Faculty of Computers and Artificial Intelligence

Embedded Systems

Lab no 05: Line Follower Robot Prototype DC motor and Infrared sensor Interfaces

The purpose of this Lab is to learn interfaces with DC motor and infrared sensor**.** To do that, we are going to design an Arduino line follower robot project using an infrared sensor for detection.

This Arduino line follower robot project aims to achieve a line follower prototype based on an Arduino board with an IR sensor that senses black-line and then sends the detection signal to Arduino. Next, Arduino drives the motor according to the sensors' output.

Parts: -

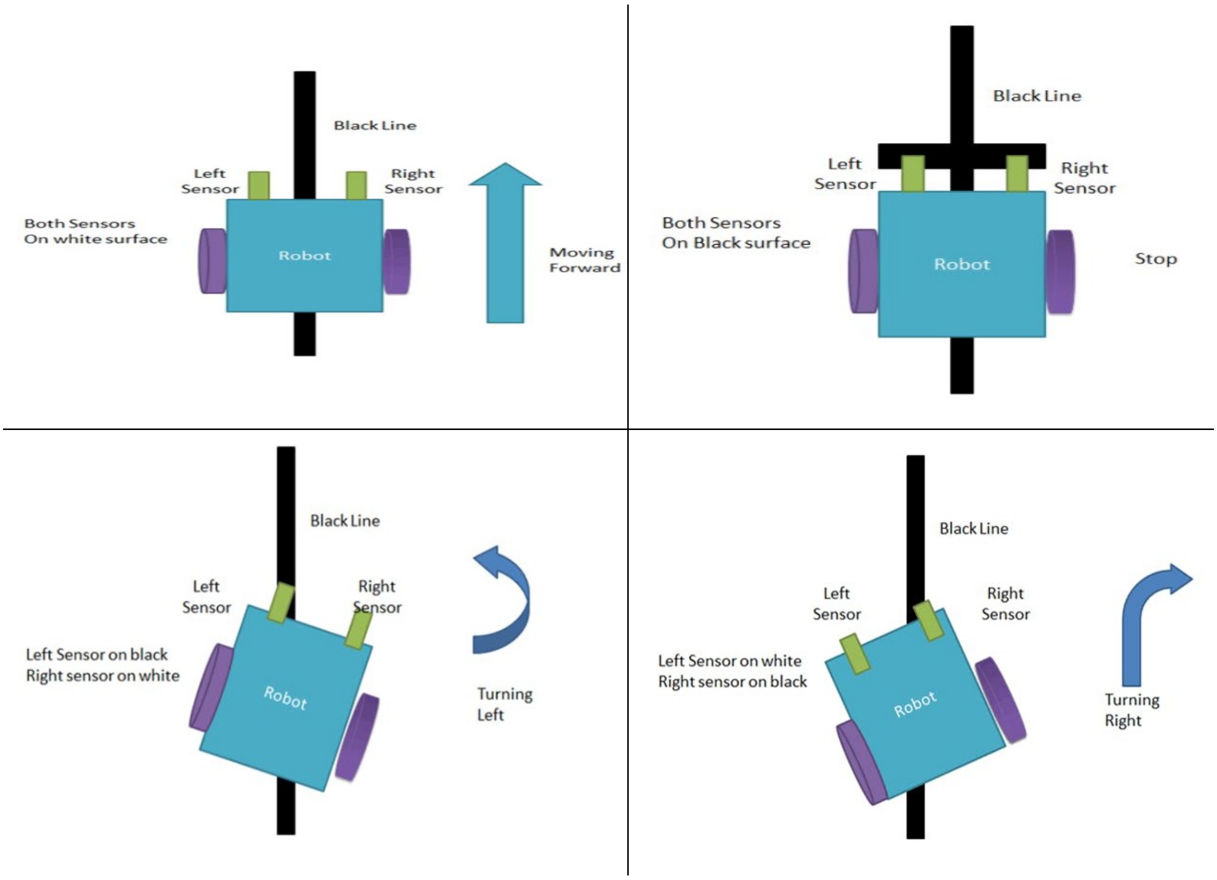
1. Concepts of the Line Follower Robot.
2. Understanding the Circuit and its Components
   a) The IR Proximity Sensor.
   b) DC Motor.
3. Hardware Connection & Preparing the Fixture.
4. Line Follower Robot Code.

Part 1. Concepts of the Line Follower Robot

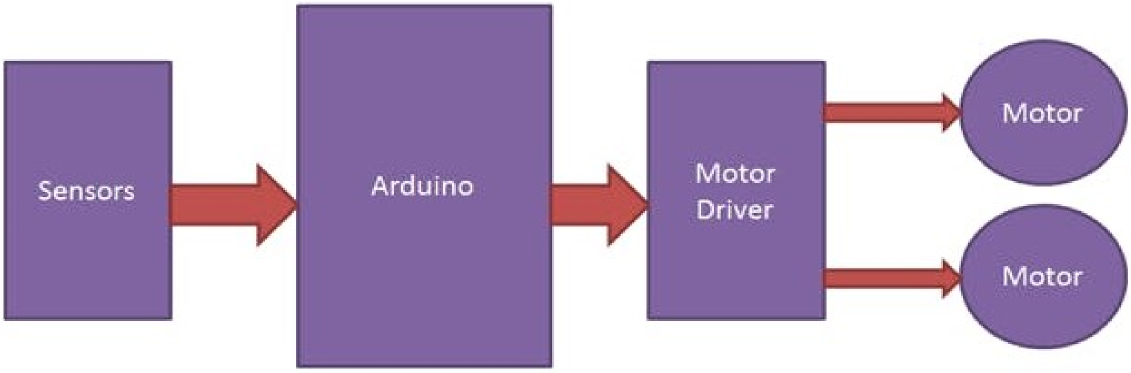
Line follower robot works based on the behavior of light on black and white surfaces. When light falls on a white surface it is almost fully reflected, while in the case of a black surface light is completely absorbed. This behavior of light is used to build a line follower robot.

Here in this project, we are using two IR sensor modules namely the left sensor and the right sensor. When both the left and right sensors sense white then the robot moves forward. If both sensors come on a black line, the robot stops. If the left sensor comes on a black line, then the robot turns to the left side.

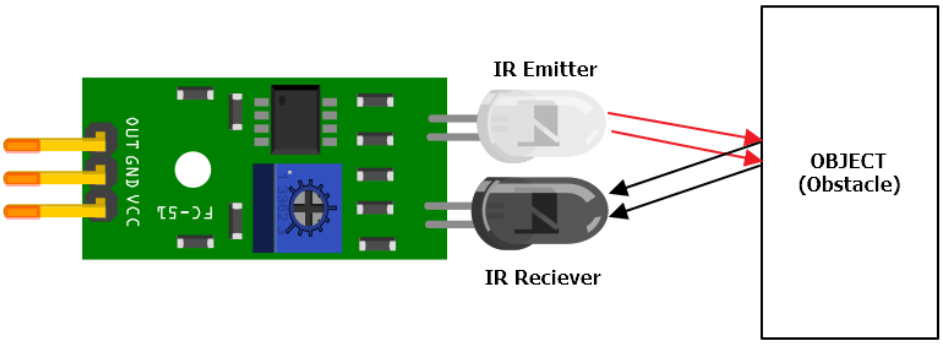
If the right sensor senses a black line, then the robot turns the right side until each sensor comes to a white surface. When the white surface comes robot starts moving on forward again.



Part 2. Understanding the circuit and its components



Part 2. a) The IR Proximity Sensors



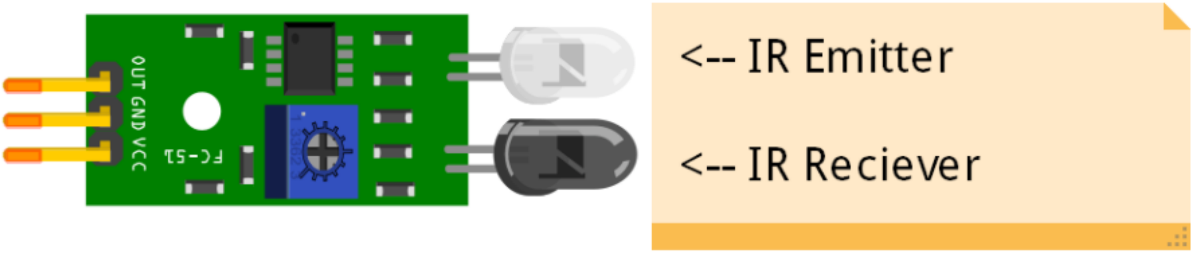
IR Proximity sensor is a simple Infrared Sensor that is used to detect obstacles. This is a multipurpose sensor and is used in many robotic applications like obstacle avoidance robots, Line Follower Robot, and any other obstacle detection projects. This is an Active Infrared sensor; it emits and detects its own IR rays. A passive infrared sensor only detects the emitted IR Rays from objects. PIR Sensor is a good example of Passive IR Sensors.

Infrared lights are Electromagnetic radiations with longer wavelengths and are not visible to human eyes. In IR Sensors we have IR emitting and receiving LEDs. In the long distance, the angle of light and sensor are meet together making the output

HIGH (+5v). If there is an obstacle the angle is less and the receiving LED can't detect the IR wave. Hence it will output a LOW Signal (0v).

IR Proximity Sensor Pinout

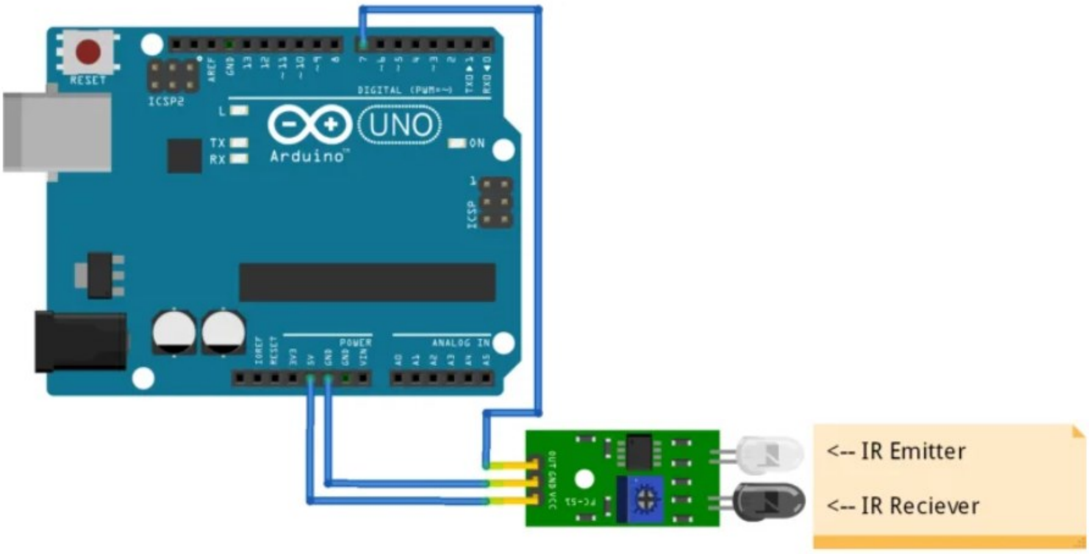
IR Sensor has three terminals, two terminals for power supply and one terminal for Output. Additionally, there is a potentiometer to calibrate the sensor.



VCC – Power Supply (+5V)

GND – Ground

OUT – Output



We use the onboard LED and Serial monitor to see the operation of IR Proximity sensor. An IR LED is connected at pin 7 of Arduino and onboard sensor is at pin 13. We defined a variable ‘sensorOut’ to store the sensor value temporarily.

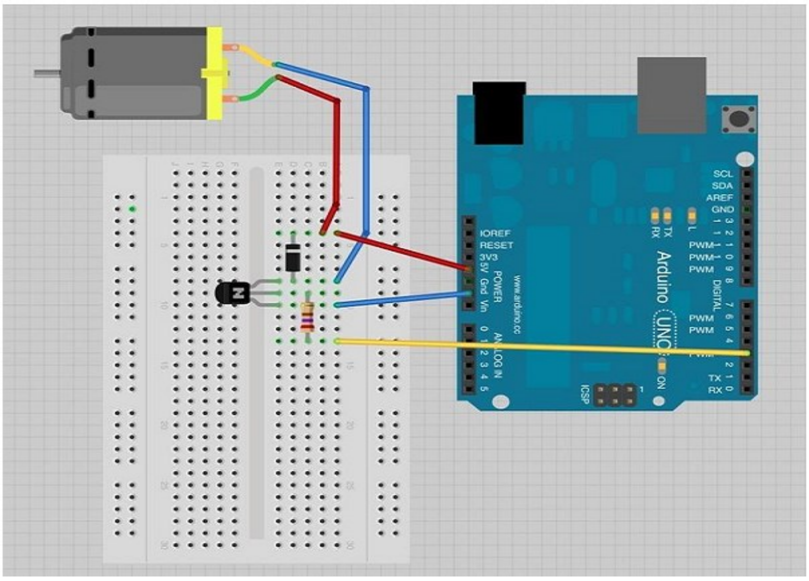
```
int LED = 13; // Onboard LED pin
int irPin = 7;  // This is our input pin (IR LED at pin 7)
int sensorOut = HIGH;  // HIGH at No Obstacle

void setup() {
  pinMode(LED, OUTPUT);
  pinMode(irPin, INPUT);
  Serial.begin(9600);
}
void loop() {
  sensorOut = digitalRead(irPin);
  if (sensorOut == LOW)
  {
    Serial.println("What is this Obstacle?");
    digitalWrite(LED, HIGH);
  }
  else
  {
    Serial.println("No Obstacle");
    digitalWrite(LED, LOW);
  }
  delay(200);
}
```

Once the code is uploaded open the serial monitor and make sure the baud rate is set to 9600. When there is no Obstacle in front of the IR Sensor it prints ‘No Obstacle’. If you place an object in front of it, then it prints ‘What is this Obstacle?’ on the serial monitor.

Part 2. b) DC motor

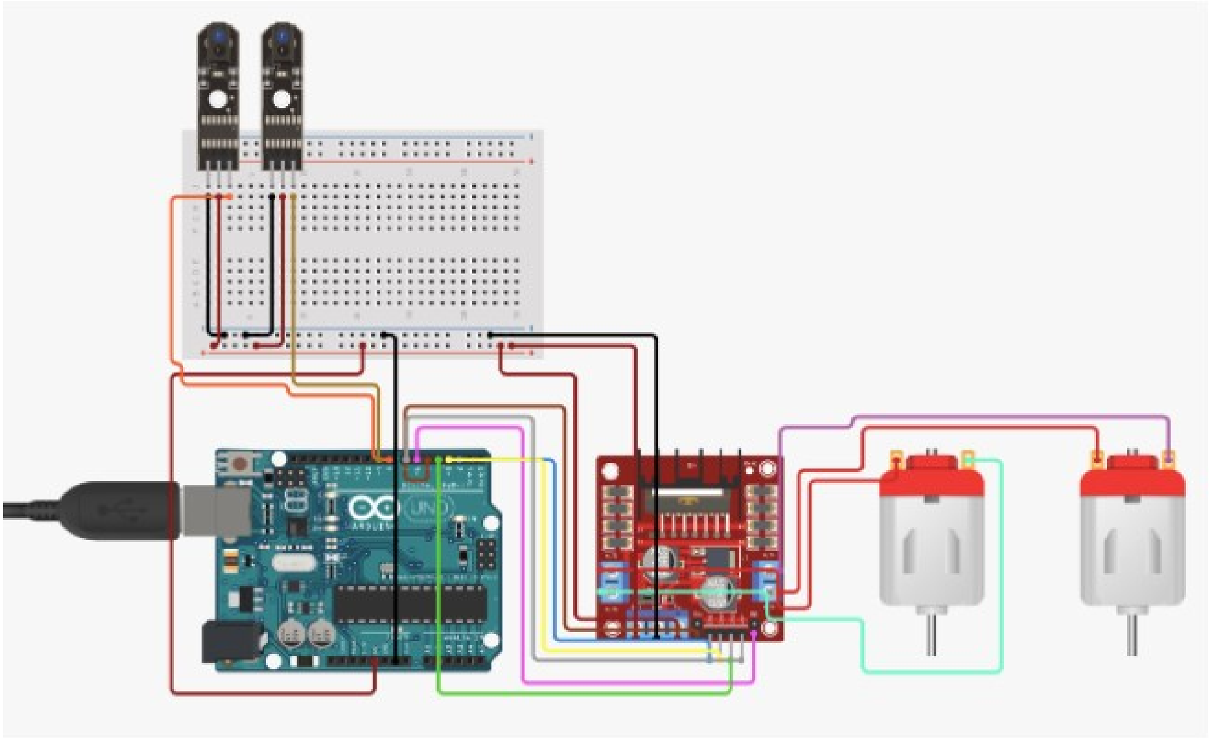
A DC motor (Direct Current motor) is the most common type of motor. DC motors normally have just two leads, one positive and one negative. If you connect these two leads directly to a battery, the motor will rotate. If you switch the leads, the motor will rotate in the opposite direction.



```
int motorPin = 3;
void setup() {
 pinMode(motorPin, OUTPUT);
 Serial.begin(9600);
 while (! Serial);
Serial.println("Speed 0 to 255");
}
 void loop()
{
 if (Serial.available())
     { int speed = Serial.parseInt();
       if (speed >= 0 && speed <= 255)
       {
         analogWrite(motorPin, speed);
       }
    }
}
```

Part 3. Hardware Connection & Preparing the Fixture.

The below figure shows the connection of the Arduino line follower robot project.



Step 1: Connect the circuit as shown in the figure above.
Step 2: Use the Arduino IDE to write your own code.
Step 3: Upload your code to the Arduino and connect it to the batteries or you can even use a power bank to run the Arduino.
Step 4: Test it on a black path.

Part 4. Line Follower Robot Code.

```
// Black Line Follower
int IR1=8;      //Right sensor
int IR2=9;    //left Sensor
// motor one
int enA = 5;    //Right motor
int MotorAip1=2;
int MotorAip2=3;
```

```
// motor two
int enB = 6;    //Left motor
int MotorBip1=4;
int MotorBip2=7;
void setup()
{
  // put your setup code here, to run once:
  pinMode(enA, OUTPUT);
  pinMode(enB, OUTPUT);
  pinMode(IR1,INPUT);
  pinMode(IR2,INPUT);
  pinMode(MotorAip1,OUTPUT);
  pinMode(MotorAip2,OUTPUT);
  pinMode(MotorBip1,OUTPUT);
  pinMode(MotorBip2,OUTPUT);
}
void loop()
{
   if(digitalRead(IR1)==HIGH && digitalRead(IR2)==HIGH) //IR
will not glow on black line
  {
    //Stop both Motors
    digitalWrite(MotorAip1,LOW);
    digitalWrite(MotorAip2,LOW);
    digitalWrite(MotorBip1,LOW);
    digitalWrite(MotorBip2,LOW);
   analogWrite (enA, 0);
   analogWrite (enB, 0);
  }

  else if(digitalRead(IR1)==LOW && digitalRead(IR2)==LOW)
//IR not on black line
  {
    //Move both the Motors
    digitalWrite(MotorAip1,HIGH);
    digitalWrite(MotorAip2,LOW);
    digitalWrite(MotorBip1,HIGH);
    digitalWrite(MotorBip2,LOW);
    analogWrite (enA, 200);
    analogWrite (enB, 200);
  }
  else if(digitalRead(IR1)==LOW && digitalRead(IR2)==HIGH)
  {
    //Tilt robot towards left by stopping the left wheel and 
moving the right one
    digitalWrite(MotorAip1,HIGH);
    digitalWrite(MotorAip2,LOW);
    digitalWrite(MotorBip1,LOW);
    digitalWrite(MotorBip2,HIGH);
    analogWrite (enA, 200);
    analogWrite (enB, 150);
   delay(100);
  }

  else if(digitalRead(IR1)==HIGH && digitalRead(IR2)==LOW)
  {
    //Tilt robot towards right by stopping the right wheel and 
moving the left one
     digitalWrite(MotorAip1,LOW);     // If I want to turn 
right then the speed of the right wheel should be less than 
that of the left wheel, here, let a be the right wheel
    digitalWrite(MotorAip2,HIGH);
    digitalWrite(MotorBip1,HIGH);
    digitalWrite(MotorBip2,LOW);
    analogWrite (enA, 150);
   analogWrite (enB, 200);
   delay(100);
  }

  else
  {
    //Stop both the motors
    digitalWrite(MotorAip1,LOW);
    digitalWrite(MotorAip2,LOW);
    digitalWrite(MotorBip1,LOW);
    digitalWrite(MotorBip2,LOW);
    analogWrite (enA, 0);
   analogWrite (enB, 0);
  }
}
```