****

**Project Name:** Line Following Robot

**Submitted By:**

YOUSRA ZAHEER(1790-2023)

MARIAM ZAHEER(1788-2023)

SYEDA MAIRA ABID GARDAZI(1940-2023)

**Program:** BSCS-03

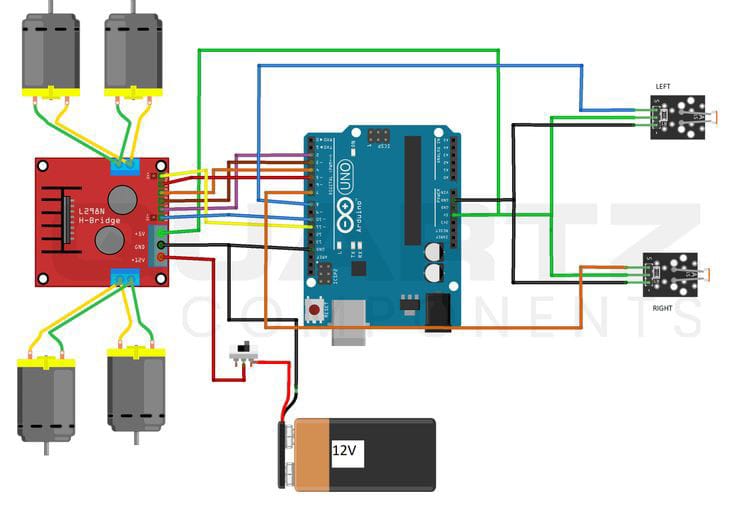
**Submitted To:** Engr. Talha Riaz

**Institute:** Hamdard University Islamabad

**Line Following Robot**

**Components:**

* Car chassis
* 4 TT Gear Motors
* 4 Wheels
* Connectors & screws
* Auduino uno circuit
* 2 IR sensors
* L298 Motor Driver
* Power supply (batteries)
* Jumper wires
* Solderable wires
* Soldering Iron



**Introduction**

### This project relates to ****Digital Logic Design (DLD)**** as it involves the practical application of digital logic principles to control a line-following robot. The robot uses ****IR sensors**** to detect the surface (black or white), which generate binary signals (0 or 1) based on the presence or absence of reflected light. These signals are processed by the ****Arduino Uno****, which acts as the control unit implementing logical decisions (using conditional statements, equivalent to logic gates). The ****L298 motor driver**** serves as an interface between the Arduino and the motors, converting digital control signals into appropriate motor movements. The project demonstrates the implementation of digital circuits, logical decision-making, and control systems, fundamental concepts in DLD.

### ****Arduino Uno Board****

The **Arduino Uno** is a microcontroller development board based on the ATmega328P microcontroller. It is widely used in robotics, automation, and electronic projects due to its open-source nature. Key features of the Arduino Uno include:

1. **Digital I/O Pins:** 14 digital pins (6 with PWM capability) to read and write digital signals.
2. **Analog Input Pins:** 6 analog pins to read sensor data as varying voltage levels.
3. **Power Supply:** Operates at 5V and can be powered via USB, an external 5V, or a 7-12V DC input.
4. **Communication:** Serial communication via USB for programming and debugging.
5. **Programming Language:** Uses a simplified version of C/C++ and can be programmed using the Arduino IDE.

### ****Implementation in the Line-Following Robot****

In this project, the Arduino Uno serves as the **central processing unit** (CPU) to control the robot’s operation. Its role includes:

#### ****1. Sensor Data Processing****

* The two **IR sensors** are connected to the analog input pins (A0 and A1) of the Arduino.
* The sensors detect surface colors and send **binary signals** (0 or 1) to the Arduino, where the signals are processed using conditional logic.

#### ****2. Decision-Making****

* Based on the sensor inputs, the Arduino decides the robot’s movement:
  + **Forward:** Both sensors detect white (binary 0, 0).
  + **Turn Right:** Right sensor detects black and left sensor detects white (binary 1, 0).
  + **Turn Left:** Right sensor detects white and left sensor detects black (binary 0, 1).
  + **Stop:** Both sensors detect black (binary 1, 1).

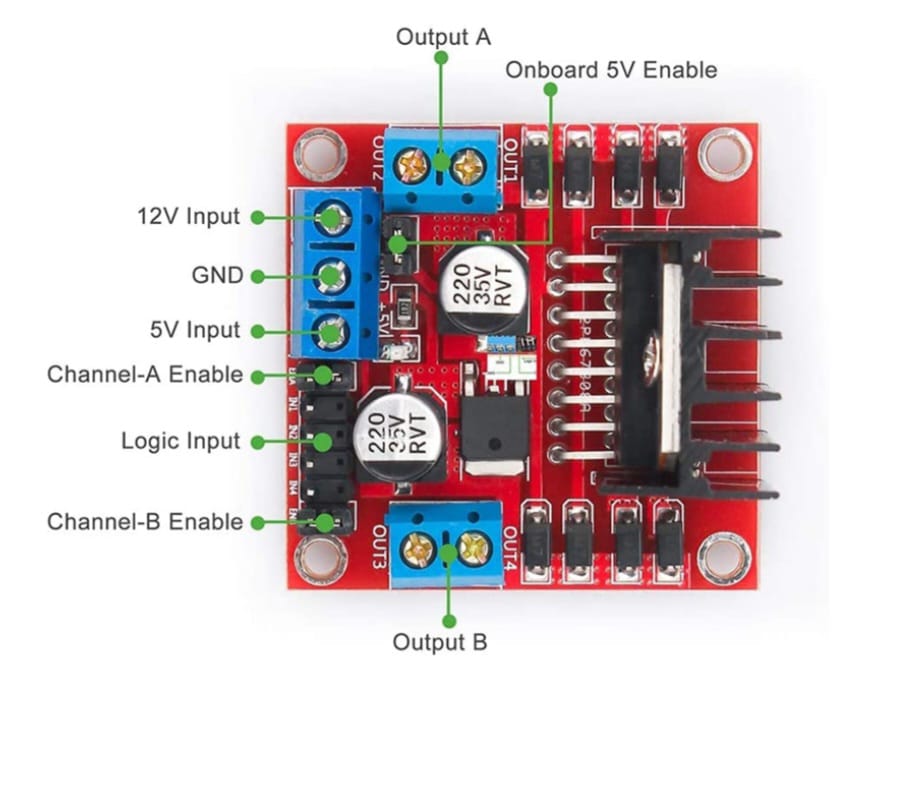
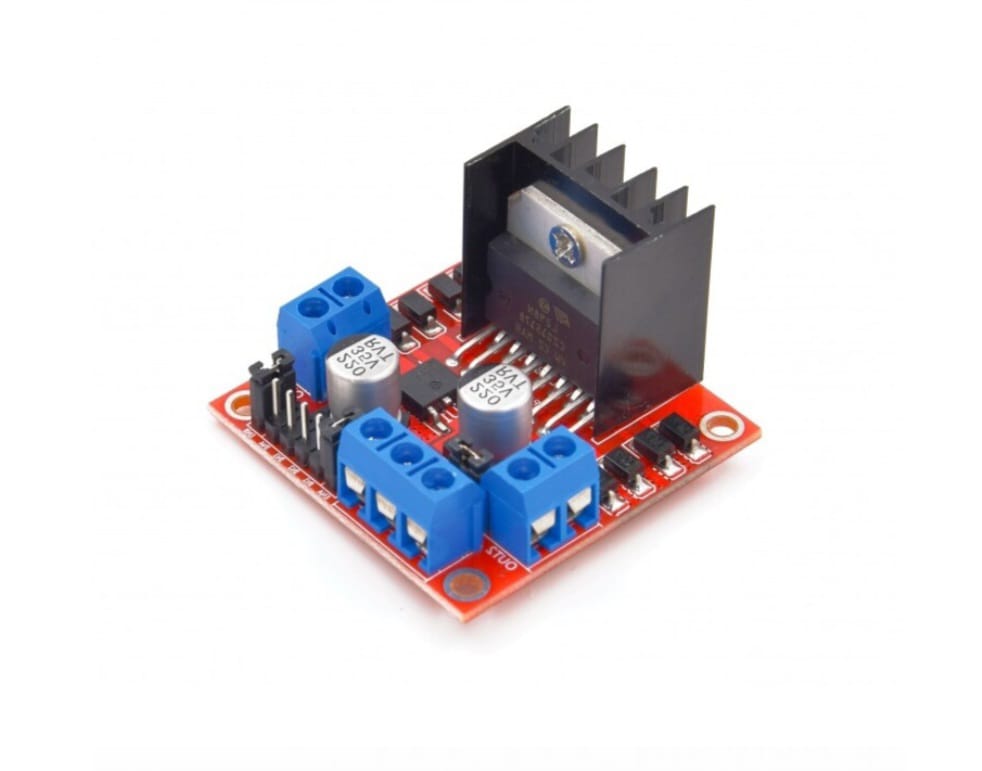
#### ****3. Motor Control****

* The Arduino sends **control signals** to the **L298 motor driver** via digital output pins.
* These signals determine:
  + **Direction:** By controlling the IN1, IN2, IN3, and IN4 pins of the motor driver.
  + **Speed:** Using PWM signals sent to the ENA and ENB pins.

### ****L298 Motor Driver Module****

The **L298 Motor Driver Module** is a dual H-bridge driver that allows the control of two DC motors in both directions. It is widely used for motor control due to its ability to handle higher current levels. Key features of the L298 motor driver include:

1. **Dual H-Bridge Configuration:** Enables control of two motors independently, including direction and speed.
2. **Voltage Range:** Can handle motor supply voltages between 5V to 35V.
3. **Current Handling:** Supports up to 2A per motor channel (continuous).
4. **Control Pins:**
   * **Input Pins (IN1, IN2, IN3, IN4):** Control the direction of the motors.
   * **Enable Pins (ENA, ENB):** Enable or disable motor operation and control speed via PWM signals.
5. **Built-in Voltage Regulator:** Provides a 5V output to power external components like the Arduino board.
6. **Heat Sink:** Enhances heat dissipation for handling high currents.



### ****Implementation in the Line-Following Robot Project****

In this project, the **L298 motor driver module** is used to control the two pairs of TT gear motors for the robot's movement. It acts as an intermediary between the Arduino Uno and the motors.

#### ****1. Motor Control****

* **Direction Control:**
  + The Arduino sends digital HIGH/LOW signals to the input pins (IN1, IN2 for the right motor and IN3, IN4 for the left motor).
  + The signals determine the polarity of the motor terminals, controlling the direction of motor rotation (forward or backward).

| **IN1** | **IN2** | **Motor Direction** |
| --- | --- | --- |
| HIGH | LOW | Forward |
| LOW | HIGH | Reverse |
| LOW | LOW | Stop |

* **Speed Control:**
  + The Arduino sends a **PWM signal** to the enable pins (ENA for the right motor and ENB for the left motor).
  + The duty cycle of the PWM signal determines the motor speed.

#### ****2. Power Distribution****

* **Motor Power:** The L298 is powered by a 12V battery pack, providing the necessary voltage and current for motor operation.
* **5V Output:** The built-in voltage regulator provides a stable 5V output, which can optionally power the Arduino board.

#### ****3. Connections****

* **To Arduino:** The Arduino is connected to the control and enable pins to send direction and speed control signals.
* **To Motors:** The output terminals (OUT1, OUT2 for the right motor and OUT3, OUT4 for the left motor) are connected to the motors.

### ****Comprehensive Jumper Wire Connections****

### ****1. IR Sensors to Arduino****

The IR sensors detect surface colors (black or white) and send signals to the Arduino.

| **IR Sensor Pin** | **Connection** | **Arduino Pin** | **Details** |
| --- | --- | --- | --- |
| **VCC (Right)** | Connect to Arduino 5V | - | Provides power to the right IR sensor. |
| **GND (Right)** | Connect to Arduino GND | - | Connects the ground of the right IR sensor to the Arduino's ground. |
| **OUT (Right)** | Connect to Arduino A0 | A0 | Sends digital signal based on surface detection (configured as R\_S). |
| **VCC (Left)** | Connect to Arduino 5V | - | Provides power to the left IR sensor. |
| **GND (Left)** | Connect to Arduino GND | - | Connects the ground of the left IR sensor to the Arduino's ground. |
| **OUT (Left)** | Connect to Arduino A1 | A1 | Sends digital signal based on surface detection (configured as L\_S). |

### ****2. Arduino to L298 Motor Driver****

The Arduino sends control signals to the L298 motor driver to manage motor speed and direction.

| **Arduino Pin** | **Connection** | **L298 Pin** | **Details** |
| --- | --- | --- | --- |
| **Pin 9** | Connect to L298 IN1 | IN1 | Controls the forward/reverse direction for the right motor. |
| **Pin 8** | Connect to L298 IN2 | IN2 | Controls the forward/reverse direction for the right motor. |
| **Pin 7** | Connect to L298 IN3 | IN3 | Controls the forward/reverse direction for the left motor. |
| **Pin 6** | Connect to L298 IN4 | IN4 | Controls the forward/reverse direction for the left motor. |
| **Pin 10** | Connect to L298 EN A | EN A | Enables speed control for the right motor using a PWM signal. |
| **Pin 5** | Connect to L298 EN B | EN B | Enables speed control for the left motor using a PWM signal. |

**Code Implementation in Arduino Uno**

The code is uploaded to the Arduino Uno using the **Arduino IDE** via USB. Once the code is successfully embedded, the Arduino reads the signals from the **IR sensors**, processes the information, and controls the direction and speed of the motors, enabling the robot to autonomously follow a line.

#define enA 10//Enable1 L298 Pin enA

#define in1 9 //Motor1 L298 Pin in1

#define in2 8 //Motor1 L298 Pin in1

#define in3 7 //Motor2 L298 Pin in1

#define in4 6 //Motor2 L298 Pin in1

#define enB 5 //Enable2 L298 Pin enB

#define R\_S A0 //ir sensor Right

#define L\_S A1 //ir sensor Left

void setup(){ // put your setup code here, to run once

pinMode(R\_S, INPUT); // declare if sensor as input

pinMode(L\_S, INPUT); // declare ir sensor as input

pinMode(enA, OUTPUT); // declare as output for L298 Pin enA

pinMode(in1, OUTPUT); // declare as output for L298 Pin in1

pinMode(in2, OUTPUT); // declare as output for L298 Pin in2

pinMode(in3, OUTPUT); // declare as output for L298 Pin in3

pinMode(in4, OUTPUT); // declare as output for L298 Pin in4

pinMode(enB, OUTPUT); // declare as output for L298 Pin enB

analogWrite(enA, 150); // Write The Duty Cycle 0 to 255 Enable Pin A for Motor1 Speed

analogWrite(enB, 150); // Write The Duty Cycle 0 to 255 Enable Pin B for Motor2 Speed

delay(1000);

}

void loop(){

if((digitalRead(R\_S) == 0)&&(digitalRead(L\_S) == 0)){forword();} //if Right Sensor and Left Sensor are at White color then it will call forword function

if((digitalRead(R\_S) == 1)&&(digitalRead(L\_S) == 0)){turnRight();} //if Right Sensor is Black and Left Sensor is White then it will call turn Right function

if((digitalRead(R\_S) == 0)&&(digitalRead(L\_S) == 1)){turnLeft();} //if Right Sensor is White and Left Sensor is Black then it will call turn Left function

if((digitalRead(R\_S) == 1)&&(digitalRead(L\_S) == 1)){Stop();} //if Right Sensor and Left Sensor are at Black color then it will call Stop function

}

void forword(){ //forword

digitalWrite(in1, HIGH); //Right Motor forword Pin

digitalWrite(in2, LOW); //Right Motor backword Pin

digitalWrite(in3, LOW); //Left Motor backword Pin

digitalWrite(in4, HIGH); //Left Motor forword Pin

}

void turnRight(){ //turnRight

digitalWrite(in1, LOW); //Right Motor forword Pin

digitalWrite(in2, HIGH); //Right Motor backword Pin

digitalWrite(in3, LOW); //Left Motor backword Pin

digitalWrite(in4, HIGH); //Left Motor forword Pin

}

void turnLeft(){ //turnLeft

digitalWrite(in1, HIGH); //Right Motor forword Pin

digitalWrite(in2, LOW); //Right Motor backword Pin

digitalWrite(in3, HIGH); //Left Motor backword Pin

digitalWrite(in4, LOW); //Left Motor forword Pin

}

void Stop(){ //stop

digitalWrite(in1, LOW); //Right Motor forword Pin

digitalWrite(in2, LOW); //Right Motor backword Pin

digitalWrite(in3, LOW); //Left Motor backword Pin

digitalWrite(in4, LOW); //Left Motor forword Pin

}