H-BRIDGE MOTOR DRIVE CIRCUIT (BI-DIRECTIONAL SPEED CONTROLLER USING MOSFETs)By

RAMOLATELA LEBAEA

(221064591)

**A**

**Laboratory Report**

**Submitted as a Partial Requirement**

**For**

**POWER ELECTRONICS**

(POWELB3)

**B. ENG. TECH: Electrical Engineering**

**In**

The Department of Electrical and Electronic Engineering Technology

**FACULTY OF ENGINEERING AND THE BUILT ENVIRONMENT**

September 2024

**Table of Content**

[1. Introduction 3](#_Toc179881364)

[2. Objectives 3](#_Toc179881365)

[3. Practical Description and Circuit Diagrams 3](#_Toc179881371)

[1.1 design choice 3](#_Toc179881372)

[1.2 Circuit Diagram and Description 4](#_Toc179881373)

[ Pin Configuration: 4](#_Toc179881374)

[ Setup Function: 4](#_Toc179881375)

[ Loop Function: 5](#_Toc179881376)

[4. Circuit Resting and Results 5](#_Toc179881377)

[5. technical challenges. 8](#_Toc179881378)

[6. Conclusion and Practical Recommendation: 9](#_Toc179881379)

[7. References List 10](#_Toc179881380)

# Introduction

In H-Bridge Motor Drive Circuit, the concept is to construct an H-bridge motor drive model which could effectively drive a 12V, 2A DC motor. The motor can only run in forward and reverse direction as the polarity of voltage is controlled in motor control circuits and hence H-bridge configuration is frequently used for motor control. This circuit will also enable variable speed control through implementation of Pulse Width Modulation (PWM) this circuit will most appropriate when used in motors that require precision control. This practical will explain the general principles of power electronics and motor drives applied to the DC motors with special reference to the control strategies employed in the operation of a motor.

# Objectives

# To develop a coupled-coil transformer for a power supply of 12V AC or DC with an RMS Current 5A.

# In order to facilitate bi-directional movement of the motor by switching on and off the appropriate H-Bridge configuration.

# Use Pulse Width Modulation Techniques: As a way of achieving speed control of the motor.

# Demonstrate the purpose of developing the functionality of the constructed circuit as well as its capability to perform to the necessary specifications for motor control.

# To design circuit that will work as the H-bridge motor drive circuit to realize simulated circuit and a working prototype for circuit is also to perform a laboratory demonstration of the H-bridge motor drive circuit.

# Practical Description and Circuit Diagrams

## design choice

* I designed and implemented a motor drive circuit that is an H bridge to determine the speed and direction of the 12 Volt, 2A DC motor. In this circuit, I employed MOSFET as switch in order to enable forward and reversed operation of the motor, and PWM control is controls the speed of the motor.

## Circuit Diagram and Description

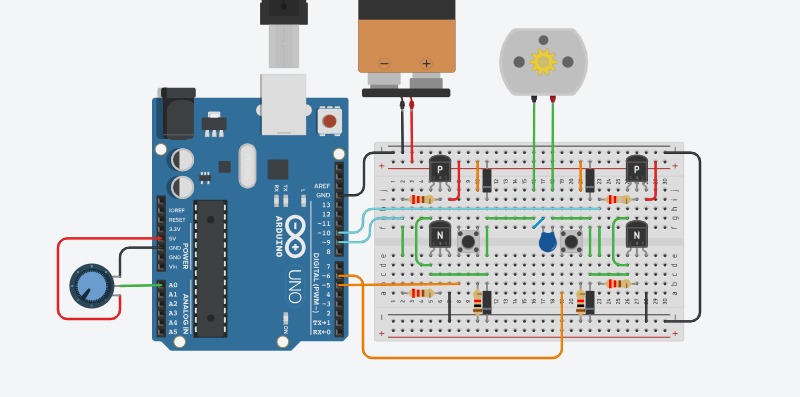


Figure 1: Simulated Motor Drive circuit diagram

* The base of this circuit is H-bridge, where four switching devices are configured in a shape of the letter H. It creates paths by allowing the current to flow in one direction or another within the motor to confer the ability of choosing the direction of the rotations.
* The Arduino creates the PWM signal to the motor to regulate the speed of the motor. PWM signal control changes the average value of the voltage supplied to the motor, as a result, the speed is affected.
* The H-bridge drives a 12V motor and the speed is regulated by the duty cycle of the PWM signal, and to analyze the shape and characteristics of signal PWM waveform is employed.

Here is a code to run the circuit:

### Pin Configuration:

* + Analog Input Pin: int potPin = A0; sets the potentiometer to analog pin A0.
  + PWM Output Pin: int pwmPin = 9; sets the PWM output to pin 9.

### Setup Function:

* + Pin Mode: pinMode(pwmPin, OUTPUT); configures pin 9 as an output pin.

### Loop Function:

* + Reading Potentiometer Value: int potValue = analogRead(potPin); reads the analog value from the potentiometer (range 0-1023).
  + Mapping Value: int pwmValue = map(potValue, 0, 1023, 0, 255); maps the potentiometer value to the PWM range (0-255).
  + Writing PWM Signal: analogWrite(pwmPin, pwmValue); outputs the PWM signal to pin 9.
  + Delay: delay(10); adds a small delay for stability.

# Circuit Testing and Results Analysis

* 1. H-Bridge Switching

In an H-bridge configuration, two switches are usually turned on diagonally (for instance Q1/Q4 or Q2/Q3) hence making it easier for the current to pass either in the forward or reverse direction to control motion. As for the switches in this experiment, the Arduino controls the gate signals of the switches (MOSFETs or BJTs) according to the set direction and speed.

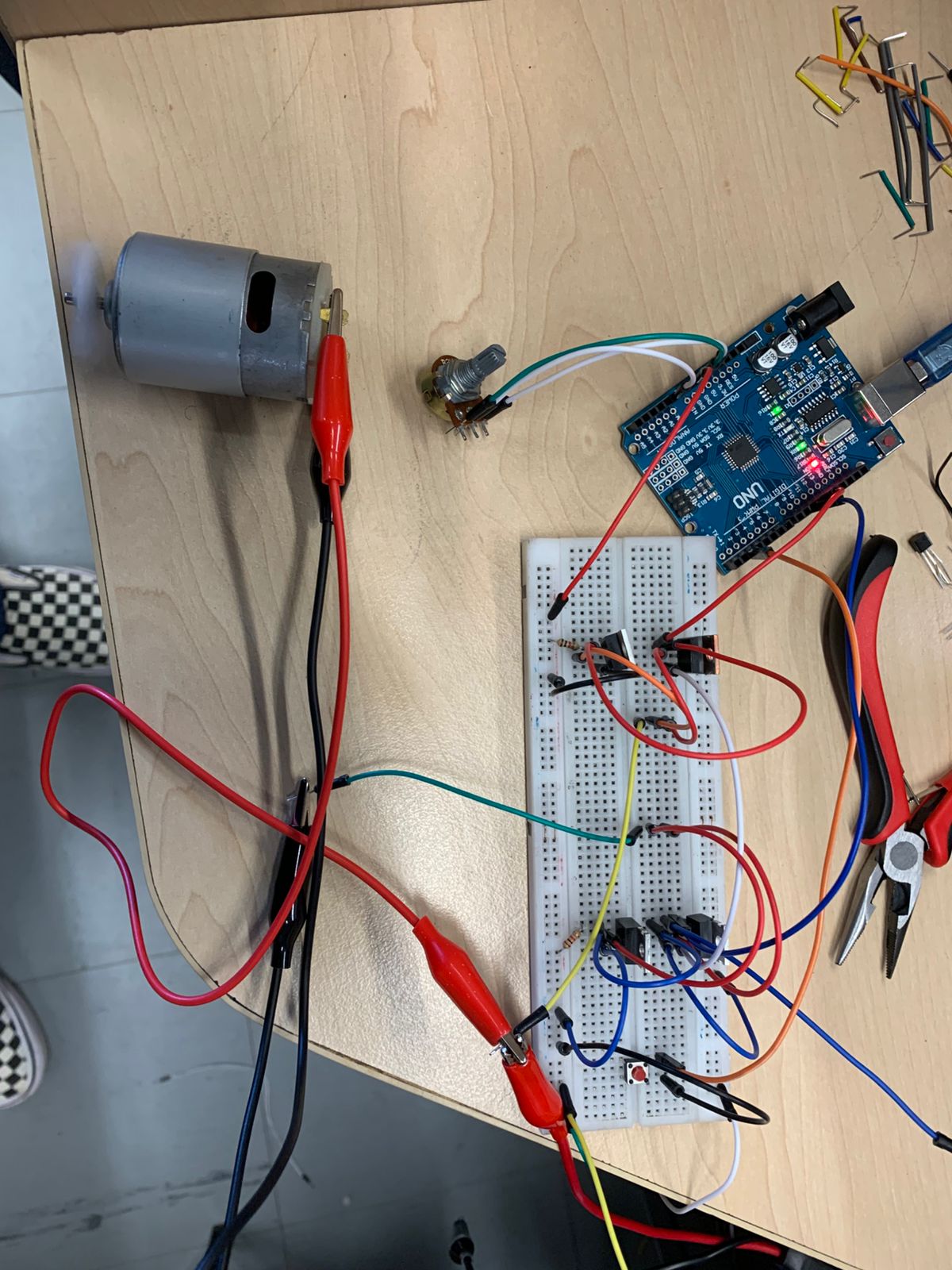
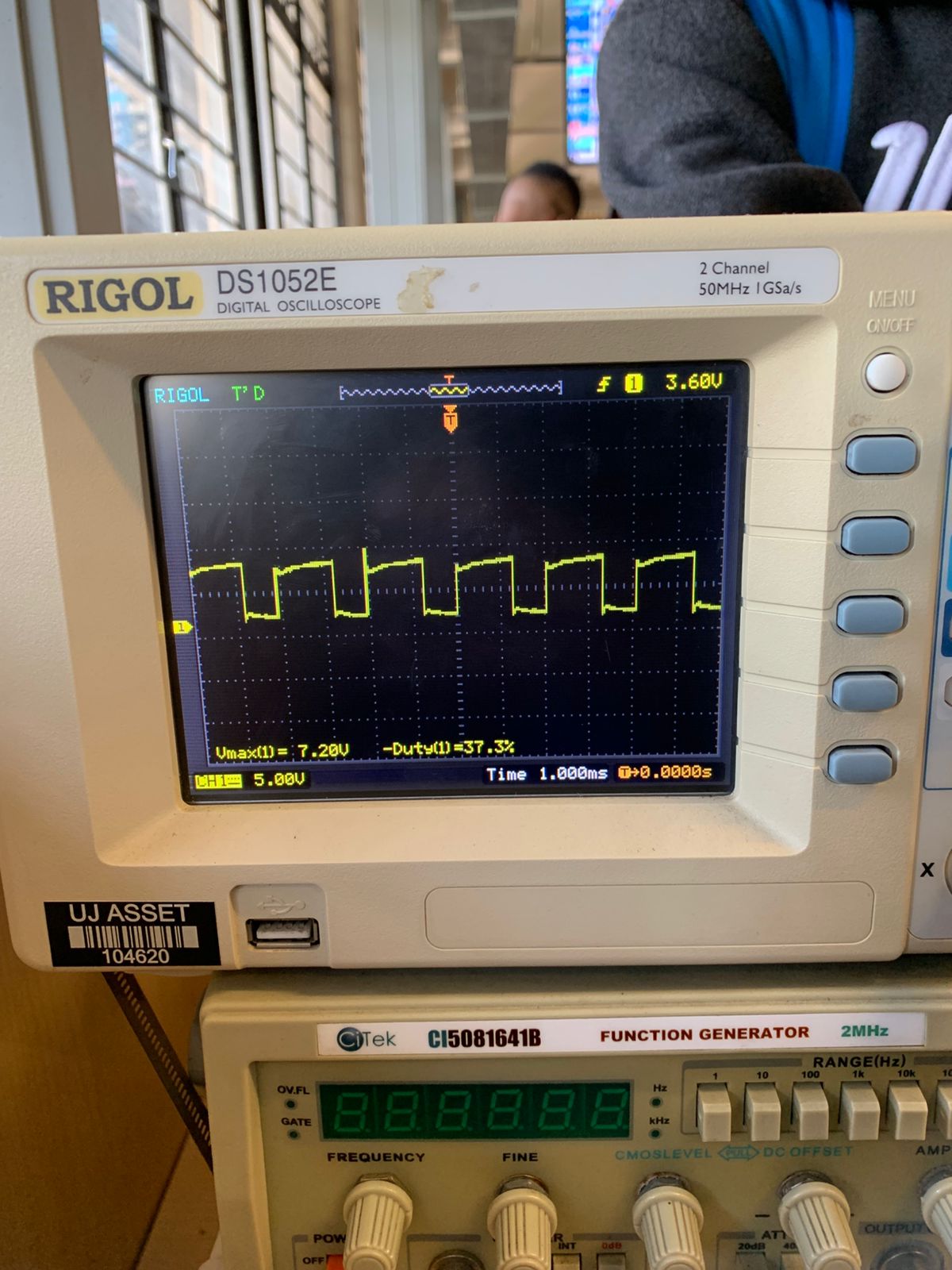


Figure 2: A Running Circuit.

For the motor to rotate back, the Q1 and Q4 switches are turned OFF and instead the Q2 and Q3 switches are turned ON therefore making current to flow in the reversed direction, and when Q2 and Q3 are ON the current flows in a different direction and therefore rotates the motor in a different direction.

* 1. PWM Signal

The Arduino is used in the modulation of the PWM signal which in turn determines the speed of the motor. The oscilloscope reveals the waveform of the PWM signal, as indicted in the figure 2, with an approximate measured frequency and duty cycle of 37.5%. The duty cycle is defined as the ratio of the time that the signal is high relative to one cycle. Higher values of duty cycle enhance the speed of the motor due to higher availability of power, and lower values constrict the speed of the motor.



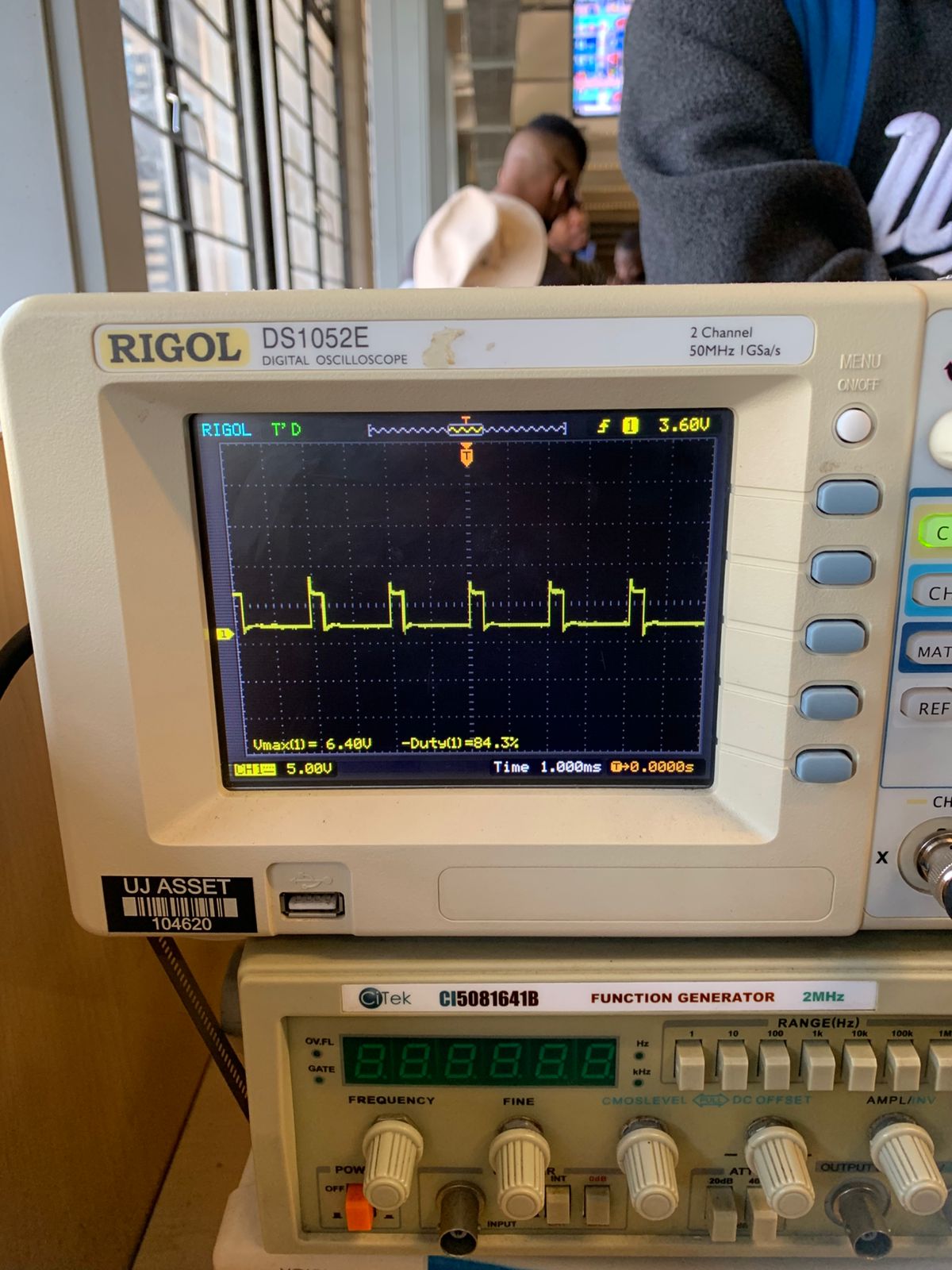


Figure 3: PWM Signal Displayed on Oscilloscope

* 1. PWM Signal:

The Arduino produced the PWM signal to regulate the speed of the motor. The oscilloscope captured the waveform, and two critical duty cycle readings were observed:

* + - Vmax = 6.40V, Duty Cycle = 84.3%: This high duty cycle indicated that the motor was supplied with power 84.3% of the time and that is was operating close to its top speed.
    - Vmax = 7.20V, Duty Cycle = 37.5%: This suggests that the motor is running at a lower speed than before because it draws power for only 37.5 % of the time.
  1. Speed Control:

The speed of the motor was regulated using the PWM duty cycle of the Arduino with the help of a potentiometer interfaced with it.

The oscilloscope captured two distinct PWM duty cycles: 84.3% as well as 37. 5% which illustrates the capability of the circuit in regulating the speed of the motor. At 84.3% duty cycle, the motor moves at a faster rate because more power is supplied and at 37.5%, the motor slows down. From the PWM modulation, both the speed and direction of the motor were regulated, thus proving the efficiency of the H-bridge circuit.

# technical challenges.

* Regarding the results of this practical as seen from the oscilloscope, there was a unique situation where the PWM signal got wiped out when the duty cycle was over 86% and below 22%, and this made it difficult to regulate the speed of the motor and analyze the waveform.
* Potential course might be the lack of PWM signal at higher and lower duty ratios above 86% and below 22% could be attributed to the following. First, the switching on and off of the H-bridge components, MOSFETs, may not be complete at these duty cycles, resulting in signal loss as a result of component saturation. The other possible problem is low gate drive voltage that may also prove inadequate to drive the MOSFETs at these ends.

# Conclusion and Practical Recommendation:

This practical effectively helped in designing and actualizing an H-bridge motor drive circuit. Thus, it was possible to control the speed and direction of the DC motor by changing the PWM duty cycle. Based on the oscilloscope analysis, proper PWM signals were generated with observed duty ratios of 84.3% and 37.5% for the different motor speeds. This experiment effectively shows how PWM control is used in motor control and explores the role of H-bridge circuits when driving motors.

Concerning the signals such as PWM, classic solutions in the LD should ensure that the voltage offers clear cut options for the MOSFETs to cut off and switch on fully for all duty cycles, and set the oscilloscope trigger to the correct value and adjust the oscilloscope bandwidth to the right setting. Assess the stability of voltage levels depending on the high or low cycles of power supply and apply capacitive filtering in voltage signals.

# References List

1. Indrek. (2020, January 27). N-Channel MOSFET as a Switch. Turn ON a 12V Motor with Arduino. (Step-By-Step Guide) [Video]. YouTube. <https://www.youtube.com/watch?v=3PkpOeHTnfo>
2. www.electronicwings.com. (n.d.). PWM in Arduino | Arduino. [online] Available at: https://www.electronicwings.com/arduino/pwm-in-arduino. [Accessed date 04 August 2024].
3. N. Mohan, *Power Electronics: A First Course*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2012.