

LAB3-MC

Stepper Motors

CMP(N)211

Spring 2021

By: Salma Abdel Monem

Lab Objectives

- ✓ Control Stepper motors using H-bridge
- ✓ Use Potentiometers

Recap

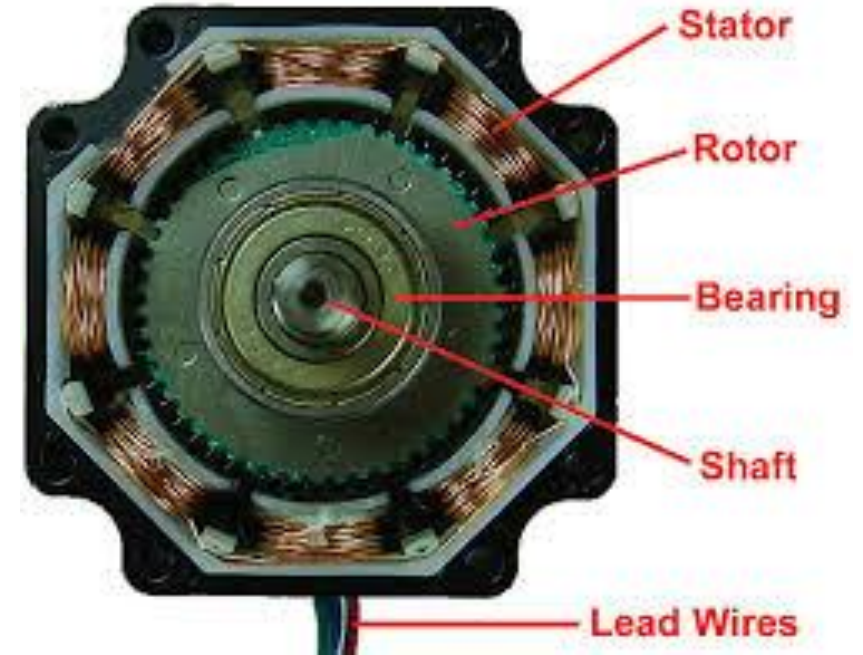
Till Now we have learnt the following:

- ✓ Read and write analog and Digital Signals
- ✓ Control Dc motors
- ✓ Temperature Sensors
- ✓ Now it's time to learn how to **control Stepper motors**

Stepper motors

They come in different sizes and different configurations :

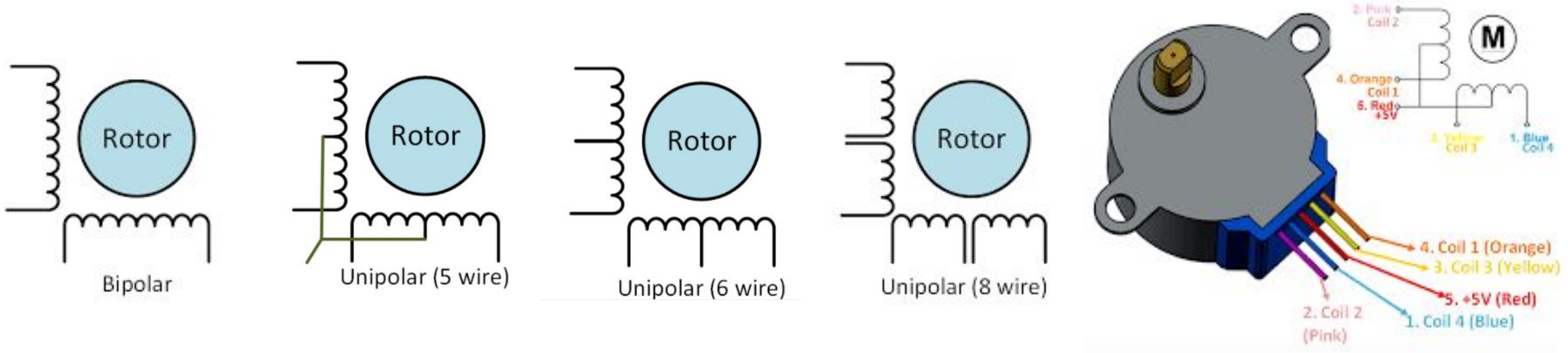
- ✓ Voltage
- ✓ Current
- ✓ Resistance
- ✓ Torque
- ✓ Unipolar/bipolar



Stepper motors

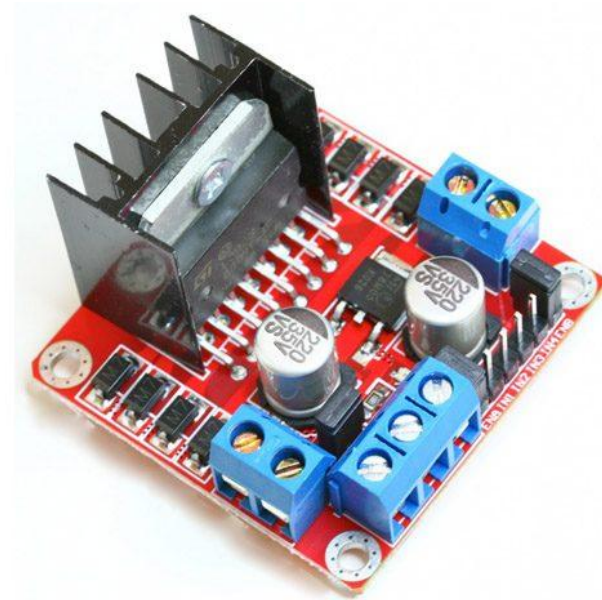
They come in different sizes and different configurations :

- ✓ Unipolar (each coil has specific polarity and maybe On/OFF)
- ✓ Bipolar (each coil changes its polarity regularly)
- ✓ A Bipolar has more torque but a Unipolar is simpler to connect.



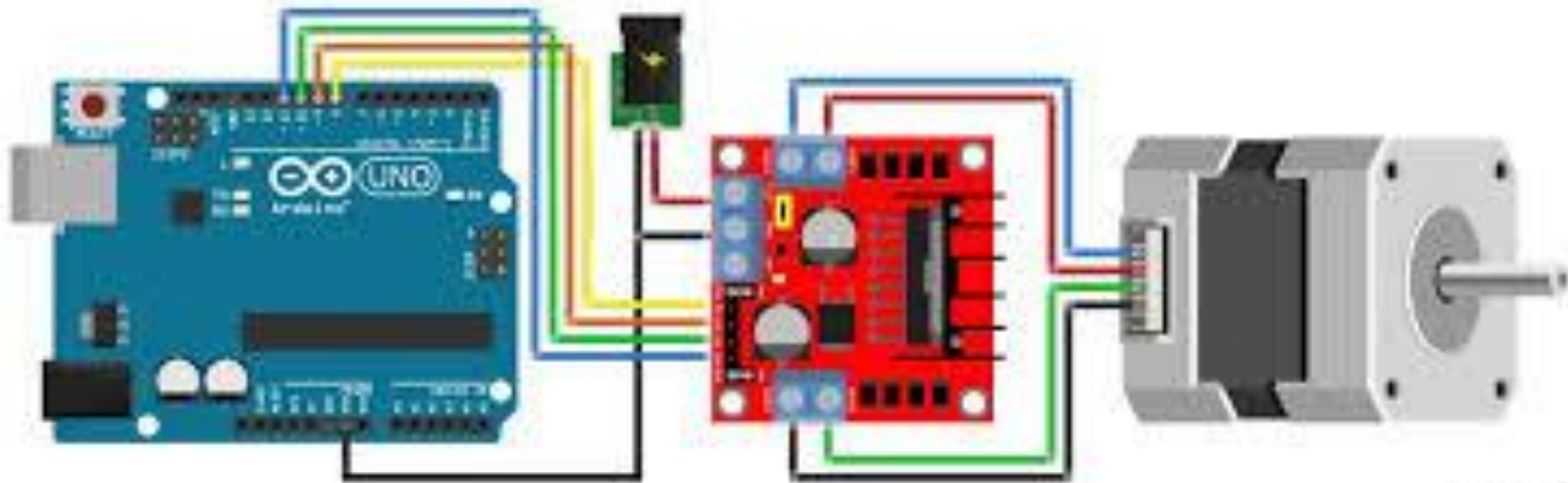
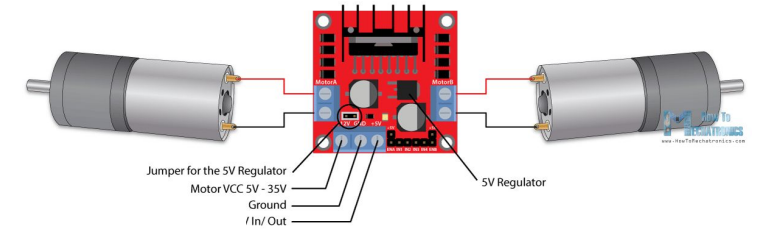
Stepper motors and H-Bridge

- ✓ The H-Bridge (L298p here) or other more complicated stepper motor drivers like (TB6600) can be used to control the direction of the stepper motor.



Stepper motors and H-Bridge

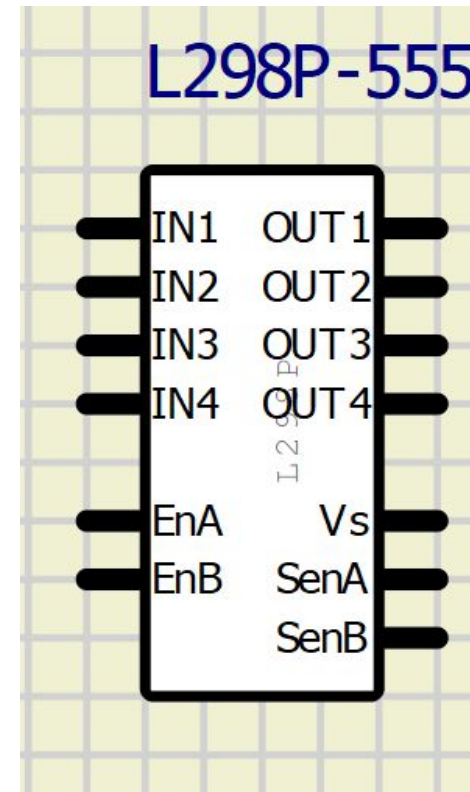
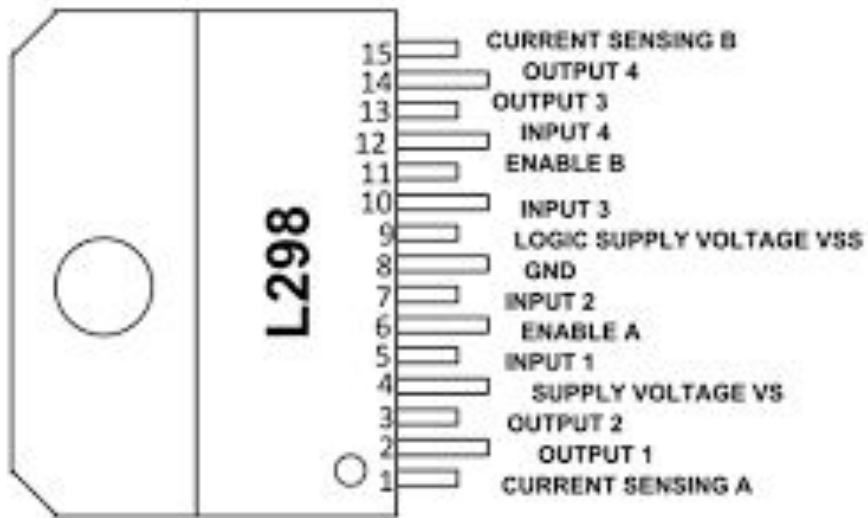
✓ L298p can connect two dc motors or one stepper motor.



fritzing

Stepper motors and H-Bridge

✓ Recall the L298p pinout



Stepper motors and H-Bridge

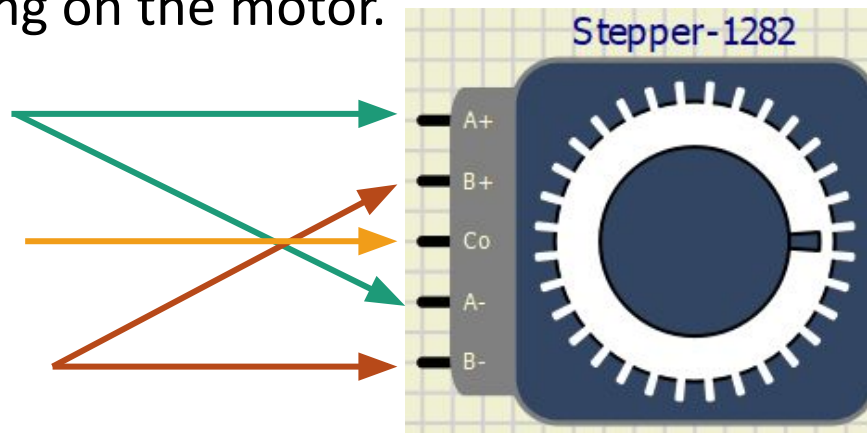
L298p Pinout

- ✓ IN1,IN2,IN3, and IN4 control direction of the motor (arduino)
- ✓ EnA and EnB Enables the motor (PWM arduino) (both should be 255)
- ✓ OUT1,OUT2 are connected to one pair of the motor
- ✓ OUT3 and OUT4 are connected to the other pair of the motor.
- ✓ Vs is connected to 5 v from arduino or external power source depending on the motor.

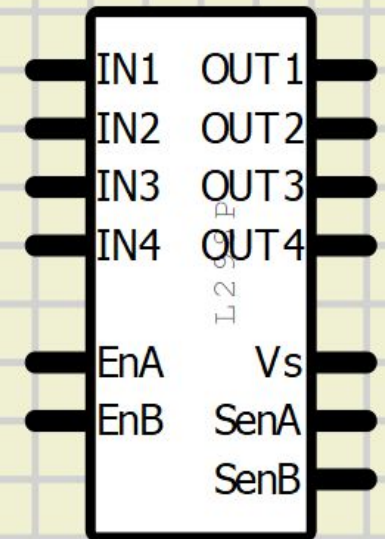
One Pair

Common =
VCC

One Pair

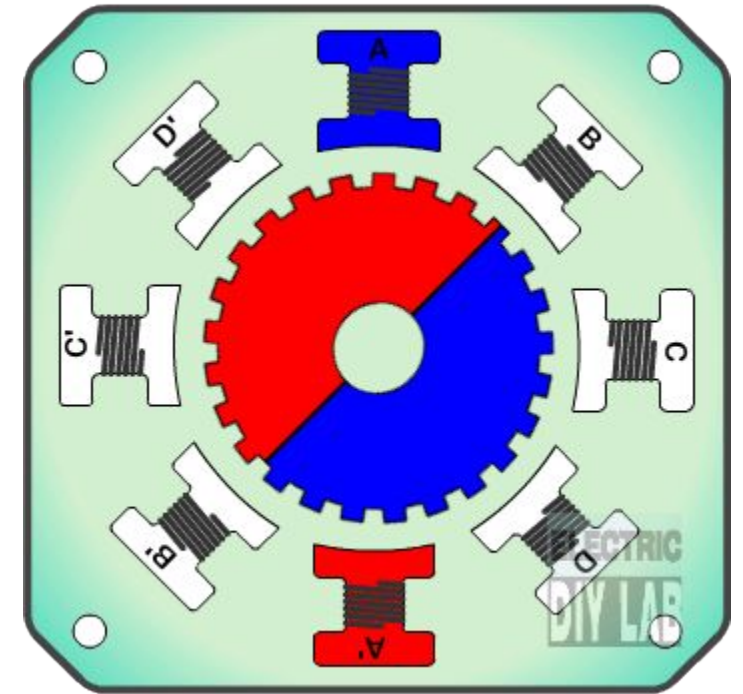


L298P-555



Stepper Speed and Steps per revolution

- ✓ The stepper's **step angle** is defined by the unit of movement (discrete steps) after a sequence of four steps.
- ✓ The **Steps per Revolution (SPR)** of the stepper defines the number of steps the motor needs to make to complete 360 degrees of motion angel.
- ✓ The (SPR) can be configured for some motors using the driver.
- ✓ **Example:** a motor with SPR=200 has a step angle of $360/200=1.8$ degrees .
- ✓ Another parameter to control is the **Speed** of the stepper movement from step to the other .



Arduino Stepper library

- ✓ Use the pins {8,9,10,11} in order (In1,In2,In3,In4) for correct movement.
- ✓ Don't forget about the Enable PWM.

```
#include <Stepper.h>

const int stepsPerRevolution = 200; // change this
//to fit the number of steps per revolution
// for your motor

// initialize the stepper library on pins 8 through 11:
Stepper myStepper(stepsPerRevolution, 8, 9, 10, 11);

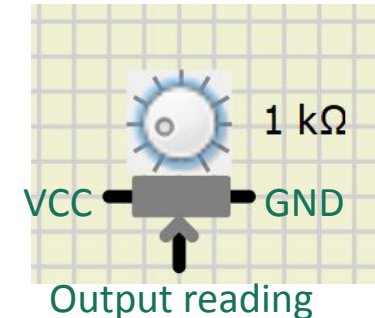
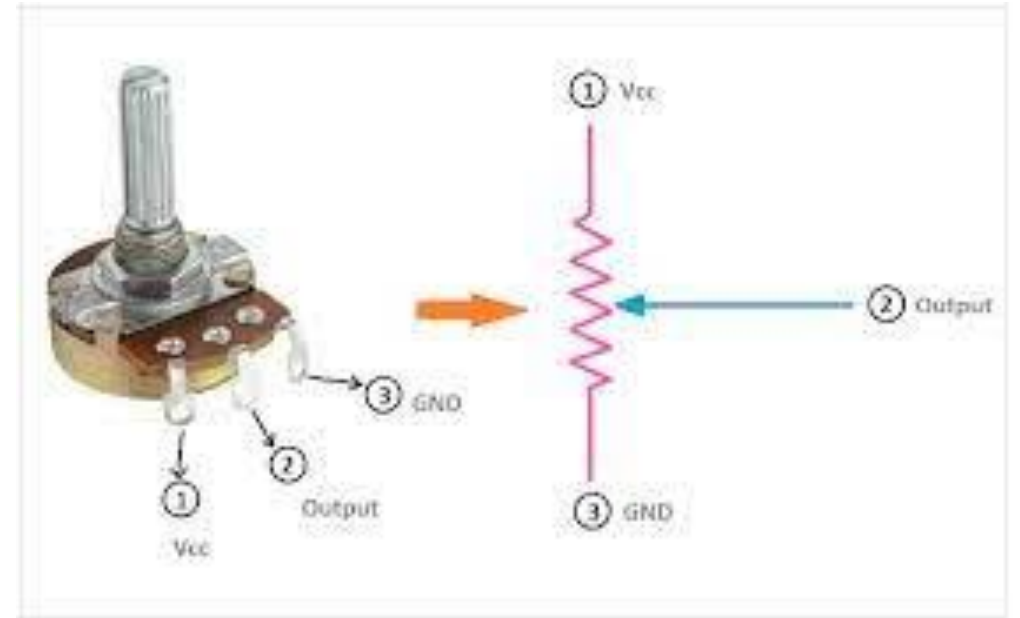
void setup() {
  // set the speed at 60 rpm:
  myStepper.setSpeed(60);
  // initialize the serial port:
  Serial.begin(9600);
}

void loop() {
  // step one revolution in one direction:
  Serial.println("clockwise");
  myStepper.step(stepsPerRevolution);
  delay(500);

  // step one revolution in the other direction:
  Serial.println("counterclockwise");
  myStepper.step(-stepsPerRevolution);
  delay(500);
}
```

Potentiometer

- ✓ A **potentiometer** is a simple knob that provides a variable resistance, which we can read into the Arduino board as an analog value.
- ✓ The analog input can vary from { 0 to 1023 } .



Exercise

- Configure the Stepper motor in the Unipolar mode and SPR =200.
- At the beginning of the program (only once), Read the speed of the motor from the serial monitor . The speed is a two digits number only.
- Recall that Serial.read() reads one byte at a time. You can convert the byte to it's integer by subtracting char '0' from it.
- Use one Tact switch to control whether the motor will move clockwise (Tact high) or counterclockwise (Tact low).
- Use the Potentiometer to control the number of steps the motor will move. Map the Potentiometer reading [0 to 1024] **linearly** to the steps numbers [0 to 200]. Hint: use the map function in arduino. `val = map(adc_val, 0, 1023, 0, 255);`
- At each loop the stepper motor should move according to the given number of **steps** , **direction** and **speed**. After each move set a 0.5 second delay to let the motor move correctly.
- Print on the serial monitor the speed, direction and steps at each move.