AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH (AIUB)
Faculty of Engineering
Department of EEE and CoE
Undergraduate Program

Course: Microprocessor and Embedded Systems **Summer 2021-22, Final**

Experiment 8: Implementation of a motor control system using Arduino: Digital input, outputs, and PWM

Submitted By:

Name	ID
MD SHAMIM SIDDIKY	20-42649-1
WASIUDDIN	15-30843-3
TANJIM HOSSAIN	18-36964-1
MD. SAKIB HOSSAIN RIJON	19-39460-1
SADIA AFRIN ETY	19-39659-1
FERDOUS SUNY	19-40485-1
DIPONKAR SUTRA DHAR	19-41004-2

Section: O
Group: 6

Submitted To

Md Ali Noor
Lecturer
Faculty of Computer Engineering
American International University-Bangladesh

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, it's logical ZERO is zero voltage, and logical HIGH is 5 voltages.

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.

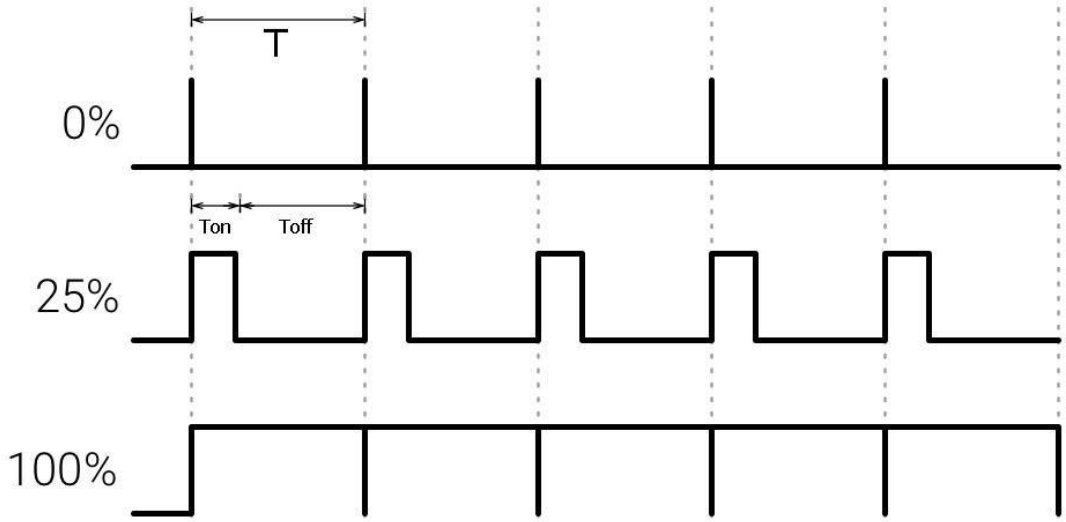

Fig 1: Output Signal of PWM

As shown in the above figure the ON time is Ton and the OFF time is Toff. T is the sum of the Ton and Toff, which is called the Period. In the concept of PWM, T is not varying and the Ton and the Toff can vary, in this way when Ton increase Toff will decrease, and Toff increase when Ton 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. 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.

Objectives:

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

Equipment List:

1) L298N Driver
2) 12V High Torque DC Motor
3) Arduino Board
4) Potentiometer
5) A power supply
6) Breadboard and Jump Wires

Circuit Diagram:

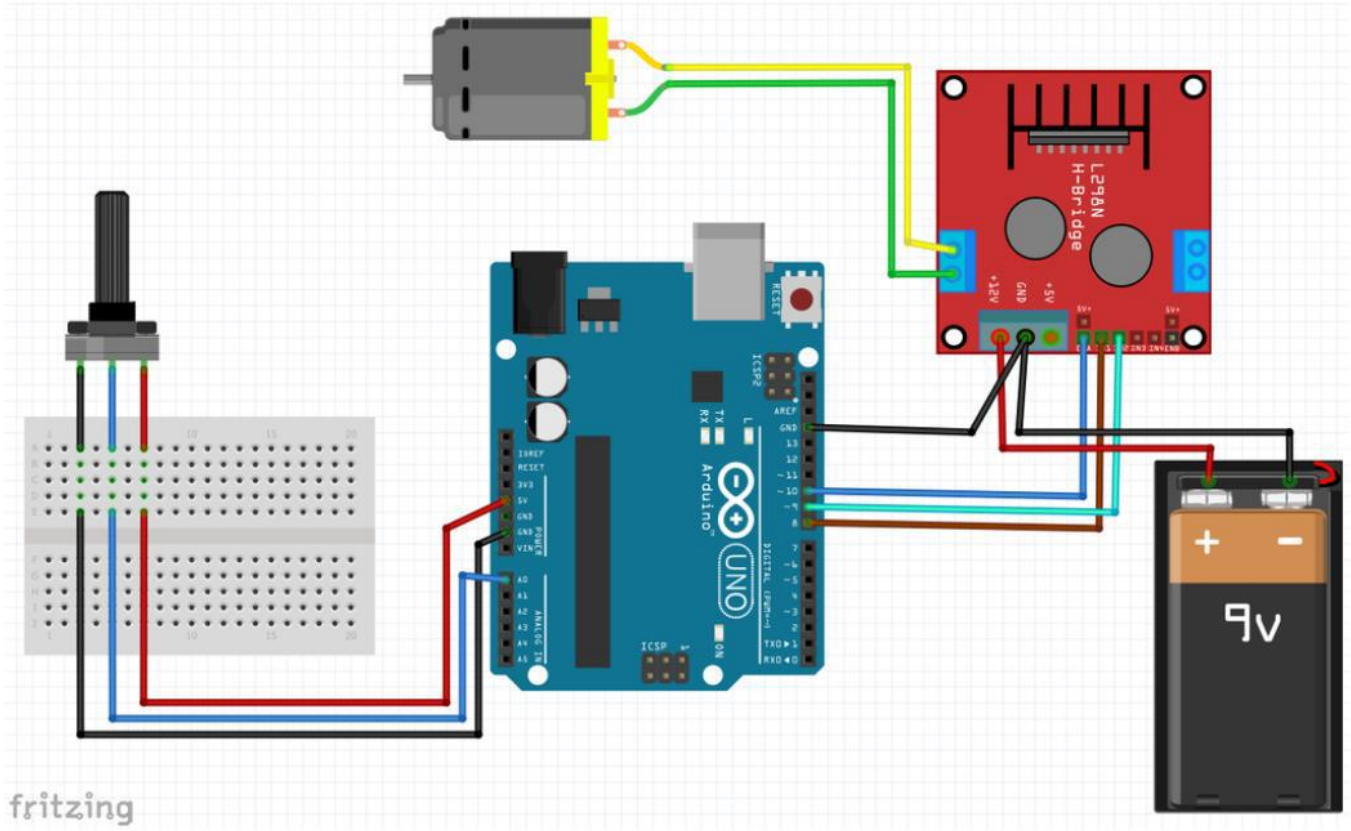

Fig 2: L298N driver, a DC motor, a potentiometer, and an Arduino board.

Hardware Setup:

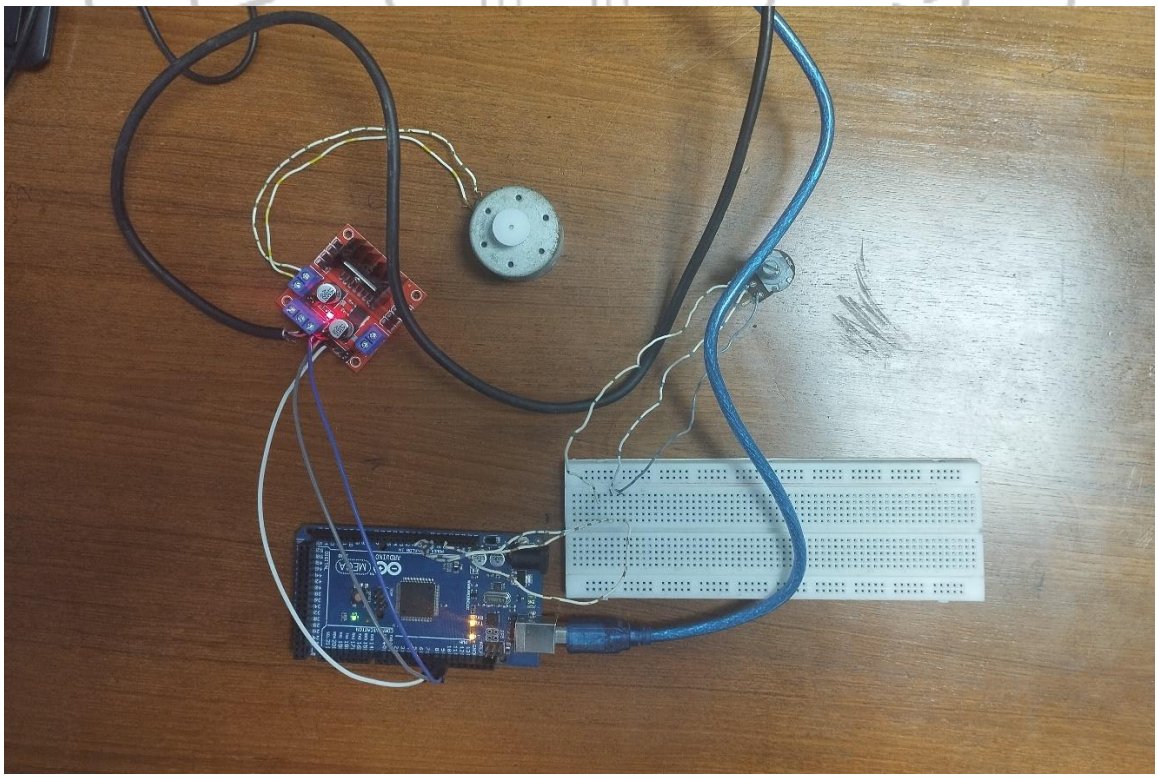

Figure 3: Hardware Set-up for motor control system

Experiment result:

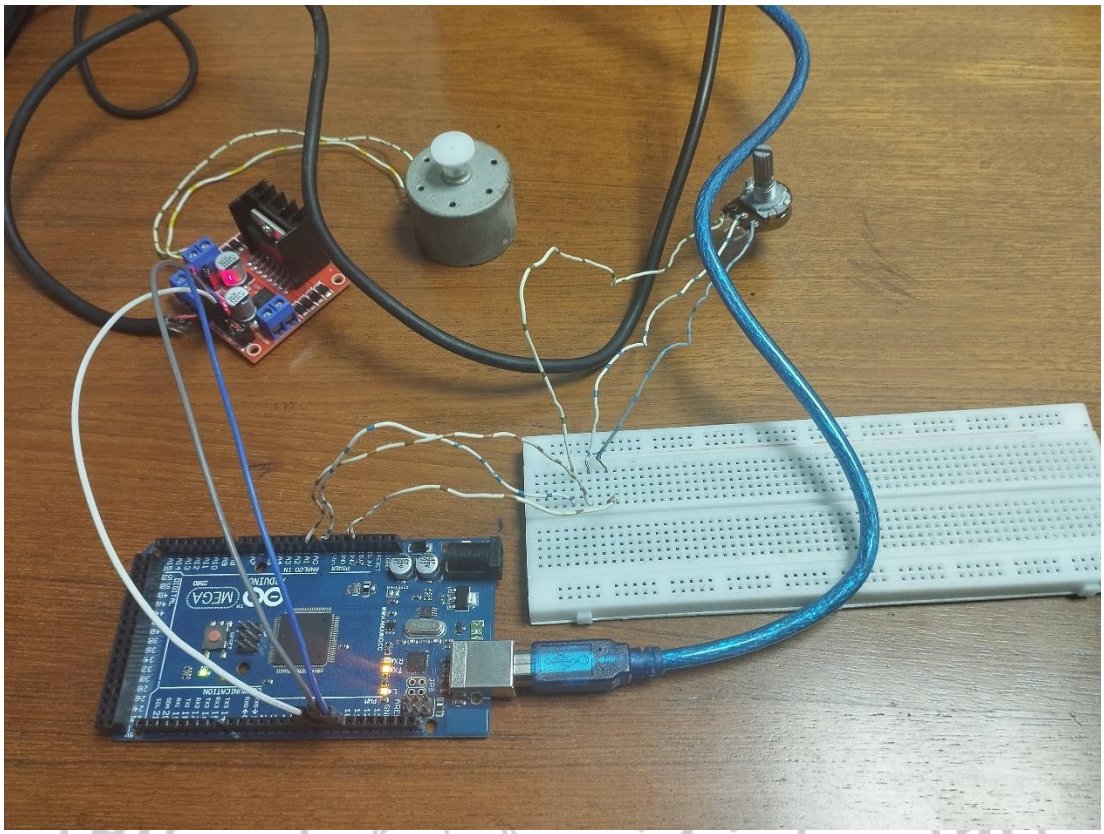

Fig 4: Motor On

Code Analysis:

First, we initialize int in1 in2 AND ConA.In the setup function, we set 5,6,7 as a OUTPUT pin. Then define a function TurnMotorA() to control the direction and speed. Then Switch between these HIGH and LOW states to change direction. After that, we calculate the speed Analog value is read from the potentiometer to calibrate it. Then declaring and reading a value from the pin and then calculating the value (value = value*0.2492668622) . Doing calibration to change range from 0-1023 to 0-255 the number and is obtained by 255/1023. And the end of the code we call the TurnMotorA(); for one function that keeps looping you can add another one with a different direction or stop.

```
int in1 = 5; //Declaring where our module is wired
int in2 = 6;
int ConA = 7;// Don't forget this is a PWM DI/DO
int speed1;
void setup() {
Serial.begin(9600);
pinMode(5, OUTPUT);
pinMode(6, OUTPUT);
```

```
pinMode(7, 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);
speed1 = analogRead(A0);
speed1 = speed1*0.2492668622; //Analog value is read from the potentiometer to calibrate it
analogWrite(ConA,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

}
```

Discussion:

In this experiment, we have learned about controlling the motor with a potentiometer. The speed of the motor is controlled using a potentiometer and changes the rotation direction using a push button. The analog value is taken from the potentiometer and the speed is assigned based on this. This concept helps in many complex engineering problems.