MECHATRONICS SYSTEM INTEGRATION (MCTA 3203)

SEMESTER 1, 2025/2026

WEEK 5: L298N Motor Driver Shield with GPIO (ver 1.0)

SECTION 1

GROUP 20

LECTURER: DR. WAHJU SEDIONO

DATE OF EXPERIMENT: Monday, 3rd November 2025

DATE OF SUBMISSION: Monday, 10th November 2025

PREPARED BY:

NAME	MATRIC NUMBER
ALI RIDHA BIN MUHAMMAD AMINUDIN	2316137
SITI NORATHIRAH BINTI RAZALLY	2319788
NUR ALYA SAKINAH BINTI MOHD MOKHTAR	2319444

ABSTRACT

In this experiment, we learnt how to control a DC motor using an L298N motor driver with an Arduino. The main goal was to see how the motor's speed and direction could be controlled through the GPIO pins and PWM signals. The motor driver was connected to the Arduino using jumper wires, then the Arduino was attached to the output terminals. After uploading the program, the motor was able to move forward and backward depending on the signal stated. The motor's speed will change when the PWM values change. This showed how the duty cycle affects its movement. At the end of this experiment, we gained a better understanding of how PWM works and how the L298N motor driver can be used to control a DC motor in a simple and effective way.

TABLE OF CONTENTS

1.0 INTRODUCTION

This experiment is mainly about learning how to control a DC motor using the L298N Motor Driver Shield together with an Arduino UNO. The idea is to see how we can make the motor move forward, backward, and at different speeds by using both hardware and coding. It also introduces the concept of Pulse Width Modulation (PWM), which is used to control the motor speed. By doing this, we can understand how simple changes in programming and wiring can affect how the motor behaves. The experiment is also meant to give hands-on experience in connecting electronic parts and understanding how they work together in a basic control system.

A DC motor changes electrical energy into movement. When current flows through the motor, it creates a magnetic field that makes the shaft spin. The L298N Motor Driver Shield helps control the direction and speed of that spinning. It has a type of circuit called an H-bridge that allows the current to move in both directions, so the motor can rotate either way. This shield is also important because it protects the Arduino board from drawing too much power from the motor, which could damage the system.

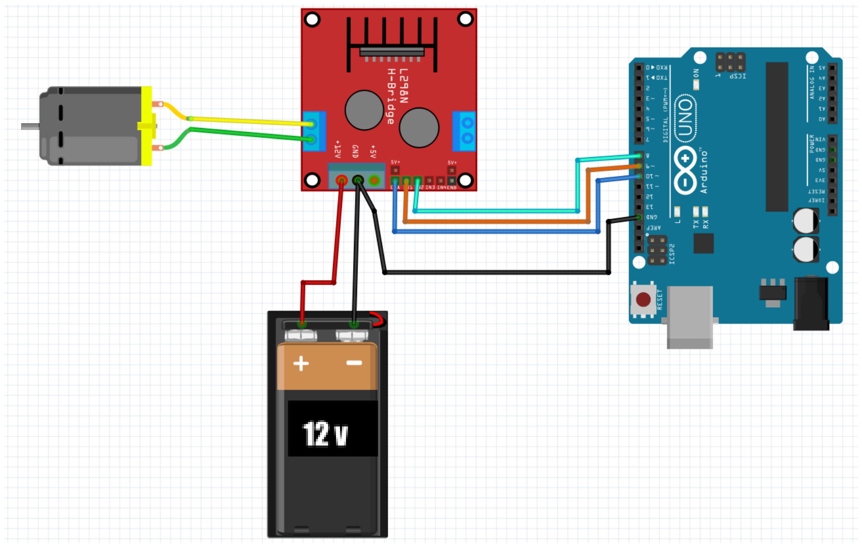
The speed of the motor is controlled using PWM, which basically turns the motor's power on and off very quickly. Even though the switching happens fast, the motor doesn't stop instantly. Instead, the average power it gets changes. If the "on" time, or duty cycle, is high, the motor spins faster. When the duty cycle is low, the motor slows down. This method is simple, efficient, and doesn't waste much energy, which makes it very useful in motor control systems.

It is expected that through this experiment, the DC motor's speed and direction can be properly controlled using the L298N Motor Driver Shield. Increasing the PWM value should make the motor turn faster, while lowering it should reduce the speed. Changing the input signals should also reverse the motor's rotation. When both inputs are high, the motor should stop, which works like a braking effect. Overall, this experiment helps show how basic electronics and programming can work together to control real mechanical systems, just like in robotics or automated machines.

2.0 MATERIALS AND EQUIPMENTS

1. Arduino UNO
2. L298N Motor Driver Shield
3. DC Motor (6V–12V) 1–2
4. External Power Supply (9V/12V)
5. Jumper Wires
6. Breadboard

3.0 EXPERIMENTAL SETUP



1. The L298N motor driver was connected to the Arduino using jumper wires.
2. A DC motor was connected to the output terminals of the motor driver.
3. The Arduino board was powered through a USB cable from the laptop.
4. An external power supply was used to provide enough voltage for the motor's operation.
5. The GPIO pins from the Arduino were connected to the input pins of the L298N to control the direction of the motor.
6. A PWM-capable pin from the Arduino was connected to the enable pin on the motor driver for speed control.
7. All connections were double-checked before powering the circuit.
8. The Arduino code was uploaded to test the motor in forward and reverse directions.
9. The PWM values in the code were adjusted to observe changes in the motor speed.

4.0 METHODOLOGY

4.1 Circuit Assembly

1. Connect motor driver L298N to Arduino Uno:
- IN1 → 12 (Motor A direction)
- IN2 → 13 (Motor A direction)
- IN3 → 10 (Motor Bdirection)
- IN4 → 11 (Motor Bdirection)
- ENA → 4 (Speed control A)
- ENB → 5 (Speed control B)
- GND → GND
2. Connect DC motor to motor driver L298N:
- Motor + (red) → OUT3
- Motor - (black) → OUT4
3. Connect encoder of DC motor to Arduino Uno:
- Green → GND
- Blue → 5V
- Yellow → Encoder A
- White → Encoder B
4. Connect motor driver L298N to external power supply
- +12V → +12V
- GND → COM

4.2 Programming Logic

1. Pin configuration & initialization
    - Define all motor control pins (ENB, IN3, IN4) for speed and direction
    - Set each pin as an output inside setup() so the Arduino can control the motor driver
2. Motor forward control
    - Set direction pins (IN3 = HIGH, IN4 = LOW) to make the motor spin forward
    - Apply full PWM speed (analogWrite(ENB, 255))
    - Maintain motion for a specific duration (delay(2000))
3. Motor reverse control
    - Reverse the direction (IN3 = LOW, IN4 = HIGH)
    - Keep the same speed (255)
    - Run for 2 seconds before stopping or changing again
4. Motor stop control
    - Cut off the PWM signal (analogWrite(ENB, 0)) to stop the motor
    - Wait for a short delay (delay(1000)) before looping again

4.3 Code Used

```
// Motor A pin definitions
const int ENB = 5; // PWM pin for speed control
const int IN3 = 10; // Direction
const int IN4 = 11; // Direction

void setup() {
pinMode(ENB, OUTPUT);
pinMode(IN3, OUTPUT);
pinMode(IN4, OUTPUT);
}
void loop() {
// Move forward
digitalWrite(IN3, HIGH);
digitalWrite(IN4, LOW);
analogWrite(ENB, 255); // Speed: 0-255
delay(2000);
// Move backward
digitalWrite(IN3, LOW);
digitalWrite(IN4, HIGH);
analogWrite(ENB, 255);
delay(2000);
// Stop
analogWrite(ENB, 0);
delay(1000);

}
```

4.4 Control Algorithm

1. Prepare the system and motor driver for operation.
    - Define all motor pins (ENB, IN3, IN4)
    - Set all pins as outputs inside setup()
    - Ensure the motor control circuit is ready before running the main loop
2. Control the motor's movement (forward and reverse) with speed adjustment.
    - Forward motion:
        - Set IN3 = HIGH, IN4 = LOW
        - Apply PWM speed signal (analogWrite(ENB, 255))
        - Run for 2 seconds
    - Reverse motion:
        - Set IN3 = LOW, IN4 = HIGH
        - Apply same speed signal (analogWrite(ENB, 255))
        - Run for 2 seconds
3. Stop the motor and control the looping operation.
    - Stop motor by cutting PWM output (analogWrite(ENB, 0))
    - Wait for a short delay (delay(1000)) before restarting the loop
    - Repeat the cycle continuously

5.0 DATA COLLECTION

PWM	Direction	RPM
255	Forward	27.8k
128	Forward	20.5k
255	Reverse	-28.8k
64	Reverse	-10.7k

6.0 DATA ANALYSIS

During the experiment, the speed of the DC motor was tested at different PWM values to observe how the duty cycle affects its performance. The results showed that when the PWM value increased, the motor's rotation speed also increased. For instance, at a lower PWM value like 64, the motor rotated slowly, while at 128 it moved faster, and at 255 it reached its maximum speed. This shows that the motor speed is directly proportional to the duty cycle of the PWM signal. The relationship can be expressed as:

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

or approximately

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

meaning the higher the duty cycle, the greater the average voltage supplied to the motor. No complex calculations were needed since the trend was clear and consistent. Small differences in speed could be caused by factors such as friction, load, or minor voltage drops. Overall, the data confirmed that PWM is an effective way to control the speed of a DC motor using the L298N driver and that the direction of rotation can be easily managed through the control signals from the Arduino.

7.0 RESULT

After assembling the circuit with the L298N motor driver and Arduino UNO, the system successfully controlled the DC motor's rotation and speed as expected. When the Arduino program was uploaded, the motor began rotating in the forward direction when the input pins (IN1 = HIGH and IN2 = LOW) were activated. At a PWM value of 255, the motor rotated at its maximum speed, producing a strong and consistent motion, resulting in the RPM being 27.8k. When the PWM value was reduced to 128, the rotation speed visibly decreased to about half of the maximum speed, showing a clear relationship between the PWM duty cycle and motor speed and resulting in the RPM being 20.5k.

When the motor's direction pins were reversed (IN1 = LOW, IN2 = HIGH), the motor changed direction and rotated in the reverse orientation. At a PWM value of 255 in reverse mode, the speed was again at its maximum, comparable to the forward direction, resulting in the RPM being -28.8k. Similarly, when the PWM was lowered to 64, the motor rotated slowly, resulting in the RPM being -10.7k. When the PWM was set to 0, the motor stopped completely, confirming that the enable pin (ENA) controls whether power is delivered to the motor. The experiment also confirmed that when both IN1 and IN2 were set to HIGH, the motor immediately stopped with a sudden halt, demonstrating the effect of active braking.

In summary, the results showed that increasing the PWM duty cycle produced higher motor speeds, reversing the input pins changed the direction, and equal logic levels at the inputs caused the motor to stop.

TASK:

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

- ENA and ENB are enable pins for Motor A and Motor B.
- When connected to a PWM output, they control the speed of each motor.
- If ENA/ENB = HIGH, the motor is enabled; if LOW, the motor is disabled.
- Varying the PWM signal on these pins adjusts the motor's speed.

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

PWM (Pulse Width Modulation) is used because it allows efficient control of motor speed without wasting power as heat.
It rapidly switches the motor ON and OFF:

- A higher duty cycle gives more average voltage → higher speed.
- A lower duty cycle gives less average voltage → slower speed.

PWM is energy-efficient and provides smooth control.

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

When both IN1 and IN2 are HIGH, both terminals of the motor are at the same voltage level, causing no voltage difference across the motor. The motor stops immediately.

4. Explain how braking can be implemented using the L298N.

Braking is implemented by setting both input pins (IN1 & IN2) to the same logic level (both HIGH or both LOW) while keeping the enable (ENA) HIGH.
This short-circuits the motor terminals, creating a counter current that stops the motor quickly, called active braking.

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

```
// Motor A pins

const int ENA = 4;

const int IN1 = 12;

const int IN2 = 13;

// Motor B pins

const int ENB = 5;

const int IN3 = 10;

const int IN4 = 11;


void setup() {

  pinMode(ENA, OUTPUT);

  pinMode(IN1, OUTPUT);

  pinMode(IN2, OUTPUT);

  pinMode(ENB, OUTPUT);

  pinMode(IN3, OUTPUT);

  pinMode(IN4, OUTPUT);

}

void loop() {

  // Forward

  digitalWrite(IN1, HIGH); digitalWrite(IN2, LOW);

  digitalWrite(IN3, HIGH); digitalWrite(IN4, LOW);

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

  delay(2000);

  // Reverse

  digitalWrite(IN1, LOW); digitalWrite(IN2, HIGH);

  digitalWrite(IN3, LOW); digitalWrite(IN4, HIGH);

  analogWrite(ENA, 200);

  analogWrite(ENB, 200);

  delay(2000);

  // Stop

  analogWrite(ENA, 0);

  analogWrite(ENB, 0);

  delay(1000);

}
```

8.0 DISCUSSION

The experiment demonstrated the fundamental operation of DC motor control using the L298N motor driver module in conjunction with an Arduino UNO. The L298N operates as a dual H-bridge driver, which allows current to flow in either direction through the motor. This bidirectional current flow enables the motor to rotate both forward and backward depending on the logic states of the input pins. In this setup, the ENA pin on the L298N module was used to control the motor's speed through PWM (Pulse Width Modulation), while IN1 and IN2 controlled the rotation direction.

PWM was a crucial component in this experiment, as it provided a method of adjusting the motor's speed without directly varying the supply voltage. Instead, PWM rapidly switches the motor's power supply on and off at high frequency. The duty cycle, or the proportion of time the signal stays high within one period, determines the effective voltage applied to the motor. A higher duty cycle results in higher average voltage and therefore greater speed, while a lower duty cycle decreases the voltage and slows the motor. This method is highly efficient and commonly used in motor control systems because it minimizes power loss and allows smooth acceleration or deceleration.

The behaviour of the motor also confirmed the theory of H-bridge control. When IN1 was set HIGH and IN2 LOW, the current flowed in one direction, producing forward rotation. Reversing these inputs reversed the current flow, changing the rotation direction. Setting both inputs to the same state (both HIGH or both LOW) caused the motor to stop due to the absence of potential difference across the terminals. When both were HIGH and the enable pin remained active, the motor performed active braking, which rapidly stopped the rotation due to the counteracting current in the circuit.

Overall, the experiment validated the principle that motor direction depends on the polarity of the input signals, while speed is determined by the PWM duty cycle on the enable pin. The use of the L298N module provided a safe and effective means of interfacing a DC motor with the Arduino for bidirectional and variable-speed control.

9.0 CONCLUSION

In conclusion, the experiment successfully achieved its objectives of controlling the speed and direction of a DC motor using the L298N motor driver module and Arduino UNO. The results clearly demonstrated that the L298N module can effectively drive a DC motor in both directions by manipulating the logic states of the input pins and that the speed can be precisely regulated through the use of PWM signals on the enable pins. The experiment also showed that setting both inputs to the same logic level provides a braking effect, allowing quick stops when necessary.

The overall performance of the system confirmed the theoretical understanding of PWM and H-bridge operation. As the PWM value increased, the speed of the motor increased proportionally, and reversing the input states changed the direction of rotation. The L298N driver proved to be a reliable and efficient component for controlling DC motors in mechatronic applications. This experiment provided valuable practical experience in interfacing electronic hardware and programming microcontrollers for motor control, which is fundamental knowledge in robotics and automation systems.

10.0 RECOMMENDATIONS

To make this experiment more interactive and comprehensive in the future, a couple of modifications could be considered. First, the provided sample code only controls the motor speed using a fixed (hard-coded) PWM value within the code, such as `analogWrite(ENA, 200)`. An engaging improvement would be to connect a potentiometer (variable resistor) to one of the "ANALOG IN" pins on the shield. We could then modify the code to read the analog value from the potentiometer and map it to the PWM range (0-255). This would allow us to physically and in real time control the motor's speed by turning a knob, making the concept of speed control much more tangible.

There are several crucial takeaways any student performing this experiment can learn. The most significant lesson revolves around power management. The safety precaution to never power motors directly from the Arduino 5V pin is paramount. Motors require significantly more current, which is why we use an external power supply (9V/12V). Along with this, ensuring the "GND" (ground) is common between the shield and the Arduino is another critical step that is often overlooked. Without a common ground, the circuit may not function correctly. Finally, always double-check your pins. The manual itself warns that pin mapping can vary by shield model. Knowing exactly which Arduino pins control speed (like D3 for ENA and D5 for ENB) and direction (like D12 and D13 for Motor A) is essential for saving time on debugging.

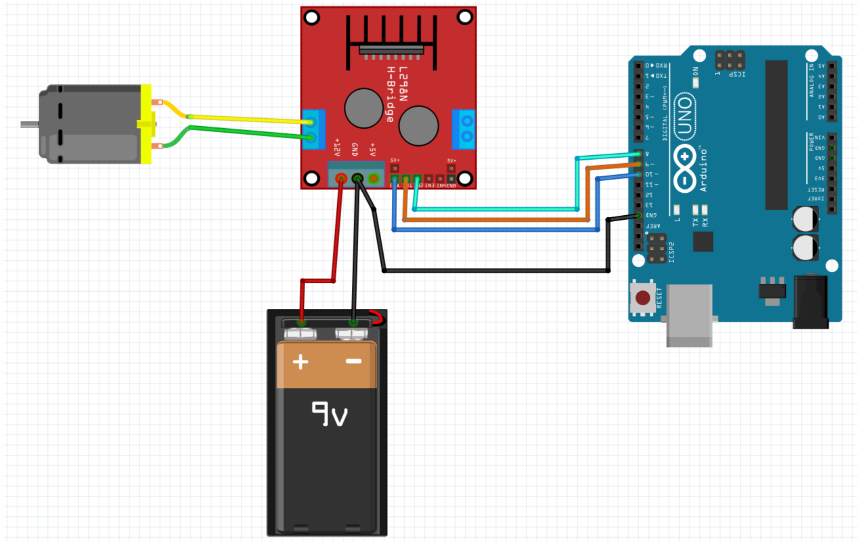
11.0 REFERENCES

Instructables. (2017, November 15). *Tutorial for L298 2Amp Motor Driver Shield for Arduino*. Instructables. https://www.instructables.com/Tutorial-for-L298-2Amp-Motor-Driver-Shield-for-Ard/

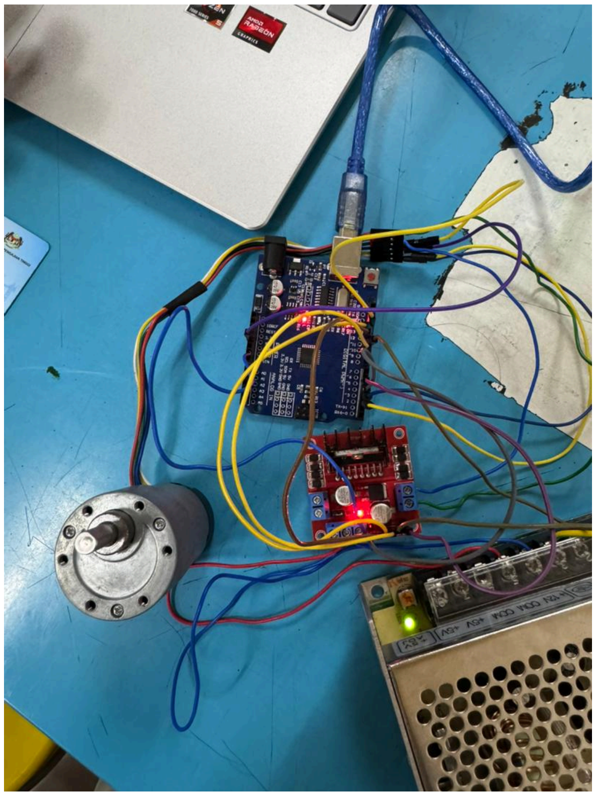
Cirkit Design. (2025). *How to Use the L298P drive shield*. Cirkitdesigner.com. https://docs.cirkitdesigner.com/component/a25f6e70-294e-42fd-b77c-9e9349b76b9a/l298p-drive-shield

LastMinuteEngineers. (2018, November 28). *In-Depth: Interface L298N DC Motor Driver Module with Arduino*. Last Minute Engineers. https://lastminuteengineers.com/l298n-dc-stepper-driver-arduino-tutorial/

12.0 APPENDICES



Wiring diagram for experiment 5



Experiment 5

ACKNOWLEDGEMENTS

Special thanks to Dr. Wahju Sediono for his guidance and support during this experiment. We would also like to extend our gratitude to our instructors, peers, and the laboratory staff for their valuable insights and assistance throughout the experiment. Their feedback and encouragement have greatly contributed to the success of this project. Finally, we appreciate the resources and learning materials provided, which enabled us to understand and implement serial communication effectively.

Certificate of Originality and Authenticity

We hereby certify that we are responsible for the work presented in this report. The content is our original work, except where proper references and acknowledgements are made. We confirm that no part of this report has been completed by anyone not listed as a contributor. We also certify that this report is the result of group collaboration and not the effort of a single individual. The level of contribution by each member is stated in this certificate. Furthermore, we have read and understood the entire report, and we agree that no further revisions are required. We collectively approve this final report for submission and confirm that it has been reviewed and verified by all group members.

Signature: [illegible]	Read [/]
Name: ALI RIDHA BIN MUHAMMAD AMINUDIN	Understand [/]
Matric Number: 2316137	Agree [/]
Signature: [illegible]	Read [/]
Name: SITI NORATHIRAH BINTI RAZALLY	Understand [/]
Matric Number: 2319788	Agree [/]
Signature: [illegible]	Read [/]
Name: NUR ALYA SAKINAH BINTI MOHD MOKHTAR	Understand [/]
Matric Number: 2319444	Agree [/]