**LINE FOLLOWING ROBOT**

A course project report submitted

partial fulfilment of the requirement of

**SMART SYSTEM DESIGN**

By

S. Charan  (2203A51518)

K. Tharun  (2203A51486)

T. Sri Harshitha  (2203A51522)

T. Sathvika  (2203A51523)

Under the guidance of

Mr P. Ramchandar Rao

Assistant Professor & Member, Center for Embedded Systems

and IOT, Department of ECE.

Mr Sumith Gupta

Assistant Professor & Member, Center for Embedded Systems

and IOT, Department



**ABSTRACT**

**This project aims to implement the algorithm and control the movement of the robot by proper tuning of the control parameters and thus achieve better performance. In addition, the LCD interface is added to display the distance travelled by the robot. It can be used industrial automated equipment carriers, small household applications, tour guides in museums and other similar applications, etc. Line Following Robot is an autonomous robot which is able to follow a black line that is drawn on the surface consisting of a contrasting colour. It is designed to move automatically and follow the line. The robot uses arrays of optical sensors to identify the line, thus assisting the robot to stay on the track. The array of four sensor makes its movement precise and flexible. The robot is driven by DC gear motors to control the movement of the wheels. The Arduino Uno interface is used to perform and implement algorithms to control the speed of the motors, steering the robot to travel along the line smoothly.**

CONTENTS

CHAPTER NO. TITLE PAGE NO.

1 Introduction 1

2 System description 2

2.1 Block Diagram 3

3 Hardware & Software Tools 4

4 Implementation 5

5 Results 6

6 Conclusion 7

References 8

**CHAPTER 1**

**INTRODUCTION**

* 1. **INTRODUCTION**

**The Line follower robot is a mobile machine that can detect and follow the line drawn on the floor. Generally, the path is predefined and can be either visible like a black line on a white surface with a high contrasted colour or it can be invisible like a magnetic field. Therefore, this kind of Robot should sense the line with its Infrared Ray (IR) sensors that installed under the robot. After that, the data is transmitted to the processor by specific transition buses. Hence, the processor is going to decide the proper commands and then it sends them to the driver and thus the path will be followed by the line follower robot. TABAR is a line follower robot designed and tested to attend at Tabrizi line follower robot competition. But it encounters with some technical and mechanical problems. In this Paper, we have illustrated the process of design, implementation and testing TABAR, a small line follower robot designed for the line follower robot competition. The technical and mechanical issues and problems also have investigated. In order to follow a black line on a white surface, the Line Following Robot interfaces the NXP LPC1758 microcontroller on the SJ One Board with a Reflectance Sensor Array and Dual Serial Motor Controller. The block diagram below illustrates the connectivity of these primary components. The diagram also illustrates the connectivity between the Dual Serial Motor Controller, two brushed DC motors, twin motor gearbox, and two tires. In order to follow a black line on a white surface, the Line Following Robot interfaces the NXP LPC1758 microcontroller on the SJ One Board with a Reflectance Sensor Array and Dual Serial Motor Controller. The block diagram below illustrates the connectivity of these primary components. The diagram also illustrates the connectivity between the Dual Serial Motor Controller, two brushed DC motors, twin motor gearbox, and two tires.**

**1.2 OBJECTIVE**

**The objective of the line following robot is to follow a line on its given path which is obtained for which it uses IR sensors which detects the line and H bridge which controls the working of the wheel's.**

**CHAPTER 2**

**PROJECT DESCRIPTION**

**2.1 BLOCK DIAGRAM OF THE PROJECT**

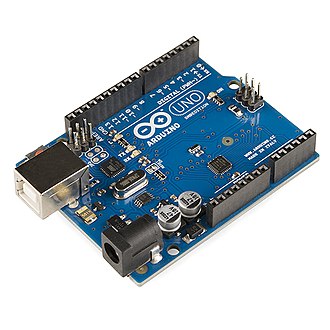
**2.2**

**As shown in the above schematic diagram it mainly consists of an Arduino, two sensors i.e., IR sensors, motor driver, 2 motors. The IR sensors are connected to Arduino. Arduino is connected to motor driver and to four motors. The two IR sensors detects the white surface and travel in black path which we draw.**

**2.3 HARDWARE DESCRIPTION**

**2.3.1 Arduino Uno**

**Arduino Uno is an 8-bit ATmega328P microcontroller. To support the microcontroller, it uses the components such as crystal oscillator, serial communication, voltage regulator It has 14 digital I/O pins (6 pins can be used as PWM pins). It has six separate analogy input pins, a USB connection, a Power barrel jack, an ICSP header, and a reset button**.



**This board is programmable with the Arduino IDE (Integrated Development Environment) platform via a type B USB cable. This board can be powered by a USB cable or an external voltage between 7 to 20 volts. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter. Revision 2 of the Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode. Revision 3 of the board has the following new features: 1.0 pinout: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible both with the board that use the AVR, which operate with 5V and with the Arduino Due that operate with 3.3V. The second one is a not connected pin, that is reserved for future purposes. Stronger RESET circuit. Atmega 16U2 replace the 8U2. "Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the index of Arduino boards.**

**Microcontroller ATmega328**

**Operating Voltage 5V**

**Input Voltage (****recommended) 7-12V**

**Input Voltage (****limits) 6-20V**

**Digital I/O Pins 14 (of which 6 provide PWM output)**

**Analog Input Pins 6**

**DC Current per I/O Pin 40 mA**

**DC Current for 3.3V Pin 50 mA**

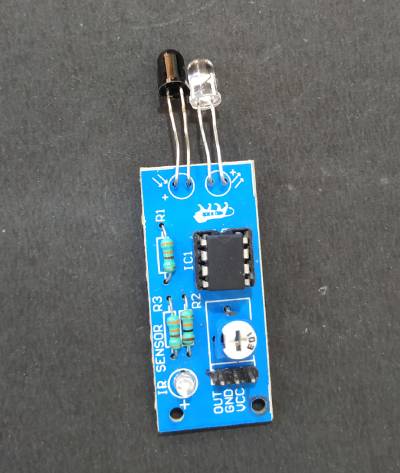
**Flash Memory 32 KB of which 0.5 KB used by bootloader**

**SRAM 2 KB (ATmega328)**

**EEPROM 1 KB (ATmega328)**

**2.3.2 Infrared sensors**

**An infrared sensor emits the light to detect some surroundings. In the infrared spectrum, all the objects radiate some form of thermal radiation that is invisible to our eyes, but an IR sensor can detect these radiations. Here, IR LED is an emitter, and the IR photodiode is a detector. An IR LED emits the IR light, and the photodiode is sensitive to this IR light. When IR light falls on the photodiode, the output voltages and the resistances will change in proportion to the magnitude of the received IR light.**



**The Infrared detection system uses the five essential elements: an infrared source, a transmission medium, optical component, infrared detectors, and signal processing. An infrared transmission can be done through the vacuum, atmosphere, and optical fibres. To know more about the IR sensor, refer to the article, working principle of IR sensor.**

**The robot travels along the line using the IR sensor. The sensor has two diodes, one diode sends infrared light, the other diode receives the reflected light from the surface. When the infrared rays fall on the white surface, they are reflected back. When infrared light falls a black surface, the light is absorbed by the black surface and no rays are reflected back, so the photodiode does not receive any light. The sensor measures the amount of reflected light and sends the value to the Arduino. There is a potentiometer on the sensor, with which we can adjust the sensitivity of the sensor.**

**2.3.3 L293D motor shield**

**The L293D is designed to provide bidirectional drive currents of up to 600-mA at voltages from 4.5 V to 36 V. Both devices are designed to drive inductive loads such as relays, solenoids, DC and bipolar stepping motors, as well as other high-current/high-voltage loads in positive-supply applications. It can drive 4 DC motors on and off or drive 2 DC motors with directional and speed control. L293D DC Motor Driver Module is fully compatible with Arduino and Raspberry Pi cards.**



**The L293D is designed to provide bidirectional drive currents of up to 600-mA at voltages from 4.5 V to 36 V. Both devices are designed to drive inductive loads such as relays, solenoids, DC and bipolar stepping motors, as well as other high-current/high-voltage loads in positive-supply applications.**

**2.3.4 BO motor**

**BO Motor is known as Battery Operated motor. These motors are commonly used in hobby-grade projects where the user requires a small DC motor as a simple actuator.BO series linear motor provides good torque and rpm at lower operating voltages. The BO motors are available in single Shaft, Dual Shaft, and DC Plastic Gear BO. These motors consume low current. In this project, we have used four single shaft BO motors. DC motor (BO) Battery Operation. Dc motor converts electrical energy into mechanical energy. Why DC gear motor used in robot Motor control circuit. DC MOTOR concept is where gears reduce the speed of the vehicle but increase its torque is known as gear reduction. In DC motor is assembled with multiple gear setup. Speed of motor is counted in terms of rotations of the soft per minute is called RPM. RPM means Revolution Per Minute. The setup assemble helps to increasing the torque and reduce the motor speed. All micro-controller-based Robots this type of DC motor can be used.**



**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.**

**2.3.5 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.**

**This Wheel is a must when you are building your own movable robot, a line follower maybe! An important component to keep your robot moving, mountable on a chassis.**



**2.3.6 Battery**

**Batteries are the most common power source for mobile robots. In today's designs, lead-acid batteries have been mostly replaced by lithium chemistries. The two most common lithium chemistries in these applications are variations on lithium-ion (Li-ion), and lithium iron phosphate (LiFePO4).**



**Battery – A battery is comprised of one or more cells. It is a complete unit which can be used to provide power to the robot. Power (Watt) – Power is the rate at which work is done. Often when we compare how power hungry something is we will look at how much power it uses.**

**2.4 Software**

**Arduino software**

**The software used here is ARDUINO SOFTWARE:**

**The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and Genuine hardware to upload programs and communicate with them.**

**Writing Sketches:**

**Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension ino. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom righthand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.**

**NB:**

**Versions of the Arduino Software (IDE) prior to 1.0 saved sketches with the extension pde. It is possible to open these files with version 1.0, you will be prompted to save the sketch with the ino extension on save.**

 **Verify**

**Checks your code for errors compiling it.**

 **Upload**

**Compiles your code and uploads it to the configured board. See uploading below for details.**

**Note: If you are using an external programmer with your board, you can hold down the "shift" key on your computer when using this icon. The text will change to "Upload using Programmer"**

**New**

**Creates a new sketch.**

**Open**

**Presents a menu of all the sketches in your sketchbook. Clicking one will open it within the current window overwriting its content.**

**Note: due to a bug in Java, this menu doesn't scroll; if you need to open a sketch late in the list, use the File | Sketchbook menu instead.**

**Save**

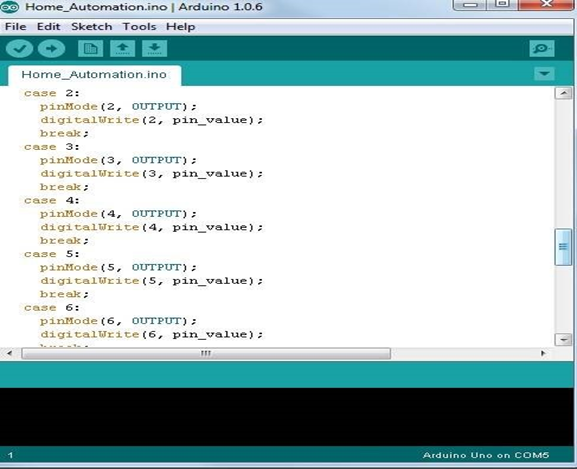
**Saves your sketch.**

**Serial Monitor**

**Opens the serial monitor.**

**Additional commands are found within the five menus: File, Edit, Sketch, Tools,and help.**

**Programming on Arduino Uno**



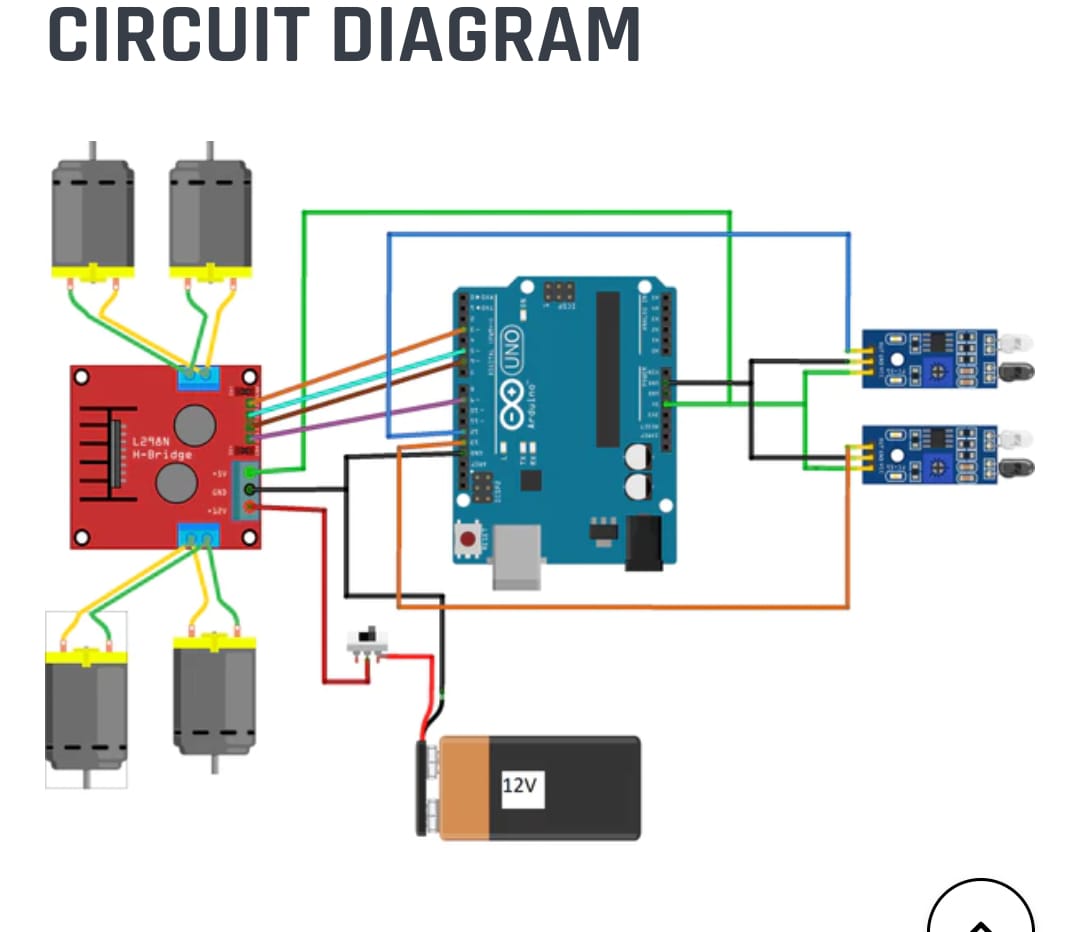
**In order for the Arduino-Uno board to be able to interact with the application used in this project certain program (code) needs to be uploaded to the Arduino-Uno.**

**CHAPTER 3**

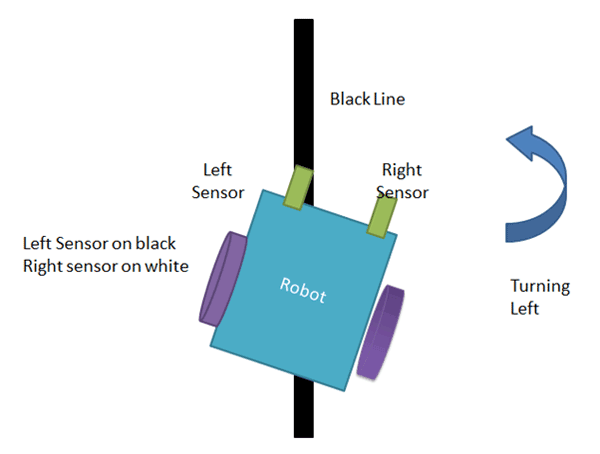
**PROJECT IMPLEMENTATION**

**3.1 Working**

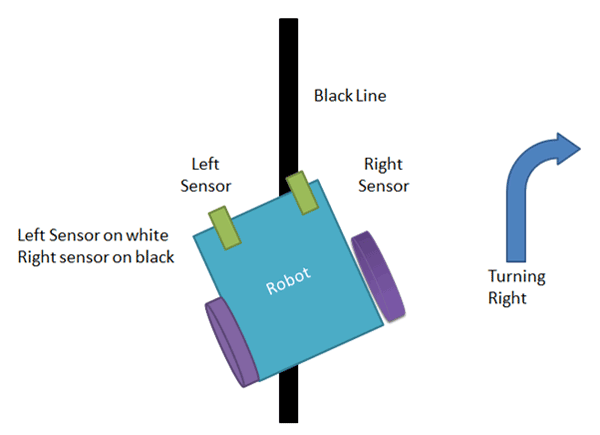
**Building a Line follower robot using Arduino is interesting. The line follower robot senses a black line by using a sensor and then sends the signal to Arduino. Then Arduino drives the motor according to sensors' output. Here in this project, we are using two IR sensor modules namely the left sensor and the right sensor. When both left and right sensor senses white then the robot moves forward.**



**If the left sensor comes on a black line, then the robot turns the left side.**



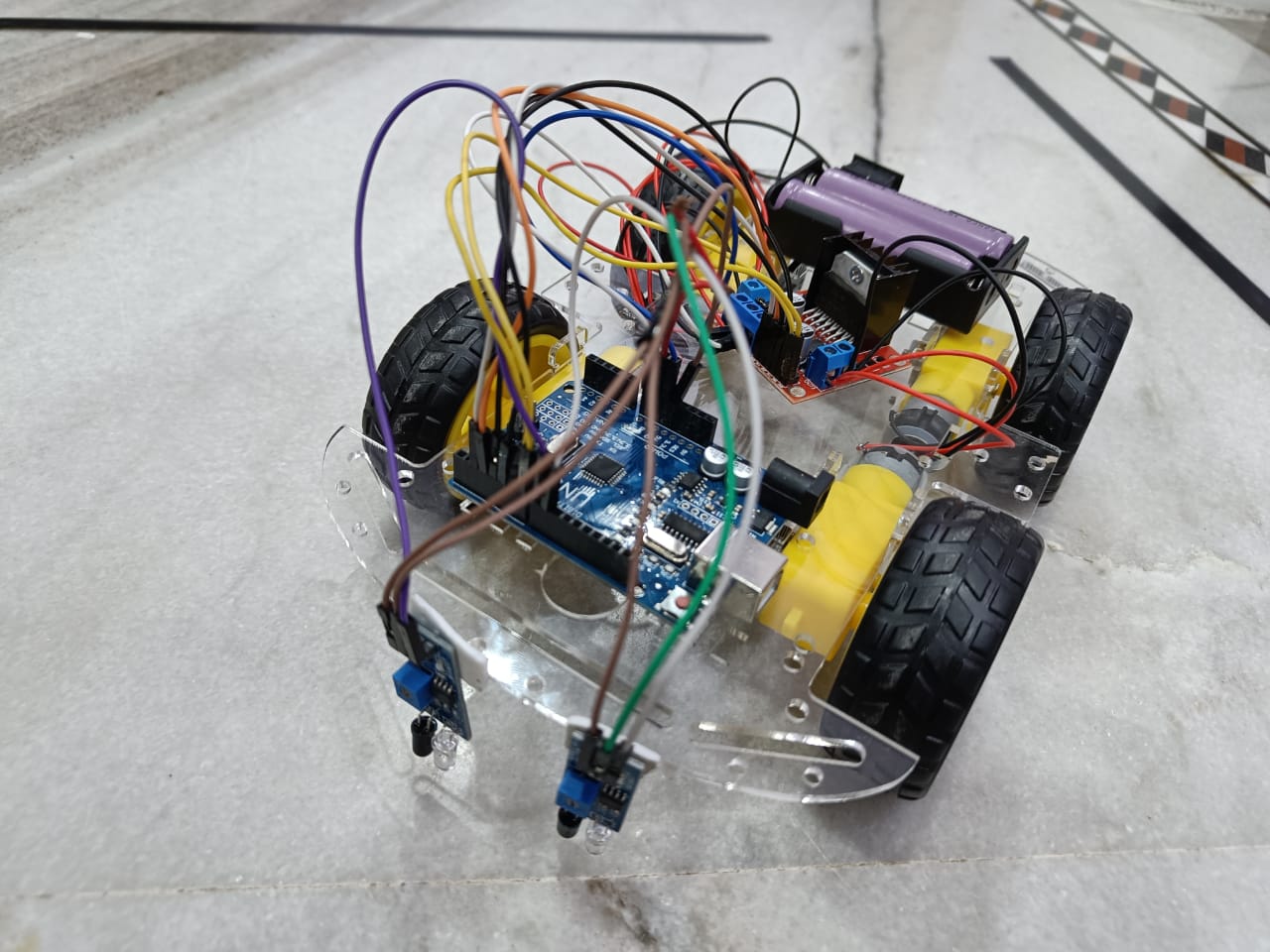
**If the right sensor sense black line, then robot turn right side until both sensors comes at the white surface. When the white surface comes robot starts moving on forward again.**



**If both sensors come on the black line, the robot stops.**

**3.2 Results**

**In result we observe that the Line follower robot works successfully to track on the black line. Above the white surface (art paper) there are some black lines in different directions. The robot still good enough to sense the line and follows the track. Also, the robot is capable to carry some load likely 500gm.**



**3.3 Advantages**

**Robot movement is automatic.**

**It is used for long distance applications.**

**Simplicity of building.**

**Fit and forget system.**

**Used in home, industrial automations etc.**

**3.4 Disadvantages**

**1.Line follower robot requires 2-3 inches broad line.**

**2.It may not move properly if the black line drawn is of low intensity.**

**3.The IR sensors may sometimes absorb IR rays from surroundings also. As a result, robots may move in improper way.**

**CHAPTER 4**

**CONCLUSION**

**4.1 Conclusion**

**Robotics has a significant role in global economy and everyday life. Another concern of robotics research is to be competitive and design patents for global industries according to their nature of applications. The demand of robotics technology is expanding in wide range of applications and human activities, especially for manufacturing, medical, service, defence, and consumer industries. The Designed robot has 2 IR sensors, Arduino microcontroller board, and L293D motor shield. Arduino mainly controls the robot to follow the line. This line follower robot is the prototype of robots for industrial use. By studying this one can build line follower robot for industrial use. Performance can be improved by using good materials and great sensing power also improves motor movement. The setup cost of line follower robot majorly depends upon the expensive machinery, land, and building and round the clock staff to maintain and use that machinery. In India where the population is humongous, resources are scarce.**

**So, it becomes difficult to set up such a capital extensive project without any financial supports from private sectors. Skilled staffs are also necessary for that. This is alternate to the existing system by replacing skilled labour with robotic machinery. This robot will be able to handle more goods in a manufacturing process in less time with better accuracy as well as lower per capital cost.**

**4.2 Future scope**

**The line follower developed is also sensing any type of obstacle in its way and can also control speed with the help of speed regulator. Further improvement can be done in the robot by using a greater number of IR sensors or an array or IR sensors.**

**BIBLIGRAPHY**

**[1]. Ahmed El Shafee and Karim Alaa Hamed (2012), “Design and Implementation of a Wi-Fi Based Home Automation System”, World Academy of Science, Engineering and Technology, Vol. 6.**

**[2]. Bader M O Al-Thabiti, Iman I MA Bosola Iman, Mahdi H M Alzahrani, Samih A Almalki and Mohamed S Soliman (2014), “Design and Implementation of Reliable Wireless Real-Time Home Automation System Based on Arduino Single Board Microcontroller”, International Journal of Control, Automation and Systems.**

**[3]. Bharani Alankar S R and Manikandan Babu S (2014), “Intelligent Home Appliance Status Intimation Control and System Using GSM”, International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 4, No. 4.**

**[4]. Naresh P Jawalkar, Vasi Ahmed, Siddharth A Ladhake and Rajesh D Thakare (2008),**

**“Micro-Controller Based Remote Monitoring Using Mobile Through Spoken Commands”, Journal of Networks, Vol. 3, No. 2**

**[5]. Prakash Kumar and Pradeep Kumar (2013), “Arduino Based Wireless Intrusion Detection Using IR Sensor and GSM”, International Journal of Computer Science and Mobile Computing, Vol. 2, no.5.**

**[6]. Rajeev Pyare (2013), “Internet of Things: Ubiquitous Home Control and Monitoring System Using Android Based Smart Phone”, International Journal of Internet of Things.**

**[7]. Rifat Shahriyar, Enamul Hoque, Sohan SM, Iftekhar Naim, Md. Mostafa Akbar and Masud Karim Khan (2008), “Remote Controlling of Home Appliances Using Mobile Telephony”, International Journal of Smart Home.**

**[8]. Shepherd R (2001), “Bluetooth Wireless Technology in the Home”, Journal of Electronics and Communication Engineering.**

**APPENDIX**

**// Arduino Line Follower Robot Code**

**#****define enA 2//Enable1 L293 Pin enA**

**#****define in1 3 //Motor****1 L293 Pin in1**

**#****define in2 4 //Motor****1 L293 Pin in1**

**#****define in3 5 //Motor****2 L293 Pin in1**

**#****define in4 6 //Motor****2 L293 Pin in1**

**#****define enB 7 //Enable2 L293 Pin enB**

**#****define R\_S 8//ir sensor Right**

**#****define L\_S 9 //ir sensor Left**

**void setup () {**

**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);**

**delay (1000);**

**}**

**void loop () {**

**if((digitalRead(R\_S) == 0) &&(digitalRead(L\_S) == 0)) {forward ()****;} //if Right Sensor and Left Sensor are at White color then it will call forward function**

**if((digitalRead(R\_S) == 1) &&(digitalRead(L\_S) == 0)) {turnRight();} //if Right Sensor is Black and Left Sensor is White then it will call turn Right function**

**if((digitalRead(R\_S) == 0) &&(digitalRead(L\_S) == 1)) {turnLeft()****;} //if Right Sensor is White and Left Sensor is Black then it will call turn Left function**

**if((digitalRead(R\_S) == 1) &&(digitalRead(L\_S) == 1)) {Stop ();} //if Right Sensor and Left Sensor are at Black color then it will call Stop function**

**}**

**void forward** **(){ //forword**

**digitalWrite(in1, HIGH); //Right Motor forword Pin**

**digitalWrite(in2, LOW****); //Right Motor backword Pin**

**digitalWrite(in3, LOW****); //Left Motor backword Pin**

**digitalWrite(in4, HIGH); //Left Motor forword Pin**

**}**

**void turnRight(****){ //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**

**}**

**void turnLeft(****){ //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**

**}**

**void** **Stop() {//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 forword Pin**

**}**