**LINE FOLLOWING ROBORT**

**Aim :** - To Desgin a line following Robort using basic tools

**Objective :** The main aim is to detect a line following robort.

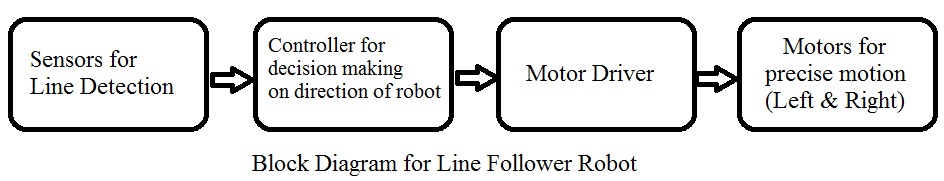
The use of line following robotic vehicle is transport the materials from one place to another place in the industries. This robot movement is completely depends on the track.The line following robots is commonly used for carry children through shopping malls, homes, entertainment places, industries.

**Components Required :**

1. Ardunio UNO & Genuiuo UNO
2. SparkFun Dual H-Bridge Motor Driver L298
3. Proximity Sensor
4. IR Sensors
5. Flexibility Wire
6. Piece of cardboard/ Acrylic sheet
7. Caster wheels

**Block Diagram of the Project:**

The line follower robot built in this project is divided in to 4 blocks. The following image shows the block diagram for line follower robot.

[](https://www.electronicshub.org/wp-content/uploads/2017/03/Block-Diagram.jpg)

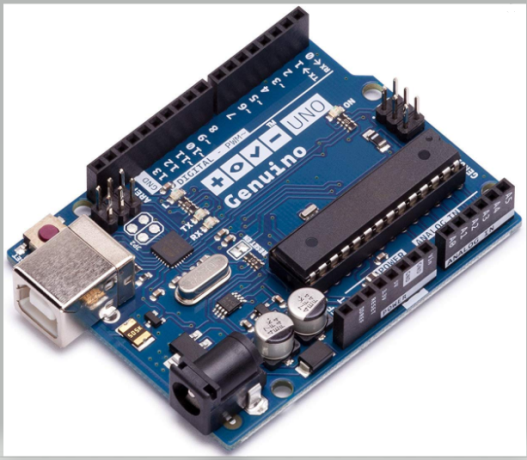
**Block Diagram Description**

**Sensors (IR Sensor)**: We have used IR Sensor Module as the line detecting sensor for the project. It consists of an IR LED and a Photo diode and some other components like comparator, LED etc.

[](https://www.electronicshub.org/wp-content/uploads/2017/03/IR-Sensor-Module.jpg)

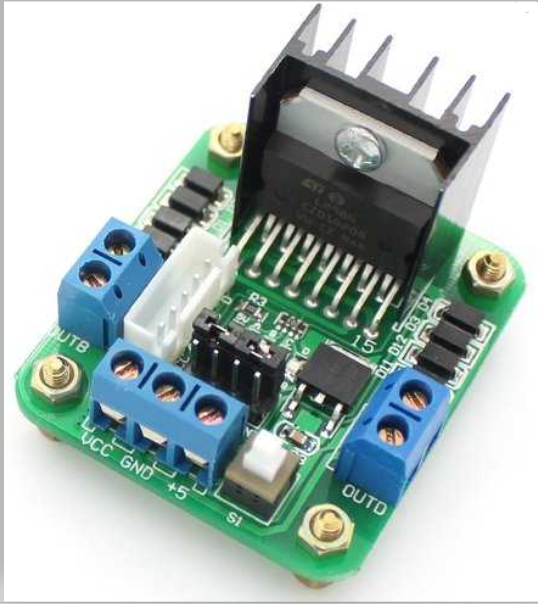
**Arduino UNO:**

The ArduinoUno is an open-source microcontroller board based on the Microchip ATmega 328P microcontroller and developed by Arduino. The board is equipped with sets of digital and analog input/output pins that may be interfaced to various expansion boards (shields) and other circuits.The board has 14 digital I/O pins (six capable of PWM output), 6 analog I/O pins, and is programmable with the Arduino IDE (Integrated Development Environment),via a type B USB cable.It can be powered by the USB cable or by an external 9-volt battery, though it accepts voltages between 7 and 20 volts. It is similar to the Arduino Nano and Leonardo.



**SparkFun Dual H-Bridge Motor Driver L298 :**

The L298N is an integrated monolithic circuit in a 15- lead Multiwatt and PowerSO20 packages. It is a high voltage , high current dual full-bridge driver de-signed to accept standard TTL logic level sand drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the in-put signals .The emitters of the lower transistors of each bridge are connected together rand the corresponding external terminal can be used for the connection of an external sensing resistor.



**Proximity Sensor:**

A proximity sensor often emits an electromagnetic field or a beam of electromagnetic radiation (infrared, for instance), and looks for changes in the field or return signal. The object being sensed is often referred to as the proximity sensor's target. Different proximity sensor targets demand different sensors. For example, a capacitive proximity sensor or

photoelectric sensor might be suitable for a plastic target; an inductive proximity sensor always requires a metal targetProximity sensors can have a high reliability and long functional life because of the absence of mechanical parts and lack of physical contact between the sensor and the sensed object.



**IR Sensors:**

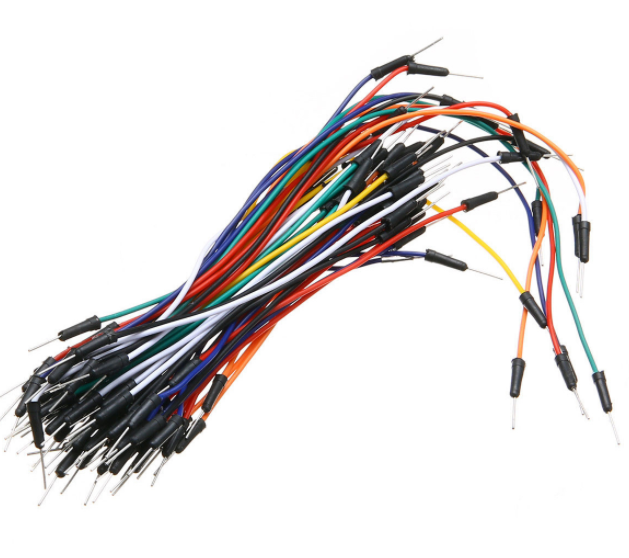
Infrared sensors can be passive or active. Passive infrared sensors are basically Infrared detectors. Passive infrared sensors do not use any infrared source and detects energy emitted by obstacles in the field of view. They are of two types: quantum and thermal. Thermal infrared sensors use infrared energy as the source of heat and are independent of wavelength. Thermocouples, pyroelectric detectors and bolometers are the common types of thermal infrared detectors.

Quantum type infrared detectors offer higher detection performance and are faster than thermal type infrared detectors. The photosensitivity of quantum type detectors is wavelength dependent. Quantum type detectors are further classified into two types: intrinsic and extrinsic types. Intrinsic type quantum detectors are photoconductive cells and photovoltaic cells.

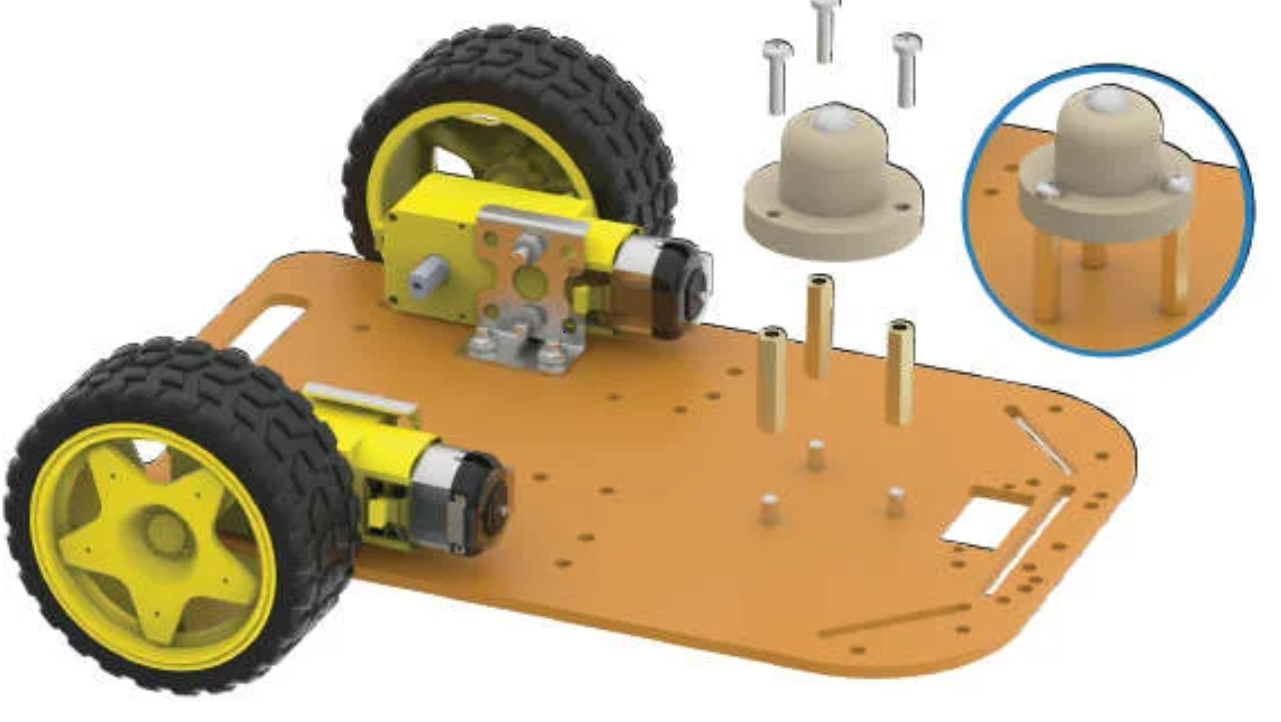


**Flexibility Wire :**

A wire is a single usually cylindrical, flexible strand or rod of metal. Wires are used to bear mechanical loads or electricity and telecommunications signals. Wire is commonly formed by drawing the metal through a hole in a die or draw plate.



**Piece of cardboard/ Acrylic sheet with Wheels:**

****

### Concepts of Line Follower:

### Concept of working of line follower is related to light. We use here the behavior of light at black and white surface. When light fall on a white surface it is almost full reflected and in case of black surface light is completely absorbed. This behavior of light is used in building a line follower robot. In this arduino based line follower robot we have used IR Transmitters and IR receivers also called photo diodes. They are used for sending and receiving light. IR transmits infrared lights. When infrared rays falls on white surface, it’s reflected back and catched by photodiodes which generates some voltage changes. When IR light falls on a black surface, light is absorb by the black surface and no rays are reflected back,

### thus photo diode does not receive any light or rays. Here in this arduino line follower robot when sensor senses white surface then arduino gets 1 as input and when senses black line arduino gets 0 as input.

### IR rx.PNG

### Circuit Explanation:

The whole **arduino line follower robot** can be divided into 3 sections: sensor section, control section and driver section.

**Sensor section:**

This section contains IR diodes, potentiometer, Comparator (Op-Amp) and LED’s. Potentiometer is used for setting reference voltage at comparator’s one terminal and IR sensors are used to sense the line and

provide a change in voltage at comparator’s second terminal. Then comparator

compares both voltages and generates a digital signal at output. Here in this **line follower circuit** we have used two comparator for two sensors. LM 358 is used as comparator. LM358 has inbuilt two low noise Op-amps.



**Connecting Sensors :-**

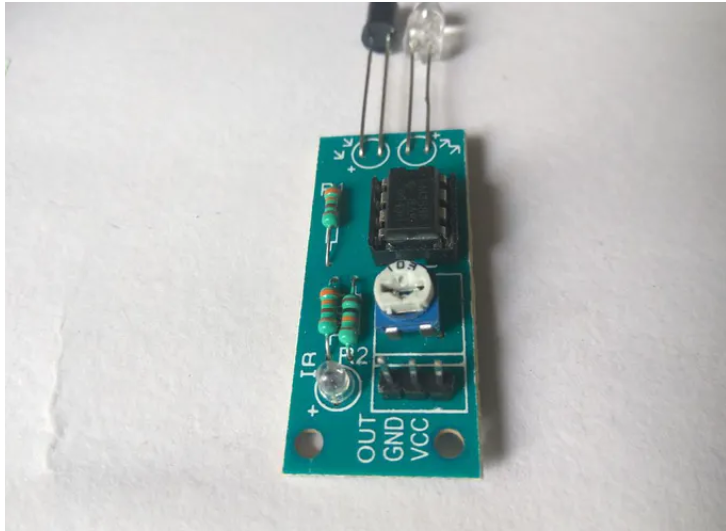
Now we will add sensors to our robot, so it can scan the for the black lines. These Infrared sensors I've used have 3 Pins,

1. VCC which is the +be terminal and should be connect to the +5v.

2. GND which is the -ve terminal and must be connected to the ground.

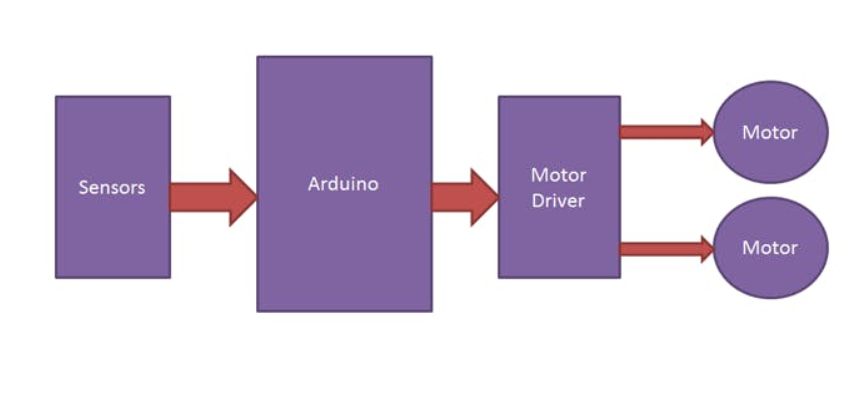
3. Signal pin. which is connected to any Analog pin on Arduino.

here we are using 2 sensors so we will need 2 Analog pins, A4, A5.

****

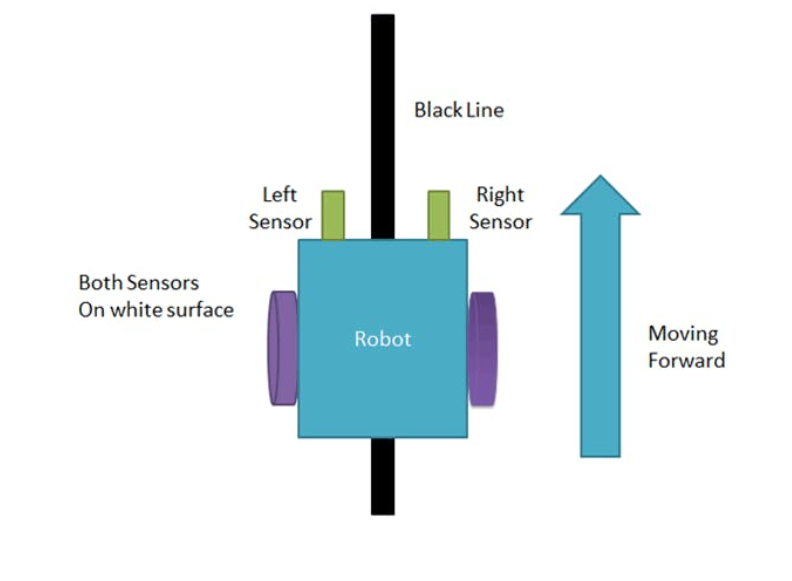
**Working of Line Follower Robot using Arduino:**

Working of line follower is very interesting. Line follower robot senses black line by using sensor and then sends the signal to arduino. Then arduino drives the motor according to sensors' output.



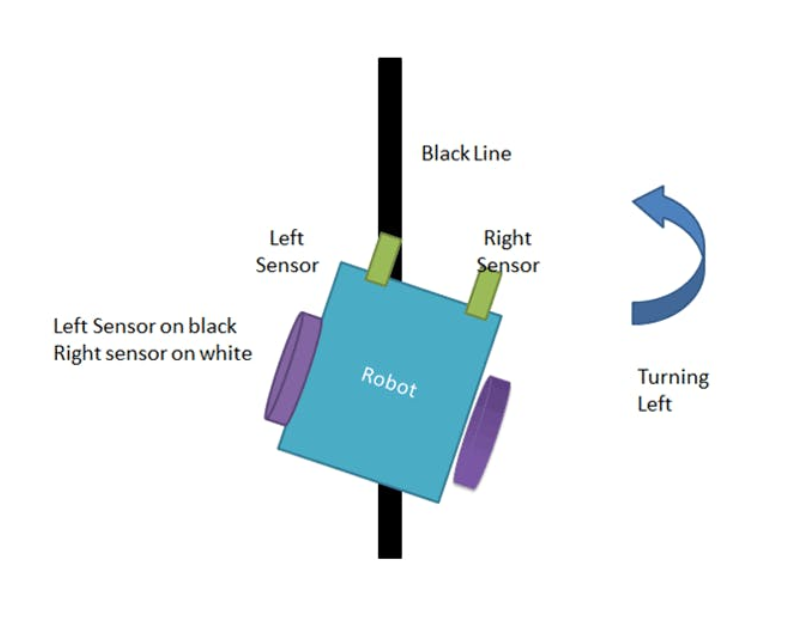
**Moving Forward:**

Here in this project we are using two IR sensor modules namely left sensor and right sensor. When both left and right sensor senses white then robot move forward.



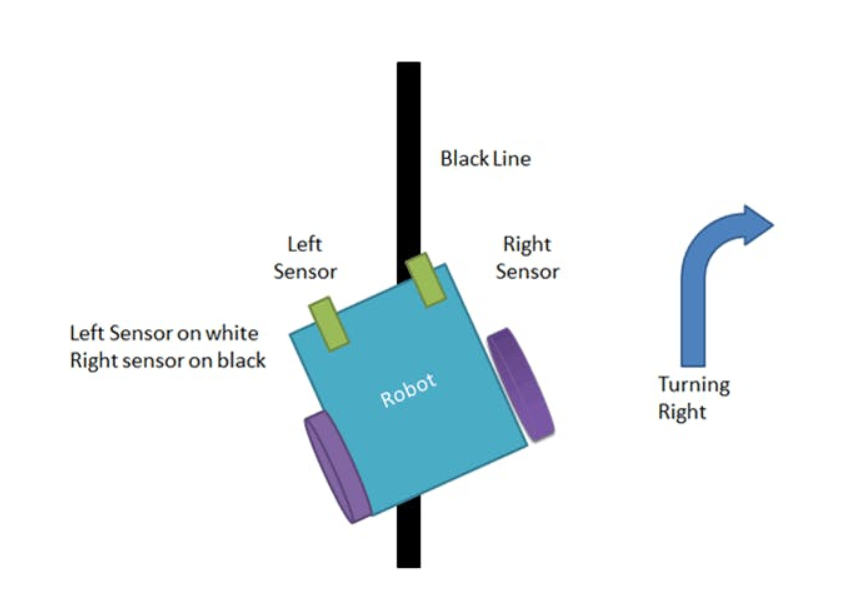
**Moving Left:**

In the IR Sensor Module , If left sensor comes on black line then robot turn left side.

****

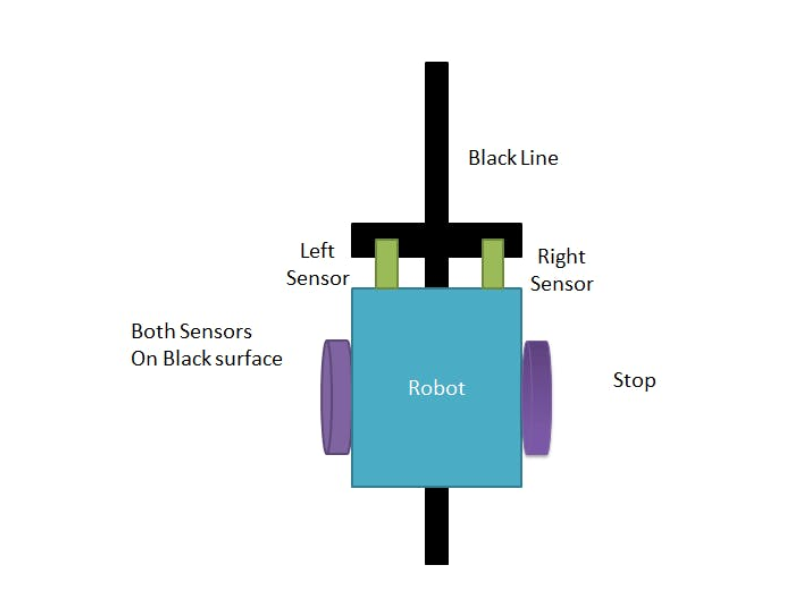
**Moving Right :**

In the IR Sensors , If right sensor sense black line then robot turn right side until both sensor comes at white surface. When white surface comes robot starts moving on forward again.



**Moving at the Junction:**

In the IR Sensors , If both sensors comes on black line, robot stops.

****

**Working with code :**

int vSpeed = 110; // MAX 255

int turn\_speed = 230; // MAX 255

int turn\_delay = 10;

//L293 Connection

const int motorA1 = 8;

const int motorA2 = 10;

const int motorAspeed = 9;

const int motorB1 = 12;

const int motorB2 = 13;

const int motorBspeed = 11;

//Sensor Connection

const int left\_sensor\_pin =A0;

const int right\_sensor\_pin =A1;

int left\_sensor\_state;

int right\_sensor\_state;

void setup() {

pinMode(motorA1, OUTPUT);

pinMode(motorA2, OUTPUT);

pinMode(motorB1, OUTPUT);

pinMode(motorB2, OUTPUT);

Serial.begin(9600);

delay(3000);

}

void loop() {

left\_sensor\_state = analogRead(left\_sensor\_pin);

right\_sensor\_state = analogRead(right\_sensor\_pin);

if(right\_sensor\_state > 500 && left\_sensor\_state < 500)

{

Serial.println("turning right");

digitalWrite (motorA1,LOW);

digitalWrite(motorA2,HIGH);

digitalWrite (motorB1,LOW);

digitalWrite(motorB2,HIGH);

analogWrite (motorAspeed, vSpeed);

analogWrite (motorBspeed, turn\_speed);

}

if(right\_sensor\_state < 500 && left\_sensor\_state > 500)

{

Serial.println("turning left");

digitalWrite (motorA1,HIGH);

digitalWrite(motorA2,LOW);

digitalWrite (motorB1,HIGH);

digitalWrite(motorB2,LOW);

analogWrite (motorAspeed, turn\_speed);

analogWrite (motorBspeed, vSpeed);

delay(turn\_delay);

}

if(right\_sensor\_state > 500 && left\_sensor\_state > 500)

{

Serial.println("going forward");

digitalWrite (motorA2,LOW);

digitalWrite(motorA1,HIGH);

digitalWrite (motorB2,HIGH);

digitalWrite(motorB1,LOW);

analogWrite (motorAspeed, vSpeed);

analogWrite (motorBspeed, vSpeed);

delay(turn\_delay);

}

if(right\_sensor\_state < 500 && left\_sensor\_state < 500)

{

Serial.println("stop");

analogWrite (motorAspeed, 0);

analogWrite (motorBspeed, 0);

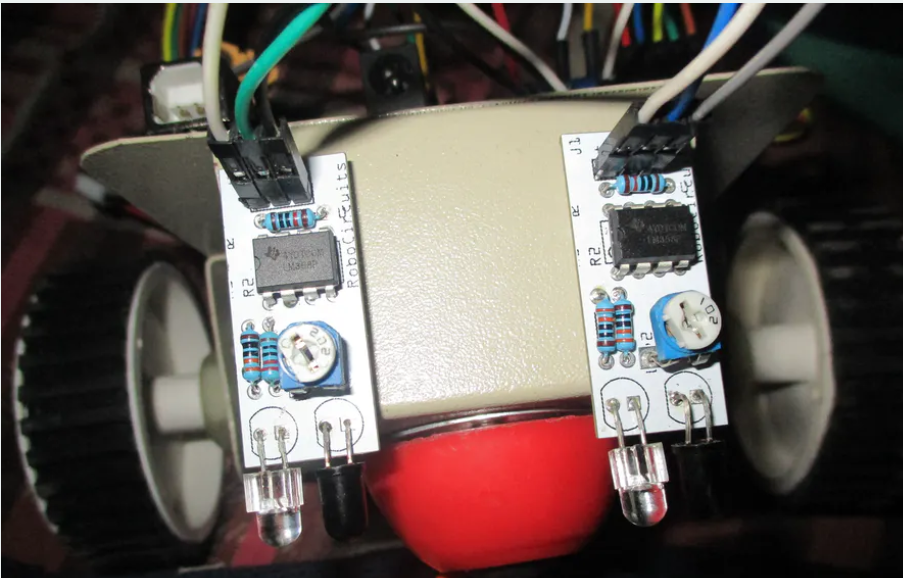
}

}

Circuit Diagram :



**Final Robot:**

****

int vSpeed = 110; // MAX 255

int turn\_speed = 230; // MAX 255

int turn\_delay = 10;

//L293 Connection

const int motorA1 = 8;

const int motorA2 = 10;

const int motorAspeed = 9;

const int motorB1 = 12;

const int motorB2 = 13;

const int motorBspeed = 11;

//Sensor Connection

const int left\_sensor\_pin =A0;

const int right\_sensor\_pin =A1;

int left\_sensor\_state;

int right\_sensor\_state;

void setup() {

pinMode(motorA1, OUTPUT);

pinMode(motorA2, OUTPUT);

pinMode(motorB1, OUTPUT);

pinMode(motorB2, OUTPUT);

Serial.begin(9600);

delay(3000);

}

void loop() {

left\_sensor\_state = analogRead(left\_sensor\_pin);

right\_sensor\_state = analogRead(right\_sensor\_pin);

if(right\_sensor\_state > 500 && left\_sensor\_state < 500)

{

Serial.println("turning right");

digitalWrite (motorA1,LOW);

digitalWrite(motorA2,HIGH);

digitalWrite (motorB1,LOW);

digitalWrite(motorB2,HIGH);

analogWrite (motorAspeed, vSpeed);

analogWrite (motorBspeed, turn\_speed);

}

if(right\_sensor\_state < 500 && left\_sensor\_state > 500)

{

Serial.println("turning left");

digitalWrite (motorA1,HIGH);

digitalWrite(motorA2,LOW);

digitalWrite (motorB1,HIGH);

digitalWrite(motorB2,LOW);

analogWrite (motorAspeed, turn\_speed);

analogWrite (motorBspeed, vSpeed);

delay(turn\_delay);

}

if(right\_sensor\_state > 500 && left\_sensor\_state > 500)

{

Serial.println("going forward");

digitalWrite (motorA2,LOW);

digitalWrite(motorA1,HIGH);

digitalWrite (motorB2,HIGH);

digitalWrite(motorB1,LOW);

analogWrite (motorAspeed, vSpeed);

analogWrite (motorBspeed, vSpeed);

delay(turn\_delay);

}

if(right\_sensor\_state < 500 && left\_sensor\_state < 500)

{

Serial.println("stop");

analogWrite (motorAspeed, 0);

analogWrite (motorBspeed, 0);

}

}