Motor Control with L298n

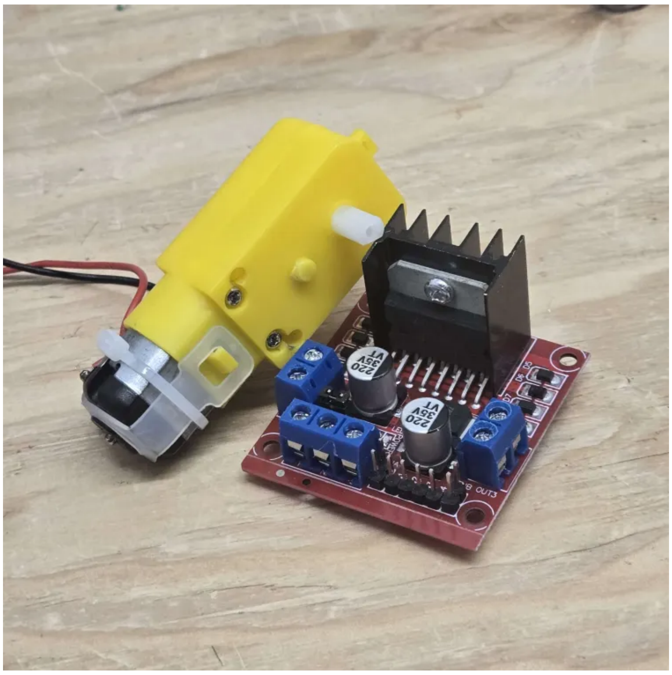

Figure 1: DC Motor and l298n Controller

In this workshop, we will explore how to control motors using the L298 motor driver and an Arduino. The L298 is a powerful driver that allows you to control the speed and direction of up to two motors independently.

Wiring the L298 to Arduino

Before beginning, ensure you have the required components: L298 motor driver, Arduino, motors, and connecting wires.

Power Supply

- **VCC** → Connect to the `5V` pin on your Arduino to power the motor driver.
- **GND** → Connect to one of the `GND` pins on your Arduino to complete the circuit.

Signal Pins for Motor A

- **ENA** → Connect to a digital pin, for example, 9 (this pin enables the motor's speed control).
- **IN1** → Connect to a digital pin, for example, 7 (this controls the motor's direction).
- **IN2** → Connect to a digital pin, for example, 6 (this also controls the motor's direction).

Motor A

- Red wire → Connect to the `OUT1` terminal on the L298.
- Black wire → Connect to the `OUT2` terminal on the L298.

Programming

Step 1: Setup Pins

Set all relevant pins to output mode using the `pinMode` function at the beginning of your program in `setup()`.

```
#define ENA 9
#define IN1 7
#define IN2 6

void setup() {
    // Set pin modes for motor control
    pinMode(ENA, OUTPUT);
    pinMode(IN1, OUTPUT);
    pinMode(IN2, OUTPUT);
}
```

Step 2: Control Motor Direction

Below is a motor behavior table based on L298 inputs. Use it to determine the necessary signal combinations for controlling motor direction and speed.

Direction	IN1	IN2
Forward	`HIGH`	`LOW`
Backward	`LOW`	`HIGH`
Stop	`LOW`	`LOW`

Example of sending signals to control motor A:

```
// "Enable A" is activated to power motor A at 50% speed (255 is full speed)
analogWrite(ENA, 127);

// Rotate motor A forward
digitalWrite(IN1, HIGH);
digitalWrite(IN2, LOW);

// Wait 1 second to observe motor action
delay(1000);

// Stop motor A
digitalWrite(IN1, LOW);
digitalWrite(IN2, LOW);
```

This code could be in the `setup()` function to test the motor’s behavior.

Step 3: Implement Control Functions

For easier control, create functions to handle motor operations:

```
void motor_forward() {
    // Activate motor A and make it move forward at 50% speed
    analogWrite(ENA, 127);
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
}

void motor_backward() {
    // Activate motor A and make it move backward at 50% speed
    analogWrite(ENA, 127);
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, HIGH);
}
```

These functions simplify the process of controlling the motor by encapsulating the logic into reusable functions. Adjust pin values according to your setup.

```
void setup() {
    // Set pin modes for motor control
    pinMode(ENA, OUTPUT);
    pinMode(IN1, OUTPUT);
    pinMode(IN2, OUTPUT);

    // Use your functions to control the motor
    motor_forward();
    delay(500);
    motor_backward();
}
```

By following these steps, you can control the L298 motor driver effectively for any project involving motor control.