American International University- Bangladesh

Department of Electrical and Electronic Engineering

EEE4103: Microprocessor and Embedded Systems Laboratory

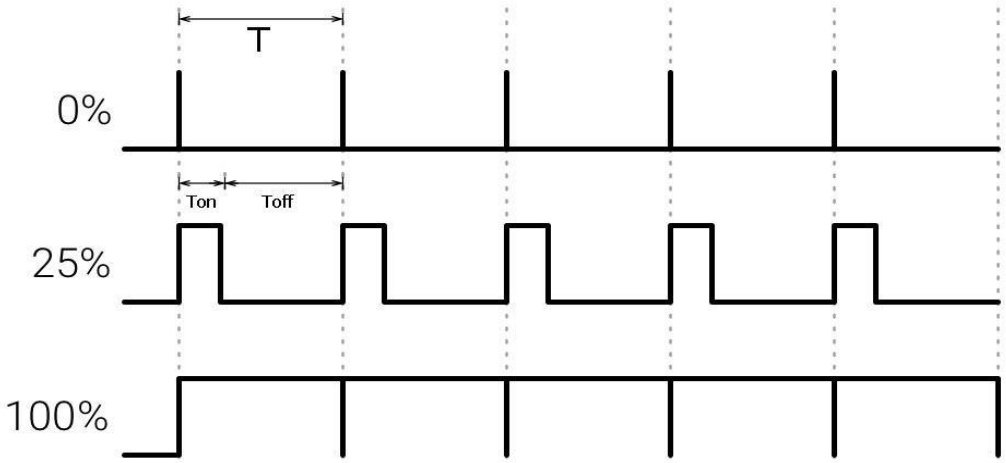
Title: Implementation of a motor control system using Arduino: Digital input, outputs, and PWM

Introduction: The objective of this experiment is to get familiarized with Microcontroller based motor speed control.

Theory and Methodology:

Microcontrollers and Arduino are digital devices; they cannot give analog output. Microcontroller gives Zero and ONE as output, where ZERO is logical LOW and ONE is logical HIGH. In our case, we are using a 5-volt version of the Arduino. So its logical ZERO is zero voltage, and logical HIGH is 5 voltage.

The digital output is good for digital devices but sometimes we need analog output. In such a case the PWM is very useful. In the PWM, the output signal switches between zero and one, on a high and fixed frequency, as shown in the figure below.



Output Signal Of PWM

As shown in the above figure the ON time is T_{on} and the OFF time is T_{off}. **T is the sum of the T_{on} and T_{off}, which is called the Period.** In the concept of PWM, T is not varying and the T_{on} and the T_{off} can vary, in this way when T_{on} increase T_{off} will decrease, and T_{off} increase when T_{on} decrease proportionally.

The duty cycle is a fraction of one Time period. The duty cycle is commonly expressed as a percentage or a ratio. A period is a time it takes for a signal to complete an on-and-off cycle. As a formula, a duty cycle may be expressed as:

DUTY CYCLE = (Ton / T) x100 %

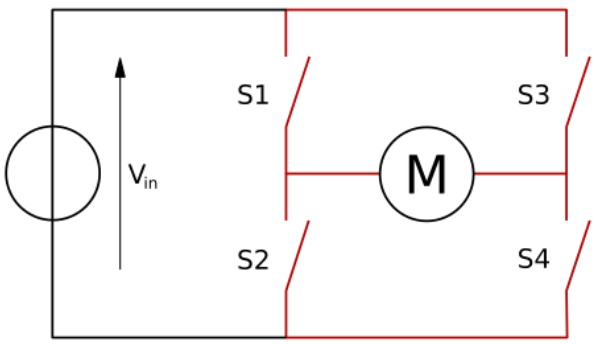
Now the motor speed varies according to the duty cycle. Suppose the duty is zero, the motor does not run, and when the duty cycle is 100 % the motor moves on maximum RPM**. But this concept is not always right because the motor starts running after giving some fixed voltage that is called threshold voltage.**

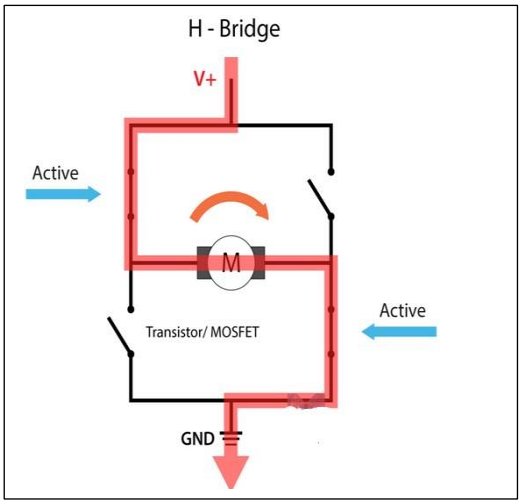
Microcontroller and the Arduino can process signals and **consumes almost 20 to 40mA current but motors need high current and voltage, so we are using the transistor for driving the motor.** Transistor is connected in ***series*** with the motor and the ***transistor's base is connected to Arduino's PWM pin through a resistance.*** PWM signal is coming from Arduino and the transistor works as a switch and it **short circuits the Emitter (E) and Collector (C) when the PWM signal is in a High state** and ***normally opens when the PWM signal is in a LOW state***. This process works continuously and the motors run at the desired speed.

H-Bridge DC Motor Control

If you're a bit familiar with DC motors (motors widely used in toys and robots) you'll notice that if you switch polarity the rotation direction will change too, so in order to go forward and backward you need to switch the wires… but that's not practical, you need something to do it electronically without modifying any wires: This is where the H-bridge comes handy.

For controlling the rotation direction, we just need to inverse the direction of the current flow through the motor, and the most common method of doing that is by using an H-Bridge. An H-Bridge circuit contains four switching elements, transistors or MOSFETs, with the motor at the center forming an H-like configuration. **By activating two particular switches at the same time we can change the direction of the current flow, thus changing the rotation direction of the motor.**



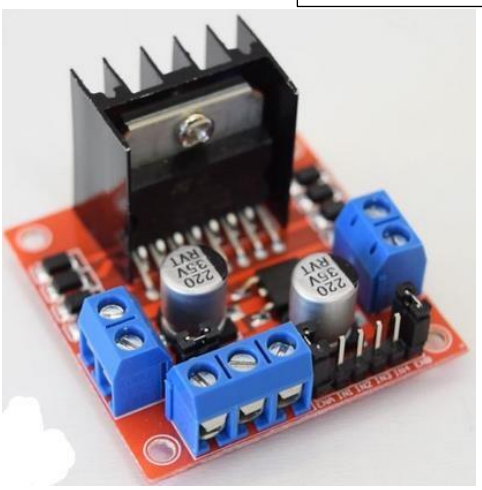


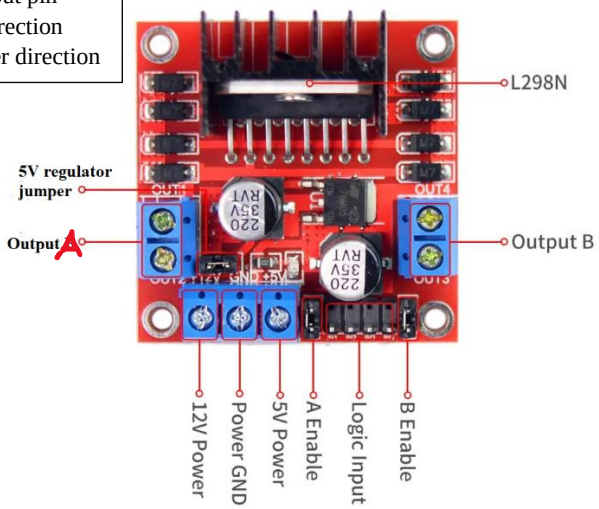
So if we combine these two methods, the PWM and the H-Bridge, we can have complete control over the DC motor. Many DC motor drivers have these features and the L298N is one of them.

L298N Driver

The module used is a L298N Dual H-bridge, it's often used with Arduino, it can control 2 DC motors at the same time, and you can control the direction and the speed as well. This module can control a Stepper motor as well. The module can drive DC motors that have voltages between 5 and 35V, with a peak current up to 2A.

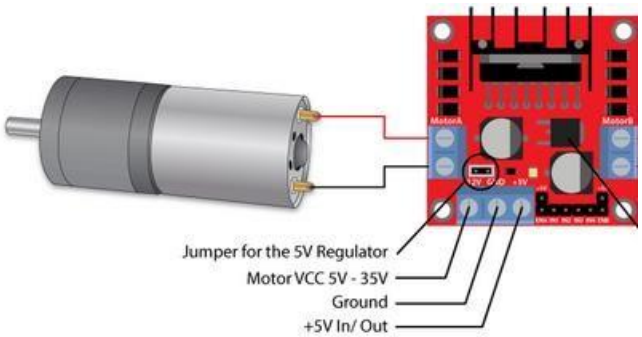
```
// in1 and in2 = for logic input pin
// HIGH and LOW = one direction
// LOW and HIGH = another direction
```



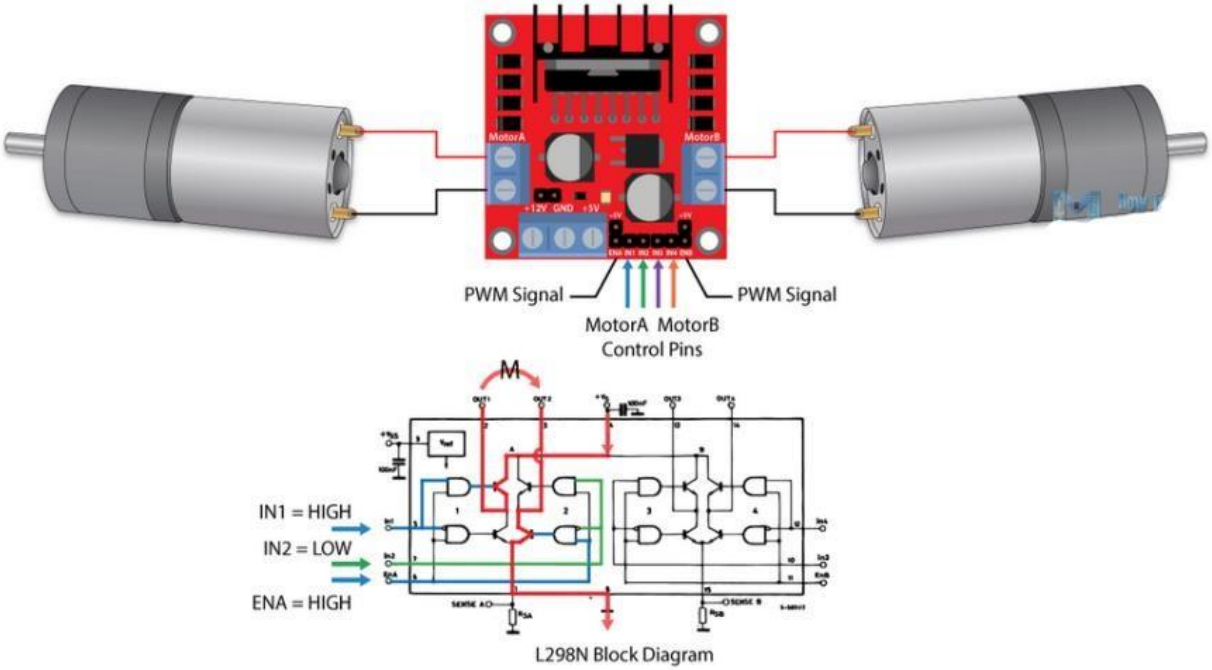


Let's take a closer look at the pinout of the L298N module and explain how it works. The module has two screw terminal blocks for motor A and B, another screw terminal block for the Ground pin, the VCC for the motor, and a 5V pin which can either be an input or output.

- The "Logic Input" pins control the directions: Forward, Backward and Stop, each two of them control A motor.
- 12V power is not always 12V it can be 9V, or it can be powered using up to 47V DC but you have to remove the regulator jumper or you'll burn it, the regulator can support only up to 12V.
- Enable A/B are for controlling the speed, if their jumpers are kept, the speed will be the maximum, they can handle up to 5V
- The module can be powered using Arduino but that's not recommended at all, it's better to use external power, and you can power the Arduino through the module too via the 5V/Gnd pins
- Don't forget that the GND pin should always be wired with Arduino.



Next are the logic control inputs. The Enable A and Enable B pins are used for enabling and controlling the speed of the motor. ***If a jumper is present on this pin, the motor will be enabled and work at maximum speed, and if we remove the jumper we can connect a PWM input to this pin and in that way control the speed of the motor. If we connect this pin to a Ground the motor will be disabled.***

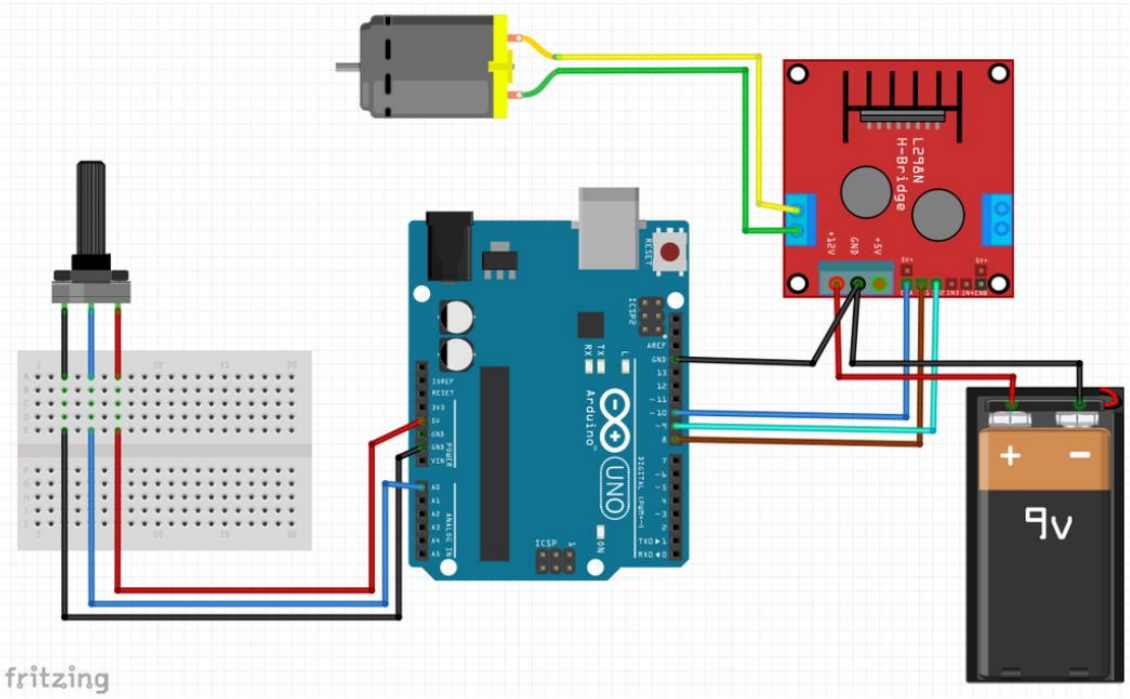


L298N Block Diagram

Next, the Input 1 and Input 2 pins are used for controlling the rotation direction of motor A, and inputs 3 and 4 for motor B. Using these pins we control the switches of the H-Bridge inside the L298N IC. If input 1 is LOW and input 2 is HIGH the motor will move forward, and vice versa, if input 1 is HIGH and input 2 is LOW the motor will move backward. In case both inputs are the same, either LOW or HIGH the motor will stop. The same applies to inputs 3 and 4 and motor B.

Arduino and L298N

Now let's make some practical applications. In the first example, we will ***control the speed of the motor using a potentiometer***. Here's the circuit schematic diagram.



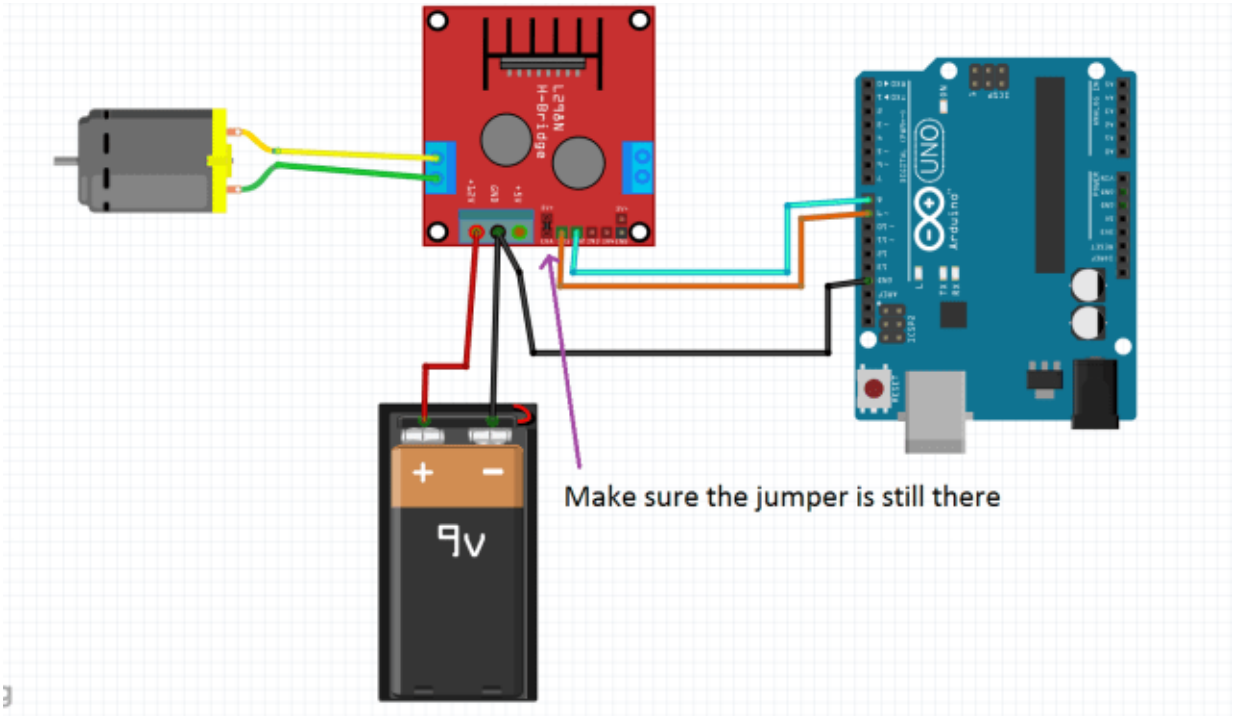
So we need an L298N driver, a DC motor, a potentiometer, and an Arduino board.

Components List

- L298N Driver
- 12V High Torque DC Motor
- Arduino Board
- Potentiometer
- A power supply
- Breadboard and Jump Wires

Program to be written in Arduino IDE:

Part-1: Wirings: (No speed control)



Code-1: (No speed control)

```
//This code is to use with L298n Dual H-bridge motor driver
//It just turns on a DC motor for a certain time and turn it off

int in1 = 9; //Declaring the pins where in1 in2 from the driver are wired
int in2 = 8; //here they are wired with D9 and D8 from Arduino

void setup() {
  pinMode(in1, OUTPUT); //Declaring the pin modes, obviously they're outputs
  pinMode(in2, OUTPUT);

}

//Before starting the loop you should create functions type "void" to control the driver's pins
//Here I created two functions, the first one turns a motor to a direction (you can change it by switching LOW and HIGH
//and the second one to stop the motor

void TurnMotorA(){
  digitalWrite(in1, HIGH);
```

```
// in1 and in2 = for logic input pin
// HIGH and LOW = one direction
// LOW and HIGH = another direction
```

```
 digitalWrite(in2, LOW);
}


void TurnOFFA(){
 digitalWrite(in1, LOW);
 digitalWrite(in2, LOW);
}
void loop() {
 TurnMotorA(); //in the loop we use the function to turn the motor for 3s and stop it for 2s
 delay(3000);
 TurnOFFA();
 delay(2000);
}
```

Code 2: Turn on and change the direction (No speed control)

```
//This code is to use with L298n Dual H-bridge motor driver
//It just turns on a DC motor for a certain time in a direction, turn it off, turn in the other direction and turn it off again

int in1 = 9; //Declaring the pins where in1 in2 from the driver are wired
int in2 = 8; //here they are wired with D9 and D8 from Arduino

// in1 and in2 = for logic input pin
// HIGH and LOW = one direction
// LOW and HIGH = another direction

void setup() {
 pinMode(in1, OUTPUT); //Declaring the pin modes, obviously they're outputs
 pinMode(in2, OUTPUT);

}

//Before starting the loop you should create functions type "void" to control the driver's pins
//Here I created three functions, one to turn the motor in a direction "#1", the other one to the other direction "#3"
//and the second one to stop the motor
//For changing directions you switch the HIGH with LOW and vice-versa

void TurnMotorA(){
 digitalWrite(in1, HIGH);
 digitalWrite(in2, LOW);
}

void TurnOFFA(){
 digitalWrite(in1, LOW);
 digitalWrite(in2, LOW);
}
void TurnMotorA2(){
 digitalWrite(in1, LOW);
 digitalWrite(in2, HIGH);
}

void loop() {
 TurnMotorA(); // We turn to direction 1 for 3s then stop for 2s
 delay(3000);
 TurnOFFA();
 delay(2000);
 TurnMotorA2(); // We turn to direction 2 for 3s then stop for 2s
 delay(3000);
 TurnOFFA();
 delay(2000);
}
```

Part-2: Wirings: For speed control

Don't forget !! Make sure 'A Enable' or/and 'B Enable' jumper is open

Don't forget !! The analog pin of the L298N (EnA/EnB) should be used with an Arduino PWM pin.

Code-3: Speed control

```
//This code is to use with L298n Dual H-bridge motor driver
//It just turns on a DC motor for a certain time with a low speed and turn it off then turn on with high speed
//refer to surtrtech.com for more information

int in1 = 9; //Declaring the pins where in1 in2 from the driver are wired
int in2 = 8; //here they are wired with D9 and D8 from Arduino
int EnA = 10; //And we add the pin to control the speed after we remove its jumper
          //Make sure it's connected to a pin that can deliver a PWM signal

void setup() {
  pinMode(in1, OUTPUT); //Declaring the pin modes, obviously they're outputs
  pinMode(in2, OUTPUT);
  pinMode(EnA, OUTPUT);
}

//Before starting the loop you should create functions type "void" to control the driver's pins
//here I created three functions, the first one is to turn the motor to a direction with speed (100)
//The second one to turn it off
//And the last one to turn it in the same direction as the first but higher speed (250)
//Speed range (0-255)

void TurnMotorA(){
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  analogWrite(EnA,100);
}

void TurnOFFA(){
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  analogWrite(EnA,0);
}
void TurnMotorA2(){
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  analogWrite(EnA,255);
}

void loop() {
  TurnMotorA();  //Sequence: turning on low speed, stop, turning again in high speed and stop
  delay(2000);

  TurnOFFA();
  delay(2000);

  TurnMotorA2();
  delay(4000);

  TurnOFFA();
  delay(2000);
}
```

```
// in1 and in2 = for logic input pin
// HIGH and LOW = one direction
// LOW and HIGH = another direction
```

Part-3: Control DC motor speed using potentiometer + L298N + Arduino

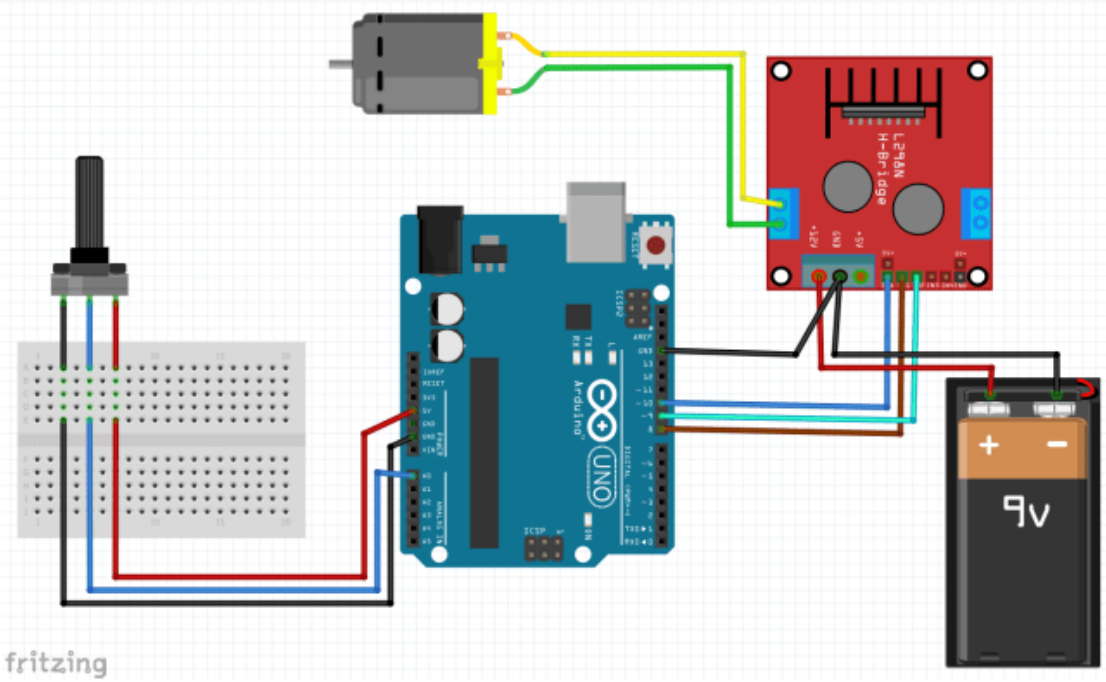
Code (a): Calibrating the potentiometer so it suits the values needed for the driver

```
Serial.begin(9600);
}

void loop() {

int value = analogRead(A0); //declaring and reading value from the pin
value = value*0.2492668622; // doing calibration to change range from 0-1023 to 0-255 the number
                            //is obtained by 255/1023

Serial.println(value);
}
```



Code-4: Speed control using potentiometer

```
//We read the value from the analog input, calibrate it then inject to the module

int in1 = 8; //Declaring where our module is wired
int in2 = 9;
int EnA = 10;// Don't forget this is a PWM DI/DO
int speed1;

void setup() {
pinMode(8, OUTPUT);
pinMode(9, OUTPUT);
pinMode(10, OUTPUT);
}

void TurnMotorA(){ //We create a function which control the direction and speed
digitalWrite(in1, LOW); //Switch between this HIGH and LOW to change direction
digitalWrite(in2, HIGH);
speed1 = analogRead(A0);
speed1 = speed1*0.2492668622; //We read the analog value from the potentiometer calibrate it
analogWrite(EnA,speed1);// Then inject it to our motor
}


void loop() {
TurnMotorA(); //one function that keeps looping you can add another one with different direction or stop
}
```

```
// in1 and in2 = for logic input pin
// HIGH and LOW = one direction
// LOW and HIGH = another direction
// LOW and LOW = motor stop
// HIGH and HIGH = motor stop
```

Questions for report writing:

1) Include all codes and scripts into the lab report following the writing template mentioned in appendix A of Laboratory Sheet Experiment 8.
2) Implement the system in the Tinkercad or proteus simulation tool by following the link: https://youtu.be/q_B3yAM4PH0

References:

[1] https://surtrtech.com/2018/01/27/control-dc-motor-speed-using-potentiometer-l298n-arduino/ accessed on November 18, 2022

[2] https://surtrtech.com/2018/01/27/step-by-step-on-how-to-use-the-l298n-dual-h-bridge-driver-with-arduino/ accessed on November 18, 2022

Appendix

```
// in1 and in2 = for logic input pin
// HIGH and LOW = one direction
// LOW and HIGH = another direction
// LOW and LOW = motor stop
// HIGH and HIGH = motor stop
```

```
int in1 = 8; //Declaring where our module is wired
int in2 = 9;
int EnA = 10; // Don't forget this is a PWM DI/DO
int speed1;
void setup() {
Serial.begin(9600);
pinMode(8, OUTPUT);
pinMode(9, OUTPUT);
pinMode(10, OUTPUT);
}
void TurnMotorA(){ //A function to control the direction and speed
digitalWrite(in1, LOW); //Switch between this HIGH and LOW states to change direction
digitalWrite(in2, HIGH);
delay(5000)
digitalWrite(in1, HIGH); //Switch between this HIGH and LOW states to change direction
digitalWrite(in2, LOW);
delay(3000)
speed1 = analogRead(A0);
speed1 = speed1*0.2492668622; //Analog value is read from the potentiometer to calibrate it
analogWrite(EnA,speed1); // To activate the motor
}
void loop() {
int value = analogRead(A0); //declaring and reading value from the pin
value = value*0.2492668622; // doing calibration to change range from 0-1023 to 0-255 the
                            // number and is obtained by 255/1023
Serial.println(value);
TurnMotorA(); //one function that keeps looping you can add another one with a different
              //direction or stop
}
```