****

***Project Report***

**Course Code:** CSE 426

**Course Title:** Principles of Robotics

**Submitted To**

Md. Hafizul Imran(MHI)

Department of CSE

**Daffodil International University**

**Submitted By**

**Anup Chandra Mondal Susmita Sarker Chaity Md Foyaz Ahmed**

ID: 193-15-13424 ID: 193-15-13442 ID: 193-15-13538

Section:- A(54 Batch)

Department of CSE

**Daffodil International University**

**Line Following Robot**

# **1 Objective**

The sole purpose of this project is that:

1. The robot must be capable of following a line.
2. It should be capable of taking various degrees of turns.

# **2 Introduction**

The line follower is a self-operating robot that detects and follows a line that is drawn on the floor. The path consists of a black line on a white surface (or it may be reverse of that). The control system used must sense a line and maneuver the robot to stay on course.

# **3 Components**

## 3.1 Modules:

1. 5 Array IR Sensor Modules
2. L298N Motor Driver
3. Arduino UNO

## 3.2 Lithium Ion Battery

1. 3 cells of 31865 Li-ion battery

## 3.3 Others

1. Jumper Wires
2. Male Headers / Connectors
3. LEDs
4. IR LEDs
5. Photo Diodes

# **4 Project Modules**

## 4.1 Arduino UNO

The Arduino UNO is a widely used open-source microcontroller board based on the [Microchip](https://en.wikipedia.org/wiki/Microchip_Technology) [ATmega328P](https://en.wikipedia.org/wiki/ATmega328P) microcontroller and developed by [Arduino.cc.](https://en.wikipedia.org/wiki/Arduino) The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board features 14 Digital pins and 6 Analog pins. It is programmable with the [Arduino IDE](https://en.wikipedia.org/wiki/Arduino#Software) (Integrated Development Environment) via a type B USB cable. It can be powered by a USB cable or by an external 9-volt battery, though it accepts voltages between 7 and 20 volts.



**Figure 4.1: Arduino UNO**

## 4.2 IR Sensors

An IR sensor is a device which detects IR radiation falling on it. There are numerous types of IR sensors that are built and can be built depending on the application. An IR sensor is basically a device which consists of a pair of an IR LED and a photodiode which are collectively called a photo-coupler or an opto-coupler. The IR LED emits IR radiation, reception and/or intensity of reception of which by the photodiode dictates the output of the sensor. If the object is reflective,

(White or some other light color), then most of the radiation will get reflected by it, and will get incident on the photodiode. For further understanding, please refer to the left part of the illustration below. If the object is non-reflective, (Black or some other dark color), then most of the radiation will get absorbed by it, and will not become incident on the photodiode. It is similar to there being no surface (object) at all, for the sensor, as in both the cases, it does not receive any radiation.

### **4.2.1 Working Mechanism**

An IR sensor is basically a device which consists of a pair of an IR LED and a photodiode which are collectively called a photo-coupler or an opto-coupler. The IR LED emits IR radiation, reception and/or intensity of reception of which by the photodiode dictates the output of the sensor. Now, there are so many ways by which the radiation may or may not be able to reach the photodiode.

### **4.2.2 Use in Line Follower Robots**

IR sensors are the main triggers of the whole line following robot’s action mechanism. IR sensors are the ones which detect the color of the surface underneath it and send a signal to the microcontroller or the main circuit which then takes decisions according to the algorithm set by the creator of the bot. The sensors used in them are based on reflective/non-reflective indirect incidence. The IR LED emits IR radiation, which in normal state gets reflected back to the module from the white surface around the black line, which gets incident on the photodiode. But, as soon as the IR radiation falls on the black line, the IR radiation gets absorbed completely by the black color, and hence there is no reflection of the IR radiation going back to the sensor module.

### **4.2.3 IR LED**

An IR LED is a type of LED which emits light in the frequency range of Infra-Red, hence the name ‘IR’ LED.

### **4.2.4 Photodiode**

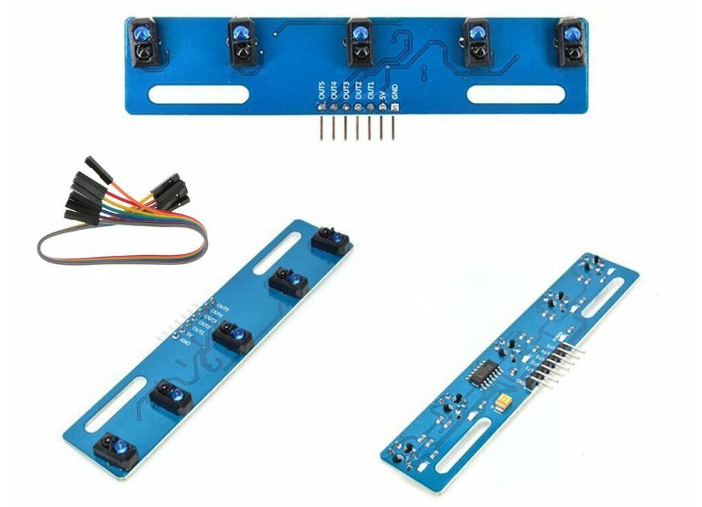
A photodiode is a type of diode which detects light. We can think of it as having a very high resistance when no light is falling on it. As we increase the intensity of light incident on it, the current through it gradually increases too. So, by increasing the incident light on a photodiode, we convert it into a normal low value resistor, which conducts current. We should note here that a photodiode looks exactly like an LED, sometimes, with a dark blue or black film on the outer casing.

### **4.2.5 LM358**

So basically, we use it to compare two voltages, one is fixed and the other varies with an environmental parameter. If the parameter-controlled voltage is higher than the fixed the voltage, then the IC should give one output, and if it is lower than the fixed voltage, then it should give another output. So, we see that the IC gives only two types of outputs, which we design to be 5 Volts and 0 Volts. This makes our sensor digital.

### **4.2.6 How does it work?**

If the IR LED emissions become incident on the photodiode, the photodiode’s resistance comes down to a finite value. The drop across the 10K series resistor is what we use as the input, which is compared with the threshold. The point to be noted here is that more the incident radiation on the photodiode, less will be the drop across it, and hence more will be the drop across the series resistor. If the voltage developed across the resistor is greater than the threshold set by us, the output of the IC will be high, else it will be low. Hence, if our reflected radiation is never strong enough to be greater than the threshold and we have a constant low as output, we can reduce the threshold voltage by turning the “minus shaped” slit in the variable resistance towards its terminal where we connected Gnd. In case our threshold is very low and the output is always high in spite of no radiation or if it is just too sensitive, then you can increase the threshold by turning the slit the other way. When the emissions are absorbed by a black surface, the resistance of the photodiode becomes very high due to no incidence of IR emissions on it, and the output remains low. I like to use an LED to indicate the output, even if I have the output going to the main circuit, but it is totally up to you when you make it.



**Figure 4.2: IR Sensor Module**

## 4.3 L298N H-Bridge

### **4.3.1 Usage**

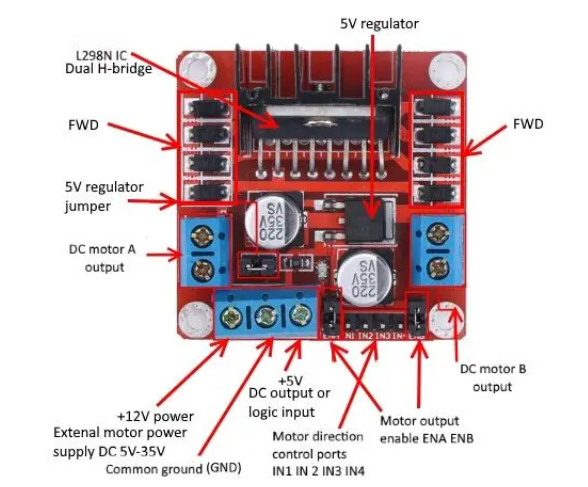
H-Bridge's are typically used in controlling motors speed and direction, but can be used for other projects such as driving the brightness of certain lighting projects such as high-powered LED arrays.

### **4.3.2 Working of L298N**

An H-Bridge is a circuit that can drive a current in either polarity and be controlled by Pulse Width Modulation (PWM).

Pulse Width Modulation is a means in controlling the duration of an electronic pulse. In motors try to imagine the brush as a water wheel and electrons as a the flowing droplets of water. The voltage would be the water flowing over the wheel at a constant rate, the more water flowing the higher the voltage. Motors are rated at certain voltages and can be damaged if the voltage is applied to heavily or if it is dropped quickly to slow the motor down. Thus PWM. Take the water wheel analogy and think of the water hitting it in pulses but at a constant flow. The longer the pulses the faster the wheel will turn, the shorter the pulses, the slower the water wheel will turn.

Motors will last much longer and be more reliable if controlled through PWM.

****

**Figure 4.3: L298N Pinout Diagram**

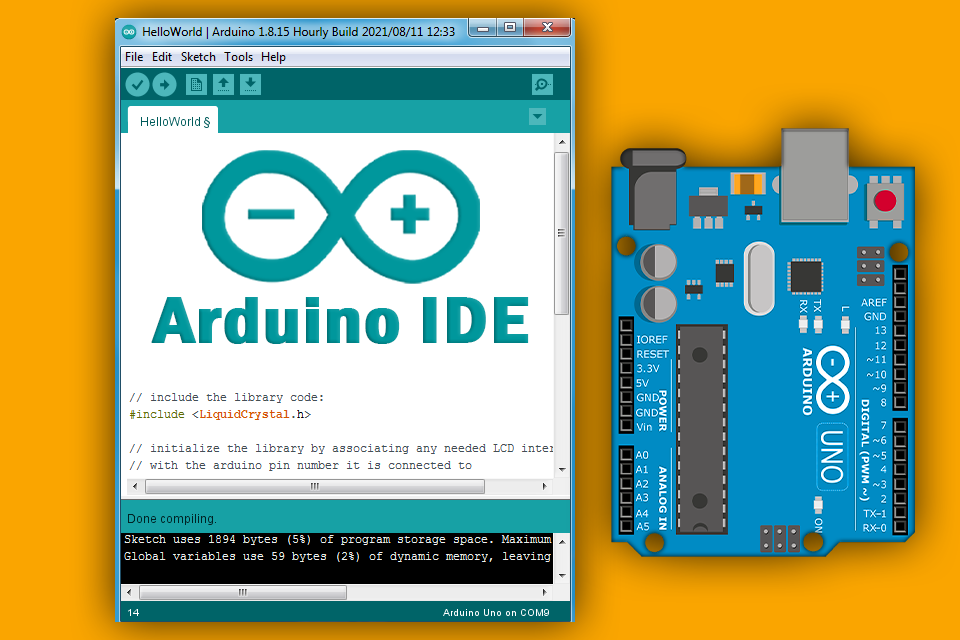
## 4.4 DC to DC Converter

The DC to DC convertor is used to convert the 12 V supply from a battery to 5 V. This 5V is supplied wherever needed in the rest of the project modules.

# **5 Software Used**

## 5.1 Arduino IDE

The Arduino integrated development environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in the programming language Java. It originated from the IDE for the languages Processing and Wiring. It includes a code editor with features such as text cutting and pasting, searching and replacing text, automatic indenting, brace matching, and syntax highlighting, and provides simple one-click mechanisms to compile and upload programs to an Arduino board. It also contains a message area, a text console, a toolbar with buttons for common functions and a hierarchy of operation menus. The source code for the IDE is released under the GNU General Public License, version 25. The Arduino IDE supports the languages C and C++ using special rules of code structuring.



#### Figure 5.1: Arduino IDE

**5.1.1 Conditions:**

1. When all of the sensors face the black surface, the motors won’t run
2. When the left and the middle sensors face the black surface, the right motor turns on, turning the robot to left
3. When the right and the left sensors would face the black surface, the condition is don’t care as it is not possible
4. When only left sensor faces the black surface, the left motor turns on, turning the robot right
5. When the right and the middle sensors face the black surface, the left motor turns on, turning the robot to right
6. When only the middle sensor faces the black surface, both the motors turn on making the robot run straight.
7. When only left sensor faces the black surface, the left motor turns on, turning the robot towards right
8. When none of the sensors face the black surface, both the motors turn on, making the robot run in straight direction

# **6 Applications of Line Follower Robot:**

### **6.1.1 Industrial Applications:**

These robots can be used as automated equipment carriers in industries replacing traditional conveyer belts.

### **6.1.2 Automobile applications:**

These robots can also be used as automatic cars running on roads with embedded magnets.

### **6.1.3 Domestic applications:**

These can also be used at homes for domestic purposes like floor cleaning etc.

### **6.1.4 Guidance applications:**

These can be used in public places like shopping malls, museums to provide path guidance.

## 6.2 Advantages:

The advantages of Line Tracking Robot are:

1. Robot movement is automatic
2. It is used for long distance applications
3. Simplicity of building
4. Fit and forget system
5. Used in home, industrial automations etc.

# **7 Building a Basic Line Follower Robot:**

Building a basic Line Follower Robot involves the following steps.

1. Designing the mechanical part or the body of the robot
2. Defining the kinematics of the robots
3. Designing the control of the robot

The mechanical part or body of the robot can be designed using AutoCAD or Workspace. A basic Line follower robot can consist of a base at the two ends of which the wheels are mounted. A rectangular sheet of hard plastic can be used as the base. Further a rigid body like a cylinder can be added along with other shaped bodies inter connected with each other by joints, and each with its defined motion in particular direction. The Line follower robot can be a wheeled mobile robot with a fixed base, a legged mobile robot with multiple rigid bodies interconnected by joints.

The next step involves defining the Kinematics of the robot. Kinematic analysis of the robot involves the description of its motion with respect to a fixed coordinate system. It is concerned mainly with the movement of the robot and with motion of each body in case of a legged robot. It generally involves the dynamics of the robot motion. The whole trajectory of the robot is set using the Kinematic analysis. This can be done using Workspace software.

The control of the robot is the most important aspect of its working. Here the term control refers to the robot motion control, i.e. controlling the movement of the wheels. A basic line follower robot follows certain path and the motion of the robot along this path is controlled by controlling the rotation of wheels, which are placed on the shafts of the two motors. So, the basic control is achieved by controlling the motors. The control circuitry involves the use of sensors to sense the path and the microcontroller or any other device to control the motor operation through the motor drivers, based on the sensor output.

A Robot is any machine which is completely automatic, i.e. it starts on its own, decides its own way of work and stops on its own. It is actually a replica of human being, which has been designed to ease human burden. It can be controlled pneumatically or using hydraulic ways or using the simple electronic control ways.

**8 Arduino Code**

#define m1 4  //Right Motor MA1

#define m2 5  //Right Motor MA2

#define m3 2  //Left Motor MB1

#define m4 3  //Left Motor MB2

#define e1 9  //Right Motor Enable Pin EA

#define e2 10 //Left Motor Enable Pin EB

//5 Channel IR Sensor Connection//

#define ir1 A0

#define ir2 A1

#define ir3 A2

#define ir4 A3

#define ir5 A4

int st=0;

void setup() {

  pinMode(m1, OUTPUT);

  pinMode(m2, OUTPUT);

  pinMode(m3, OUTPUT);

  pinMode(m4, OUTPUT);

  pinMode(e1, OUTPUT);

  pinMode(e2, OUTPUT);

  pinMode(ir1, INPUT);

  pinMode(ir2, INPUT);

  pinMode(ir3, INPUT);

  pinMode(ir4, INPUT);

  pinMode(ir5, INPUT);

}

void loop() {

  int speed = 70;

  int turn\_speed = 100;

  //Reading Sensor Values

  int s1 = digitalRead(ir1);  //Left Most Sensor

  int s2 = digitalRead(ir2);  //Left Sensor

  int s3 = digitalRead(ir3);  //Middle Sensor

  int s4 = digitalRead(ir4);  //Right Sensor

  int s5 = digitalRead(ir5);  //Right Most Sensor

  //if only middle sensor detects black line

  if(((s1 == 1) && (s2 == 1) && (s3 == 0) && (s4 == 1) && (s5 == 1))||((s1 == 1) && (s2 == 0) && (s3 == 0) && (s4 == 0) && (s5 == 1)))

  {

    //going forward with full speed

    analogWrite(e1, speed);

    analogWrite(e2, speed);

    digitalWrite(m1, HIGH);

    digitalWrite(m2, LOW);

    digitalWrite(m3, HIGH);

    digitalWrite(m4, LOW);

  }

  //if only left sensor detects black line

  else if((s1 == 1) && (s2 == 0) && (s3 == 1) && (s4 == 1) && (s5 == 1))

  {

    //going right with full speed

    analogWrite(e1, turn\_speed);

    analogWrite(e2, turn\_speed);

    digitalWrite(m1, HIGH);

    digitalWrite(m2, LOW);

    digitalWrite(m3, LOW);

    digitalWrite(m4, LOW);

  }

  //if only left most sensor detects black line

  else if((s1 == 0) && (s2 == 1) && (s3 == 1) && (s4 == 1) && (s5 == 1))

  {

    //going right with full speed

    analogWrite(e1, turn\_speed);

    analogWrite(e2, turn\_speed);

    digitalWrite(m1, HIGH);

    digitalWrite(m2, LOW);

    digitalWrite(m3, LOW);

    digitalWrite(m4, HIGH);

  }

  //if only right sensor detects black line

  else if((s1 == 1) && (s2 == 1) && (s3 == 1) && (s4 == 0) && (s5 == 1))

  {

    //going left with full speed

    analogWrite(e1, turn\_speed);

    analogWrite(e2, turn\_speed);

    digitalWrite(m1, LOW);

    digitalWrite(m2, LOW);

    digitalWrite(m3, HIGH);

    digitalWrite(m4, LOW);

  }

  //if only right most sensor detects black line

  else if((s1 == 1) && (s2 == 1) && (s3 == 1) && (s4 == 1) && (s5 == 0))

  {

    //going left with full speed

    analogWrite(e1, turn\_speed);

    analogWrite(e2, turn\_speed);

    digitalWrite(m1, LOW);

    digitalWrite(m2, HIGH);

    digitalWrite(m3, HIGH);

    digitalWrite(m4, LOW);

  }

  //if middle and right sensor detects black line

  else if((s1 == 1) && (s2 == 1) && (s3 == 0) && (s4 == 0) && (s5 == 1))

  {

    //going left with full speed

    analogWrite(e1, turn\_speed);

    analogWrite(e2, turn\_speed);

    digitalWrite(m1, LOW);

    digitalWrite(m2, LOW);

    digitalWrite(m3, HIGH);

    digitalWrite(m4, LOW);

  }

  //if middle and left sensor detects black line

  else if((s1 == 1) && (s2 == 0) && (s3 == 0) && (s4 == 1) && (s5 == 1))

  {

    //going right with full speed

    analogWrite(e1, turn\_speed);

    analogWrite(e2, turn\_speed);

    digitalWrite(m1, HIGH);

    digitalWrite(m2, LOW);

    digitalWrite(m3, LOW);

    digitalWrite(m4, LOW);

  }

  //if middle, left and left most sensor detects black line

  else if((s1 == 0) && (s2 == 0) && (s3 == 0) && (s4 == 1) && (s5 == 1))

  {

    //going right with full speed

    analogWrite(e1, turn\_speed);

    analogWrite(e2, turn\_speed);

    digitalWrite(m1, HIGH);

    digitalWrite(m2, LOW);

    digitalWrite(m3, LOW);

    digitalWrite(m4, LOW);

  }

  //if middle, right and right most sensor detects black line

  else if((s1 == 1) && (s2 == 1) && (s3 == 0) && (s4 == 0) && (s5 == 0))

  {

    //going left with full speed

    analogWrite(e1, turn\_speed);

    analogWrite(e2, turn\_speed);

    digitalWrite(m1, LOW);

    digitalWrite(m2, LOW);

    digitalWrite(m3, HIGH);

    digitalWrite(m4, LOW);

  }

  //if all sensors are on a black line

  else if((s1 == 0) && (s2 == 0) && (s3 == 0) && (s4 == 0) && (s5 == 0))

  {

    //stop

if(st==1){

    digitalWrite(m1, LOW);

    digitalWrite(m2, LOW);

    digitalWrite(m3, LOW);

    digitalWrite(m4, LOW);

}else{

st=1;

}

delay(170);

return;

  }

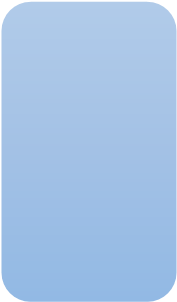
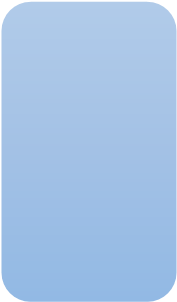
st=0;

}

# **9 Overall Assembly of Robot**

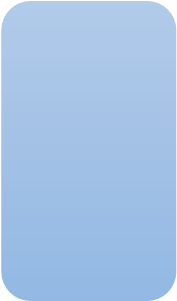
The power supply provides the 5 V to IR sensors, motor driver, timer circuit and combinational logic board. The outputs of the IR sensors are fed into the combinational logic board as inputs. This combinational logic board gives two outputs to the two input pins of the motor driver circuit. The output of the motor driver is fed into the left and the right motors of the robot which are connected to the wheels. The outputs of the PWM are fed into the two enable pins of the motor driver IC and it is used to vary the speed of the motors by changing its duty cycle with the help of potentiometer.

## 9.1 Block Diagram

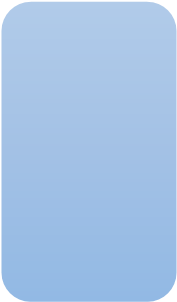


**IR**

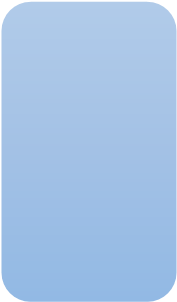
**IR**



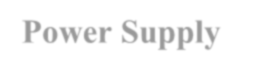
**IR**



**IR**



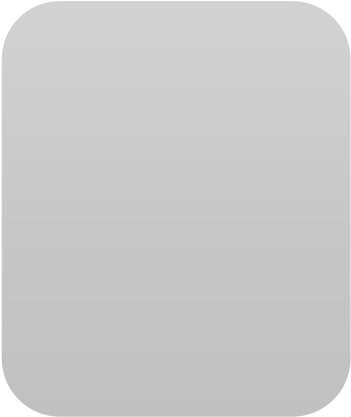
**IR**



**Power Supply**



**Arduino UNO**



**MOTOR**

**DRIVER**

M

L

M

R

# **9.2 Chassis Design**

# 

# **10 Hardware Assembly**



**Figure 10.1: Final Product**