AMEIRCAN INTERNATIONAL UNIVERSITY-BANGLADESH (AIUB)

FACULTY OF ENGINEERING

MICROPROCESSOR AND EMBEDDED SYSTEMS [F]
Spring 2021-2022

Lab Report on

Experiment 06: Controlling a motor through the application of PWM.

Supervised by
DR. NADIA ANAM (ID: 1305-1442-2)
Assistant Professor, EEE

Group – 05

Submitted by:

	NAME	ID	Department
1.	AKTER, ALAM MORIUM	19-39957-1	CSE
2.	NABIL, MD. JOBAIR AHMED	18-38837-1	CSE
3.	BHUIYAN, FAHIM MAHMUD	20-42970-1	CSE
4.	MOBARAK, BHUIYAN SAAD BIN	19-41059-2	EEE
5.	SAKIB, A.B.M. NAZMUS	19-41582-3	EEE

Date of submission: MARCH 30, 2022

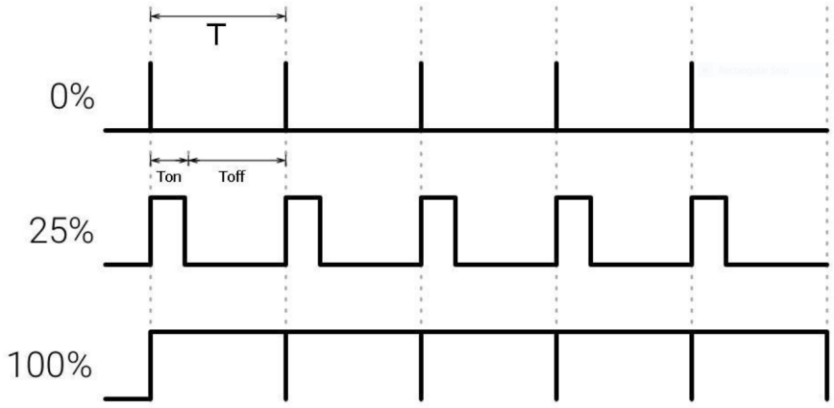
Title: Controlling a motor through the application of PWM.

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

Theory and Methodology:

Microcontroller and Arduino are digital devices and they are unable to provide analog output. The microcontroller outputs ZERO and ONE, where ZERO denotes logical LOW and ONE denotes logical HIGH. In our case, we are utilizing the Arduino 5-volt version. As a result, logical ZERO equals zero voltage and logical HIGH equals five voltage.

While digital output is ideal for digital equipment, analog output is sometimes required. PWM is extremely handy in this situation. As illustrated in the diagram below, the output signal flips between zero and one on a high and fixed frequency in the PWM.



Output Signal Of PWM

The ON time is Ton, and the OFF time is Toff, as indicated in the diagram above. The Time Period is defined as the sum of the Ton and Toff. T is not variable in the PWM concept, but Ton and Toff are. As a result, when Ton increases, Toff decreases, and Toff increases when Ton decreases correspondingly.

The duty cycle is a fraction of a period. The duty cycle is usually stated as a percentage or as a ratio. The time it takes for a signal to complete an on-and-off cycle is called a period. A duty cycle can be calculated using the following formula:

DUTY CYCLE = (Ton / T) x100 %

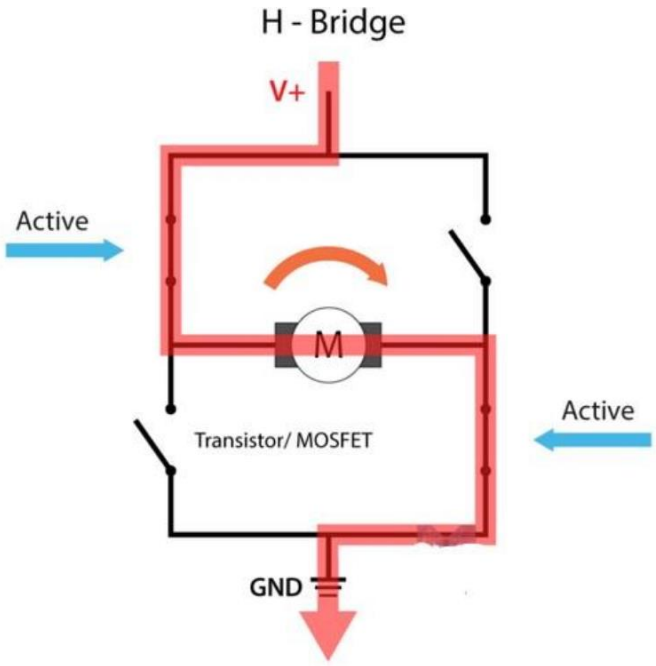
The motor speed now fluctuates depending on the duty cycle. If the duty is zero, the motor will not operate, and if the duty is 100 percent, the motor will run at maximum RPM. However, this concept is not necessarily correct because the motor only starts operating until a certain fixed voltage, known as the threshold voltage, is applied.

Microcontrollers and Arduinos can process signals and use about 20 to 40 milliamps of current, but motors require much more current and voltage, so we use a transistor to drive the motor. The transistor is in series with the motor, and the transistor's base is connected to the Arduino's PWM pin via a resistor. The transistor acts as a switch, short-circuiting the Emitter (E) and Collector (C) when the PWM signal is in the HIGH state and usually opening when the PWM signal is in the LOW state. This procedure continues indefinitely, with the motors running at the specified speed.

H-Bridge DC Motor Control

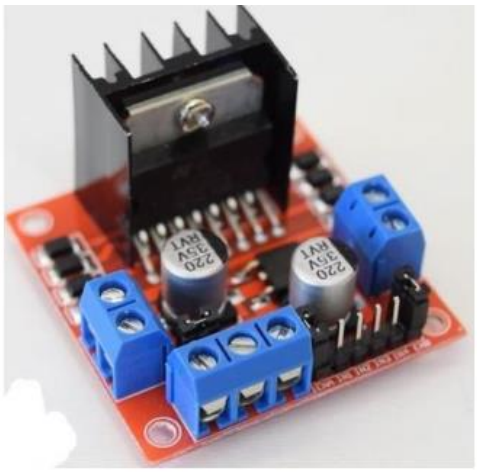
We only need to reverse the direction of the current flow through the motor to regulate the rotation direction, and the most popular method for doing so is to use an H-Bridge. An H-Bridge circuit is made up of four switching devices, such as transistors or MOSFETs, with the motor in the middle, producing a H shape. We can change the direction of the current flow and consequently the rotation direction of the motor by activating two specific switches at the same time.

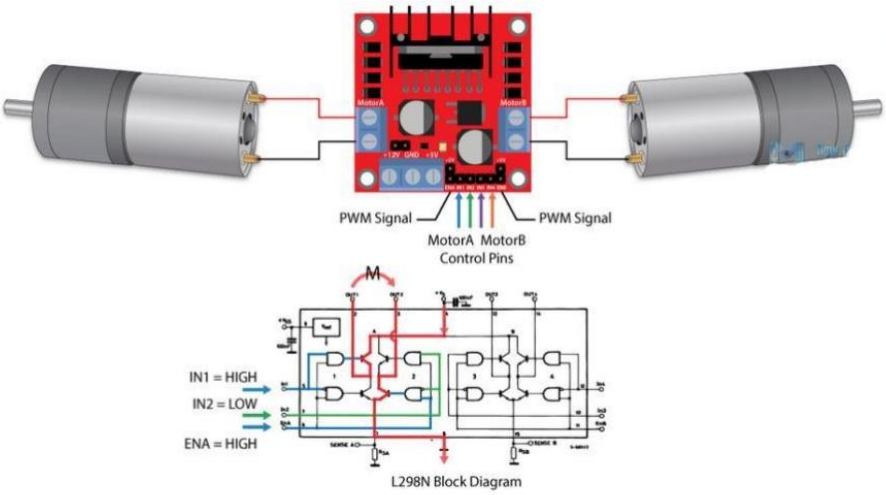
Combining these two methods, the PWM and the H-Bridge, we can have complete control over the DC motor. L298N DC motor has been used here.



L298N Driver:

The L298N is a dual H-Bridge motor driver that allows for simultaneous speed and direction control of two DC motors. The module can power DC motors with voltages ranging from 5 to 35V and peak currents of up to 2A. The module has two screw terminal blocks for the motor A and B, and another screw terminal block for the Ground pin, the VCC for motor and a 5V pin which can either be an input or output.





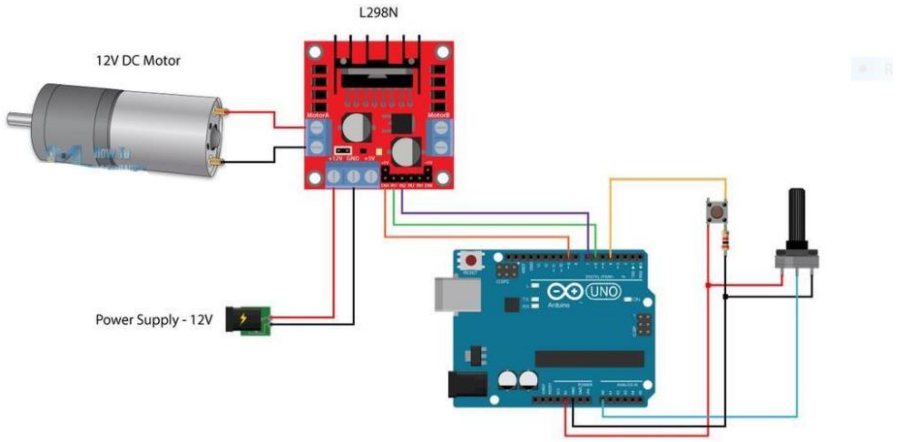
The logic control inputs come next. Enable A and Enable B pins are used to control and enable the motor's speed. If a jumper is present on this pin, the motor will be enabled and run at full speed; however, if the jumper is removed, we may attach a PWM input to this pin and control the motor's speed. The motor will be disabled if this pin is connected to Ground.

Following that, the Input 1 and Input 2 pins are used to control the rotation direction of the motor A, while the inputs 3 and 4 are used to control the rotation direction of the motor B. We control the H-Bridge switches inside the L298N IC using these pins. The motor will go forward if input 1 is LOW and input 2 is HIGH, and the motor will move backward if input 1 is HIGH and input 2 is LOW.

The motor will halt if both inputs are the same, either LOW or HIGH. The same is true for inputs 3 and 4 as well as motor B.

Arduino and L298N:

In the first example, we will use a potentiometer to control the motor's speed and a push button to adjust the rotation direction. So, we need an L298N driver, a DC motor, a potentiometer, a push button, and an Arduino board. The schematic diagram is shown below.



Apparatus List

- L298N Driver
- 12V High Torque DC Motor
- Arduino Board
- Breadboard and Jump Wires

Using Arduino IDE to write code:

```
int in1 = 6; //Declaring where our module is wired

int in2 = 7;

int ConA = 9;// Don't forget this is a PWM DI/DO

int speed1;

void setup() {

 Serial.begin(9600);

pinMode(6, OUTPUT);
```

```
pinMode(7, OUTPUT);
pinMode(9, 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 thea analog value from the potentiometer calibrate it
analogWrite(ConA,speed1); // Then inject it to our 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 is obtained by 255/1023
Serial.println(value);
TurnMotorA(); //one function that keeps looping you can add another one with
              different direction or stop
}
```

Discussions:

In this experiment, when the button is not pressed motor rotates in the backward direction, we control the speed of the motor using a potentiometer. If we keep the button pressed it will start rotating in the forward direction. Again, if we release the switch it will start to rotate to the backward direction. The same design was also

implemented by using a flowchart. It was quite easy because all the work was done step by step, which can be understood by anyone.

Reference(s):

1) https://www.arduino.cc/.
2) https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutoriall298npwm-h-bridge/