American International University- Bangladesh
Department of Electrical and Electronic Engineering
EEE 4103: Microprocessor and Embedded Systems Laboratory

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

Introduction:

The objectives of this experiment are to-

1. Familiarize the students with the PWM signals generated by the Arduino.
2. Control the speed of a DC motor using the PWM signals generated by the Arduino.
3. Change the direction of rotation of a DC motor using the input push switch.

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 in Fig. 1.

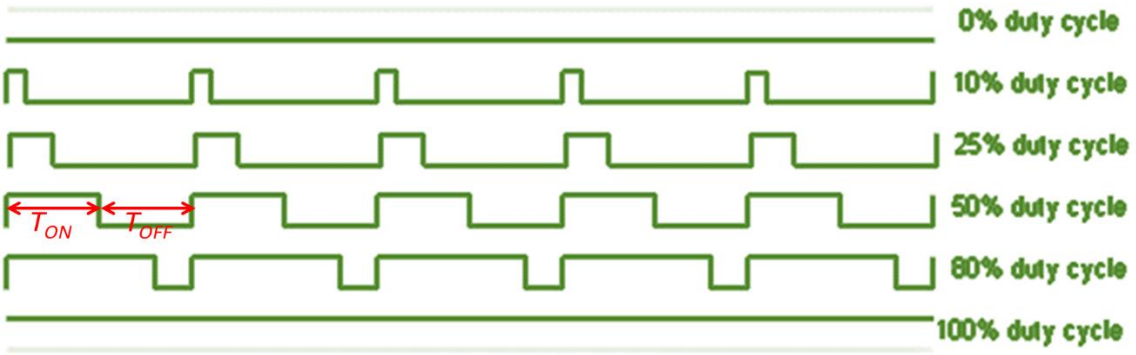

Figure 1: PWM signals at different duty cycles.

Output Signals of PWM

As shown in the above figure, the ON time is T_{ON} and the OFF time is T_{OFF}. Here, 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, D 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, D is commonly expressed as a percentage or a ratio. A time 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.

The Arduino microcontroller and 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. The 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 motor runs at the desired speed. The circuit is given in Fig. 2.

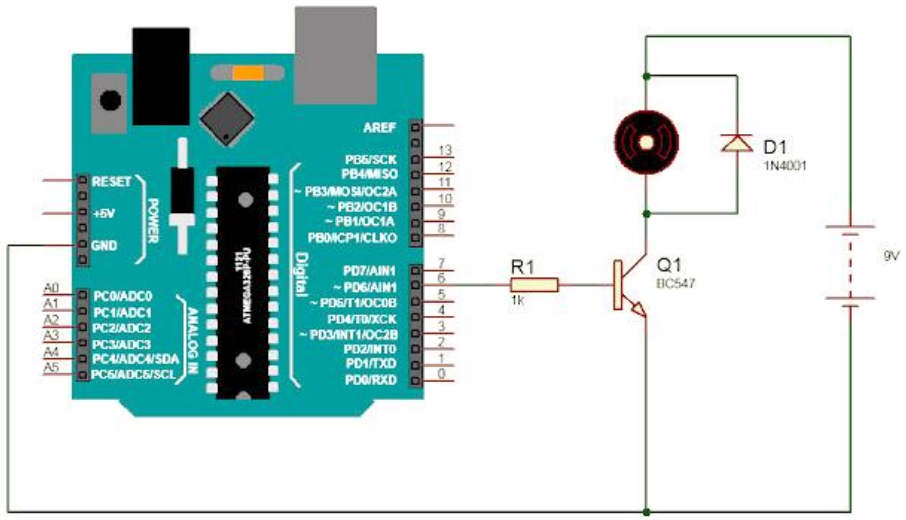

Figure 2: An Arduino microcontroller is controlling a DC motor's speed through an npn transistor.

H-Bridge DC Motor Control

To control the direction of rotation, we just need to reverse 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. Figure 3 shows the current flow in one direction when one upper and one lower switches are turned on.

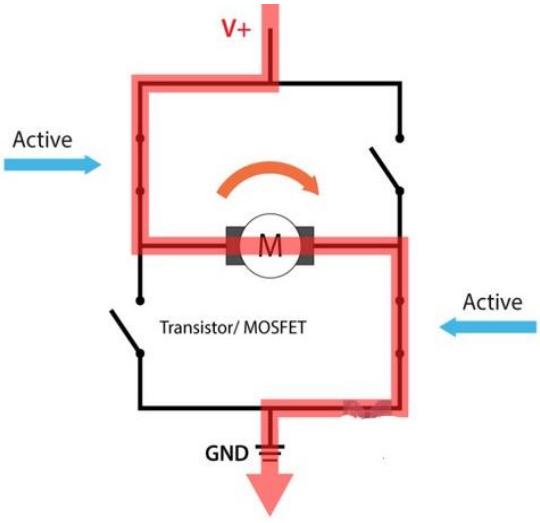

Figure 3: An H-bridge is controlling a DC motor's direction of rotation.

So, if we combine these two methods, the PWM and H-Bridge, we can have a 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 shown in Fig. 4. The module can drive DC motors that have voltages between 5 and 35 V, with a peak current of up to 2 A.

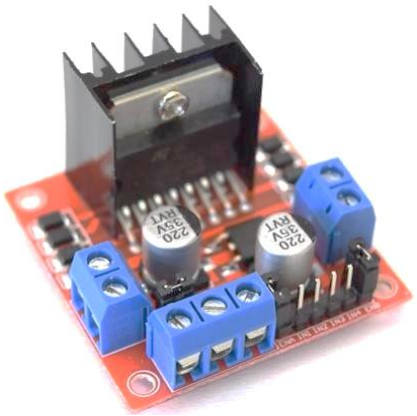

Figure 4: A dual H-bridge motor drive circuit, L298N

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 V_{CC} for the motor, and a 5 V pin which can either be an input or output as shown in Fig. 5.

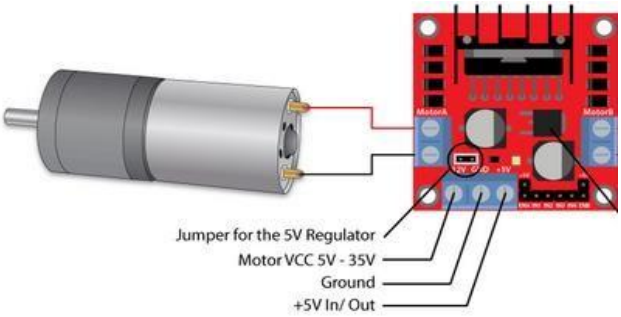

Figure 5: Pin configuration of a dual H-bridge motor drive circuit, L298N

Next. are the logic control inputs. The Enable A and B pins are used to enable and control the speed of the motor as shown in Fig. 6. 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.

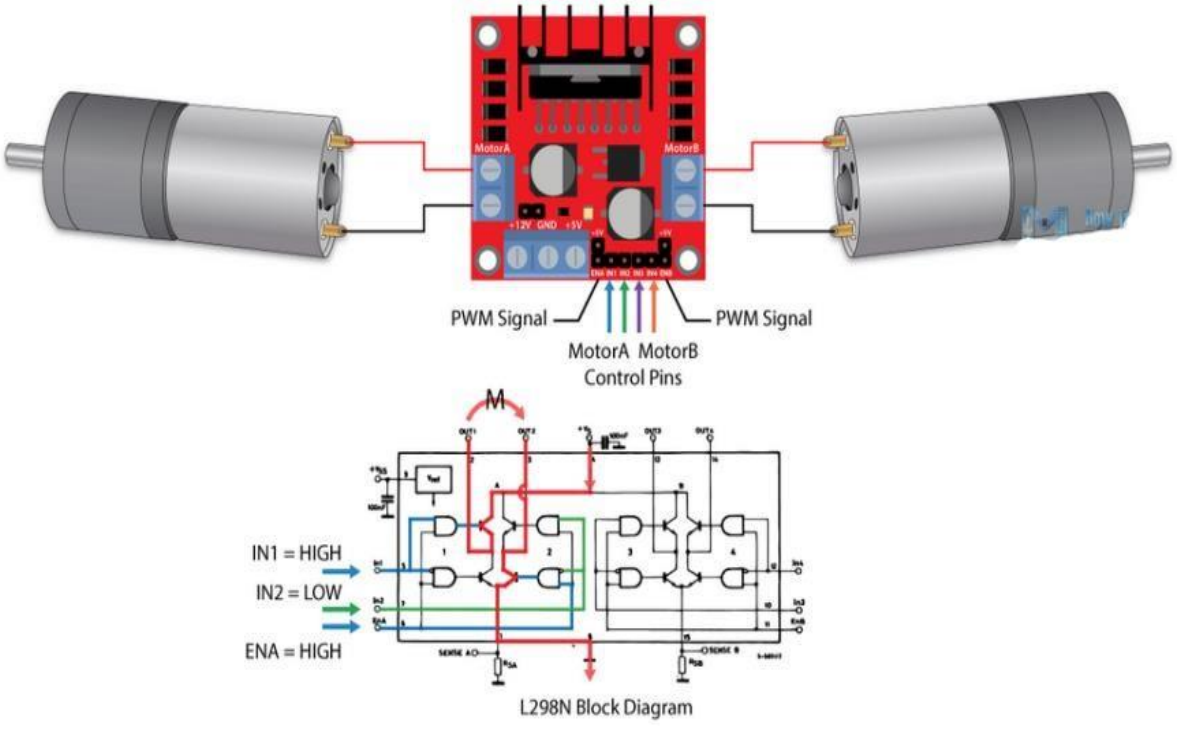

Figure 6: Pin configuration and internal circuitry of a dual H-bridge motor drive circuit, L298N

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 as shown in Fig. 6. 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 Uno and L298N Driver

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.

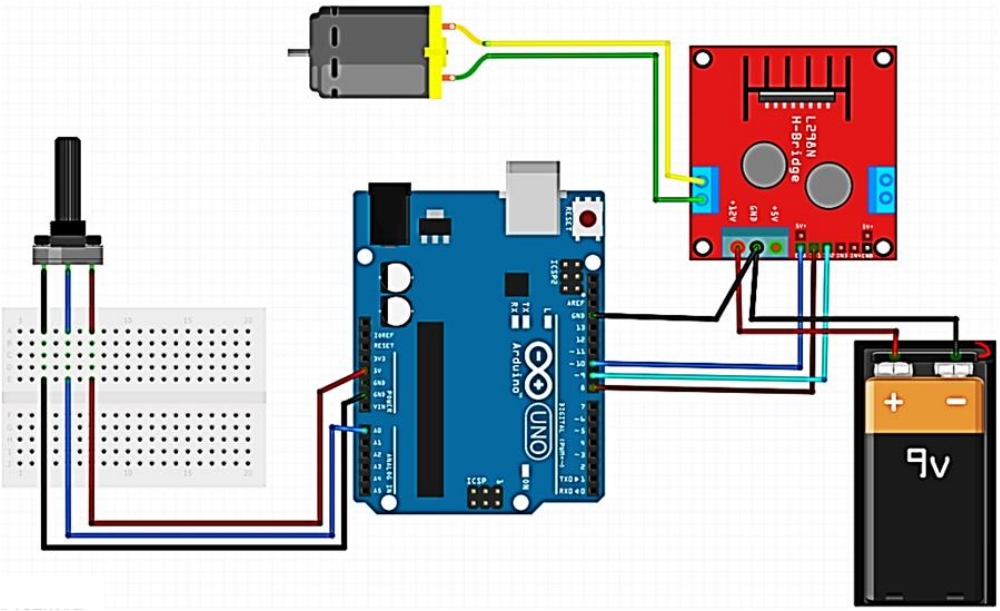

Figure 7: An Arduino microcontroller is controlling a DC motor's speed through a dual H-bridge.

Apparatus:

1) Arduino Uno Board
2) L298N Driver
3) 12 V High Torque DC Motor with Fan Blades connected to it.
4) Potentiometer, push switch, and a resistor of 10 kΩ
5) A DC Power Supply
6) Breadboard
7) Jumper Wires

Experimental Circuit Diagram:

The circuit diagrams of Figs. 7-8 show how to connect a motor control system using a DC motor, push switch, DC power supply, potentiometer, and DC motor driver circuit to the Arduino UNO board, but you can follow the same procedure for the Arduino Mega or Arduino Nano. Almost everything will remain the same except for the pin numbers. You must check the pinouts of Arduino Mega or Nano.

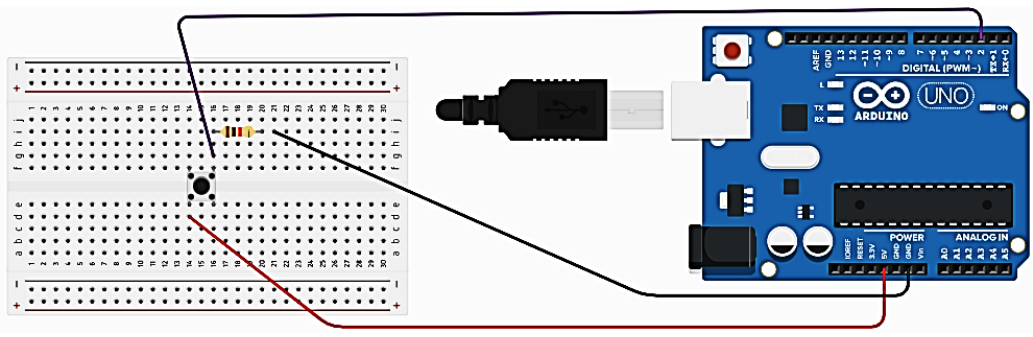

Figure 8: Arduino board's pin connections with a push button switch.

Arduino Program:

Please set up the Tools with the appropriate COM port before uploading the code.

```
int switchrotation = 2; // input pin to switch the direction of rotation
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(switchrotation, INPUT);
pinMode(in1, OUTPUT);
pinMode(in2, OUTPUT);
pinMode(ConA, 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

Serial.println("The motor is running in the clockwise direction."); // May need to change
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(analogVoltage);
}

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

Serial.println("The motor is running in the anticlockwise direction."); // May need to change
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(analogVoltage);
}


void loop() {
    if (digitalRead(switchrotation) == LOW) {

    TurnMotorA1();        // function that keeps looping to run the motor continuously.
    // you can add another one to stop through the delay() function to run for a certain duration.
    }
```

```
    else if (digitalRead(switchrotation) == HIGH) {
    TurnMotorA2();
    }

}
```

Experimental Procedure:

The main task of this experiment is to implement a motor control system using a DC motor, push switch, potentiometer, DC motor driver circuit. The implementation procedures are as follows:

1. Connect the circuit diagram as per Fig. 7. Connect the motor, driver, potentiometer, DC power supply, and push switch.
2. Write or copy the code into the IDE and then Save the file.
3. Compile the program. Then plug the Arduino microcontroller board into the PC.
4. Upload the program onto the Arduino board and run the program.
5. Observe the motor speed and direction of rotation at different conditions by varying the knob of the potentiometer on the serial communication screen.

Questions for Report Writing:

1) Include all codes and scripts in the lab report following the lab report writing template.
2) Show the output/results in the form of images. Give their captions and descriptions.
3) Configure the port numbers for outputs and inputs according to your ID. Consider four digits from your ID (if your ID is XY-PQABC-Z then consider In1, In2, switchrotation, and ConA from your ID's PQAB positions, if any digit is zero then use C digit. Include all the programs and results in your lab report.
4) Include the **Proteus and TinkerCad simulation** of the same program of this manual. Explain the simulation methodologies of both simulators.

Reference(s):

[1] Arduino IDE, https://www.arduino.cc/en/Main/Software accessed on 2nd July 2023.
[2] https://www.tinkercad.com/things/b6oU31mFyQa-brilliant-snaget/editel?tenant=circuits, accessed on 2nd July 2023.