

Design and Implementation of Line Follower Robot using Arduino Microcontroller

Quazi Taif Sadat, Syed Khalid Rahman, Tabassum E Nur, Neelanjana Subin Ferdous

Abstract— This paper describes the basic architecture, operation of individual components, basic algorithm, working principle and applications of a line follower robot. A line follower robot is a device that detects and follows back and forth a specified line with a contrasted color drawn on the floor. The line can also be normal. It uses one or more sensors in the IR array to detect the line and keep the robot strictly on the track. DC motors controls the movement of the wheels of the robot. Arduino Uno R3 interface has been used to perform and implement algorithms to control the speed and direction of the robot. It can be used industrial automated equipment carriers, small household applications, tour guides in museums and other similar applications, etc. In this paper, the authors explain about the robot design, implementation, coding and relevant problems they faced along with their possible solutions.

Index Terms— Arduino Uno R3, DC motor driver circuit, IR Array, Liquid crystal display, Transistor-transistor logic, Ultrasonic sensor, Voltage regulator circuit.

1 INTRODUCTION

Recently, many line following robots have been designed and utilized. Line follower robot is a self operating mobile machine that follows a line drawn on the floor. The path can be a visible black line on white surface or vice versa. Using ultrasonic sensors and IR array sensors which are set up inside the robot, the path location can be easily detected and followed.

Line follower robot can be used in many industrial purposes. It can be used in carrying heavy and risky products. It can also transport radioactive material in a nuclear power plant. In restaurant, this robot can take orders and serve food accordingly.

In this paper, the detailed component wise construction will be discussed in section 2, circuit diagram and working principle will be explained in section 3, flow chart and relevant explanations 4 will be discussed in section and the programming part 5 will be illustrated in section.

2 COMPONENTS OF LINE FOLLOWER ROBOT

2.1 Arduino UNO R3 and IDE

Arduino Uno R3 is a microcontroller board based on the ATmega328P (datasheet). It has 6 analog input pins and 13 digital all of which can be used as PWM outputs, 16 MHz quartz crystal, a USB connection, power jack and a reset button. It contains everything needed to support the microcontroller; by simply connecting it to a computer with a USB cable one can program it. It supports 5V DC to 12 V DC. The safe power supply is 9V DC. Arduino controls the whole robot actions. The motor shield is placed above it. Motor shield's pins are connected to Arduino's pins.

Arduino IDE is used for writing a program and inserting the

code to the Arduino is done by USB cable which connects Arduino and the computer.

2.2 Digital IR Array Sensor and Ultrasonic Sensor

The line follower robot uses 7 array digital IR sensor array to sense the line. Among them, five IR sensors are used with a suitable distance from each other because there are six analog pins on Arduino. The IR sensors contain an infrared transmitter and infrared receiver pair. For balancing left and right side four sensors are used and one middle sensor for line detection. On each IR sensor, there are two diodes. One of them send Infrared rays and another one receives it. If the receiver receives more reflected light than it is on the white surface and if receives less reflected light (or doesn't receive any reflected light) that means it is on the black surface. One IR sensor includes one infrared transmitter and one receiver. IR sensor array is the combination of five IR reflectance-sensors. The distance between the surface and the IR sensors array should be less than 5mm and distance between two IR sensors depends on line width. If the line width is narrow, the distance between the sensors must be reduced; otherwise, while curving the line, the robot will not be turned on time. The sensors work well if it is shielded from ambient light. The ultrasonic sensor detects obstacles from a predefined distance and sends a signal to Arduino UNO R3 sensor through trigger pin to respond accordingly. The sensor operates on 5V DC. This sensor protects the robot from collisions due to obstacles.

2.3 DC Motor Driver Circuit

A well-known and suitable motor driver is IC L298 which can be used to control two motors. It is a high voltage, high-current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as DC and stepping motors [3]. Two enable inputs of 5 volts are provided to enable or disable the device independently of the input signals. L298 has 2 amperes per channel current capacity and it can support from 3 volts to 30 volts for outputting. Moreover, L298 works well up to 16 volts without any heat sink.

2.4 Voltage Regulator Circuit

The voltage regulator circuit contains a diode 1N4007, two capacitors of 100uF, one resistor of 10k and a BC547 model transistor. It converts the incoming DC signal of 12 volts from

-
- Quazi Taif Sadat is currently working as Director, Innovation Lab in Bangladesh University, Bangladesh, E-mail: taifsadat@hotmail.com
 - Syed Khalid Rahman is currently working as Senior Lecturer in Bangladesh University, Bangladesh, E-mail: Rahman.khalide@bu.edu.bd
 - Tabassum E Nur is currently working as Senior Lecturer in Bangladesh University, Bangladesh, E-mail: tabassum.vnsc@gmail.com
 - Neelanjana Subin Ferdous is currently working as Lecturer in Bangladesh University, Bangladesh, E-mail: neelanjanabeece6@gmail.com

5 volts for proper component wise operation. The circuit is connected to other components to ensure accurate supply of voltage.

2.5 Chassis Board

Robot body is another important thing. One can use many kinds of chassis but it should be kept in mind that the chassis has to support all devices and has to be strong. It can be made from glass, plastic, Aluminium or any other lightweight materials.

The authors have made car shape line follower robot. Devices are installed above the chassis and motors and sensors are installed below the chassis by screws.

3 CIRCUIT DIAGRAM

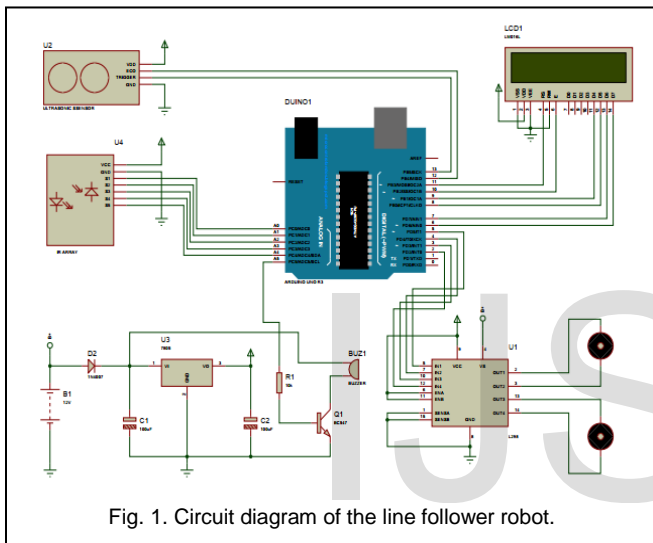


Fig. 1. Circuit diagram of the line follower robot.

In the circuit diagram, the Arduino UNO R3 microcontroller receives the approximate location of the line on surface from the signals received from IR array and ultrasonic array. As the IR array consists of five sensors, the location of the line is specified from the intensity of responses from the five sensors. At the same time, the voltage regulator circuit maintains a constant 9V input in the microcontroller. The current location, motion speed and other parameters are constantly updated in LCD display. Based on responses of IR and ultrasonic sensors, the programmable microcontroller generates an output which is driven to motor driver circuit. The motor driver circuit controls the direction of rotation of motor according to the inputs received from the Arduino UNO R3 microcontroller.

4 FLOW CHART

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 specific

values accordingly. Lastly, the reading achieved from ultrasonic sensors indicates that whether there is an obstacle in following the line or not.

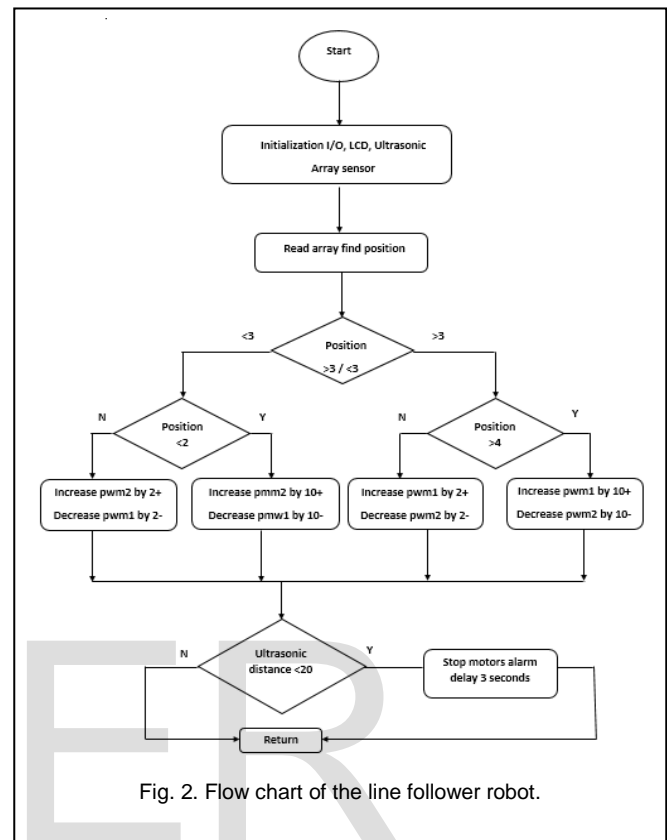


Fig. 2. Flow chart of the line follower robot.

5 SOURCE CODE

Arduino microcontroller has its own IDE for coding. It supports C as a programming language. Arduino IDE has its own facility to connect Arduino to computer via USB cable and passes enough current and code through it.

```
#include<LiquidCrystal.h>
LiquidCrystal lcd(11, 10, 9, 8, 7, 6);
const int buzzer = A5;
const int trigPin = 13;
const int echoPin = 12;
unsigned int avg_distance = 0;
unsigned int avg_distance1 = 0;
long duration;
int distance;
double alpha = 0.75;
int period = 100;
double change = 0.0;
double minval = 0.0;
int d = 0;
int readY = 0;
int LL, LM, M, RM, RR;
int right_duty = 100, left_duty = 100;
void beep()
{
    digitalWrite(buzzer, HIGH); delay(100);
```

```
digitalWrite(buzzer, LOW); delay(100);
}

void setup()
{
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(buzzer, OUTPUT);
  pinMode(A0, INPUT_PULLUP);
  pinMode(A1, INPUT_PULLUP);
  pinMode(A2, INPUT_PULLUP);
  pinMode(A3, INPUT_PULLUP);
  pinMode(A4, INPUT_PULLUP);
  beep();
  lcd.begin(16, 2);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Line Follower");
  delay(2000);
  lcd.clear();
}

void loop()
{
  read_line();
  Distance_measurement();
  if (reaDy == 1 && avg_distance1 < 20) // obstacle ahead
  {
    beep();
    StoP();
    lcd.setCursor(0, 1);
    lcd.print("Obstacle! ");
    delay(1000);
    reaDy = 0;
  }
  else if (reaDy == 1 && avg_distance1 > 20)
  {
    lcd.setCursor(0, 1);
    lcd.print("Clear ahead ");
    delay(100);
    drive();
  }
}

} //end of void loop

void drive()
{
  if (LL == HIGH && LM == HIGH && M == LOW && RM == HIGH && RR == HIGH) //in line
  {
    go_forward();
    right_duty = 255;
    left_duty = 255;
  }
  else if (LL == HIGH && LM == HIGH && M == HIGH && RM == LOW && RR == HIGH) //in line+full right
  {
    go_forward();
    right_duty = 130;
    left_duty = 255;
  }
  else if (LL == HIGH && LM == HIGH && M == HIGH && RM == HIGH && RR == LOW) //in line+full right
  {
    go_forward();
    right_duty = 0;
    left_duty = 255;
  }
  else if (LL == HIGH && LM == LOW && M == HIGH && RM == HIGH && RR == HIGH) //in line+little left
  {
    go_forward();
    right_duty = 255;
    left_duty = 130;
  }
  else if (LL == LOW && LM == HIGH && M == HIGH && RM == HIGH && RR == HIGH) //in line+full left
  {
    go_forward();
    right_duty = 255;
    left_duty = 0;
  }
  else if (LL == LOW && LM == LOW && M == LOW && RM == LOW && RR == LOW) //in line + end point
  {
    StoP();
    beep();beep();
    go_back();
    delay(1000);
    StoP();
    delay(1000);
    right_duty = 0;
    left_duty = 0;
  }
}

void read_line()
{
  LL = digitalRead(A0);
  LM = digitalRead(A1);
  M = digitalRead(A2);
  RM = digitalRead(A3);
  RR = digitalRead(A4);
  lcd.setCursor(11, 0);
  lcd.print(LL);
  lcd.print(LM);
  lcd.print(M);
  lcd.print(RM);
  lcd.print(RR);
}

void go_forward()
{
  analogWrite(2, right_duty); //right forward
  digitalWrite(3, LOW);
}
```

```

    analogWrite(4, left_duty); //left forward
    digitalWrite(5, LOW);
}

void go_back()
{
    digitalWrite(2, LOW); //right backward
    digitalWrite(3, HIGH);

    digitalWrite(4, LOW); //left backward
    digitalWrite(5, HIGH);
}

void turn_right()
{
    digitalWrite(2, LOW); //right backward
    analogWrite(3, right_duty);

    analogWrite(4, left_duty); //left forward
    digitalWrite(5, LOW);
}

void turn_left()
{
    analogWrite(2, right_duty); //right forward
    digitalWrite(3, LOW);

    digitalWrite(4, LOW); //left backward
    analogWrite(5, left_duty);
}

void StoP()
{
    digitalWrite(2, LOW); //right stop
    digitalWrite(3, LOW);

    digitalWrite(4, LOW); //left stop
    digitalWrite(5, LOW);
}

void Distance_measurement()
{
    // Clears the trigPin//Sonar module HC-SR04
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    // Sets the trigPin on HIGH state for 10 micro seconds
    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; // extract distance into cm
    value
    avg_distance += distance;

```

```

d++;
if (d > 2)
{
    avg_distance /= 2;
    avg_distance1 = avg_distance;
    avg_distance = 0;
    // Prints the distance on the Serial Monitor
    d = 0;
    readY = 1;
    lcd.setCursor(0, 0);
    lcd.print("Dist:");
    lcd.print(avg_distance1);
    lcd.print("cm ");
}
Serial.print("Distance: ");
Serial.println(avg_distance1);
}

```

//

6 END SECTION

6.1 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.

ACKNOWLEDGMENT

The authors acknowledge financial and technical aid for this research received from Bangladesh University.

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 Tousei, 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