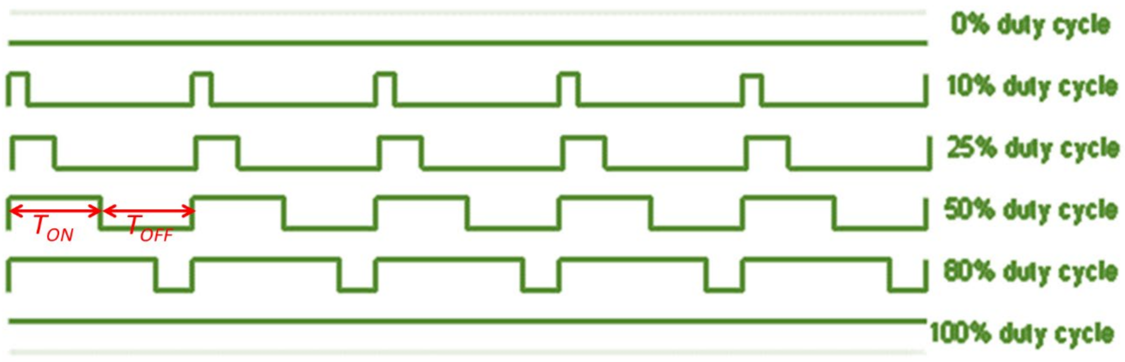
American International University- Bangladesh
Department of Electrical and Electronic Engineering
EEE4103: Microprocessor and Embedded Systems Laboratory

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

Objective:
The objectives of this experiment are to get familiarized the students with the PWM signals generated by the Arduino and the microcontroller-based DC motor's 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 V as the logical HIGH of the Arduino and 0 V as the logical LOW signal.

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. Pulse Width Modulation (PWM) is a technique by which the width of a pulse is varied while keeping the frequency or time period of the wave constant. It is a method for generating an analog signal using a digital source. A PWM signal consists of two main components that define its behavior: a duty cycle and a frequency. Pulse Width Modulated signals with different duty cycles are shown below:



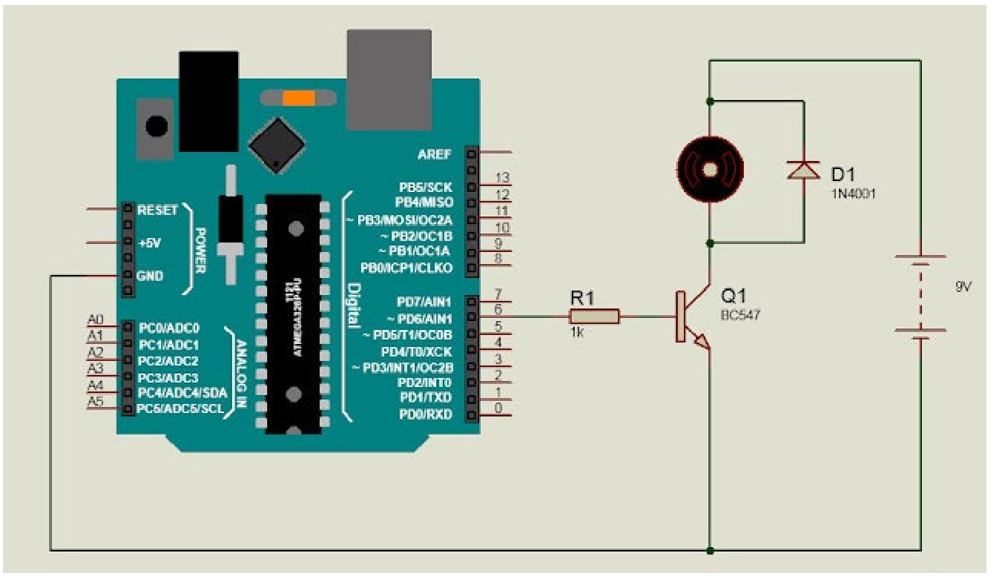
Output Signal Of PWM
As shown in the above figure the ON time is T_{on} and the OFF time is T_{off}. **T is the sum of the T_{on} and T_{off}, which is called the Period, i.e., Time period, $T = T_{ON} + T_{OFF}$.** In the concept of PWM, T is not varying, and the T_{on} and the T_{off} can vary, in this way when T_{on} increases T_{off} decreases, and vice-versa, but T remains fixed.

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

$$D = \frac{T_{ON}}{T} \times 100\%$$

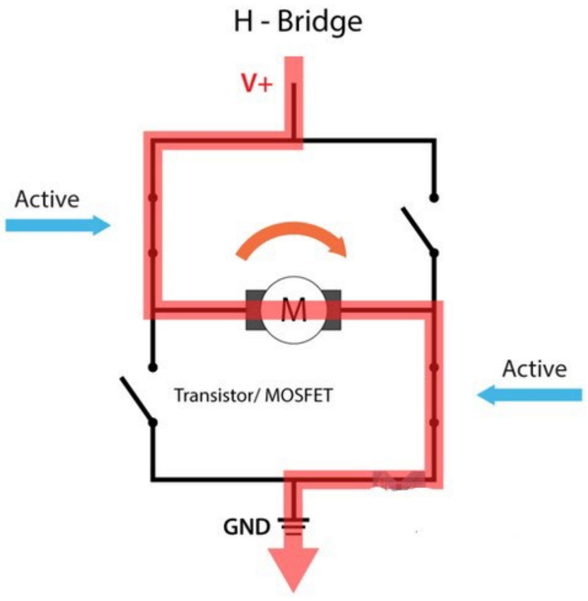
Now, the motor speed varies according to the duty cycle. Suppose the duty is zero, the motor does not run, and when the duty cycle is 100 % the motor runs at its maximum speed. **But this concept is not always right because the motor starts running after giving some fixed voltage which is called the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

Microcontroller and the Arduino can process signals and **consumes almost 20 to 40 mA 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.** The PWM signal is coming from Arduino and the transistor works as a switch that **short circuits the Emitter (E) and Collector (C) terminals 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. The circuit is given below:



H-Bridge DC Motor Control

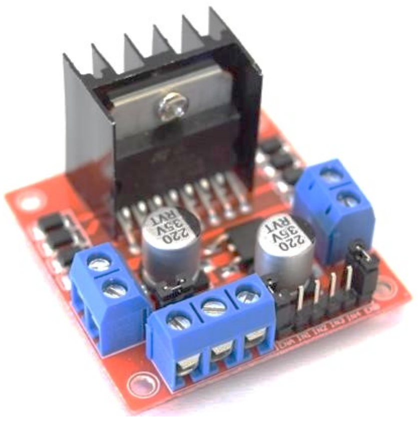
To control the direction of rotation, we just need to inverse the direction of the current flow through the motor, and the most common method of doing that is by using an H-Bridge. An H-Bridge circuit contains four switching elements, transistors, or MOSFETs, with the motor at the center forming an H-like configuration. **By activating two switches at the same time, we can change the direction of the current flow, thus changing the rotation direction of the motor.** The figure below is showing the current flow in one direction when one upper and one lower switches are turned on.



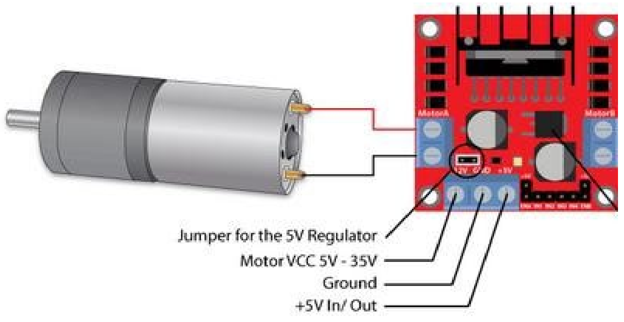
So, if we combine these two methods, the PWM and H-Bridge, we can have complete control over the DC motor. Many DC motor drivers have these features and the L298N is one of them.

L298N Driver

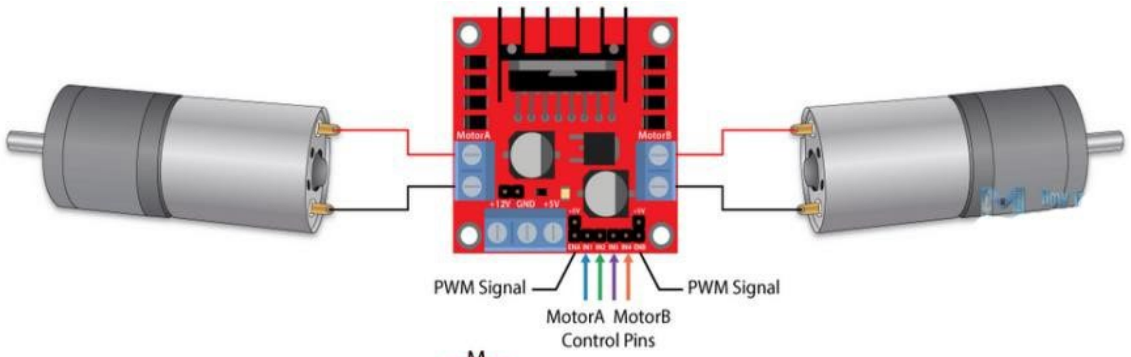
The L298N is a **dual H-Bridge motor driver which allows speed and direction control of two DC motors at** the same time as given below. The module can drive DC motors that have voltages between 5 and 35 V, with a peak current of up to 2 A.

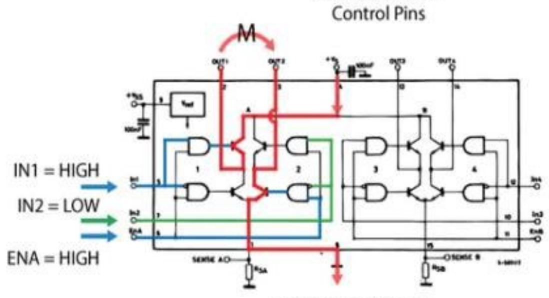


Let's take a closer look at the pinout of the L298N module and explain how it works. The module has two screw terminal blocks for motors A and B, another screw terminal block for the ground pin, the VCC for the motor, and a 5 V pin which can either be an input or output as shown below.



Next. are the logic control inputs. The Enable A and B pins are used to enable and control 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 the ground, the motor will be disabled.**



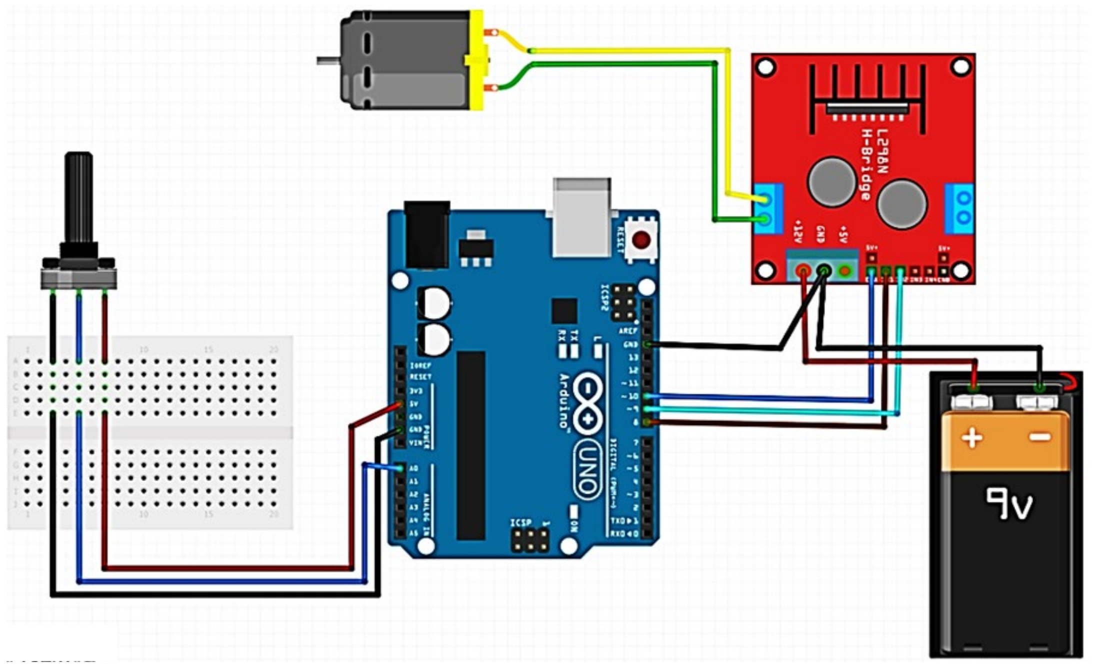


L298N Block Diagram

Next, the Input 1 and Input 2 pins are used for controlling the direction of rotation of motor A, and inputs 3 and 4 for motor B. Using these pins, we can 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. If input 1 is HIGH and input 2 is LOW, the motor will move backward. When 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 a circuit to **control the speed of the motor using a potentiometer** and **change the rotation direction using a push button**. Here's the circuit's schematic diagram. You need to connect a push button switch at pin 2 of the Arduino with this circuit.



Components List

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

Program to be written in ArduinoIDE:

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

void setup() {
Serial.begin(9600);
pinMode(2, INPUT);
pinMode(8, OUTPUT);
pinMode(9, OUTPUT);
pinMode(10, OUTPUT);
}
```

```
void TurnMotorA(){ //A function to control the direction and speed

if (digitalRead(2) == LOW){
digitalWrite(in1, LOW); //Switch between these HIGH and LOW states to change direction
digitalWrite(in2, HIGH);
float analogvalue = analogRead(A0); // declaring and reading an analog voltage value from the pin
int PWMvalue = map(analogvalue, 0, 1023, 0, 255); // mapping the analog readings to change
                        // range from 0-1023 to 0-255 to divide the value by 4 to get a PWM value
analogWrite(ConA, PWMvalue);// To activate the DC motor
}

else if (digitalRead(2) == HIGH){
digitalWrite(in1, HIGH); //Switch between these HIGH and LOW states to change direction
digitalWrite(in2, LOW);
float analogvalue = analogRead(A0); // declaring and reading an analog voltage value from the pin
int PWMvalue = map(analogvalue, 0, 1023, 0, 255); // mapping the analog readings to change
                        // range from 0-1023 to 0-255 to divide the value by 4 to get a PWM value
analogWrite(ConA, PWMvalue);// To activate the DC motor
}
}

void loop() {
float analogvalue = analogRead(A0); // declaring and reading an analog voltage value from the pin
int PWMvalue = map(analogvalue, 0, 1023, 0, 255); // mapping the analog readings to change
                        // range from 0-1023 to 0-255 to divide the value by 4 to get a PWM value

Serial.print("Digital Value = ");
Serial.print(PWMvalue);         //print digital value on serial monitor
                                //convert digital value to analog voltage
float analogVoltage = (PWMvalue *5.00)/255.00;
Serial.print("  Analog Voltage = ");
Serial.println(analogvalue);

TurnMotorA();           // function that keeps looping to run the motor continuously
                        //you can add another one with a different direction or stop
}
```

Alternative version:

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

void setup() {
Serial.begin(9600);
pinMode(2, INPUT);
pinMode(8, OUTPUT);
pinMode(9, OUTPUT);
pinMode(10, OUTPUT);
}

void TurnMotorA1(){ //A function to control the direction and speed in one direction

digitalWrite(in1, LOW); //Switch between these HIGH and LOW states to change direction
digitalWrite(in2, HIGH);
float analogvalue = analogRead(A0); // declaring and reading an analog voltage value from the pin
int PWMvalue = map(analogvalue, 0, 1023, 0, 255); // mapping the analog readings to change
                              // range from 0-1023 to 0-255 to divide the value by 4 to get a PWM value
analogWrite(ConA, PWMvalue);// To activate the DC motor
}

void TurnMotorA2(){ //A function to control the direction and speed in another direction

digitalWrite(in1, HIGH); //Switch between these HIGH and LOW states to change direction
digitalWrite(in2, LOW);
float analogvalue = analogRead(A0); // declaring and reading an analog voltage value from the pin
int PWMvalue = map(analogvalue, 0, 1023, 0, 255); // mapping the analog readings to change
                              // range from 0-1023 to 0-255 to divide the value by 4 to get a PWM value
analogWrite(ConA, PWMvalue);// To activate the DC motor
}


void loop() {
float analogvalue = analogRead(A0); // declaring and reading an analog voltage value from the pin
int PWMvalue = map(analogvalue, 0, 1023, 0, 255); // mapping the analog readings to change
                              // range from 0-1023 to 0-255 to divide the value by 4 to get a PWM value

Serial.print("Digital Value = ");
Serial.print(PWMvalue);       //print digital value on serial monitor
                              //convert digital value to analog voltage
float analogVoltage = (PWMvalue *5.00)/255.00;
Serial.print("Analog Voltage = ");
Serial.println(analogvalue);

if (digitalRead(2) == LOW){

TurnMotorA1();          // function that keeps looping to run the motor continuously
                        //you can add another one with a different direction or stop
}

else if (digitalRead(2) == HIGH){
TurnMotorA2();
}

}
```

Questions for report writing:

1. Include all codes and scripts in the lab report following the writing template.
2. Implement the system in the Proteus simulation tool by following the video link as follows: https://youtu.be/q_B3yAM4PH0