***Using the stepper motor***

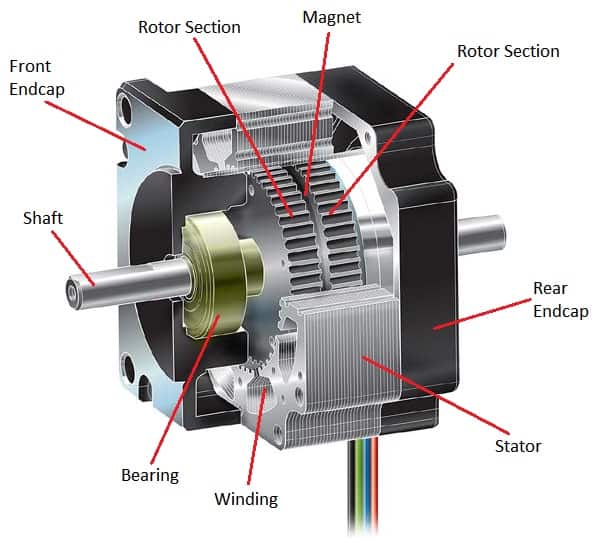
***Stepper Motor –***

* Stepper motor is a BLDC motor whose direction and speed can be easily handled using the PWM.
* Basically, it has commutation controlled electrically using the driver.
* Driver basically takes the information from microcontroller and accordingly it acts as a level shifter to provide the appropriate voltage and current to the motor.
* It uses the technique called micro stepping to further increase the step of the motor.

***Component used –***

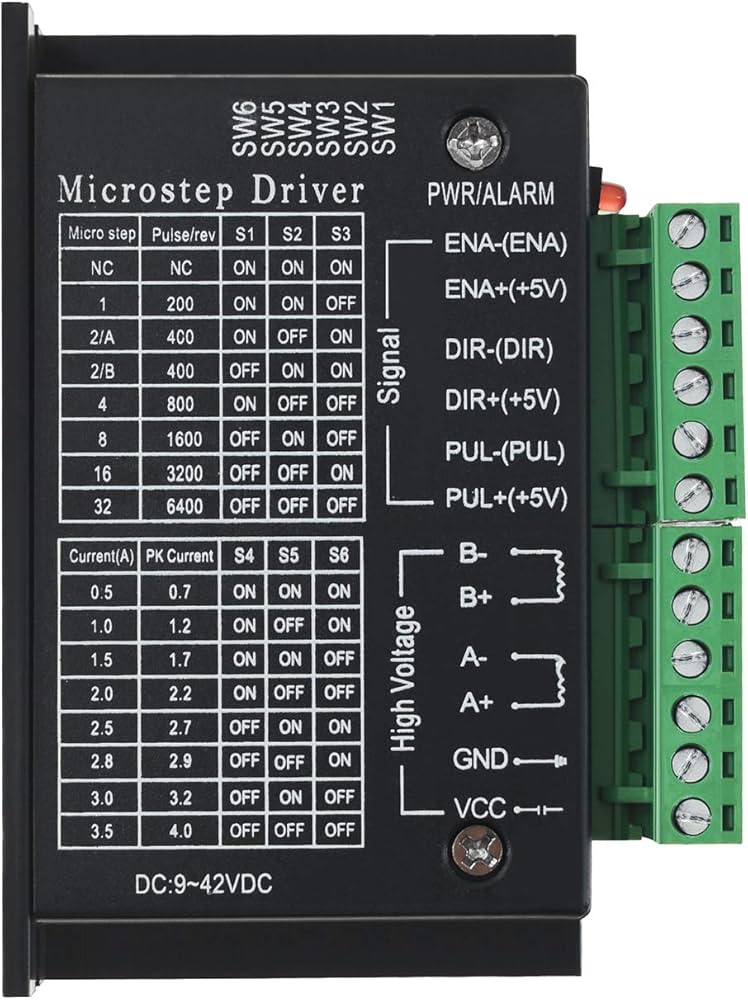
***Nema 17 Stepper*** –

* It’s the small stepper motor of 200 step means step angle of 1.8 degree.
* I used the Nema 17 of Three pole pair stator coil and 4 wires.



***TB6600 Driver –***

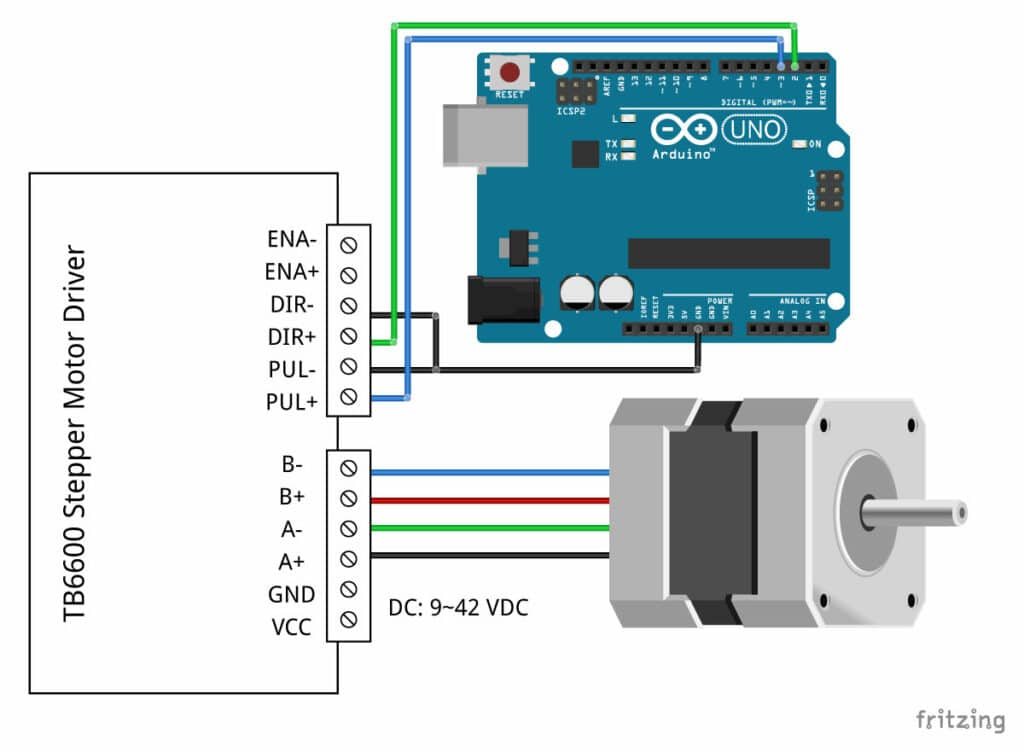
It’s a gate driver that can drive the motor of different current rating and the number of steps that we want to use.



***Connections*** –

* There are two type of connections that can be made of the driver

1. Common cathode connection
2. Common anode connection

* I did the common cathode connection for our use.
* I shorted the DIR- and PUL- pin and connected it to the ground.
* And direction pin to the pin 2 of the Arduino uno and pulse pin to pin 3 of the Arduino uno.
* Now A+ and A- to the two shorted pins of the motor and for B did same.
* Gave the Vcc supply of the 12 volts.
* 

***Arduino code for controlling the angle of the Stepper using the potentiometer***-

*#define pwmpin 9*

*#define dirpin 8*

*#define stepsPerRevolution 3200*

*#define analogpin A0*

*volatile int current\_angle = 0;*

*volatile int previous\_angle = 0;*

*volatile bool forward = false;*

*void setup() {*

*// put your setup code here, to run once:*

*pinMode(pwmpin,OUTPUT);*

*pinMode(dirpin,OUTPUT);*

*}*

*void loop() {*

*// put your main code here, to run repeatedly:*

*int stear\_input = analogRead(analogpin);*

*current\_angle = map(stear\_input,170,520,0,stepsPerRevolution);*

*if(previous\_angle < current\_angle){*

*for(int i = previous\_angle;i<current\_angle;i++){*

*digitalWrite(dirpin,HIGH);*

*digitalWrite(pwmpin,HIGH);*

*delayMicroseconds(100);*

*digitalWrite(pwmpin,LOW);*

*delayMicroseconds(100);*

*previous\_angle++;*

*}*

*}else if(previous\_angle>current\_angle){*

*for(int i = previous\_angle;i>current\_angle;i--){*

*digitalWrite(dirpin,LOW);*

*digitalWrite(pwmpin,HIGH);*

*delayMicroseconds(100);*

*digitalWrite(pwmpin,LOW);*

*delayMicroseconds(100);*

*previous\_angle--;*

*}*

*}*

*}*