INTEGRATING ARDUINO SOFTWARE WITH PHYSICAL ROBOT AS A SMART WAITER

Dr. H K Shankarananda
HOD, E&C Department,
TMAES Polytechnic, Hosapete
hks.anand@gmail.com

Abstract:

Automation & Robotics is the present demand in the society of technology. There is noticeable advancement happening everyday in the fields of science and technology to fulfill the necessity of the common public. This paper describes the design and functionality of robotic waiter for restaurants. It is a robotic assistant used to help restaurant staff to deliver food to the tables. A central microcontroller will be accommodated in the kitchen which sends instructions to the robot. Based on these instructions, it will calculate its path to the destination and navigate its way there. The central microcontroller will also be connected to a display that will show the table that is currently being served. The final implementation is completely functional by integrating the Arduino software with the physical robot. However the individual parts work in coordination of both hardware and software.

Keywords: Arduino, IR sensors, Microcontroller, Buzzer, VFD, VSD, DC motors, lithium batteries, servo motors

INTRODUCTION

Manpower is still an issue in the industry with workers being inefficient and having inexperienced. They would still need to be trained and paid. We also estimate that in the long term, robots would be cheaper (fixed cost at ~₹118500) versus paying workers minimum wage (~₹ 8000 monthly). With robots, restaurants could hire fewer staff and have lesser problems. This paper emphasizes the need of automation in all the fields. It proposes the idea of creating a robotic waiter that would assist restaurant staff in delivering food to the customers without any major issues.

The suggested project is a food serving robot used to serve food in hotels, homes, malls, hospitals etc., In this pandemic condition, we can use our robot to serve medicines and food to patients. In our home, we can use the robot to help our parents and also serving food to disable persons.

Line follower robots can be used in many industrial purposes. It can be used in carrying heavy and risky products. Radioactive products transportation inside a factory is very much risky for human life. A line follower robot can help in that section. Also in a hospital, it can monitor patients and inform doctors in critical situations.

In the restaurant business it can also help in many sections such as food servers and order taking jobs can be easily done by this kind of robot. In this paper, the robot architecture, design, coding, implementation and other objectives are explained.

ROBOT EQUIPMENT:

This robot is made by several parts:

Arduino Uno - 1Nos

L293D motor driver- 1Nos

IR sensor module -2 Nos

7.4 or 9V battery -1 Nos

BO motor - 2 Nos

Motor wheel - 2 Nos

Castor wheel - 1 Nos

Hobby robot chassis - 1 Nos

Wires

Screws

ROBOT EQUIPMENT:

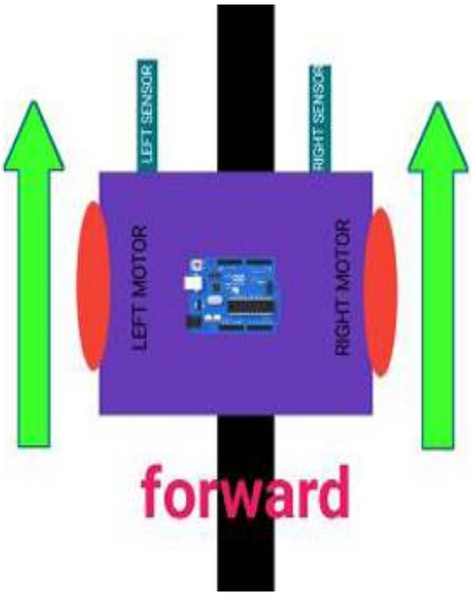
- This robot is made by several parts:
- Arduino Uno - 1Nos
- L293D motor driver- 1Nos
- IR sensor module -2 Nos
- 7.4 or 9V battery -1 Nos
- BO motor - 2 Nos
- Motor wheel - 2 Nos
- Castor wheel - 1 Nos
- Hobby robot chassis - 1 Nos
- Wires
- Screws

Navigation of Robot

How does a Line Follower Robot Navigates?

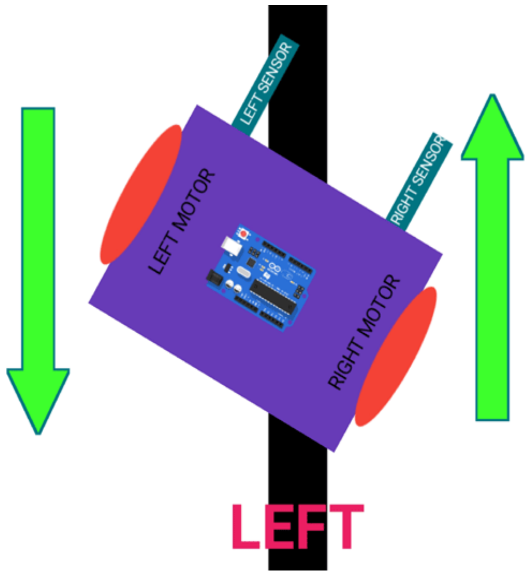
A typical line follower robot has two sets of motors, let's call them left motor and right motor. Both motors rotate on the basis of the signal received from the left and the right sensors respectively. The robot needs to perform 4 sets of motion which includes moving forward, turning left, turning right and coming to a halt. The description about the cases are given below.

Moving Forward:



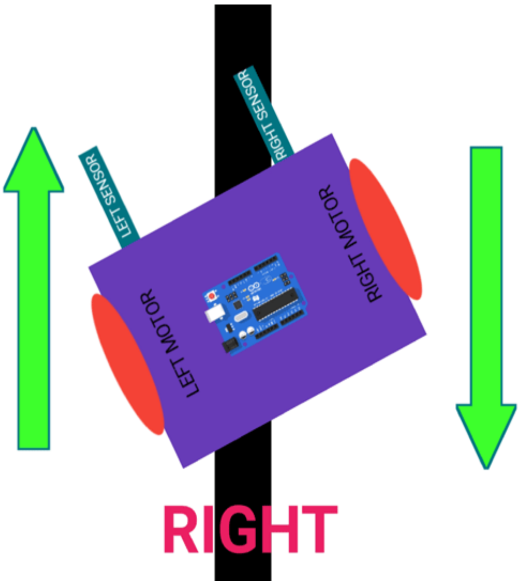
In this case, when both the sensors are on a white surface and the line is between the two sensors, the robot should move forward, i.e., both the motors should rotate such that the robot moves in forward direction (actually both the motors should rotate in the opposite direction due to the placement of motors in our setup. But for the sake of simplicity, we will call the motors rotating forward.)

Turning LEFT:



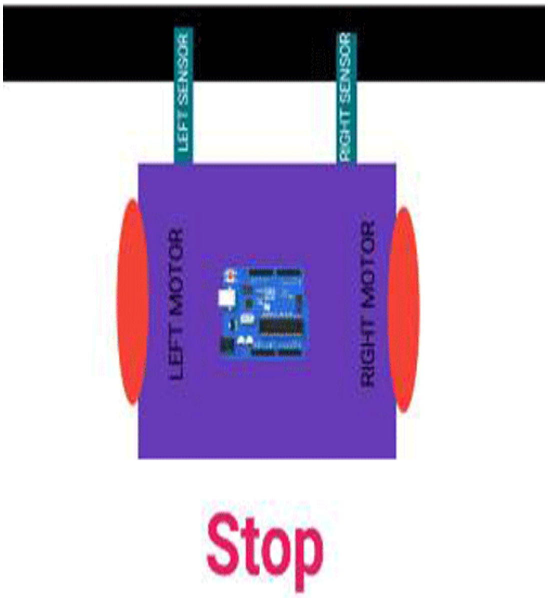
In this case, the left sensor is on top of the dark line, whereas the right sensor is on the white part, hence the left sensor detects the black line and gives a signal, to the microcontroller. Since, signal comes from the left sensor, the robot should turn to the left direction. Therefore, the left motor rotates backwards and the right motor rotates in forward direction. Thus, the robot turns towards left side.

Turning RIGHT:



This case is similar to the left case, but in this situation only the right sensor detects the line which means that the robot should turn in the right direction. To turn the robot towards the right direction, the left motor rotates forward and the right motor rotates backwards and as a result, the robot turns towards the right direction.

Stopping:

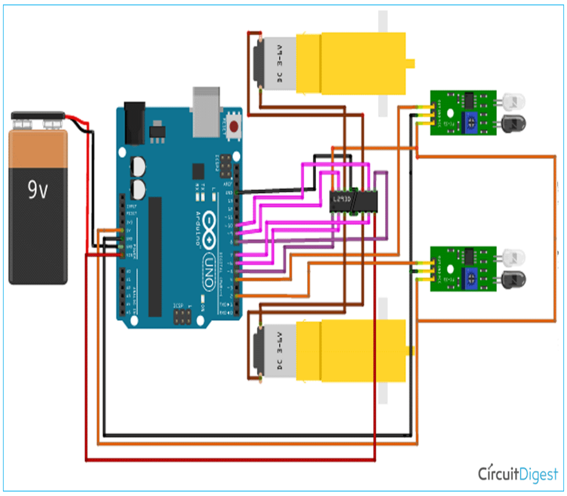


In this case, both the sensors are on top of the line and they can detect the black line simultaneously, the microcontroller is fed to consider this situation as a process for halt. Hence, both the motors are stopped, which causes the robot to stop moving.

After initializing the LCD and ultrasonic sensors, the microcontroller will read the IR array to locate the position of the path. Of the five IR array sensors, if the response is achieved from sensor from position greater or less than 3, then it is checked that whether the input is received from sensor position 2 or 4. Depending on the number of position from which the input signal is received from microcontroller, the PWM of the motor driver circuit is increased or decreased to specificvalues accordingly. Lastly, the reading achieved from ultrasonic sensors indicates that whether there is an obstacle in following the line or not.

CIRCUIT DIAGRAM

Circuit Diagram and Assembling the Arduino based Line Follower Robot



The circuit consists of mainly four parts: Two IR sensors, one motor drive, two motors, one Arduino, a battery and few connecting wires. The sensor senses the IR light reflected from the surface and feeds the output to the onboard op-amp comparator. When the sensor is situated over the white background, the light emitted by the sensor is reflected by the white ground and is received by the receiver. But when the sensor is above the black background, the light from the source doesn't reflect to it. The sensor senses the intensity of reflected light to give an output. The sensor's output is fed to the microcontroller, which gives commands to the motor driver to drive the motor accordingly. In our project, the Arduino Uno is programmed to make the robot move forward, turn right or turn left and stop according to the input coming from the sensor. The output of the Arduino is fed to the motor driver

Arduino Line Follower Robot Code

The programming part of line follower robot is very simple and we require only basic Arduino functions. The complete program for this project can be found at the bottom of this page. The explanation of this program is as follows:

First step is to defined every Arduino pin that we are using. I started with motoring the driver pins and sensor pins. Here, I have commented on each line of code for your easy understanding.

```
#define enA 5 //Enable1 L293 Pin enA
#define in1 6 //Motor1  L293 Pin in1
#define in2 7 //Motor1  L293 Pin in1
#define in3 9 //Motor2  L293Pin in1
#define in4 10 //Motor2  L293 Pin in1
#define enB 8 //Enable2 L298 Pin enB
#define R_S 4 // Right sensor
#define L_S 2 // Left sensor
```

In the loop section, declare the pin modes of each pin. Here, we need to read the output of IR sensors, hence I have defined those pins as an input. The motor needs to be driven by the Arduino, thus defining the motor driver pins as output. Finally, I pulled enable pin to high.

```
pinMode(R_S, INPUT);
pinMode(L_S, INPUT);
 pinMode(enA, OUTPUT);
pinMode(in1, OUTPUT);
 pinMode(in2, OUTPUT);
 pinMode(in3, OUTPUT);
 pinMode(in4, OUTPUT);
 pinMode(enB, OUTPUT);
digitalWrite(enA, HIGH);
digitalWrite(enB, HIGH);
```

In the loop section, we first read the values of IR sensors and then using 'if condition' control the movement of motor as per our requirements. The four movement conditions are explained below.

Moving forward:

```
if((digitalRead(R_S) == 0)&&(digitalRead(L_S) == 0)){forward();}
```

In this case, if Right Sensor and Left Sensor are on top ofwhite colour then the robot should move in the forward direction, so we call the forward function.

(Note : here 0 means output of IR sensor is high as the sensor is on white surface)

Turning towards Right Side:

```
if((digitalRead(R_S) == 1)&&(digitalRead(L_S) == 0)){turnRight();}
```

if Right Sensor is detecting Black and Left Sensor is not detecting any black line, then it will call the Right function, initiating the protocol for making right turn by calling turnRight function.

Turning towards Left side:

```
if((digitalRead(R_S) == 0)&&(digitalRead(L_S) == 1)){turnLeft();}
```

if Right Sensor is over white and the Left Sensor is detecting Black then it will call turnLeft function. It will initiate the steps for turning the robot in left direction.

Stopping the Robot:

```
if((digitalRead(R_S) == 1)&&(digitalRead(L_S) == 1)){Stop();}
```

If the Right Sensor and Left Sensor are on the Black color then it will call the Stop function. In this case, the robot will come to a complete halt.

We have defined the 4 functions of the robot as Forward, turnLeft, turnRight and Stop. The code to perform these functions are given below:

Forward function:

```
void forward(){
digitalWrite(in1, HIGH);
 digitalWrite(in2, LOW);
 digitalWrite(in3, LOW);
 digitalWrite(in4, HIGH);
}
```

This means we are pulling to high the input 1 and 4 of the motor driver, as a result, both motors will move forward same as this I made other functions

Turnright function:

```
void 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
}
```

TurnLeft function:

```
void 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
}
```

Stop function:

```
void 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 forward Pin
```

So, that's all about coding. Now we need to upload the code to the Arduino and to do so, connect your Arduino with the computer using USB cable and upload this code using the Arduino IDE.

PROGRAM CODE

Here code is developed in the same view represented by the flowchart. Code accompany with

Algorithm side of it.

```
;============================================;

; ============LINE FOLLOWING ROBOT =========;

; ==================Aurduino==================;

int irl=7;

int irr=8;

int ML=5;

int MR=6;

int irlval;

int irrval;

int buz=12;

#define echoPin 2 // attach pin D2 Arduino to pin Echo of HC-SR04

#define trigPin 3 //attach pin D3 Arduino to pin Trig of HC-SR04
```

```
// defines variables
long duration; // variable for the duration of sound wave travel
int distance;
void setup() {
   pinMode(trigPin, OUTPUT); // Sets the trigPin as an OUTPUT
  pinMode(echoPin, INPUT); // Sets the echoPin as an INPUT
  Serial.begin(9600); // // Serial Communication is starting with 9600 of baudrate speed
  Serial.println("Ultrasonic Sensor HC-SR04 Test"); // print some text in Serial Monitor
  Serial.println("with Arduino UNO R3");
  pinMode(7,INPUT);
  pinMode(8,INPUT);
  pinMode(5,OUTPUT);
  pinMode(6,OUTPUT);
    pinMode(12,OUTPUT);
  Serial.begin(9600);
  // put your setup code here, to run once:
}
void loop() {
  // put your main code here, to run repeatedly:
irlval=digitalRead(7);
irrval=digitalRead(8);
  Serial.print(irlval);
if(digitalRead(irlval)==HIGH&& digitalRead(irrval)==HIGH)
{analogWrite(6,70);
analogWrite(5,70);}
if(digitalRead(irlval)==LOW && digitalRead(irrval)==HIGH)
{analogWrite(5,70);
analogWrite(6,0);}
if(digitalRead(irlval)==LOW&& digitalRead(irrval)==LOW)
 {analogWrite(6,LOW);
  analogWrite(5,LOW);
}
if(digitalRead(irlval)==HIGH && digitalRead(irrval)==LOW)
{analogWrite(6,70);
analogWrite(5,0);}
  // Clears the trigPin condition
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  // Sets the trigPin HIGH (ACTIVE) for 10 microseconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);

  digitalWrite(trigPin, LOW);

  // Reads the echoPin, returns the sound wave travel time in microseconds

  duration = pulseIn(echoPin, HIGH);

  // Calculating the distance

  distance = duration * 0.034 / 2; // Speed of sound wave divided by 2 (go and back)

  // Displays the distance on the Serial Monitor

  Serial.print("Distance: ");

  Serial.print(distance);

  Serial.println(" cm");

  if(distance<=10)

  {digitalWrite(12,HIGH);

  analogWrite(6,LOW);

  analogWrite(5,LOW);

  }

  else{digitalWrite(12,LOW);

  }

}
```

Application:

Hotels, cinema theatre, malls, home(for disabled persons). In hotels we can use this robot for serving food for costumers and also this will attract the costumers more effectively. We can use it for cleaning purposes too.

ADVANTAGES:

- Effective and efficient work as we are using robots.
- Reduces customer waiting time.
- One time investment in the system.
- Work can be faster and may reduce the cost of laboring.

DISADVANTAGES:

- Failure of robot may fail whole system.
- Cost of maintenance is high for robots.
- Robots need the précised values to work in the environment

Future Scope:

Based on the work presented in this paper, there are various advanced features that we can add to the system. We could work on field mapping of the restaurant, for which we can use rotary encoders. With the help of field mapping, the error in traversal can be minimized and the time for traversal of the robot can also be reduced to a great extent. As we can see in recent times, everyone is trying to avoid human contact in the COVID-19 pandemic. These robots can be modified as per the requirement and deployed in any field, like medical, industries, households, etc. Robots like these can be used in the isolation wards to serve the medicine or food to the affected person. It can also be used in industries as a helping hand.

CONCLUSION

The robot is capable of following any curve or cycle. Highly efficient stepper motors can be used to control the speed of line follower robot .For better detection of obstacles along the line.

REFERENCES:

1. *N.M.A Ghani, F. Naim, T.P. Yon, "Two Wheels Balancing Robot with Line Following Capability," World Academy of Science, Engineering and Technology, pp-634-638, 2011.*
2. *A. Pathak, R.K Pathan, A.U. Tutul, N.T Tousi, A.S. Rubaba, N.Y. Bithi, "Line Follower Robot for Industrial Manufacturing Process", International Journal of Engineering Inventions, pp-10-17, October 2017.*
3. *M. Engin1, D. Engin, "Path Planning of Line Follower Robot", Proceedings of the 5th European DSP Education and Research Conference, 2012.*
4. *M.M. Myint, N.K. Soe, "Line Follower Robot using Line Tracing Sensors", International Journal of Science, Engineering and Technology Re-search (IJSETR), pp-561-563, August, 2018*