



AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

Faculty of Engineering

Lab Report

Experiment # 09

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

Date of Perform:	14 th May 2025	Date of Submission:	21 st May 2025
Course Title:	Microprocessor and Embedded Systems Lab		
Course Code:	EEE4103	Section:	R
Semester:	Spring 2024-25	Degree Program:	BSc in CSE
Course Teacher:	Prof. Dr. Engr. Muhibul Haque Bhuyan		

Declaration and Statement of Authorship:

1. I/we hold a copy of this Assignment/Case Study, which can be produced if the original is lost/damaged.
2. This Assignment/Case Study is my/our original work; no part has been copied from any other student's work or any other source except where due acknowledgment is made.
3. No part of this Assignment/Case Study has been written for me/us by any other person except where such collaboration has been authorized.
by the concerned teacher and is acknowledged in the assignment.
4. I/we have not previously submitted or am submitting this work for any other course/unit.
5. This work may be reproduced, communicated, compared, and archived to detect plagiarism.
6. I/we permit a copy of my/our marked work to be retained by the Faculty Member for review by any internal/external examiners.
7. I/we understand that Plagiarism is the presentation of the work, idea, or creation of another person as though it is your own. It is a form of cheating and is a very serious academic offense that may lead to expulsion from the University. Plagiarized material can be drawn from, and presented in, written, graphic, and visual forms, including electronic data, and oral presentations. Plagiarism occurs when the origin of the source is not appropriately cited.
8. I/we also understand that enabling plagiarism is the act of assisting or allowing another person to plagiarize or copy my/our work.

* Student(s) must complete all details except the faculty use part.

** Please submit all assignments to your course teacher or the office of the concerned teacher.

Group # 05

Sl No	Name	ID	PROGRAM	SIGNATURE
1	Shahriar Hossain	22-48990-3	BSc in CSE	
2	Al Muhtasim	22-49002-3	BSc in CSE	
3	Adiba Tanzila	22-49012-3	BSc in CSE	
4	MD.Rakib Hasan	22-49029-3	BSc in CSE	
5	Md.Imdadul Hasan(Ayon)	22-49959-3	BSc in CSE	

Faculty use only

FACULTY COMMENTS	Marks Obtained	
	Total Marks	

Table of Contents

Objectives	3
Equipment List	3
Circuit Diagram	3-4
Experimental Output Results (Color Photographs)	4-9
Simulation Output Results (Color Photographs)	9-12
Answers to the Questions in the Lab Manual	12-14
Discussion	14
References	14

Marking Rubrics (to be filled by Faculty):

Level Category	Excellent [5]	Proficient [4]	Good [3]	Acceptable [2]	Unacceptable [1]	No Response [0]
Title and Objectives	Able to clarify the understanding of the lab, no issues are missing and formatting is good.	Able to clarify the understanding of the lab experiment, no issues are missing but its formatting is not good.	Able to clarify the understanding of the lab experiment, but a few issues are wrong, and its formatting is bad.	Able to clarify the understanding of the lab experiment, but it lacks a few important issues of the experiment without maintaining the format.	Unable to clarify the understanding of the lab experiment.	No Response/ copied from others/ identical submissions with gross errors/image file printed
Codes and Methods	Able to explain the experimental codes and simulation methods using Proteus very well.	Able to explain the experimental codes and simulation methods using Proteus but is not formatted well.	Able to explain the experimental codes but simulation method using Proteus is not explained well.	Presents the experimental codes but didn't explain simulation methods using Proteus clearly.	Presents the experimental codes but didn't explain simulation methods using Proteus.	
Results	Key results and images are there. Figures/Tables have all identifications and refer to them properly in the texts.	Key results and images are there. Figures/Tables have all identifications, such as the axis labels, numbers, and captions with a few minor errors; the texts refer them.	Key results and images are there. Figures/Tables lack a few identifications, such as the axis labels, numbers, and captions; the texts refer them.	Misses several key results and images. Figures/Tables lack identification, such as the axis labels, numbers, and captions; the texts don't refer them.	Major results, such as experimental and simulation results' images are not included. Figures and tables are poorly constructed or not presented.	
Discussion and Conclusion	Proper interpretation of results and summarizes the results to draw a conclusion, discusses its applications in real-life situations to connect with the report's conclusion.	Proper interpretation of results and summarizes the results to draw a conclusion but didn't discuss its applications in real-life situations to connect with the conclusion of the report.	Interpretation of results is presented. However, there is a disconnect between the results and discussion.	Misses the interpretation of key results. There is little connection between the results and discussion.	Very poor interpretation of the results. No connection between results and discussions.	
Question and Answer	Able to produce all questions' answers correctly maintaining the lab report format.	Able to produce all questions' answers but didn't maintain the lab report format.	Able to produce all questions' answers but wrong answers to a few questions.	Able to produce all questions' answers but wrong/missing answers to multiple questions.	Unable to produce all questions' answers and completely wrong answers.	
Comments						Total Marks (25)

Objectives:

The objectives of this experiment are to-

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

Equipment List:

- 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 k Ω
- 5) A DC Power Supply
- 6) Breadboard
- 7) Jumper Wires

Circuit Diagram:

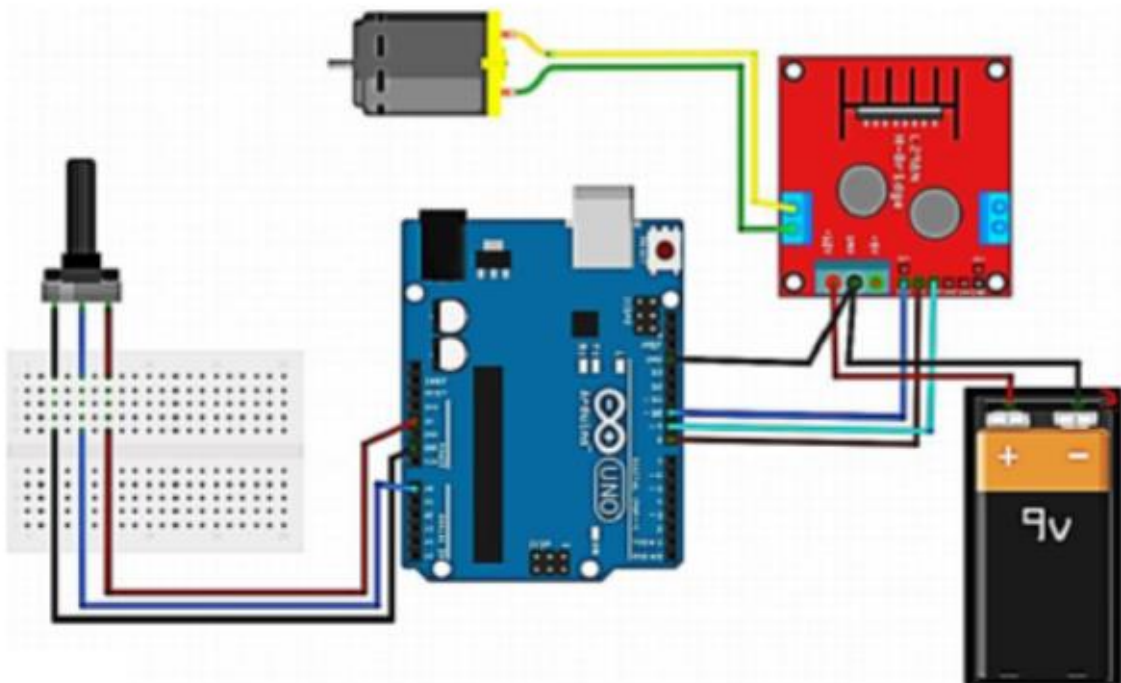


Fig : An Arduino microcontroller is controlling a DC motor's speed through a dual H-bridge.

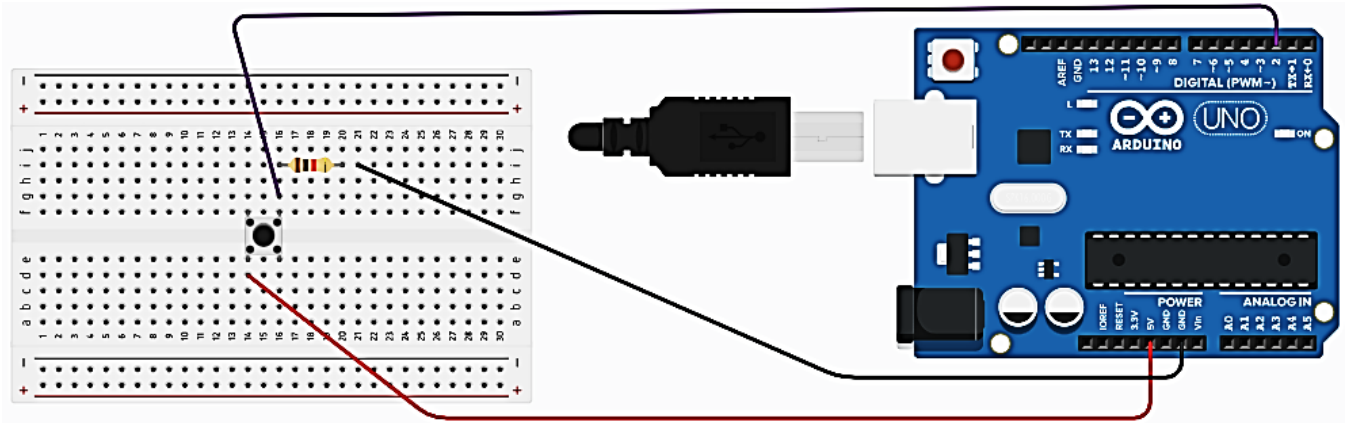


Figure : Arduino board's pin connections with a push button switch.

Experimental Output Results:

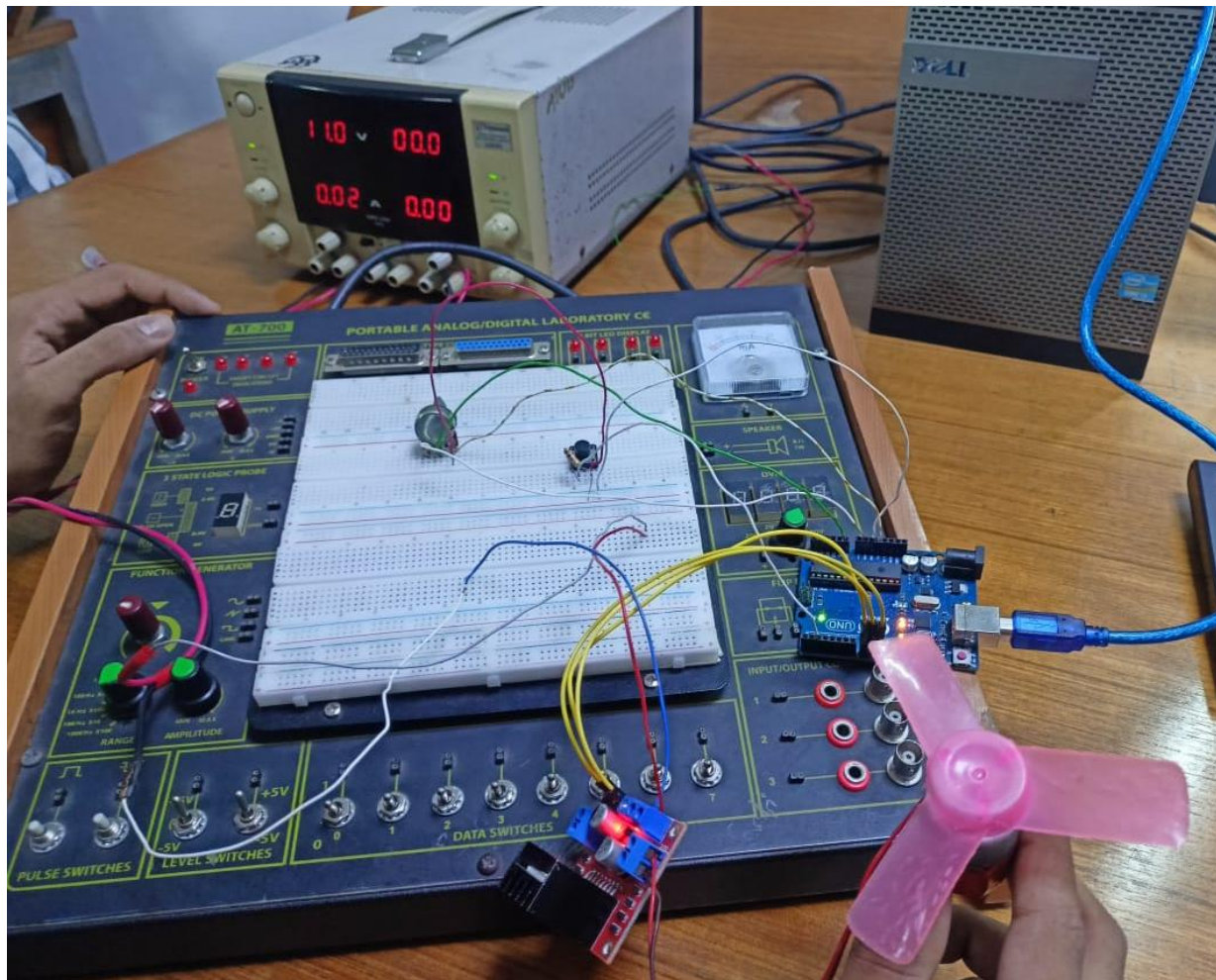


Figure : A motor control system using Arduino in Clockwise direction(Off State)

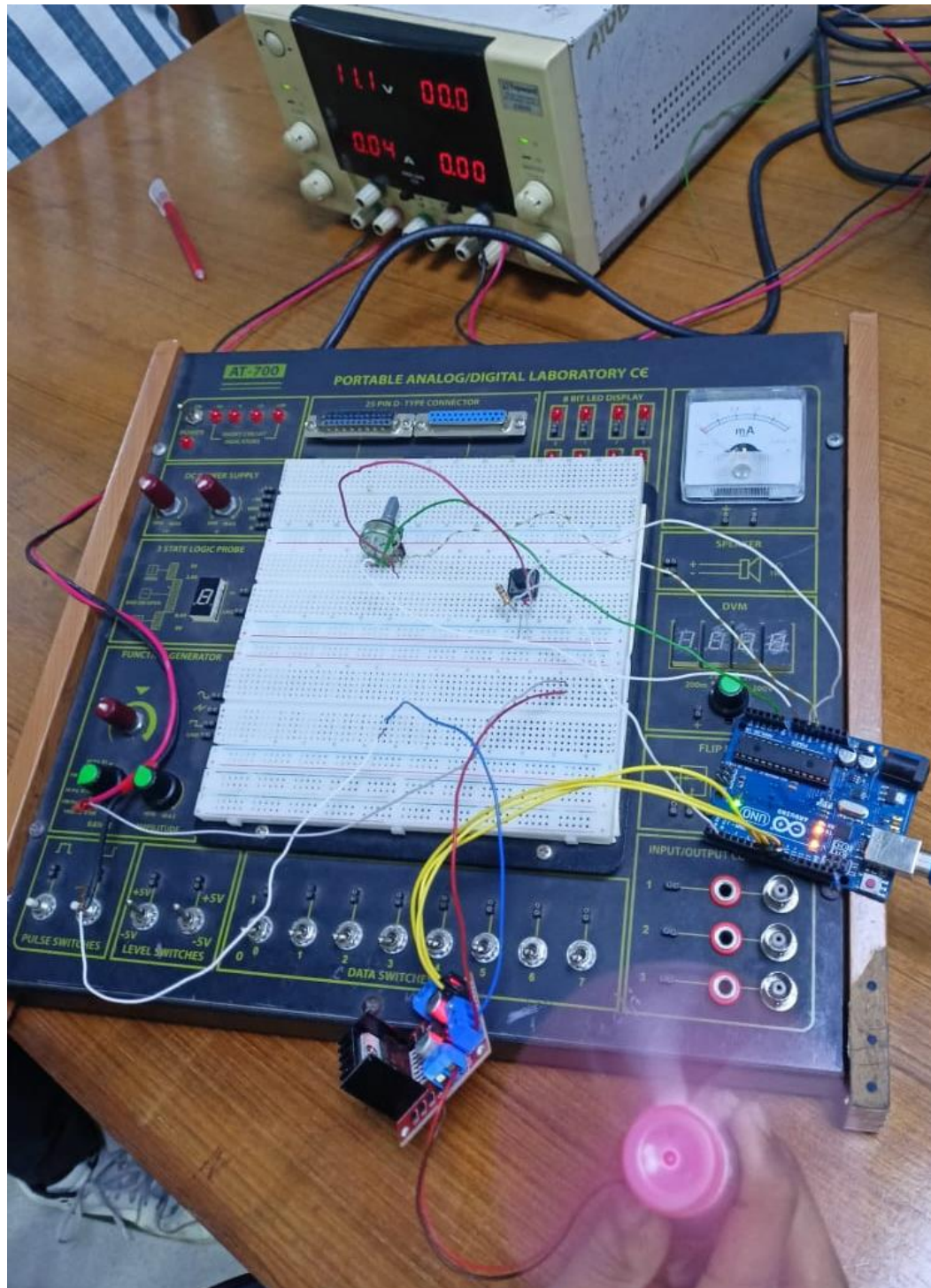


Figure : A motor control system using Arduino in Clockwise direction(On State)

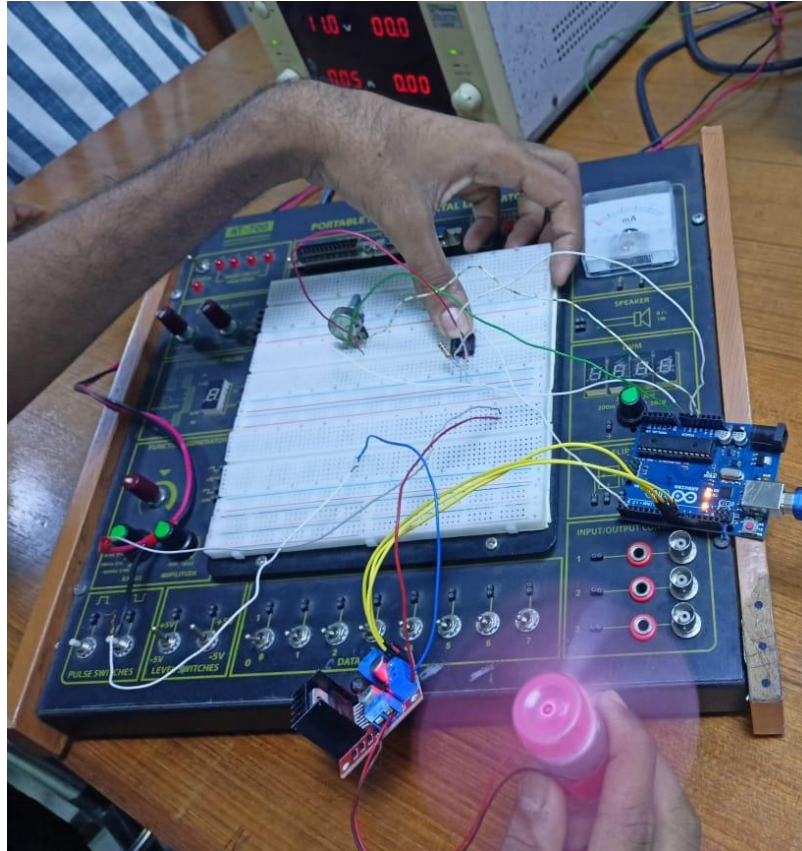


Figure : A motor control system using Arduino in Anticlockwise direction(On State-while switch press)

```

sketch_may07b
Serial.begin(9600);
pinMode(switchrotation, INPUT);
pinMode(in1, OUTPUT);
pinMode(in2, OUTPUT);
pinMode(ConA, OUTPUT);
}
void TurnMotorA1() { //A function to
digitalWrite(in1, LOW); //Switch be
digitalWrite(in2, HIGH);
float analogvalue = analogRead(A0);
int PWMvalue = map(analogvalue, 0,
// range from 0-1023 to 0-255 to di
analogWrite(ConA, PWMvalue); // To a
Serial.println(" The motor is runni
Serial.print("Digital Value = ");
Serial.print(PWMvalue);
}

```

COM7

```

Digital Value = 4 Analog Voltage = 0.00
The motor is running in the clockwise direction.
Digital Value = 4 Analog Voltage = 0.0
The motor is running in the clockwise direction.
Digital Value = 4 Analog Voltage = 0.0
The motor is running in the clockwise direction.
Digital Value = 4 Analog Voltage = 0.0
The motor is running in the clockwise direction.
Digital Value = 4 Analog Voltage = 0.0
The motor is running in the clockwise direction.
Digital Value = 4 Analog Voltage = 0.0
The motor is running in the clockwise direction.
Digital Value = 4 Analog Voltage = 0.0
The motor is running in the clockwise direction.
Digital Value = 4 Analog Voltage = 0.0
The motor is running in the clockwise direction.

```

Sketch uses 4526 bytes (14%) of program memory. Global variables use 340 bytes (16%) of dynamic memory.

Autoscroll Show timestamp Newline 9600 baud Clear output

Figure : A motor control system(Serial Monitor) using Arduino in Clockwise direction(Off State)

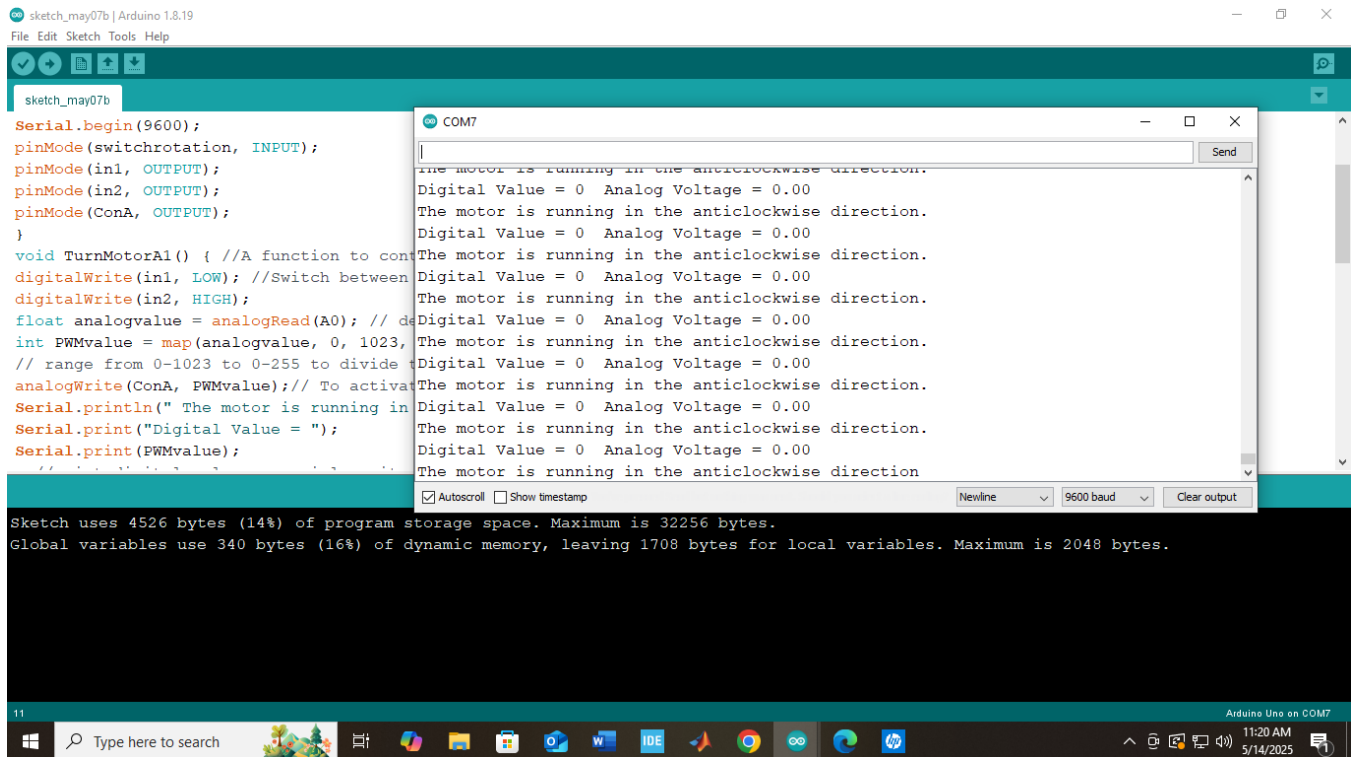


Fig : A motor control system(Serial Monitor) using Arduino in Anticlockwise direction(Off State)

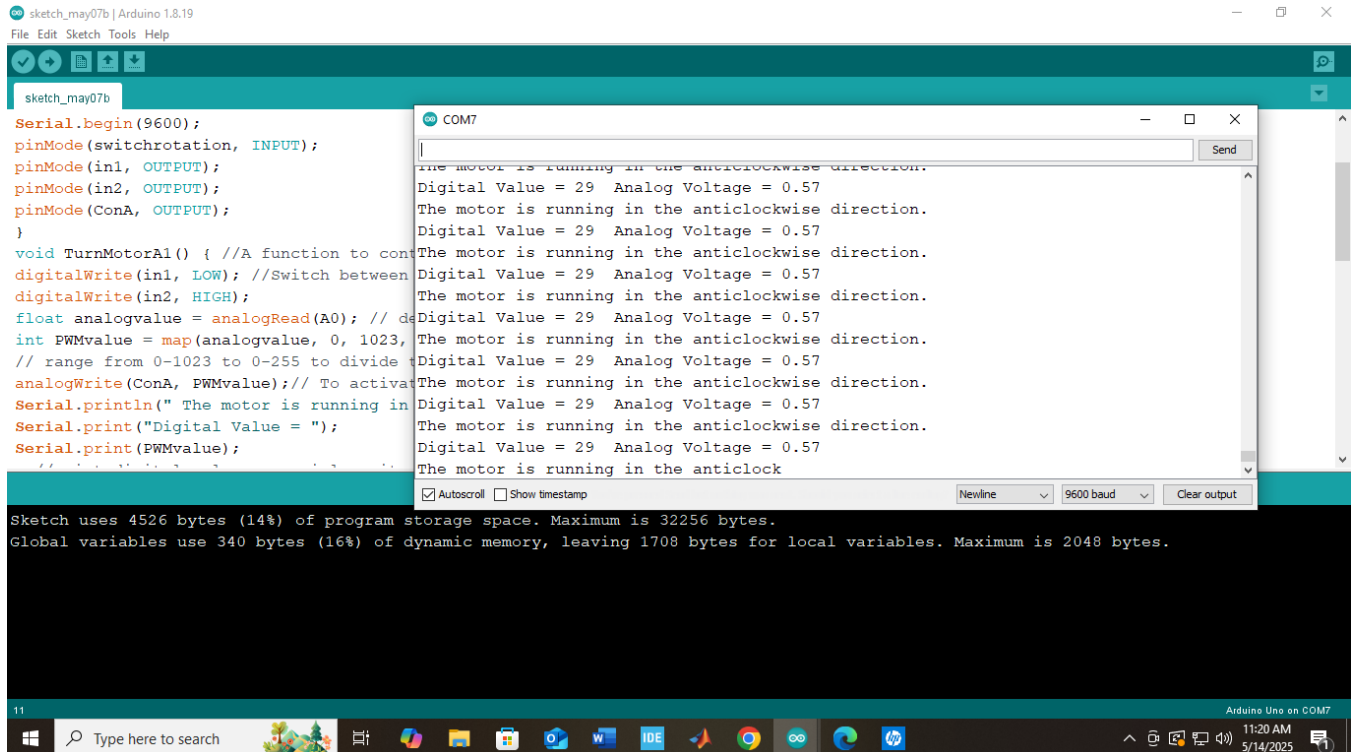


Fig : A motor control system(Serial Monitor) using Arduino in Anticlockwise direction(On State-Medium Speed)

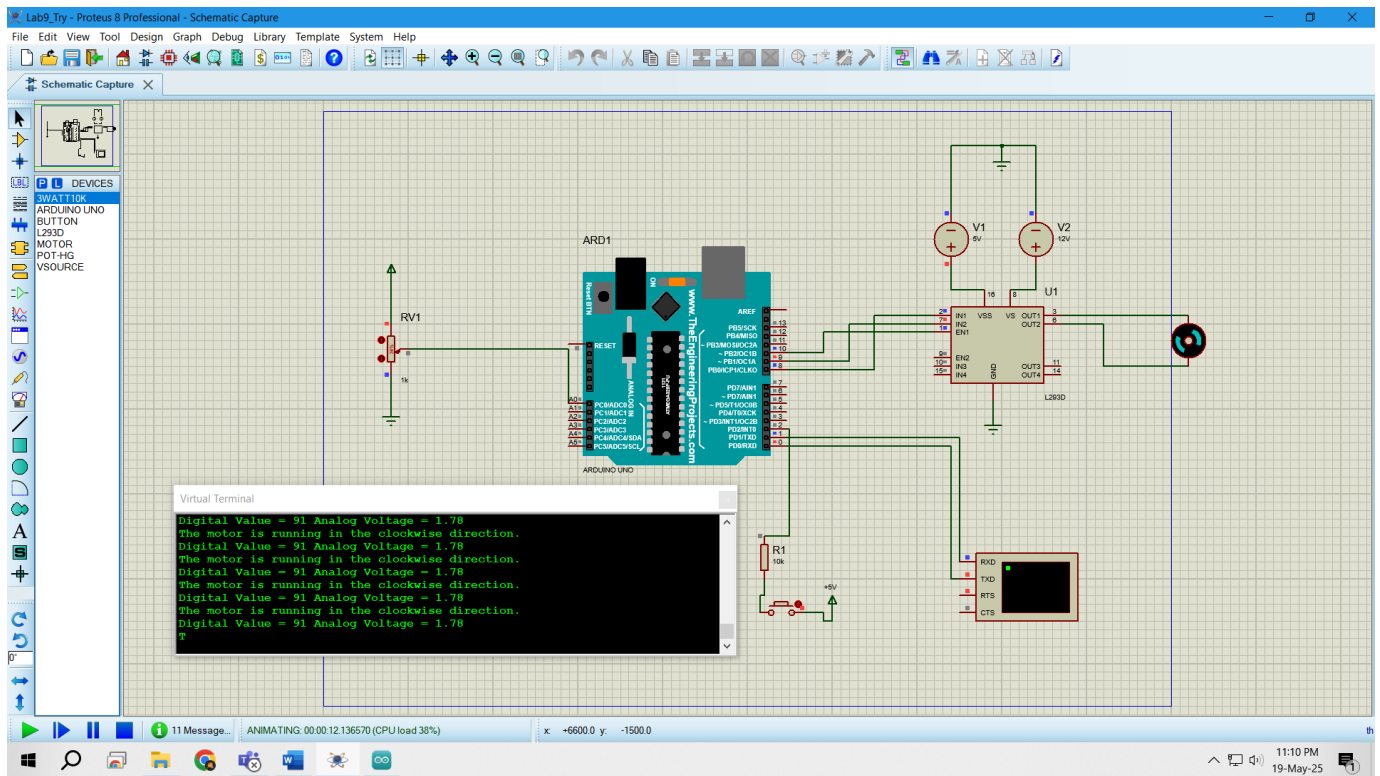


Figure : Simulation of a motor control system using Arduino in Clockwise direction(Mid speed)

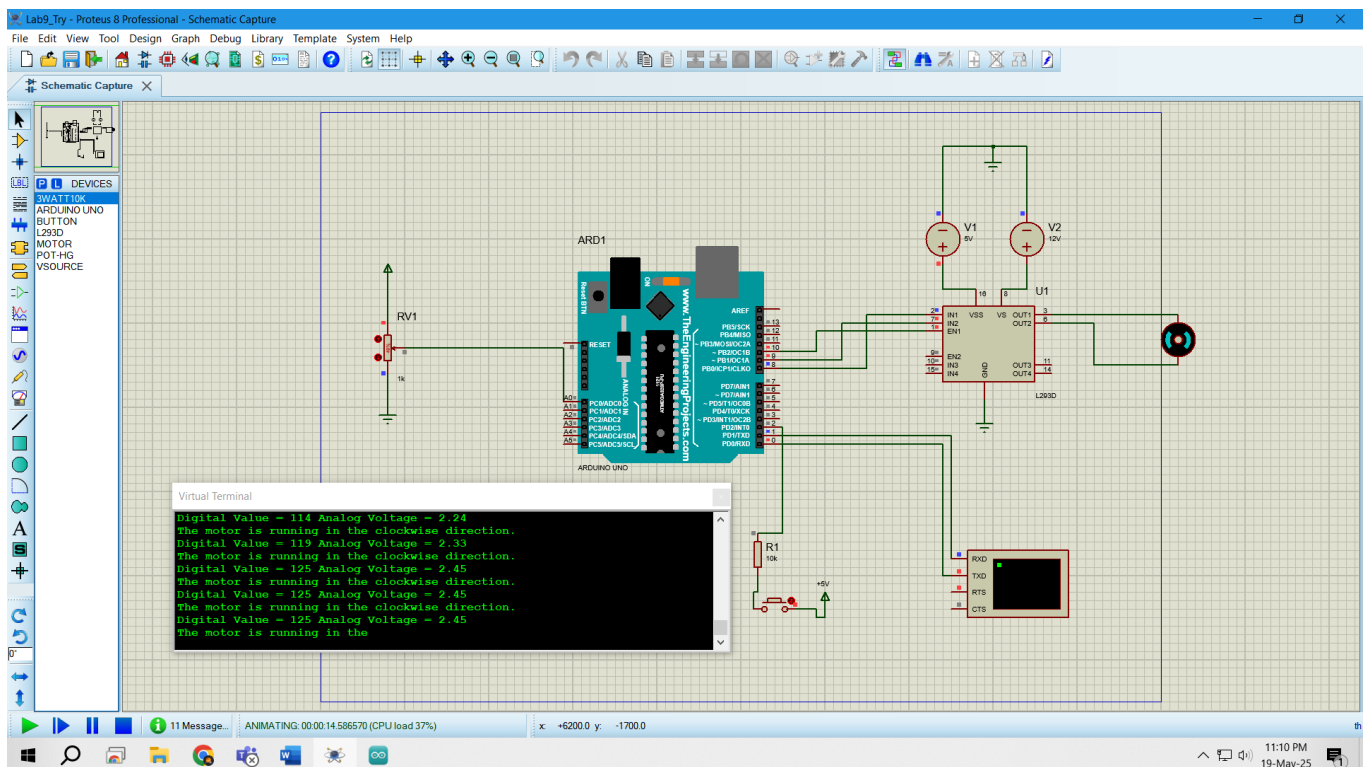
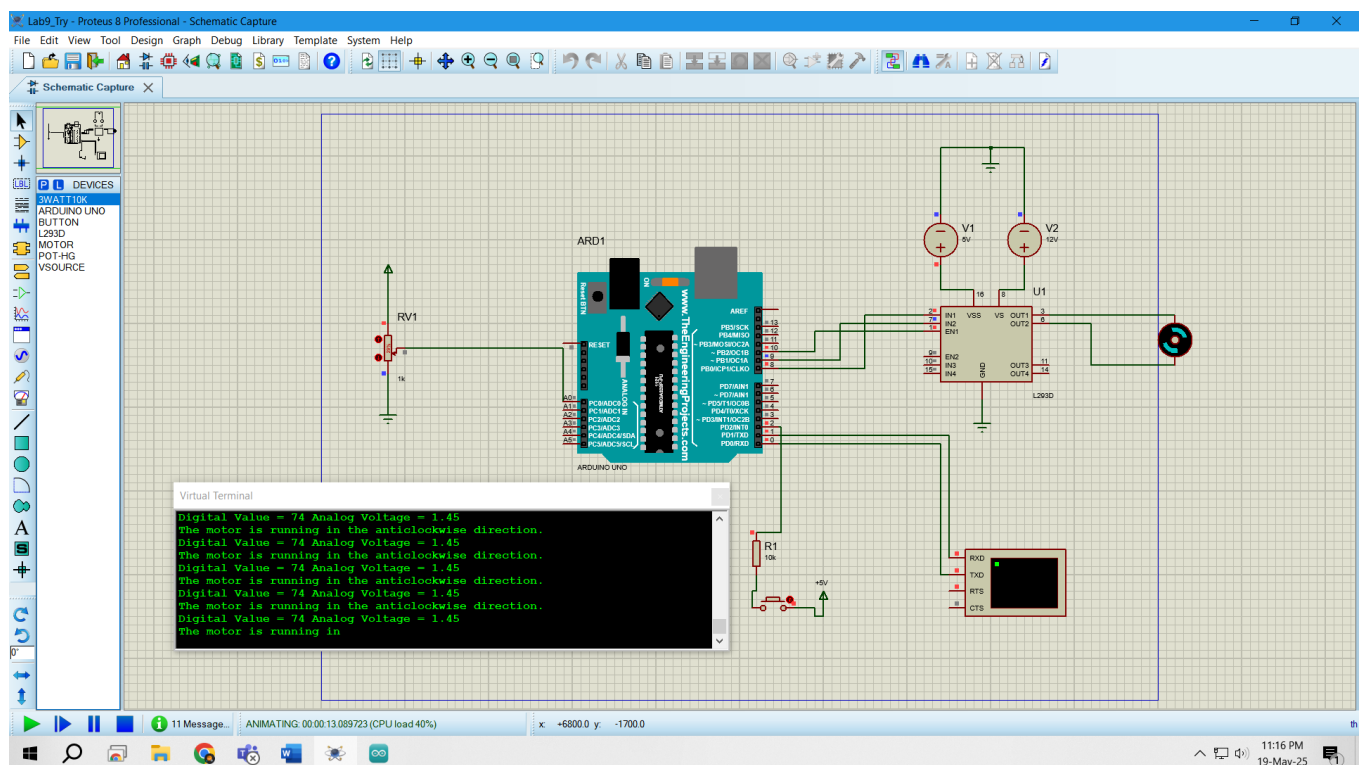
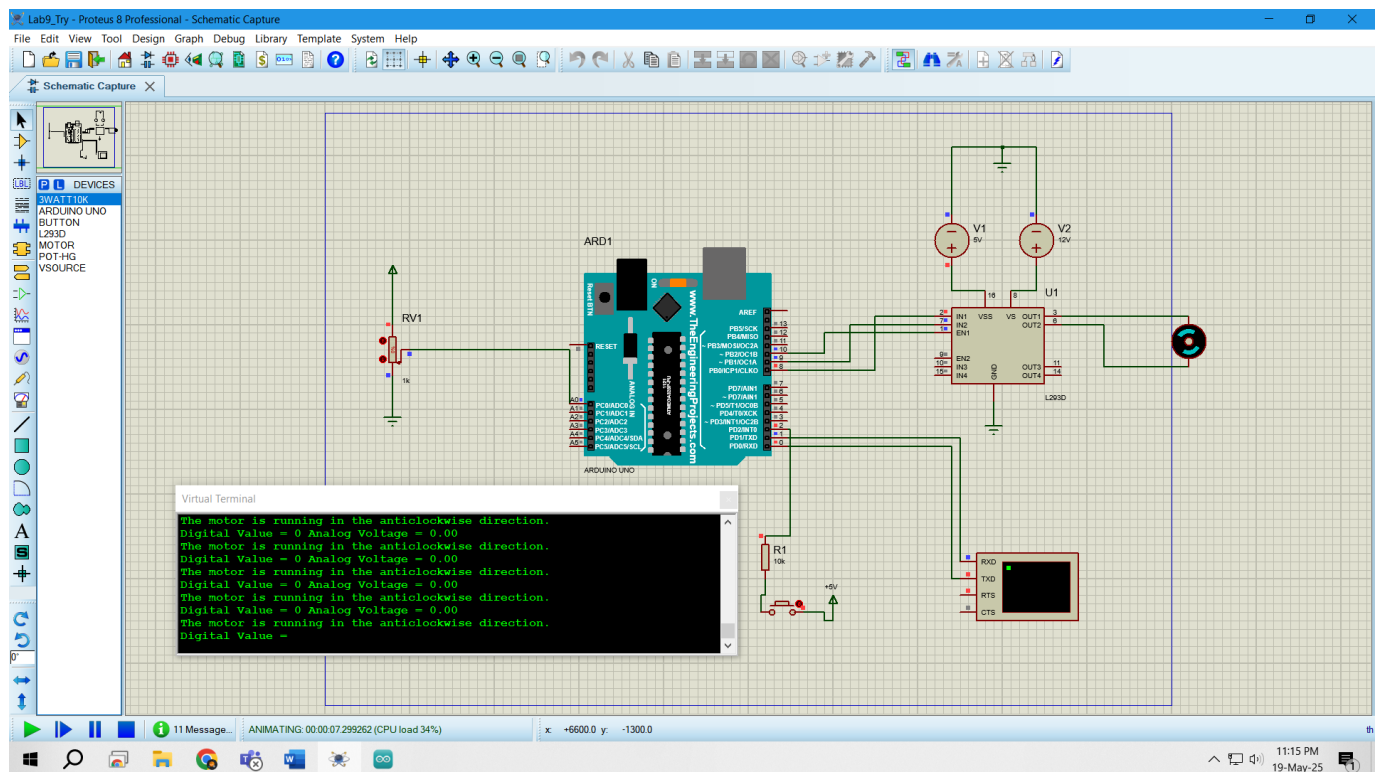


Figure : Simulation of a motor control system using Arduino in Clockwise direction(High speed)



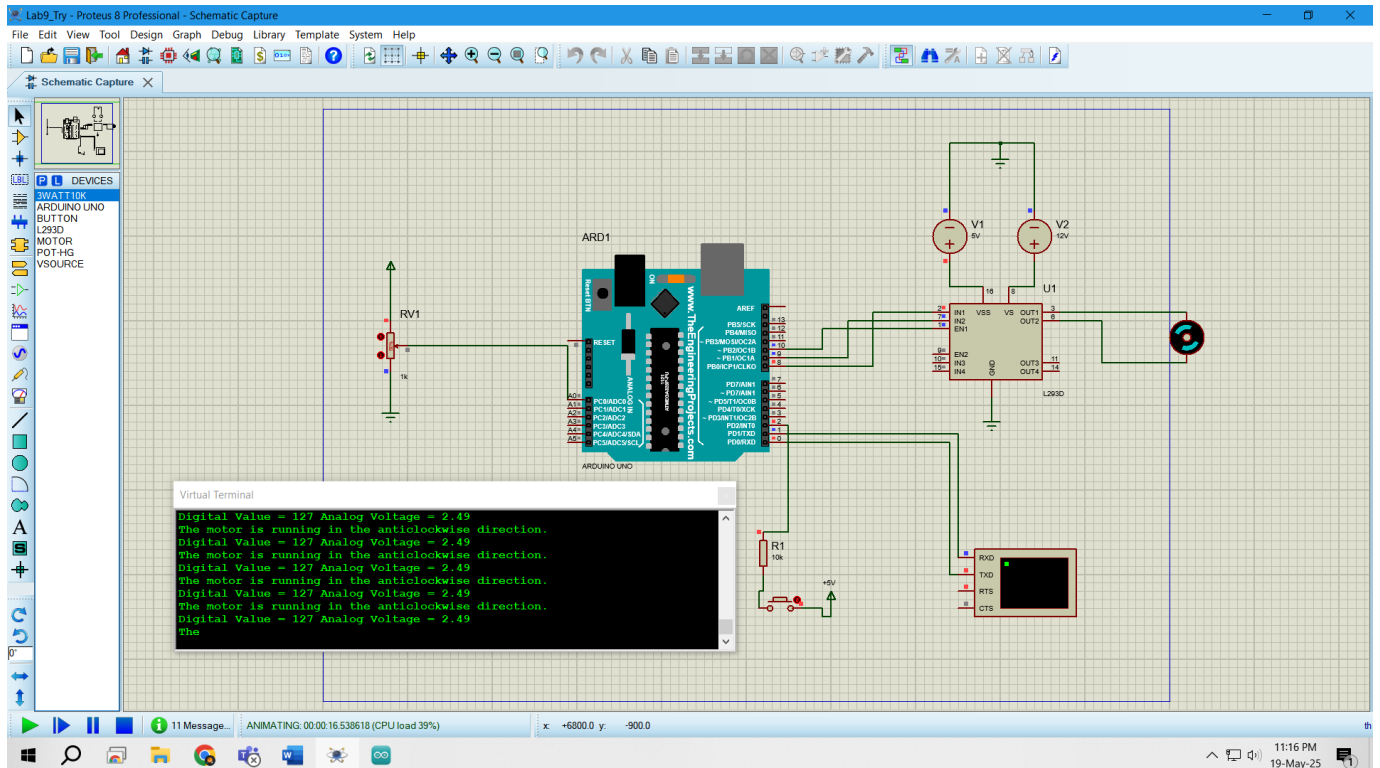


Figure : Simulation of a motor control system using Arduino in Anticlockwise direction(High speed)

Explanation:

- Proteus simulation software was used for circuit simulation.
- Arduino IDE was used to write and compile the code.
- The circuit was first designed in Proteus.
- Then the HEX file generated by Arduino IDE was used in the simulation
- 3 different speeds for both clockwise and anticlockwise direction were observed.

Answers to the Questions in the Lab Manual:

```
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();
}}

```

Explanation:

- int switchrotation = 2;
 - o Assigns digital pin 2 as the input pin to control the motor's rotation direction using an external switch.
- int in1 = 8; and int in2 = 9;
 - o Define digital pins 8 and 9 as control inputs for the H-bridge motor driver to set the motor rotation direction.
- int ConA = 10;
 - o Assigns digital pin 10 (PWM-enabled) to control the motor speed using pulse-width modulation.

- `int speed1;`
 - o Declares a variable (currently unused) to store motor speed.
- `void setup()`
 - o `Serial.begin(9600);` initializes serial communication at 9600 baud rate for debugging.
 - o `pinMode(...)` sets switchrotation as an input pin, and `in1`, `in2`, `ConA` as output pins to interface with the motor driver.
- `void TurnMotorA1()`
 - o Controls the motor to rotate in one direction (e.g., clockwise).
 - o `digitalWrite(in1, LOW); digitalWrite(in2, HIGH);` sets the direction by setting logic levels on H-bridge input pins.
 - o `float analogvalue = analogRead(A0);` reads analog input voltage from pin A0 (typically from a potentiometer).
 - o `int PWMvalue = map(analogvalue, 0, 1023, 0, 255);` converts analog input (0–1023) to PWM range (0–255).
 - o `analogWrite(ConA, PWMvalue);` applies PWM signal to control motor speed.
 - o Serial prints provide real-time feedback on motor direction, PWM value, and equivalent analog voltage.
- `void TurnMotorA2()`
 - o Controls the motor to rotate in the opposite direction (e.g., counter-clockwise).
 - o `digitalWrite(in1, HIGH); digitalWrite(in2, LOW);` reverses the motor direction using H-bridge.
 - o Analog reading and PWM mapping logic is identical to `TurnMotorA1()`.
 - o Serial monitor shows current rotation direction and power level as voltage.
- `void loop()`
 - o Continuously checks the state of the switchrotation pin.
 - o `if (digitalRead(switchrotation) == LOW) → calls TurnMotorA1() to rotate the motor clockwise.`
 - o `else if (digitalRead(switchrotation) == HIGH) → calls TurnMotorA2() to rotate the motor counter-clockwise.`
 - o This allows changing motor direction based on a digital switch input.

Discussions: In this experiment, a motor control system was developed using an Arduino Uno to demonstrate the use of digital input/output and Pulse Width Modulation (PWM). A DC motor was connected through an H-bridge driver, allowing control over its speed and rotation direction. Digital inputs were used to receive external signals, while digital outputs determined the motor's direction by switching voltage polarity. PWM was employed to vary motor speed by adjusting the duty cycle of digital pulses. This method enabled smooth speed control and efficient power usage. Challenges included managing PWM frequency to avoid noise and performance issues. Overall, the experiment successfully achieved its objectives, enhancing our understanding of motor control using microcontroller-based systems.

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.