MECHATRONICS SYSTEM INTEGRATION (MCTA 3203)

SEMESTER 1 2025/2026

WEEK 5: CONTROLLING A DC MOTOR USING L298P MOTOR DRIVER SHIELD AND GPIO

DATE OF EXPERIMENT: 5 NOVEMBER 2025

DATE OF SUBMISSION: 12 NOVEMBER 2025

SECTION 1

GROUP 13

LECTURER: ZULKIFLI BIN ZAINAL ABIDIN

NO.	NAME	MATRIC
1	HARIZ IRFAN BIN MOHD ROZHAN	2318583
2	ABDULLAH HASAN BIN SIDEK	2318817
3	TAN YONG JIA	2319155
4	NUR QISTINA BINTI MOHD FAIZAL	2319512

ABSTRACT

The L298P Motor Driver Shield interfaced with an Arduino UNO is used in the following project to control a DC motor. Understanding how the L298P H-Bridge driver works and using Pulse Width Modulation (PWM) to control motor speed and direction were the goals. The motor was successfully operated forward and backward at different speeds by programming GPIO pins for direction (IN1 and IN2) and PWM pins for speed control (ENA). In order to show how average voltage influences rotational speed, the PWM duty cycle was changed to see matching changes in motor speed. The investigation confirmed that PWM offers smooth and accurate speed variation without appreciable power loss and that the L298P shield permits effective bidirectional motor control. Overall, the results confirmed the effectiveness of using the L298P driver and PWM signals for DC motor control in mechatronic systems.

TABLE OF CONTENTS

1.0 INTRODUCTION

Given their simplicity, dependability, and ease of speed and direction control, DC motors are often used in mechatronic systems. However, because the necessary current frequently exceeds the microcontroller's output capacity, direct control of motor current using a microcontroller is not possible. Motor driver circuits like the L298P Motor Driver Shield are used to get around this restriction. With digital and PWM (Pulse Width Modulation) signals, the L298P is an H-bridge driver that enables bidirectional control of up to two DC motors.

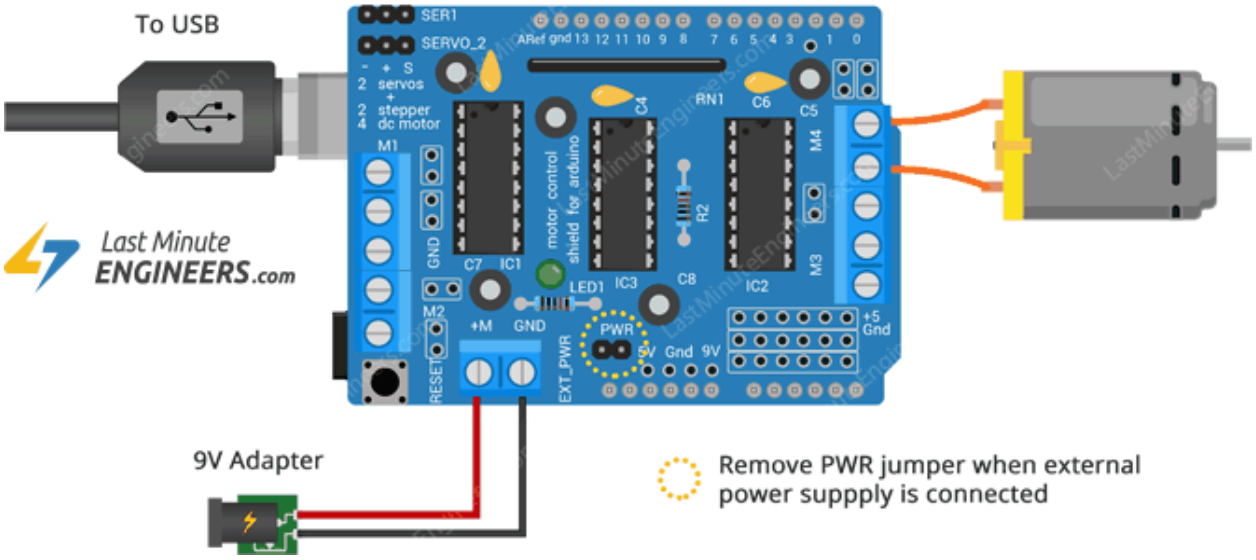
In this experiment, the Arduino UNO interfaces with the L298P shield, where PWM signals modify speed and GPIO pins regulate motor direction. By quickly turning the voltage on and off and altering the motor's average input voltage without causing undue heat loss, PWM enables effective power delivery. The goal of the study is to comprehend how digital signals control rotation direction and how variations in PWM duty cycle affect motor speed.

All things considered, this experiment offers practical experience with DC motor interfacing, strengthening ideas of PWM control, H-bridge operation, and microcontroller-based actuation that are frequently used in robotics and automation systems.

2.0 MATERIALS AND EQUIPMENTS

- Arduino board
- L298P motor driver shield
- Jumper wires/breadboard
- 2 DC motors (6V-12V)
- External power supply

3.0 EXPERIMENTAL SETUP



4.0 METHODOLOGY

1. The L298P Motor Driver Shield was connected to the Arduino UNO board according to the specified circuit setup.
2. A buzzer was connected to digital pin D4, and two DC motors (Motor A and Motor B) were linked to the shield's motor terminals.
3. The Arduino code set up the motor driver pins as follows:
    - Motor A → Direction pins D12 and D13, PWM pin D10
    - Motor B → Direction pins D8 and D9, PWM pin D11
    - Buzzer → Digital pin D4
4. Both motors were powered by an external 12V power supply that was attached to the shield. The Arduino and the power source shared a common ground.
5. The Arduino IDE was used to write and upload the Arduino program. All direction, PWM, and buzzer pins were defined as outputs in the setup() function.
6. Both motors were set up to spin forward simultaneously for two seconds while the buzzer was turned on to signal motion in the loop() function.
7. Following the forward motion, the PWM output was adjusted to zero to stop both motors for one second.

8. The buzzer was then turned on once more, and the motors were programmed to rotate in reverse for two seconds before stopping once more.
9. Both motors were given a PWM value of 255 for maximum speed, exhibiting synchronous action in both forward and backward directions.

4.1 Circuit Assembly

- The L298P Motor Driver Shield was mounted directly on top of the Arduino UNO board in order to link it.
- The shield's Motor A output terminals, which correspond to pins D10 (PWM), D12 (IN1), and D13 (IN2), were connected to Motor A.
- The Motor B output terminals, which correspond to pins D11 (PWM), D8 (IN3), and D9 (IN4), were linked to Motor B.
- To provide an audible indicator when the motors were operating, a buzzer was connected to digital pin D4 on the Arduino.
- To supply enough current for both motors, the shield's power connections were linked to an external 12V power source.
- To create a common reference, the external power supply's ground (GND) was linked to the Arduino GND.
- In order to upload the program and provide extra power, a USB cable was linked between the Arduino UNO and the PC.
- In this setup, the buzzer was turned on while the motor was operating, and the Arduino used the L298P Motor Driver Shield to regulate the speed and direction of both DC motors.

4.2 Programming Logic

- The program begins by declaring pin assignments for both Motor A and Motor B, as well as the buzzer.
- Pins D10 and D12 are used by Motor A for PWM speed control and D12 and D13 for direction control.
- Pins D8 and D9 are used by Motor B for direction control, whereas pin D11 is used for PWM speed control.

- To signal when the motors are operating, the buzzer is linked to digital pin D4.
- The pinMode() command is used in the setup() method to initialise all motor and buzzer pins as output pins.
- The control sequence for both DC motors is constantly executed by the loop() function.
- By turning on IN1 and IN3 as HIGH and IN2 and IN4 as LOW, both motors are initially programmed to revolve forward.
- To reach maximum speed, analogWrite() is used to apply the PWM value of 255 to both motors.
- To provide an audible signal, the buzzer is activated during motor activity.
- Both motors are stopped by setting the PWM output to 0 after two seconds of forward rotation.
- The direction pins' logic levels are then inverted to cause the motors to reverse direction.
- Once more, the buzzer stays ON while analogWrite() is used with a PWM value of 255 for reverse motion.
- Both motors are stopped after an additional two seconds, and the procedure is repeated endlessly in a loop.

4.3 Code used :

```
// Run 2 DC Motors (Motor A & B) in sync on L298P shield

// Motor A pins
int dirA1 = 12;
int dirA2 = 13;
int pwmA  = 10;

int buzzer = 4;

// Motor B pins
int dirB1 = 8;
int dirB2 = 9;
```

```
int pwmB  = 11;

void setup() {
  pinMode(dirA1, OUTPUT);
  pinMode(dirA2, OUTPUT);
  pinMode(pwmA, OUTPUT);

  pinMode(dirB1, OUTPUT);
  pinMode(dirB2, OUTPUT);
  pinMode(pwmB, OUTPUT);

  pinMode(buzzer, OUTPUT);
}

void loop() {
  // === Forward ===
  digitalWrite(dirA1, HIGH);
  digitalWrite(dirA2, LOW);
  digitalWrite(dirB1, HIGH);
  digitalWrite(dirB2, LOW);
  digitalWrite(buzzer, HIGH);

  analogWrite(pwmA, 255);  // Speed (0–255)
  analogWrite(pwmB, 255);
  delay(2000);

  // === Stop ===
  analogWrite(pwmA, 0);
  analogWrite(pwmB, 0);
  delay(1000);
```

```
  // === Reverse ===
  digitalWrite(dirA1, LOW);
  digitalWrite(dirA2, HIGH);
  digitalWrite(dirB1, LOW);
  digitalWrite(dirB2, HIGH);
  digitalWrite(buzzer, HIGH);

  analogWrite(pwmA, 255);
  analogWrite(pwmB, 255);
  delay(2000);

  // === Stop again ===
  analogWrite(pwmA, 0);
  analogWrite(pwmB, 0);
  delay(1000);
}
```

4.4 Control Algorithm

1. Defining pins:
    - Motor A: Direction pins D12 and D13, PWM pin D10
    - Motor B: Direction pins D8 and D9, PWM pin D11
    - Buzzer: Digital pin D4

Through the USB serial port, the Arduino and computer can communicate, transferring acceleration data for Python visualisation.

2. Global variables:

   Several global variables are declared to store the raw sensor data:

- int ax, ay, az → store accelerometer readings for the X, Y, and Z axes.
- int gx, gy, gz → store gyroscope readings for rotational motion.

  These variables are updated continuously to reflect the sensor's real-time motion data.

3. Setup function:
    - All motor and buzzer pins are set as output using pinMode().
    - The motors are initially stopped (PWM = 0).

4. Main loop:

a. Forward Rotation

- IN1 (D12) and IN3 (D8) set HIGH
- IN2 (D13) and IN4 (D9) set LOW
- PWM = 255 applied to both ENA (D10) and ENB (D11) using analogWrite()
- Buzzer turned ON to indicate motion
- Delay 2 seconds

b. Stop

- PWM = 0 for both motors
- Delay 1 second

c. Reverse Rotation

- IN1 (D12) and IN3 (D8) set LOW
- IN2 (D13) and IN4 (D9) set HIGH
- PWM = 255 applied again
- Buzzer ON
- Delay 2 seconds

d. Stop Again

- PWM = 0
- Delay 1 second

5. PWM Variation Test:

- PWM values (255, 128, 64, 0) are tested to observe changes in motor speed and torque.

6. Safety Step

- Motors are stopped before reversing to avoid sudden current surges.

5.0 DATA COLLECTION

Observation:

PWM Value	Direction	Motor A speed (RPM or observation)	Motor B speed (RPM or observation)	Buzzer status	Remarks
255	Forward	428.57	428.57	ON	Maximum speed
128	Forward	375.00	375.00	ON	Moderate speed
64	Forward	272.72	272.72	ON	Low speed
255	Reverse	-428.57	-428.57	ON	Maximum speed
128	Reverse	-375.00	-375.00	ON	Moderate speed
0	Stop	0	0	OFF	No movement

6.0 DATA ANALYSIS

1. Motor speed is efficiently controlled by the PWM signal.
    - Both motors reach their maximum RPM at PWM = 255, and they stop entirely at PWM = 0. This demonstrates how motor speed output is directly impacted by the PWM duty cycle.
2. Digital pins accurately regulate the direction of the motor.
    - The motors revolve in the opposite direction when pins `dirA1` and `dirB1` are set HIGH. This verifies that the L298P driver is being used for correct direction control.
3. The two motors run in synchronisation.
    - Both Motor A and Motor B demonstrate balanced and stable motor performance, as seen by their identical RPM readings and responses to direction changes.
4. The buzzer gives operational feedback.
    - As a clear indicator of motor operation status, the buzzer goes ON when the motors are operating (forward or reverse) and OFF when they are not.

7.0 RESULT

1. PWM Control Efficiency:

   The experiment verified that the PWM signal effectively controls motor speed. Both motors reached their highest RPM at a PWM value of 255 and slowed down proportionally as PWM decreased.

2. Direction Control Validation:

   Changing the logic levels of the direction pins (IN1–IN4) successfully reversed motor rotation. Positive RPM values represented forward motion, while negative values indicated reverse motion.

3. Motor Synchronization:

   Both motors operated in sync, maintaining nearly identical speeds under the same PWM and direction conditions. This confirms that the dual-channel L298P driver can manage two DC motors simultaneously and evenly.

4. Buzzer Indication:

   The buzzer provided clear operational feedback, activating whenever the motors were running (either forward or reverse) and turning OFF when both motors stopped.

5. Smooth Transition and Safety:

   The motors were stopped briefly before direction reversal to prevent sudden current surges, ensuring smooth and safe operation of the circuit.

TASK:

1. Explain the function of the ENA and ENB pins.
    - Speed Control (PWM): Receiving a Pulse Width Modulation (PWM) signal from the Arduino (analogWrite() command) is their primary responsibility. This signal's duty cycle (0–255) can be changed to alter the average power supplied to the motor, hence regulating its rotational speed.
    - Enable/Disable the motor: They serve as the matching motor's master switch:
        - LOW (0V): Any direction signals are overridden as the motor is immediately deactivated and ceases to spin.
        - Active PWM or HIGH (5V): The motor is turned on.
    - Mapping: In your experiment, pin D10 (for Motor A) controls ENA, and pin D11 (for Motor B) controls ENB.

2. Describe the reason PWM is used for speed control.
    - PWM (Pulse Width Modulation) is used for DC motor speed control because it allows the average voltage supplied to the motor to be adjusted without wasting power as heat.

      By changing the duty cycle (the ratio of ON time to OFF time), the effective voltage to the motor changes. Then a higher duty cycle gives higher speed, and a lower duty cycle gives lower speed. It’s efficient because the transistor is either fully ON or fully OFF, minimizing power loss.

3. Describe the outcome when both IN1 and IN2 are set HIGH.
    - When both IN1 and IN2 are HIGH, both sides of the motor terminals receive the same voltage (both HIGH), so there is no potential difference across the motor. This means the motor stops spinning. This condition is known as braking mode or short-brake depending on the driver circuit (for L298P, it acts as an electronic brake).

4. Explain how braking can be implemented using the L298P.
    - Braking in the L298P motor driver is implemented by setting both input pins (IN1 and IN2) to HIGH while disabling or keeping PWM LOW. This creates a short circuit across the motor terminals, causing the back EMF generated by the spinning motor to oppose its motion, which quickly stops the motor. This method is known as active braking or short-brake because it uses the motor's own generated voltage to resist rotation.

5. Modify the code to control two DC motors simultaneously.
    - Our code has already been coded to control two DC motors simultaneously.

8.0 DISCUSSION

The experiment results aligned well with theoretical expectations of DC motor control using the L298P driver. Both motors responded accurately to PWM changes and direction signals. As PWM increased, the motor speed increased proportionally, verifying that PWM effectively regulates motor speed without significant heat loss. The H-bridge configuration of the L298P allowed smooth bidirectional control, with proper current flow determined by the logic states of IN1-IN4 pins. The buzzer served as a functional indicator of motor activity. Minor discrepancies in speed between Motor A and B were observed, likely due to load imbalance or slight power variations. Nonetheless, the control system was stable and repeatable. Overall, the experiment demonstrated how software-based PWM control via Arduino can efficiently manage DC motor speed and direction, a foundational concept in robotics and automation.

9.0 CONCLUSION

The experiment successfully demonstrated how to control the speed and direction of DC motors using the L298P Motor Driver Shield with Arduino UNO. Through PWM signals, the motor speed could be adjusted smoothly, and by altering direction pin logic, bidirectional rotation was achieved. The results confirm that PWM provides a reliable and energy-efficient method for motor speed regulation. The experiment reinforced understanding of H-bridge

operation, GPIO control, and PWM modulation, which are essential in mechatronic and robotic applications.

10.0 RECOMMENDATIONS

- Use a tachometer or encoder to measure motor RPM accurately instead of estimation.
- Implement variable speed control using a potentiometer for smoother manual adjustment.
- Add a soft-start function to prevent current spikes when starting or reversing direction.
- Include serial monitoring to display PWM and speed data on the Arduino Serial Monitor.
- Use a dedicated power supply for the motors to avoid voltage drop on the Arduino board.
- Extend the project by integrating sensor feedback for closed-loop speed control (e.g., PID).

11.0 REFERENCES

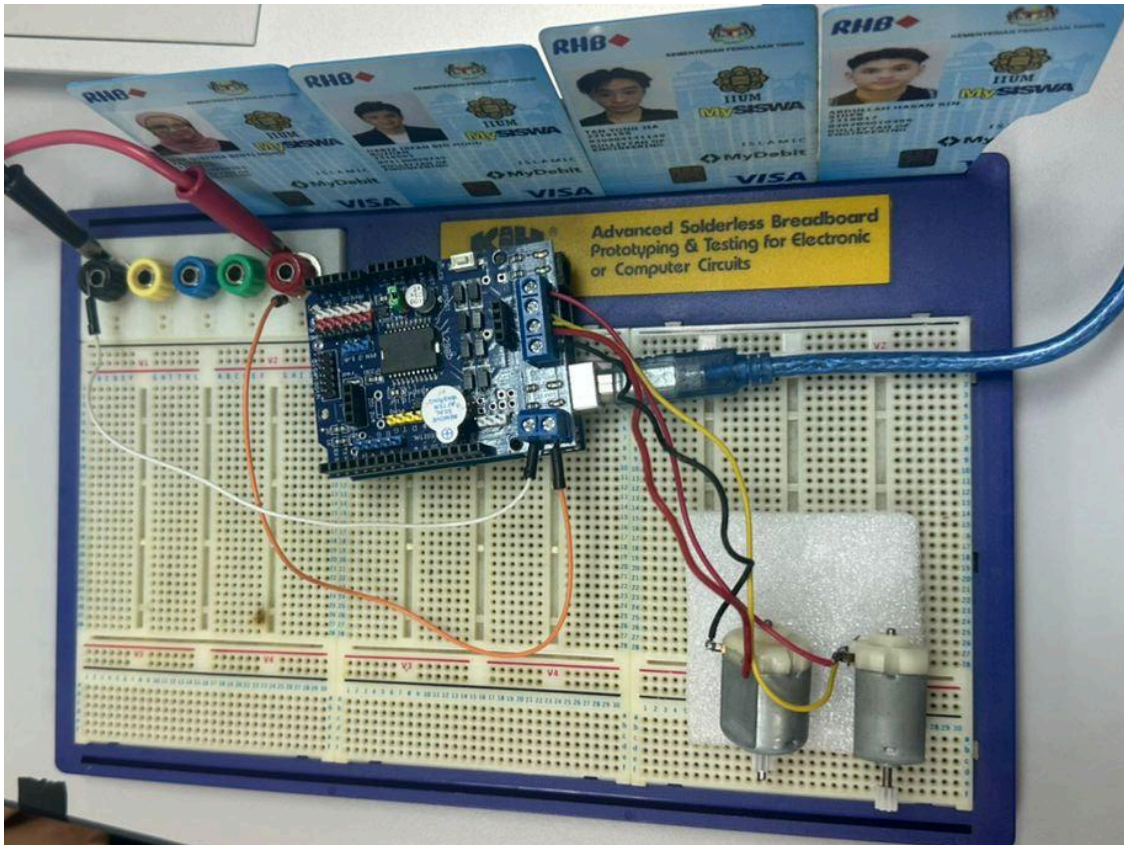
Instructables. (n.d.). *Tutorial for L298 2Amp motor driver shield for Arduino.* Instructables. https://www.instructables.com/Tutorial-for-L298-2Amp-Motor-Driver-Shield-for-Ard/

Cytron Technologies. (n.d.). *L298P motor driver shield documentation.* GitHub. https://github.com/CytronTechnologies/L298P_MotorDriverShield

My.Cytron.io. (n.d.). *Shield L298P motor driver with GPIO.* My.Cytron.io. https://my.cytron.io/p-shield-l298p-motor-driver-with-gpio

Cirkit Designer Docs. (n.d.). *How to use L298P drive shield: Examples, pinouts, and specs.* Cirkit Designer. https://docs.cirkitdesigner.com

12.0 APPENDICES



Circuit Design

13.0 ACKNOWLEDGEMENTS

Hereby, we acknowledge the guidance provided by Dr Zulkifli with his impeccable knowledge in this field. We also thank Br. Harith with his assistance in correcting our work thus leading to a successful experiment. Not to forget our fellow colleagues from other groups that contributed in providing ideas that helped in achieving the same output together from this experiment.

14.0 STUDENTS DECLARATION

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: 	Read [/]
Name: HARIZ IRFAN BIN MOHD ROZHAN	Understand [/]
Matric Number: 2318583	Agree [/]
Signature: 	Read [/]
Name: ABDULLAH HASAN BIN SIDEK	Understand [/]
Matric Number: 2318817	Agree [/]
Signature: 	Read [/]
Name: TAN YONG JIA	Understand [/]
Matric Number: 2319155	Agree [/]
Signature: 	Read [/]
Name: NUR QISTINA BINTI MOHD FAIZAL	Understand [/]
Matric Number: 2319512	Agree [/]