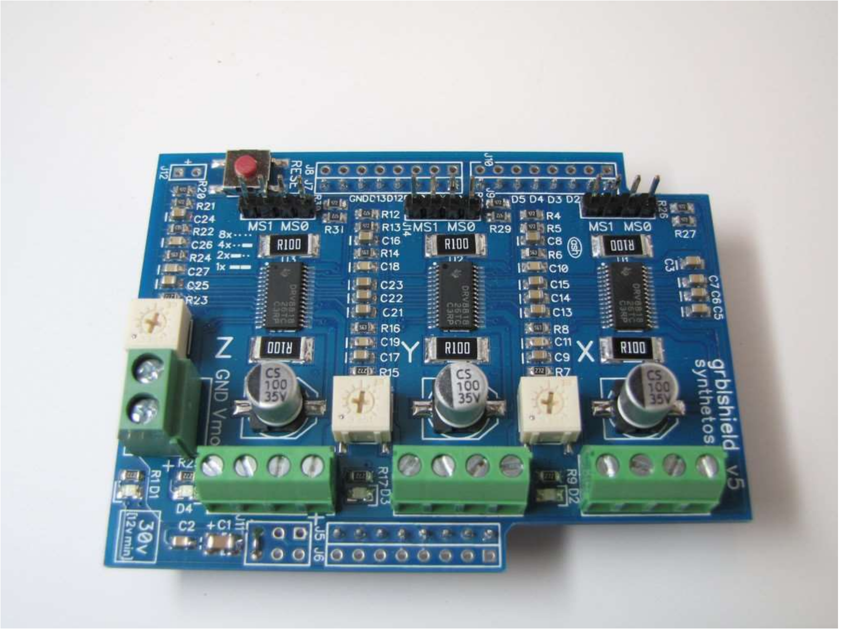




Model	DRV8825	A4988	TMC2208	ULN2003
Type	Bipolar	Bipolar	Bipolar	Unipolar
Max Current per coil	2.2 A	2.0 A	2.0 A	600 mA
Continuous current per coil	1.5 A	1.0 A	1.2 A	
Max step resolution	1/32	1/16	1/256	N/A
Logic range	2.5 – 5.25 V	3 – 5.5 V	3 – 5 V	5 V
Supply Voltage Range	8.2 – 45 V	8 – 35 V	5.5 – 36 V	50 V
Interface	Step/Dir	Step/Dir	Step/Dir	Channels
Approximate cost	$9.00	$2.00	$15.00	$1.50

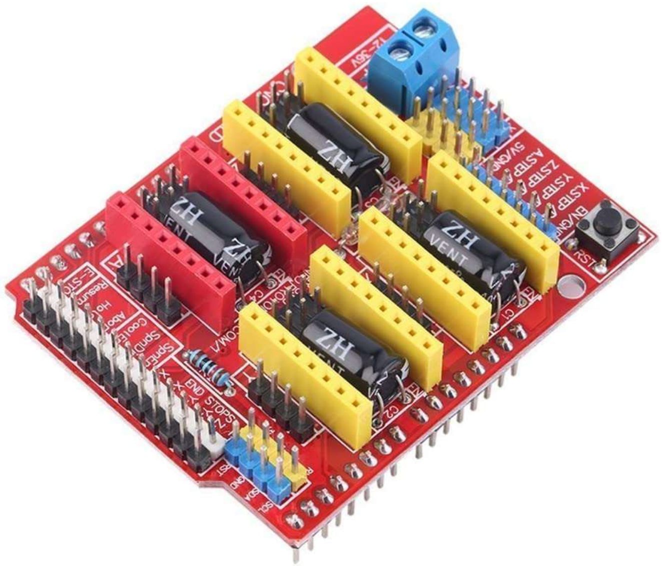
Integrated Stepper Motor Driver

- gShield v5
  - ~ $50
  - Arduino Uno Shield
  - 3x TI DRV8818 Stepper Drivers
  - Runs grbl firmware
  - https://github.com/gnea/grbl
  - Application: CNC Router



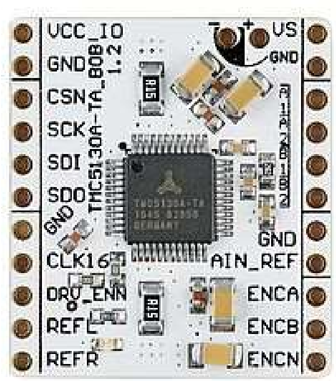
Integrated Stepper Motor Driver

- RAMPS 1.4
  - ~ $20 including DRV8825 Motor Drivers
  - Arduino Mega Shield
  - Different drivers may be used
  - Runs Marlin firmware
  - https://marlinfw.org/
  - Application: 3D printer



Motion Controller

Model	TMC5130A-TA
Type	MOSFET
Max Current per coil	2.5A
Continuous current per coil	2A
Max step resolution	1/256
Logic range	Internal or 4.6 – 5.25 V external
Supply Voltage Range	4.75 – 46V
Interface	SPI & Single Wire UART
Approximate cost	$20.00

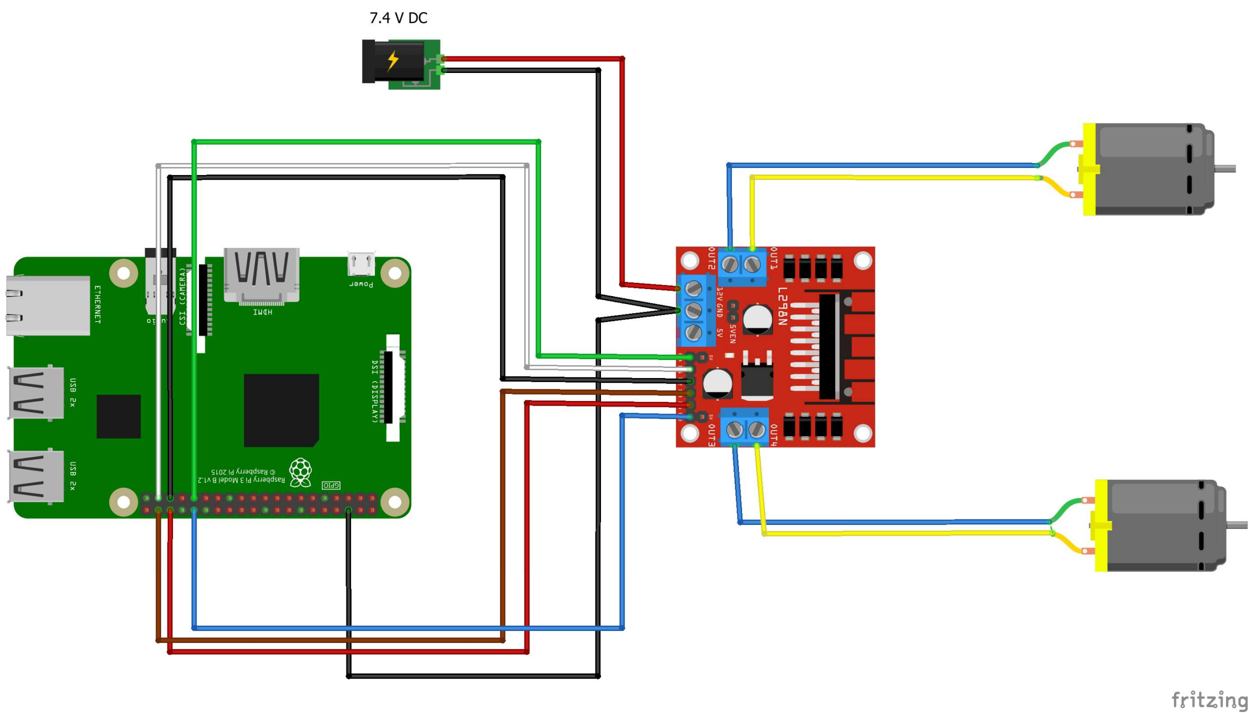


Project IDE

- Arduino IDE
  - Programming -> Arduino IDE
  - Arduino UNO or Arduino UNO clone
    - Tools -> Board -> Arduino UNO
  - Arduino Leonardo or Arduino Leonardo clone (Velleman - white board)
    - Tools -> Board -> Arduino Leonardo
- Raspberry Pi IDE
  - Programming -> Geany (for Brushed DC and Servo Projects)
  - Programming -> Thonny (for Raspberry Pi Pico project)

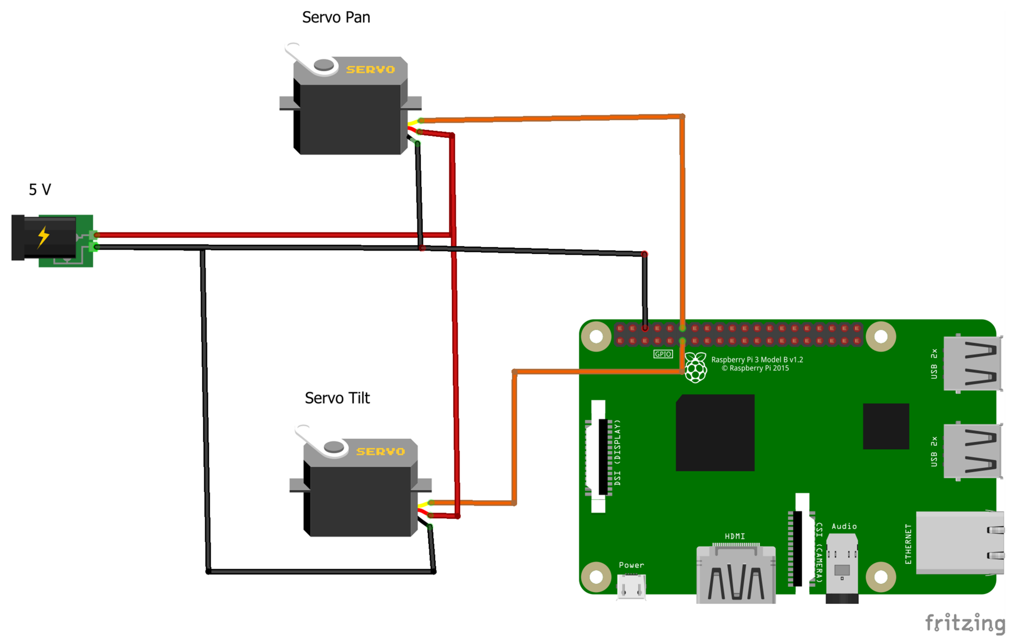
Project – Brushed DC Motor

- Driver: L298N
- Raspberry Pi Wiring
  - IN1 to GPIO26
  - IN2 to GPIO19
  - IN3 to GPIO20
  - IN4 to GPIO16
  - ENA to GPIO6
  - ENB to GPIO12
- Set power supply output to 7.4 V.
- Open MotionControlClass/BrushedDC/key_brushedDc.py using the Geany IDE.
- Run key_brushedDc.py.
- Put cursor focus in terminal window that pops up.
- Arrow keys to move motors.
- 's' key to slow the motor, 'f' key to speed up the motor.
- Enter key to stop.
- 'q' or ctrl-c to terminate program.
- Consider how to control the speed of the motors.



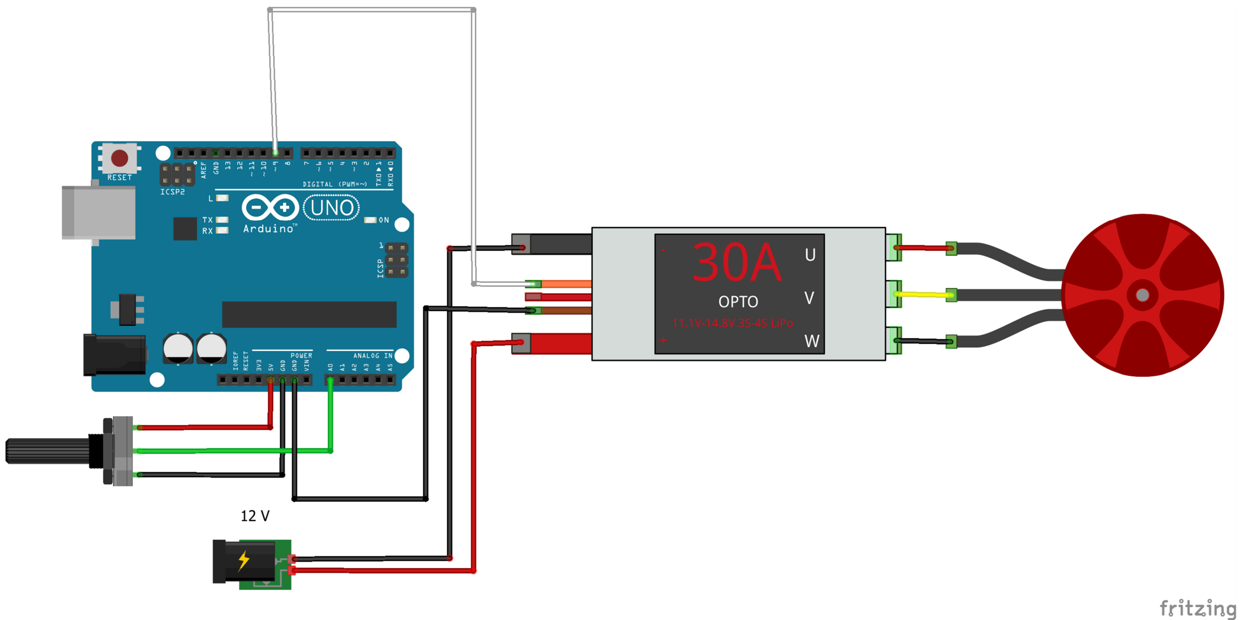
Project – Servo Motor

- Use mechanical crane
- Raspberry Pi Wiring:
  - Servo Pan: GPIO18
  - Servo Tilt: GPIO17
- Set power supply output to 5V
- Open MotionControlClass/ServoMotor/key_ServoMotor.py using the Geany IDE
- Run key_ServoMotor.py
- Put cursor focus in terminal window that pops up.
- Arrow keys to move motors.
- Enter key to stop
- ‘q’ to terminate program



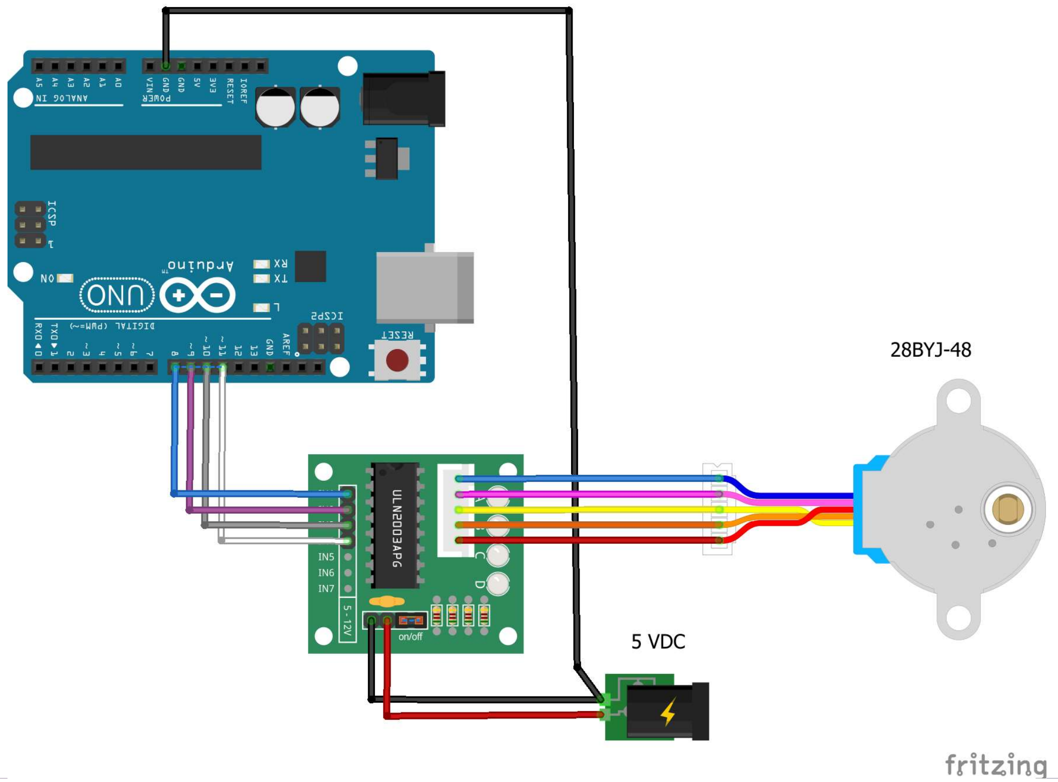
Project – Brushless DC Motor

- Driver: ESC
- Guide: https://howtomechatronics.com/tutorials/arduino/arduino-brushless-motor-control-tutorial-esc-bldc/
- Arduino Wiring:
  - Speed control: Pin 9 (White)
  - Pot Wiper: A0 (Green)
- Set power supply output to 12V but do not enable the power supply
- Open MotionControlClass/BldcMotor/BldcMotor.ino using the Arduino IDE
- Upload the sketch
- Turn pot full clockwise
- Enable the power supply
- The motor speed is controlled with the pot. If the pot is full on (counter clockwise) on then the motor will not start and must be calibrated (see Guide).



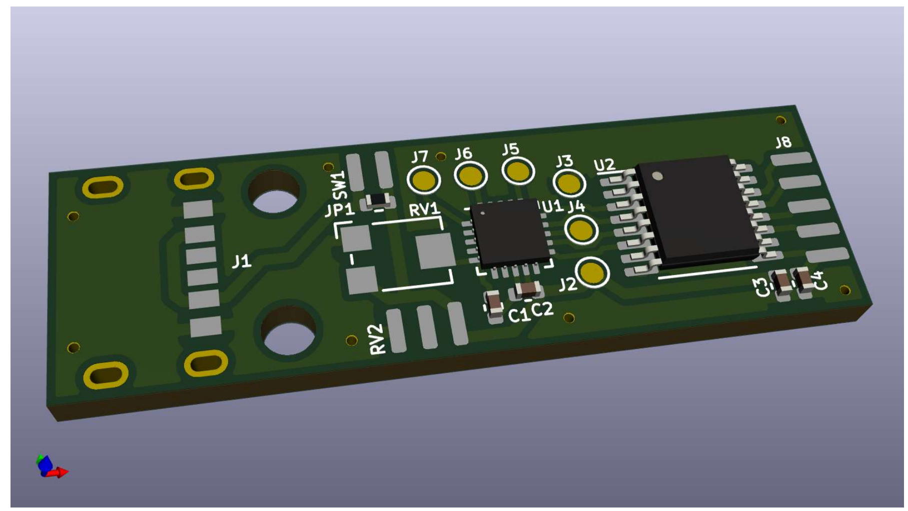
Project – Unipolar Stepper Motor

- Driver: ULN2003
- Guide: https://www.makerguides.com/28byj-48-stepper-motor-arduino-tutorial/
- Arduino Wiring
  - IN1 to Pin 11
  - IN2 to Pin 10
  - IN3 to Pin 9
  - IN4 to Pin 8
- Set power supply output to 5V
- Open MotionControlClass/UnipolarStepper/BasicUnipolarStepper/BasicUnipolarStepper.ino using the Arduino IDE
- Upload the sketch
- The motor will rotate 1 revolution, reverse and rotate 1 revolution in the opposite direction and repeat forever.
- Try changing the motor speed and/or the number of revolutions
- Observe in the Serial Monitor when the motor changes direction



Project: Unipolar with ATTiny

- Driver: ULN2003
- Plug in 28BYJ-48 motor with red wire nearest C3/C4
- Power board with USB-C power block
- The motor will rotate in the clockwise direction looking at the shaft side of the motor.



Project: Unipolar with ATTiny

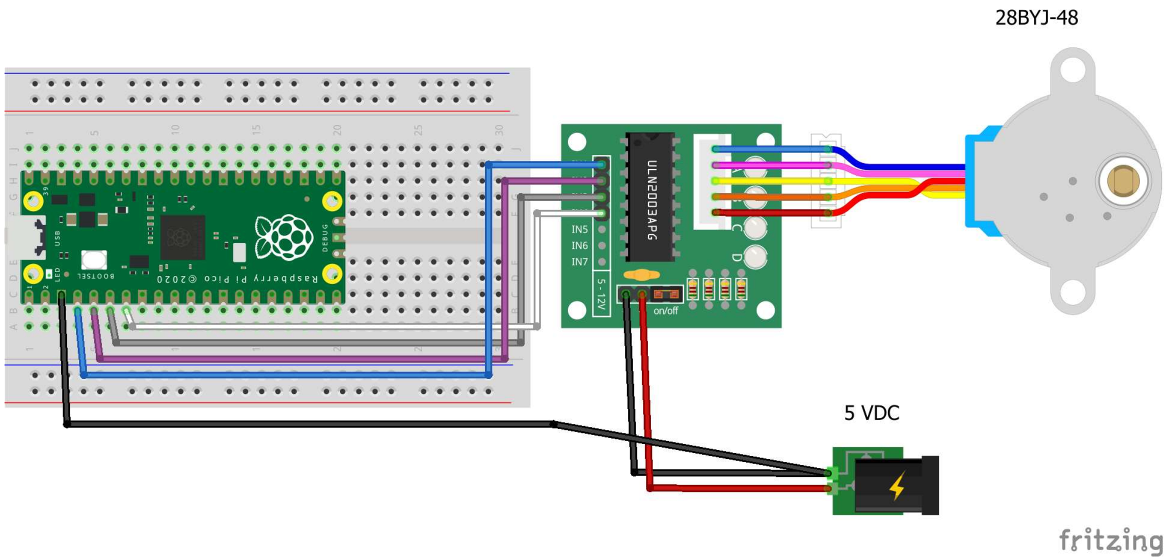
This ATTiny code snippet demonstrates how to half-step sequence the ULN2003 driver. This function is called in a forever loop that includes a delay:

```
// ------ Event loop ------ //
while (1) {
    _delay_us( StepDelay_us );
    decrMotor();
}
```

```
static inline void decrMotor( void ){
  switch( MotorState ){ //prior to decrement
  case 7: //1001->X___->0001
    clear_bit( S1_PORT, S1 );
    break;
  case 6: //0001->__^_->0011
    set_bit( S3_PORT, S3 );
    break;
  case 5: //0011->___X->0010
    clear_bit( S4_PORT, S4 );
    break;
  case 4: //0010->_^__->0110
    set_bit( S2_PORT, S2 );
    break;
  case 3: //0110->__X_->0100
    clear_bit( S3_PORT, S3 );
    break;
  case 2: //0100->^___->1100
    set_bit( S1_PORT, S1 );
    break;
  case 1: //1100->_X__->1000
    clear_bit( S2_PORT, S2 );
    break;
  case 0: //1000->___^->1001
    set_bit( S4_PORT, S4 );
    MotorState = 8;
    break;
  }  MotorState--;
}
```

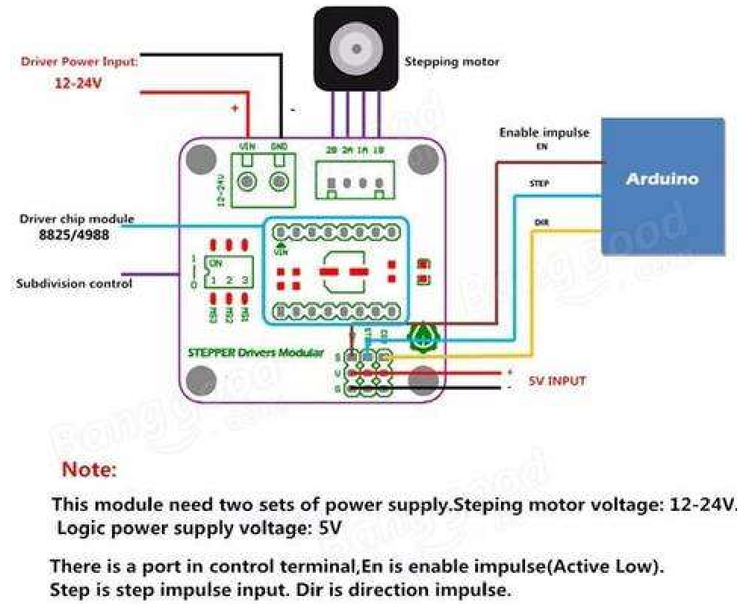
Project – Unipolar Stepper w/Pi Pico

- Driver: ULN2003
- Pi Pico Wiring
  - IN1 to GP2
  - IN2 to GP3
  - IN3 to GP4
  - IN4 to GP5
  - Power supply GND to GND on Pico
- Connect Pi Pico to Raspberry Pi Model B via USB
- Start the Thonny IDE and select MicroPython (Raspberry Pi Pico) in the bottom left corner.
- Load stepper.py from the Pi Pico
- Connect 5 V power
- Execute the program stepper.py
- The motor will rotate 1 revolution CW, reverse and rotate 1 revolution CCW and repeat for 50 seconds.
- Try changing the program to half step mode and observe what happens.



Project – Bipolar Stepper Motor

- Driver: DRV8825 or A4988
- Guide: https://www.makerguides.com/drv8825-stepper-motor-driver-arduino-tutorial/
- Arduino Wiring (from 9 pin block labeled S-V-G / EN-STEP-DIR):
  - EN (Enable impulse): jumper to GND (done: black wire)
  - DIR: Arduino Pin 2 (orange wire)
  - STEP: Arduino Pin 3 (yellow wire)
  - 5V INPUT: Arduino 5V and GND (red-V and brown-G wires)
- Set power supply output to 12V but do not enable the power supply
- If necessary, adjust current limit according to Guide: $V_{ref} = I_{max} / 2$ for DRV8825 or $V_{ref} = I_{max} * 8 * R_s$ for A4988 (R_s = 0.1 ohm for demo driver)
- Open MotionControlClass/BipolarStepper/BasicBipolarStepper/BasicBipolarStepper.ino using the Arduino IDE
- Upload the sketch
- Enable the power supply output
- The motor will move slowly one revolution (full step mode), a little faster in reverse direction and then quite fast in both directions for 5 revolutions and then repeat.
- Try changing subdivision control: Disable power supply, change DIP switch, Enable power supply. Consider how to improve the program by using variables to control the speed and number of revolutions.



Project – Bipolar Stepper Motor

- Acceleration library example
- Driver: DRV8825 or A4988
- Guide: https://www.makerguides.com/drv8825-stepper-motor-driver-arduino-tutorial/
- Arduino Wiring (from 9 pin block labeled S-V-G / EN-STEP-DIR):
  - EN (Enable impulse): jumper to GND (done: black wire)
  - DIIR: Arduino Pin 3 (orange wire)
  - STEP: Arduino Pin 2 (yellow wire)
  - 5V INPUT: Arduino 5V and GND (red-V and brown-G wires)
- Set power supply output to 12V but do not enable the power supply
- If necessary, adjust current limit according to Guide: $V_{ref} = I_{max} / 2$ for DRV8825 or $V_{ref} = I_{max} * 8 * R_s$ for A4988 (R_s = 0.1 ohm for demo driver)
- Open MotionControlClass/BipolarStepper/BasicBipolarStepper/AccelBipolarStepper.ino using the Arduino IDE
- Upload the sketch
- Enable the power supply output
- The motor will rotate 600 steps (3 revolutions in full step mode) but accelerate up to speed and decelerate when stopping.
- Try changing the number of total steps, max speed and acceleration: Disable power supply, make change and upload new sketch and enable power supply.