American International University- Bangladesh (AIUB)
Faculty of Engineering (EEE)

Course Name:	MICROPROCESSOR AND EMBEDDED SYSTEMS	**Course Code:**	EEE 4103
Semester:	2023-2024, Spring	**Sec:**	F
Lab Instructor:	Md Sajid Hossain	**Group:**	06

Experiment No:	08
Experiment Name:	Implementation of a motor control system using Arduino: Digital input, outputs, and PWM.

Submitted by (NAME):	**TRIDIB SARKAR**	**Student ID:**	**22-46444-1**

Group Members	**ID**	**Name**
1.	22-46588-1	AZMINUR RAHMAN
2.	22-46444-1	TRIDIB SARKAR
3.	19-41203-2	MD. IMTIAZ HOSSAIN
4.	19-40315-1	S.M. FAYSAL MAHMUD
5.	21-45397-3	SOWRABH CHANDRA DAS
6.	21-45489-3	AVISHEK CHANDA PRATYAY
7.	21-45408-3	KOWSHIK HALDER

Performance Date:	**April 03, 2024**	**Due Date:**	

Marking Rubrics (to be filled by Lab Instructor)

Category	Proficient [6]	Good [4]	Acceptable [2]	Unacceptable [1]	Secured Marks
Theoretical Background, Methods & procedures sections	All information, measures and variables are provided and explained.	All Information provided that is sufficient, but more explanation is needed.	Most information correct, but some information may be missing or inaccurate.	Much information missing and/or inaccurate.	
Results	All of the criteria are met; results are described clearly and accurately;	Most criteria are met, but there may be some lack of clarity and/or incorrect information.	Experimental results don't match exactly with the theoretical values and/or analysis is unclear.	Experimental results are missing or incorrect;	
Discussion	Demonstrates thorough and sophisticated understanding. Conclusions drawn are appropriate for analyses;	Hypotheses are clearly stated, but some concluding statements not supported by data or data not well integrated.	Some hypotheses missing or misstated; conclusions not supported by data.	Conclusions don't match hypotheses, not supported by data; no integration of data from different sources.	
General formatting	Title page, placement of figures and figure captions, and other formatting issues all correct.	Minor errors in formatting.	Major errors and/or missing information.	Not proper style in text.	
Writing & organization	Writing is strong and easy to understand; ideas are fully elaborated and connected; effective transitions between sentences; no typographic, spelling, or grammatical errors.	Writing is clear and easy to understand; ideas are connected; effective transitions between sentences; minor typographic, spelling, or grammatical errors.	Most of the required criteria are met, but some lack of clarity, typographic, spelling, or grammatical errors are present.	Very unclear, many errors.	
Comments:				Total Marks (Out of):	

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

Abstract:

This study focuses on the familiarization with the Microcontroller based motor speed control and the implementation of motor control system using Arduino: Digital input, outputs, and PWM. In this experiment we were able to know L298N Dual H-bridge, DC motor, Potentiometer, and Arduino. The L298N Dual H bridge used with Arduino, it can control 2 DC motors at the same time, also control the direction and the speed as well. 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 results of the simulation and measurements showed that motor speed control working properly. The Tinker cad software was used for the simulation to build the motor control system.

Introduction:

The objective of this experiment is:

1. To get familiarized with Microcontroller based motor speed control.
2. To Implementation of motor control system using Arduino: Digital input, outputs, and PWM.

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 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 **[2].**

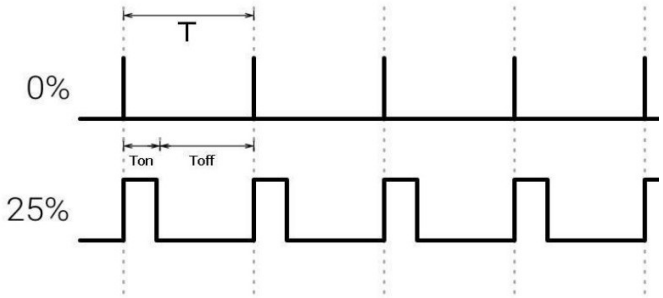

Figure 1: Output Signal of PWM **[3].**

H-Bridge DC Motor Control:

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 switches at the same time, we can change the direction of the current flow. thus, changing the rotation direction of the motor **[4].**

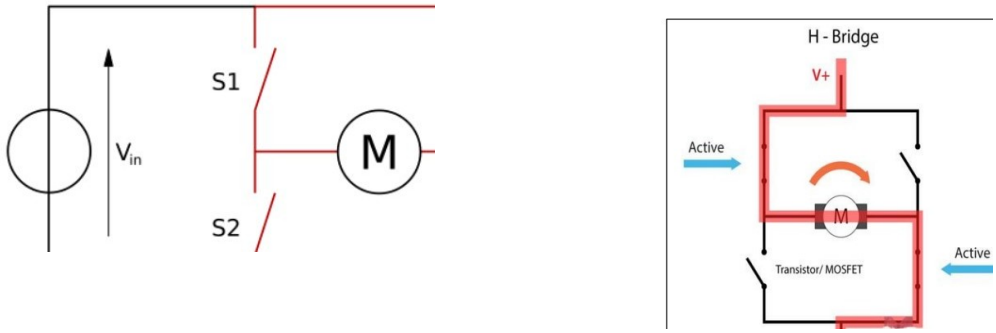

Figure 2: H-Bridge DC Motor Control **[3].**

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. The Enable A and Enable B pins are used for enabling and controlling the speed of the motor and 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. 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 **[4].**

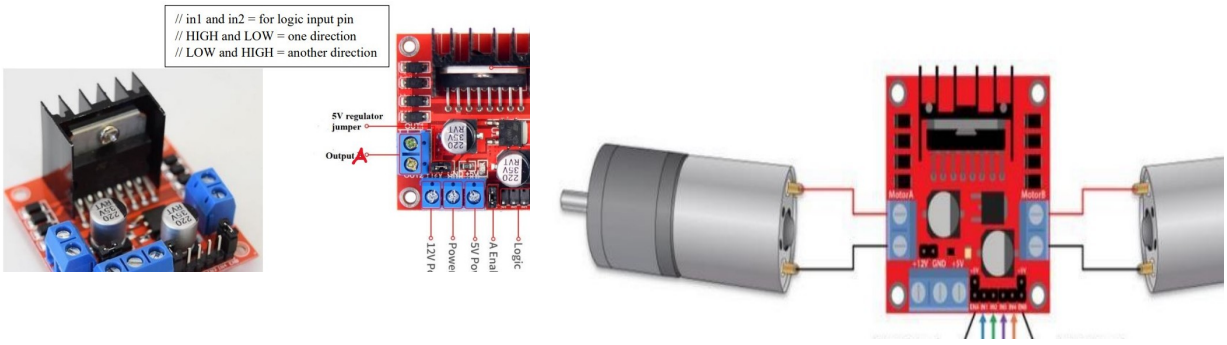

Figure 3: L298N Dual H-bridge **[3].**

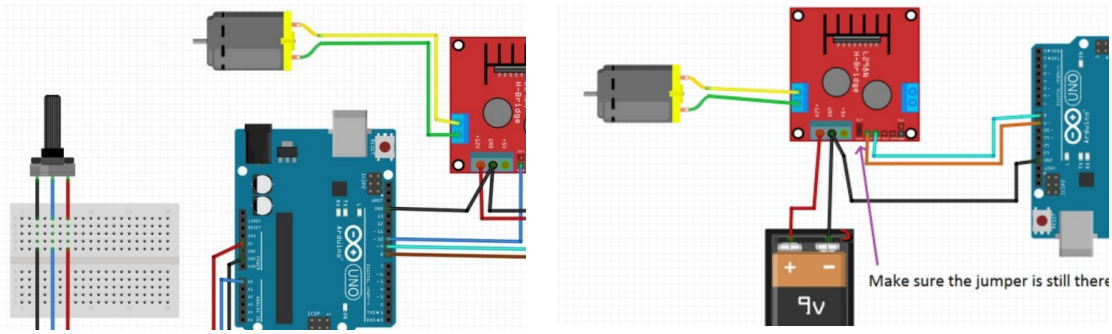

Figure 4: Experimental setup of a motor control system using Arduino **[3].**

Apparatus:

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

Precautions:

1. Safety precautions were observed during the experiment, including the proper handling of electroniccomponents and materials to prevent accidents or damage.
2. Wiring and connections were carefully attended to prevent short circuits or incorrect wiring.
3. Keep an eye out for the make sure the power supply connected to the L298N module and bothmotors are connected to the appropriate output terminals.
4. Simulations were run in Proteus software to validate the circuits and code before physical implementation to reduce the risk of errors.
5. Power sources were disconnected when changes were made to the circuit.

Experimental Procedure:

1. The Arduino board is connected to the computer using a USB cable, and the Arduino IDE is opened.
2. The breadboard is prepared, and the motor driver and Arduino are placed on it.
3. An Arduino sketch is written to control the direction of the DC motor using digital output pins and the sketch is uploaded to the Arduino, and the motor's movement is observed as the logic on theoutput pins is changed.
4. PWM control is implemented to vary the motor's speed using a potentiometer.
5. The sketch is uploaded to the Arduino, and the change in motor speed is observed as the potentiometer is adjusted.

Simulation and Measurement:

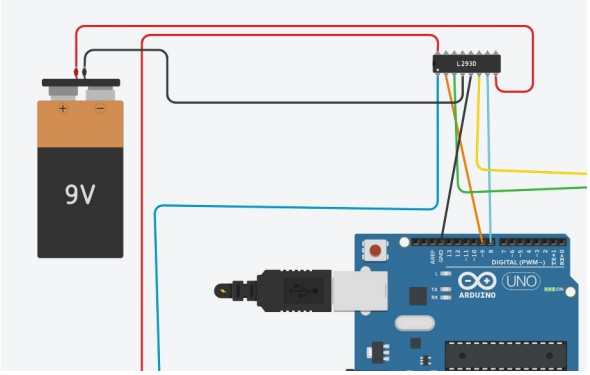

Figure 5: Simulation of No speed control.

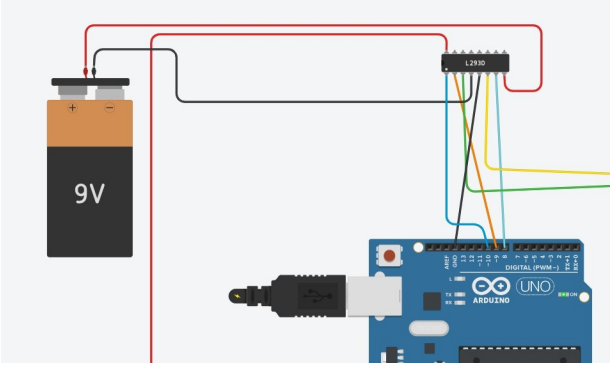

Figure 6: Simulation of turn on and change the direction.

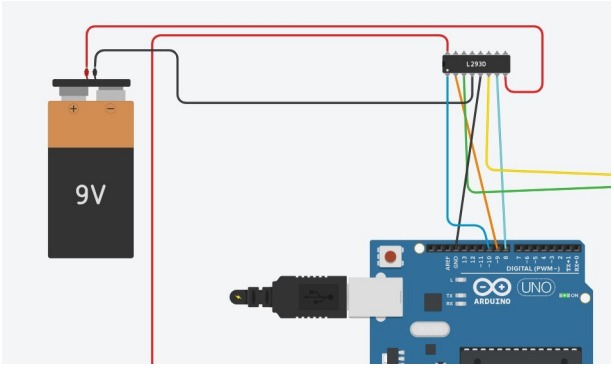

Figure 7: Simulation of Motor rotates forward.

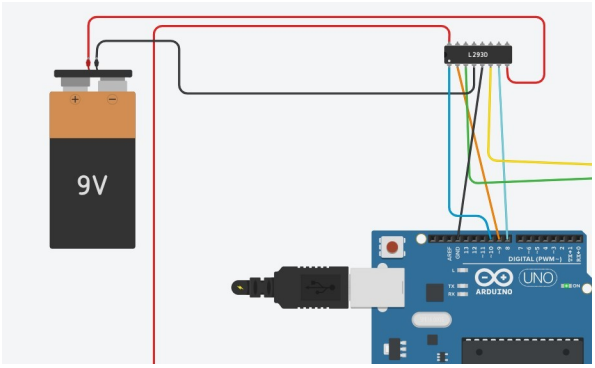

Figure 8: Simulation of Motor rotates backward.

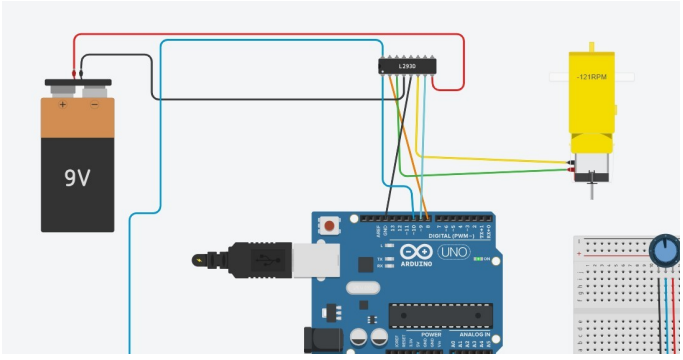

Figure 9: Simulation of Speed control using potentiometer.

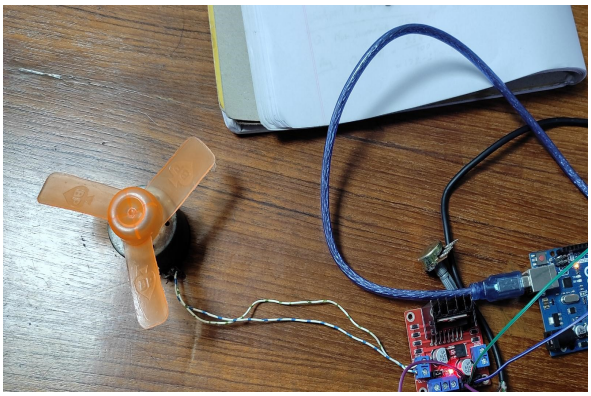

Figure 10: Image of No speed control.

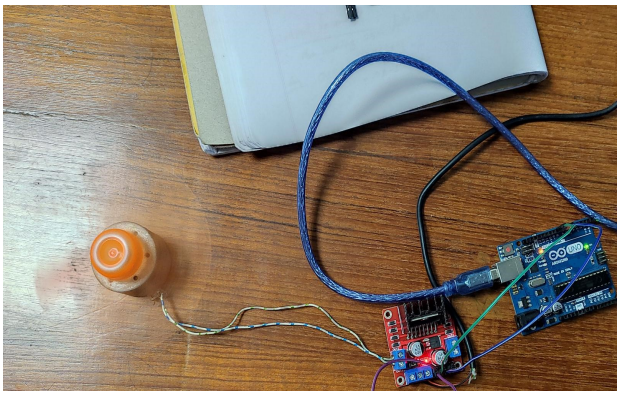

Figure 11: Image of Motor rotates forward.

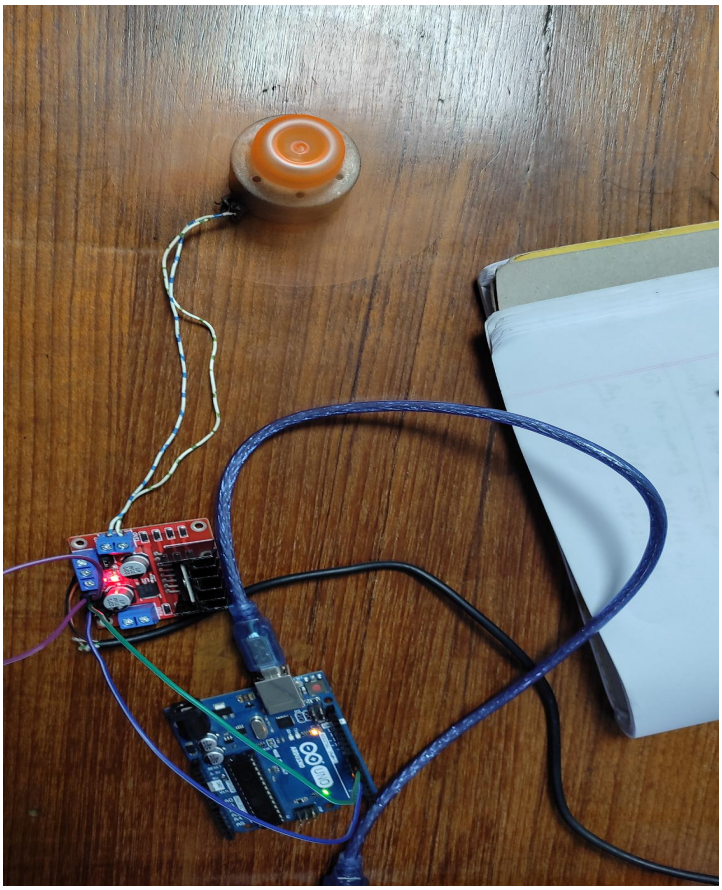

Figure 12: Image of Motor rotates backward.

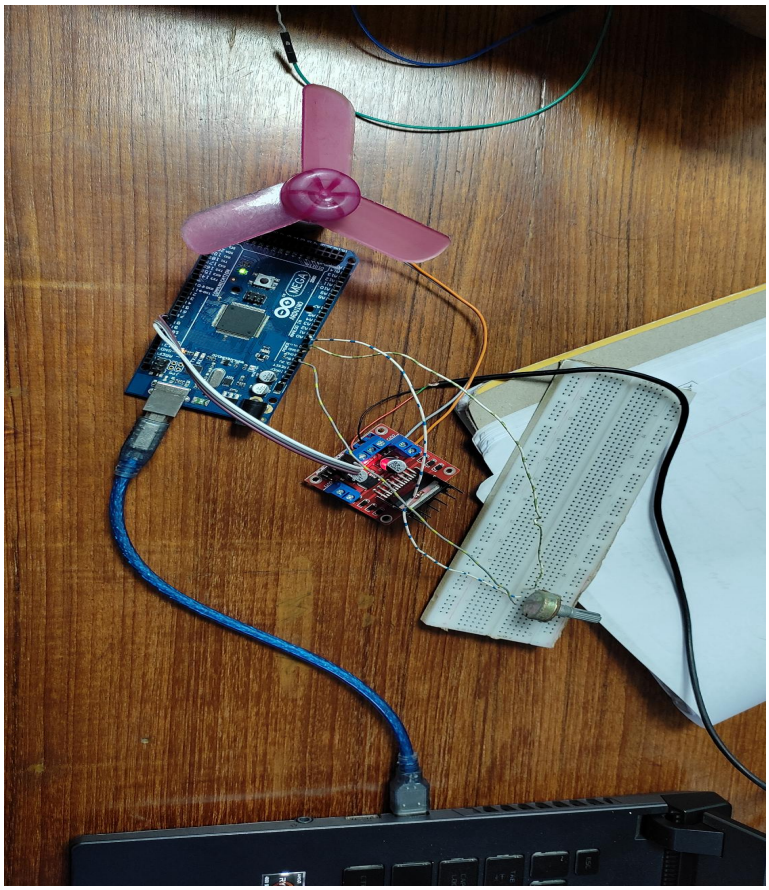

Figure 13: Image of speed control.

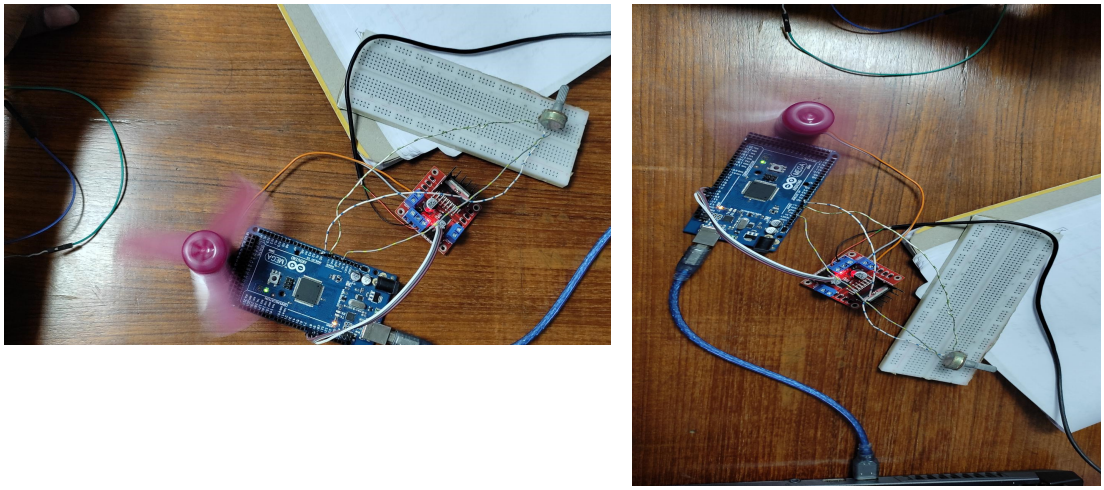

Figure 14: Images of Motor rotates forward backward and backward.

Code snippet for the System:

```
//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);
  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);
}
```

Figure 15: Code for No speed control Circuit.

```
//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
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);
}
```

Figure 16: Code for turn on and change direction Circuit.

```
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 ConA = 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(ConA, OUTPUT);
}
void TurnMotorA() {
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  analogWrite(ConA, 100);
}
void TurnOFFA() {
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  analogWrite(ConA, 0);
}
}
void TurnMotorA2() {
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  analogWrite(ConA, 250);
}
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);
}
```

Figure 17: Code for speed control Circuit.

```
//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 ConA = 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(ConA, 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
}
```

Figure 18: Code for control DC motor speed using potentiometer.

Discussion:

In this microprocessor lab experiment, an Arduino board was utilized to implement a motor control system, emphasizing digital inputs, outputs, and Pulse Width Modulation (PWM) techniques. The Arduino platform, known for its user-friendly interface and community support, served as a versatile tool for control systems. The experiment successfully created a functional motor control system, integrating the Arduino with a DC motor for dynamic speed and direction adjustments using digital inputs. The Arduino's digital output pins played a crucial role in governing the DC motor's direction through an H-bridge motor driver, enabling alteration of the voltage polarity, and determining rotational direction. The experiment introduced PWM to regulate the motor speed by adjusting the duty cycle of digital pulses, emulating analog voltage. This technique offers smooth speed control while minimizing power consumption in motor systems. Encountered challenge involved managing PWM frequency and duty cycle for efficient motor control, requiring careful consideration and testing to balance performance and noise generation during operation.

Conclusion:

The lab effectively showcased the practical utilization of Arduino-based motor control systems, focusing on digital input, outputs, and PWM. By delving into the intricate interplay between these components, participants gained a profound understanding of how to engineer efficient motor control mechanisms. The successful execution of experiments and projects underscored the importance of robust digital input mechanisms for sensory interactions, the significance of binary digital outputs in precise motor direction control, and the pivotal role of PWM in regulating speed with finesse.

Reference:

1. Arduino, "Arduino - Home," Available: https://www.arduino.cc , Accessed On: 18-April-2024. [Online].
2. "Basics of PWM(Pulse Width Modulation) | Arduino Documentation." https://docs.arduino.cc/learn/microcontrollers/analog-output , Accessed On: 18-April-2024. [Online].
3. AIUB Lab Manual.
4. Dejan, "L298N Motor Driver – Arduino Interface, how it works, codes, schematics." https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control tutorial-l298n-pwm-h-bridge/, Accessed On: 18- April-2024. [Online].