MECHATRONICS SYSTEM INTEGRATION

MCTA 3203

LAB 05:

DC MOTOR LSS05

SECTION 1

SEMESTER 1, 2025/2026

INSTRUCTOR:

ASSOC. PROF. EUR. ING. IR. TS. GS. INV. DR. ZULKIFLI BIN ZAINAL ABIDIN

DR. WAHJU SEDIONO

DATE OF EXPERIMENT: 5 NOVEMBER 2025

DATE OF SUBMISSION: 12 NOVEMBER 2025

GROUP 3

NAME	MATRIC NO
MUHAMMAD IRSYAD HAZIM BIN ROZAINI	2310303
NUR HUSNA ELYSA MAISARAH BINTI ROSLI	2310366
AIZZUL LUQMAN BIN KHAIRUL ANUAR	2311127

ABSTRACT

This experiment focused on controlling the speed and direction of a DC motor using the LSS05 motor driver and an Arduino microcontroller. The purpose was to understand how Pulse Width Modulation (PWM) and digital control signals can be applied to achieve variable motor motion. By adjusting the PWM duty cycle, we were able to vary the average voltage supplied to the motor, which directly affected its speed. The direction of rotation was controlled using GPIO pins that determined the polarity of the current flow. Overall, the experiment successfully demonstrated the relationship between PWM control and motor speed, while reinforcing practical skills in circuit assembly and Arduino programming.

TABLE OF CONTENTS...3

1.0 INTRODUCTION.. 4

2.0 EQUIPMENTS.. 5

3.0 EXPERIMENTAL SETUP..6

4.0 METHODOLOGY.. 7

5.0 DATA COLLECTION.. 11

6.0 DATA ANALYSIS...12

7.0 RESULTS..13

8.0 DISCUSSION...14

8.1 TASK... 16

9.0 CONCLUSION...21

10.0 RECOMMENDATIONS... 22

11.0 REFERENCES..23

APPENDICES.. 24

ACKNOWLEDGEMENTS..26

STUDENT'S DECLARATION...27

1.0 INTRODUCTION

In mechatronic systems, DC motors play an important role in providing smooth and precise motion control. However, since microcontrollers like the Arduino can only supply limited current, a motor driver is needed to act as an interface between the controller and the motor. In this experiment, the LSS05 motor driver was used to control the speed and direction of a DC motor. The driver works using an H-bridge configuration, which allows the current flow through the motor to be reversed, enabling it to rotate both forward and backward.

The main focus of this experiment was to control motor speed using PWM (Pulse Width Modulation) and to observe how duty cycle variations influence rotation. The session involved setting up the circuit, uploading Arduino code, and observing how different PWM duty cycles affected the motor's rotation speed. By modifying the PWM value in the Arduino code (from 0 to 255), the motors received different average voltages. During testing, both motors rotated properly at higher PWM values, but at lower values, one motor failed to start — possibly due to differences in internal resistance or friction. Since a tachometer was unavailable, a small piece of tape was attached to each motor shaft to visually estimate and compare the speeds at different PWM levels.

2.0 MATERIALS AND EQUIPMENTS

Here are the list of all equipments used in the experiment:

1. 1 Arduino UNO
2. 1 USB Cable
3. 1 LSS05 Motor Driver Shield
4. 2 DC Motors
5. Jumper Wires
6. Breadboard
7. Computer with Arduino IDE

3.0 EXPERIMENTAL SETUP

- Attach the LSS05 Shield on top of the Arduino board.
- Connect the DC motors to Motor A and Motor B terminals.
- Supply external power to the shield’s screw terminal (GND and +9V).
- Ensure GND is common between shield and Arduino.
- Upload the Arduino code using Arduino IDE.

Pin reference (Arduino Uno):

Shield Pin	Arduino Pin	Function
ENA	5	PWM (Speed A)
IN1	4	Motor A Direction
ENB	6	PWM (Speed B)
IN2	7	Motor B Direction

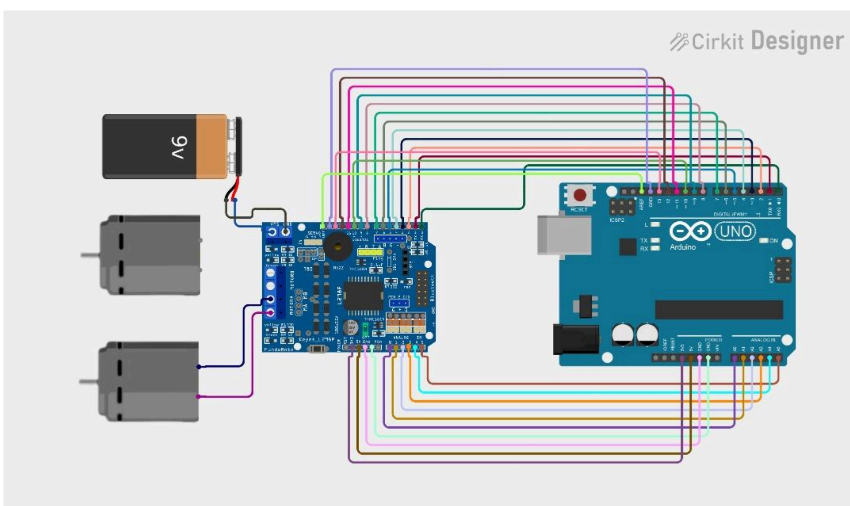

Figure 3.1 The circuit configuration of the motor, motor driver, and Arduino Uno

4.0 METHODOLOGY

1. **Circuit Assembly**
   - The Arduino Uno board was connected to the computer using a USB cable.
   - The LSS05 Motor Driver Shield was carefully stacked on top of the Arduino Uno, ensuring correct pin alignment.
   - The DC motors were connected to the Motor A and Motor B terminal block on the shield using the screw connectors.
   - A 9–12 V external DC power supply was connected to the shield's power terminal (+9V and GND).
   - A common ground between the external power supply and the Arduino was verified to maintain stable operation.
2. **Control Pins Configuration**
   - The directional input pins (IN1 and IN2) and speed control pins (ENA and ENB) were mapped according to the shield's pin configuration.
   - Arduino pins D4 and D7 were used for direction control, while D5 and D6 (PWM capable) were used for speed control.
   - Wiring was checked to ensure no loose connections or reversed polarity.
3. **Programming Logic**
   - The Arduino IDE was launched, and the provided motor control code was opened.
   - The correct board (Arduino Uno) and COM port were selected from the IDE settings.
   - The code was uploaded to the Arduino via USB.

4. Code Used

```
const int DIR_A = 4;
const int EN_A  = 5;
const int DIR_B = 7;
const int EN_B  = 6;

void setup() {
  pinMode(DIR_A, OUTPUT);
  pinMode(EN_A, OUTPUT);
  pinMode(DIR_B, OUTPUT);
  pinMode(EN_B, OUTPUT);
}

void loop() {
  // Motor A – forward
  digitalWrite(DIR_A, HIGH);
  analogWrite(EN_A, 150);  // speed (0-255)
  // Motor B – forward
  digitalWrite(DIR_B, HIGH);
  analogWrite(EN_B, 255);
  delay(2000);

  // Motor A – reverse
```

```
  digitalWrite(DIR_A, LOW);
  analogWrite(EN_A, 150);
  // Motor B – reverse
  digitalWrite(DIR_B, LOW);
  analogWrite(EN_B, 255);
  delay(2000);

  // Stop both motors
  analogWrite(EN_A, 0);
  analogWrite(EN_B, 0);
  delay(1000);
}
```

5. Running the Experiment

- The motor was allowed to rotate in the forward direction with a predetermined PWM value.
- After a fixed duration, the motor direction was reversed by swapping the digital signals for IN1 and IN2.
- The PWM value was systematically changed (e.g., 255, 128, 64), and the motor's response at each value was observed.
- The experiment was repeated to confirm reproducibility and stability of results.
- Observations were recorded in a data table, including direction, speed estimation, and motor behavior.

6. Safety and Precautions

- The external power supply was double-checked before activation to avoid overvoltage.
- The motor was monitored to prevent overheating during prolonged operation.
- Wires were secured to avoid disconnection during rotation.
- The device was powered off before adjusting wiring or motor terminals.

5.0 DATA COLLECTION

PMW	DIRECTION	(RPM)	Remarks
255	Forward	7.3k	Full speed rotation, strong torque
128	Forward	5.9k	Medium speed, moderate torque
255	Reverse	-6.8k	Full speed in reverse direction
64	Reverse	2.3k	Slow rotation in reverse direction

6.0 DATA ANALYSIS

The data obtained from the experiment clearly demonstrated that the speed of the DC motor is directly influenced by the PWM duty cycle applied to the ENA pin of the LSS05 Motor Driver Shield. A higher PWM duty cycle corresponds to a greater average voltage supplied to the motor, thereby increasing its speed. When the PWM value was set to 255, the motor received nearly full voltage and rotated at maximum speed. Reducing the PWM value to 128, which represents roughly half of the duty cycle, decreased the motor speed proportionally, while further lowering it to 64 significantly reduced the rotation speed. This relationship shows that the motor speed is approximately proportional to the PWM value, although slight deviations may occur due to friction, internal resistance, and mechanical load. By this information, it can expressed as

$$Speed \propto Duty\ Cycle\ (\%)$$

or approximately

$$Speed\ (RPM) = k \times Duty\ Cycle\ (\%),$$

The experiment also confirmed how digital logic controls the motor's direction. Setting the IN1 pin to HIGH and IN2 to LOW resulted in forward rotation, while reversing the logic caused the motor to rotate in the opposite direction. When both IN1 and IN2 were set HIGH, the motor terminals were shorted internally, which produced a braking effect that immediately stopped the motor. This verified that the LSS05 shield not only supports smooth speed variation through PWM control but also provides a reliable way to reverse direction and apply braking. Overall, the behavior of the motor was consistent with the theoretical operation of an H-Bridge motor driver.

7.0 RESULTS

The results of the experiment confirmed that the LSS05 Motor Driver Shield successfully controlled the speed and direction of a DC motor through GPIO and PWM signals. The motor's rotational speed increased proportionally with higher PWM duty cycles, while its direction could be reversed simply by changing the logic states of the input pins. The experiment also showed that braking could be achieved by setting both input pins to HIGH simultaneously. These results validated the theory that PWM control allows efficient motor speed regulation by adjusting the average voltage delivered to the motor, without wasting energy through resistive losses.

In conclusion, the experiment achieved all its objectives. The system effectively demonstrated the working principles of PWM-based speed control, direction control through GPIO signals, and the practical use of the LSS05 Motor Driver Shield in mechatronic systems. The findings illustrate that by using PWM and digital control, it is possible to manage a DC motor's operation smoothly and precisely, which is essential for applications such as robotics, automation, and mobile platforms.

8.0 DISCUSSION

The experiment demonstrated successful control of the DC motor using the LSS05 motor driver, which differed from the L298P module introduced in the class notes. Because the LSS05 has a different pin configuration and operating method, the Arduino code and wiring setup had to be adjusted. This required changing the defined pin numbers, reassigning direction and PWM control pins, and ensuring that the control signals matched the specifications of the LSS05 driver. Despite these adjustments, the motor responded correctly to the programmed commands, showing accurate speed and direction control.

Although most results matched theoretical expectations, several discrepancies were observed. For instance, the motor did not respond smoothly at very low PWM values. It occasionally required a slightly higher PWM threshold to start rotating due to static friction and minimum startup voltage requirements. The motor speed also did not scale perfectly linearly with PWM values, which is common for brushed DC motors and can be influenced by internal resistance, mechanical friction, and supply voltage fluctuations. These small variations did not affect the overall understanding of the motor behavior but are worth noting.

There were several sources of error and limitations during the experiment. First, speed estimation was done manually without a tachometer, which introduced observation errors. The absence of a tachometer or encoder made speed measurement less precise. Second, voltage drops and internal losses within the LSS05 motor driver may have affected the actual voltage reaching the motor terminals, especially under heavier load conditions. Third, minor fluctuations in external power supply output could lead to inconsistent torque and rotational speed. Additionally, motor heating over time may have reduced efficiency slightly, causing variation between repeated trials.

Overall, the experiment confirmed that the LSS05 motor driver can effectively control motor speed and direction using PWM signals and custom pin mapping. Despite having to modify the code to adapt to the different motor drivers and the minor limitations observed, the results strongly support the theoretical concepts and demonstrate a clear understanding of DC motor control principles.

8.1 TASK

1. Explain the function of the ENA and ENB pins.

- ENA and ENB are the enable pins for Motor A and Motor B, it controls the speed and activation of Motor A and Motor B. They receive the PWM signal directly from Arduino. A higher PWM signal increases the motor speed, while setting them low makes the motor decrease in terms of speed or if it is too low it will stop the motor.

2. Describe the reason PWM is used for speed control.

- PWM (Pulse Width Modulation) is used for speed control because it allows the motor to receive a controlled amount of average voltage without wasting power. Instead of supplying a constant voltage, PWM rapidly switches the motor's power ON and OFF at high frequency. By adjusting the duty cycle (the percentage of time the signal is ON), the motor receives different levels of average power:
  - Higher duty cycle → more average voltage → faster speed
  - Lower duty cycle → less average voltage → slower speed

  This method is efficient, precise, and prevents overheating or energy loss compared to using resistors or lowering the supply voltage directly. PWM also provides smooth and stable speed control even at low speeds.

3. Describe the outcome when both IN1 and IN2 are set HIGH.

- When both IN1 and IN2 are set HIGH, the motor driver activates the braking mode. In this condition, both terminals of the motor receive a HIGH signal, causing the motor to stop quickly. Instead of freely slowing down (coasting), the motor experiences an electronic brake because both inputs apply the same voltage level, preventing the motor from rotating.

4. Explain how braking can be implemented using the LSS05.

- Braking on the LSS05 motor driver is implemented by applying the same logic level to both motor input pins (IN1 and IN2). When both inputs are set either HIGH–HIGH or LOW–LOW, the motor terminals receive the same voltage, resulting in no potential difference across the motor. This condition forces the motor to stop rapidly because the induced back EMF is shorted internally, creating an electronic braking effect.

  In summary:

  - IN1 = HIGH and IN2 = HIGH → active braking
  - IN1 = LOW and IN2 = LOW → active braking

  This braking method makes the motor decelerate faster compared to coasting (where one pin is HIGH and the other is LOW or left floating). It is commonly used when precise stopping or quick response is required.

  .

5. Modify the code to control two DC motors simultaneously.

```
// Motor A pins for LSS05
const int ENA = 5;   // PWM for Motor A
const int IN1 = 4;   // Direction pin 1
const int IN2 = 8;   // Direction pin 2

// Motor B pins for LSS05
const int ENB = 6;   // PWM for Motor B
const int IN3 = 7;   // Direction pin 3
const int IN4 = 9;  // Direction pin 4

void setup() {
  pinMode(ENA, OUTPUT);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);

  pinMode(ENB, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
}

void loop() {
  // Move both motors forward
```

```
digitalWrite(IN1, HIGH);
digitalWrite(IN2, LOW);
digitalWrite(IN3, HIGH);
digitalWrite(IN4, LOW);

analogWrite(ENA, 200); // Speed motor A
analogWrite(ENB, 200); // Speed motor B

delay(2000);

// Move both motors backward
digitalWrite(IN1, LOW);
digitalWrite(IN2, HIGH);
digitalWrite(IN3, LOW);
digitalWrite(IN4, HIGH);

analogWrite(ENA, 200);
analogWrite(ENB, 200);

delay(2000);

// Stop both motors
analogWrite(ENA, 0);
```

```
  analogWrite(ENB, 0);


  delay(1000);
}
```

9.0 CONCLUSION

The experiment demonstrated the use of the LSS05 motor driver to control two DC motors via PWM and directional GPIO signals. The results showed that PWM effectively controls motor speed by adjusting the average voltage supplied to the motors. However, it was also observed that not all motors respond equally at lower PWM values, as one motor stopped rotating around PWM 100. This suggests that slight differences in motor torque or internal characteristics can affect performance at lower voltages. Using the tape method gave a rough idea of the speed difference between PWM values, but a tachometer would have provided more precise results. Overall, the experiment successfully met its objectives and gave practical insight into real-world limitations of DC motor control.

10.0 RECOMMENDATIONS

1. Use a tachometer in future experiments to obtain accurate RPM readings for all PWM levels.
2. Test each motor individually to identify any performance variations or mechanical friction before running them together.
3. Include current and voltage measurements to better understand torque behavior at low PWM values.
4. Implement serial monitoring to display PWM and estimated speed values during operation for clearer analysis.
5. Ensure all wiring connections and ground references are secure to maintain consistent motor performance.

11.0 REFERENCES

REEES52. (n.d.). *LSS05 Auto-Calibrating Infrared Line Tracking Sensor*. https://rees52.com/products/lss05-auto-calibrating-infrared-line-tracking-sensor-reflective-line-sensor-module-intelligent-line-tracking-sensor#:~:text=Description:,Raspberry%20Pi%2C%20and%20other%20microcontrollers.

jhdpcb. (2024, February 28). Motor Driver Fundamentals: Your Guide to Efficient Motor Control - JHDPCB. *jhdpcb*. https://jhdpcb.com/blog/efficient-motor-control/#:~:text=The%20key%20role%20of%20the,commands%20and%20real%2Dworld%20actuation.

Robocraze. (2022, June 9). *What is Motor Driver*. Robocraze. https://robocraze.com/blogs/post/what-is-motor-driver?srsltid=AfmBOorMnMBboXFKkQS-YZN6Jiq_7SpuGjrHy4lMEXKy-rJPITw-6C2J

APPENDICES

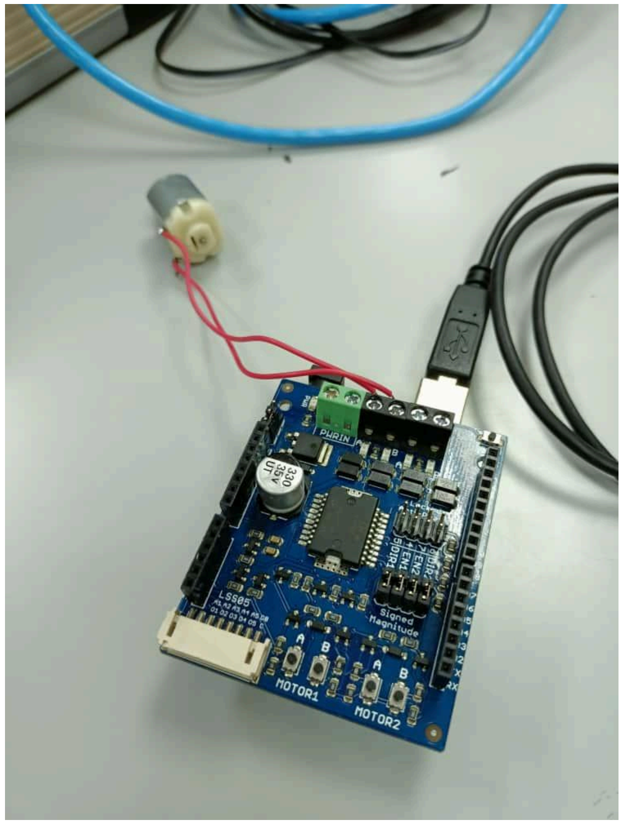

Figure 12.1 Real-Life Circuit Design

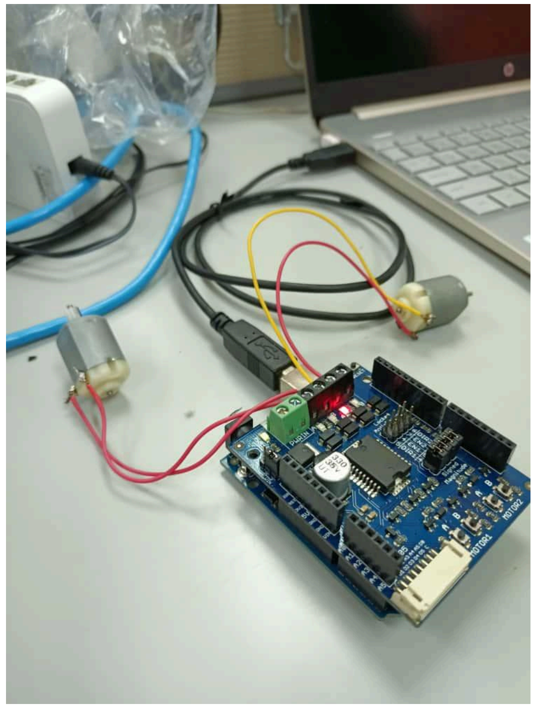

Figure 12.2 Real-Life Circuit Design using 2 motors

ACKNOWLEDGEMENTS

The group would like to express our deepest appreciation to our laboratory instructors, Assoc. Prof. Eur. Ing. Ir. Ts. Gs. Inv. Dr. Zulkifli bin Zainal Abidin and Dr. Wahju Sediono for their continuous guidance, encouragement, and clear explanations throughout the course of this week's experiment on DC Motors. Their expertise and detailed feedback greatly enhanced our understanding in Mechatronics and System Integration applications. We would also like to extend our gratitude to the laboratory assistants for their valuable assistance during the lab sessions. Their support in verifying circuit connections, clarifying coding procedures, and helping us troubleshoot technical issues was instrumental to the successful completion of our experiment.

In addition, we would like to thank our classmates and also group members, Muhammad Irsyad Hazim bin Rozaini, Nur Husna Elysa Maisarah binti Rosli and Aizzul Luqman bin Khairul Anuar for their cooperation and willingness to share insights, suggestions, and resources during the experiment. Collaborative discussions and teamwork played a crucial role in identifying and resolving problems efficiently. Lastly, we appreciate the opportunity provided by the Department of Mechatronics Engineering to conduct this experiment, which allowed us to strengthen our practical skills in Arduino programming, electronic interfacing, and logical circuit design. The collective guidance and support from all parties involved contributed significantly to the success and learning outcome of this project.

STUDENT'S DECLARATION

Certificate of Originality and Authenticity

This is to certify that we are **responsible** for the work submitted in this report, that **the original work** is our own except as specified in the references and acknowledgement, and that the original work contained herein have not been untaken or done by unspecified sources or persons.

We hereby certify that this report has **not been done by only one individual** and **all of us have contributed to the report.** The length of contribution to the reports by each individual is noted within this certificate.

We also hereby certify that we have **read** and **understand** the content of the total report and no further improvement on the reports is needed from any of the individual's contributors to the report.

We therefore, agreed unanimously that this report shall be submitted for **marking** and this **final printed report** has been **verified by us.**

Signature: — Read [/]
Name: Muhammad Irsyad Hazim bin Rozaini — Understand [/]
Matric Number: 2310303 — Agree [/]
Contribution: Abstract, Introduction, Conclusion, Recommendations

Signature: — Read [/]
Name: Nur Husna Elysa Maisarah binti Rosli — Understand [/]
Matric Number: 2310366 — Agree [/]
Contribution: Materials and Equipments, Experimental Setup, Methodology, Discussion

Signature: Read [/]
Name: Aizzul Luqman bin Khairul Anuar Understand [/]
Matric Number: 2311127 Agree [/]
Contribution: Data Collection, Data Analysis, Results, Appendices