Project Title: Line Following Robot

Semester Project: Digital Logic Design

Course: BSCS-B

Teacher:

Warda Saeed

Submitted by:

Muhammad Mobeen 200901097

Hamaad Ali 200901044

Muhammad Shamaas 200901017

Abstract

Humans have always consulted the nature to find hints and solutions to their problems. So, if we want to make a robot that follows a predetermined path, we can take example from nature as well. Ants are considered the most complex and sophisticated line following organisms. Ants sensory vision is not capable of helping it to navigate so it uses another way to stay on track and never get lost. Ants have a special organ that secretes chemical trails while the ant is on the move. The ant senses these chemical trails using their antennas and tend to follow them. That is why ants are the best line following organisms and a good example for our case.

So, if we want to make a robot that navigates itself in a given area, the “Line Following” is the only feasible way for its application because it is the simplest method to implement and with the highest effectiveness.

A Line Following Robot is an autonomous robot which can follow a black line that is drawn on the surface consisting of a contrasting color. It is designed to move autonomously and follow the line. The robot uses array of optical sensors to identify the line, thus assisting the robot to stay on the track. The array of two sensors makes its movement precise and flexible. The robot is driven by four DC geared 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.

Line Following Robots are one of the most important aspects of robotics. Line Following Robots have massive applications in today’s world. They can be used to replace human work force for simple tasks like logistics from point A to B in a predefined environment. These robots are highly applicable in today’s industrial applications.

Building a Line Following Robot is usually the first step towards robotics. So, we wanted to experience the process of building a fully autonomous robot and to learn the required skills along the way of its development. The process of building it involved a lot of technical skills like the choosing the best components for the project, getting familiar with Arduino Developing Environment, planning the structure of the project etc.

Our project is a perfect or near perfect mimic of nature that also solves a problem. 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.

List of Figures:

Table of Contents

Contents

Chapter 1: Introduction

1.1 Overview of Project

The project is Arduino based which means every input will be received by the Arduino Board for processing and then the outputs will be forwarded to L293D motor driver shield which will drive individual motors according to the logic defined in the code. Following are the details about the Inputs and Outputs.

Input: The two IR Sensors are placed in front of the robot car in such a configuration that the sensors always faces downwards so that the line can be detected. These IR sensors provide the Arduino with either “0” or “1” logics depending upon the scenario. Following table defines the different scenarios.

Input	Output
If the sensor detects a bright surface.	**High**
If the sensor detects the line (dark surface).	**Low**

Processing: Arduino receives the output from the two IR sensors and performs the indicated operations according to the code and then sends the output to the L293D motor driver shield.

Output: The L293D motor driver shield drives the individual motors according to the logic sent by the Arduino. The logic can make the motors reverse their directions and also change their speeds. All this is done by the L293D motor driver shield.

1.2 Block Diagram of Complete System (without using ICs, just use simple blocks)

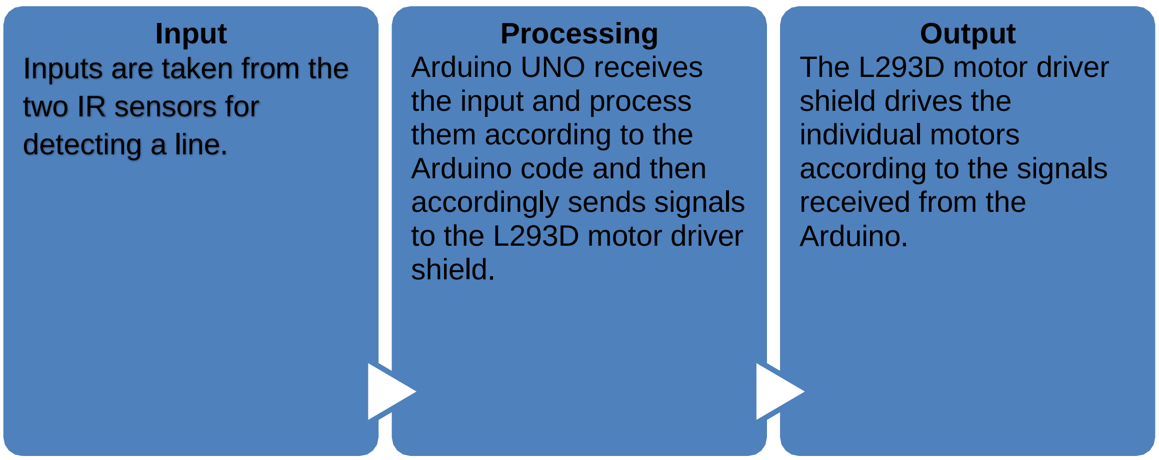

Figure 1: Block Diagram

1. Clear Work Division

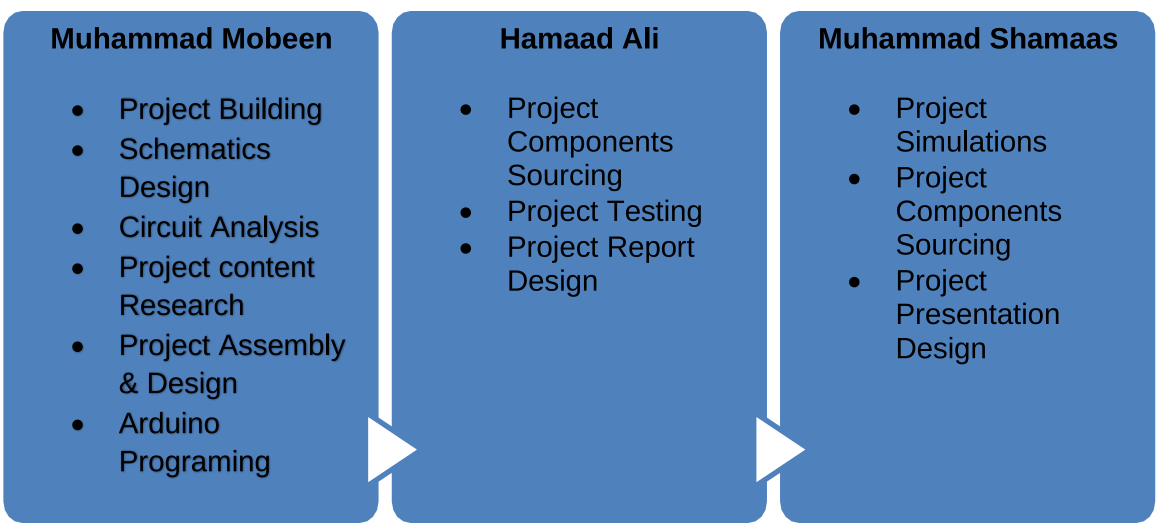

Figure 2: Work Division

Chapter 2: Design

2.1 Problem Statement

The line follower is a self-operating robot that detects and follows a line that is drawn on the floor. The track consists of a black path drawn on a white surface (contrasting colors). To make the robot follow a line we need to have a line detecting system. For that we can use a combination of IR-LED and Photodiode called an Optical sensor. This makes line detection process of high resolution and high robustness.

Robot requires steering and maneuvering mechanism for tracking the line, for that we are using four geared motors mounted on either side of the car chassis. These motors can provide us full 180˚ of rotation.

The brain of the robot is Arduino UNO. All the decision processes are handled by the Arduino. We are choosing Arduino over using just simple microcontroller IC's because Arduino provides the flexibility to manipulate things easily, which makes prototyping the project a lot easier and we can easily make changes by editing the code in Arduino IDE if we want to.

To drive the motors, we need a high voltage supply between 7V – 12V and for that we are using two 18650 Li-ion cells. Each cell output voltage is around 3.7V, since we are using two cells it will give us voltage of around 7.4V.

The problem is Arduino cannot handle voltages above +5V so to drive our four motors we need a separate motor driving system. Motor driver is a current enhancing device; it can also act as a Switching Device. For our project we are using L293D motor driver IC's which can derive 2 motors simultaneously, since our project requires to drive four motors so instead, we will be using a L293D motor driver shield that is pre-equipped with two L293D IC's and also voltage regulators to supply +5V to the Arduino UNO. It is specifically made to be mounted on top of Arduino UNO.

2.2 Truth Table / State Diagram

Analog Inputs:-

A0: Left IR Sensor input.

A1: Right IR Sensor input.

Input		Output						State In	Action
Left	Right	Digital PWM		Left Motors		Right Motors			
A0	A1	Pin-9	Pin-7	Motor 1	Motor 2	Motor 3	Motor 4		
0	0	1	1	**Low**	**Low**	**Low**	**Low**	Both sensors are on the track	**Stop all motors**
0	1	1	0	**High (Backward)**	**High (Backward)**	**High (Forward)**	**High (Forward)**	Only left sensor is detecting track	**Sharp turn left**
1	0	0	1	**High (Forward)**	**High (Forward)**	**High (Backward)**	**High (Backward)**	Only right sensor is detecting track	**Sharp turn right**
1	1	0	0	**High (Forward)**	**High (Forward)**	**High (Forward)**	**High (Forward)**	Both sensors are not detecting the track	**Go straight**

Figure 3: Truth Table for Arduino Logic

2.3 Simplification of Functions / K-Maps & Equations

For our understanding we can make K-Maps for each action of the robot. Similarly, we can derive equations for each action made by the robot. Following is the representation of it.

Forward Function:-

K-Map:-

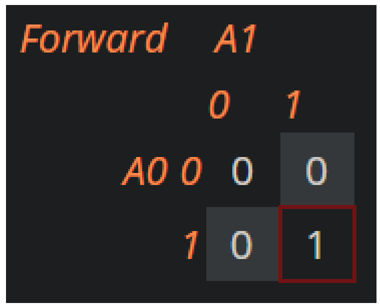

Figure 4.1: K-Map Forward Function

Simplified Equation:-

Forward(A0, A1) = A0A1

Right Function:-

K-Map:-

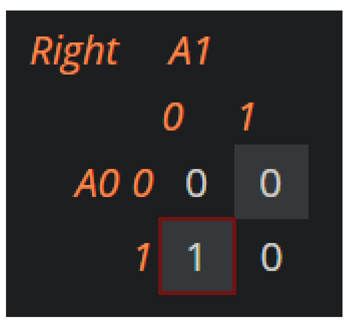

Figure 4.2: K-Map Right Function

Simplified Equation:-

Right(A0, A1) = A0A1'

Left Function:-

K-Map:-

Figure 4.3: K-Map Left Function

Simplified Equation:-

Left(A0, A1) = A0'A1

Stop Function:-

K-Map:-

Figure 4.4: K-Map Stop Function

Simplified Equation:-

Stop(A0, A1) = A0'A1'

2.4 Complete Logic Diagram

Logic Diagram generated in Proteus Software.

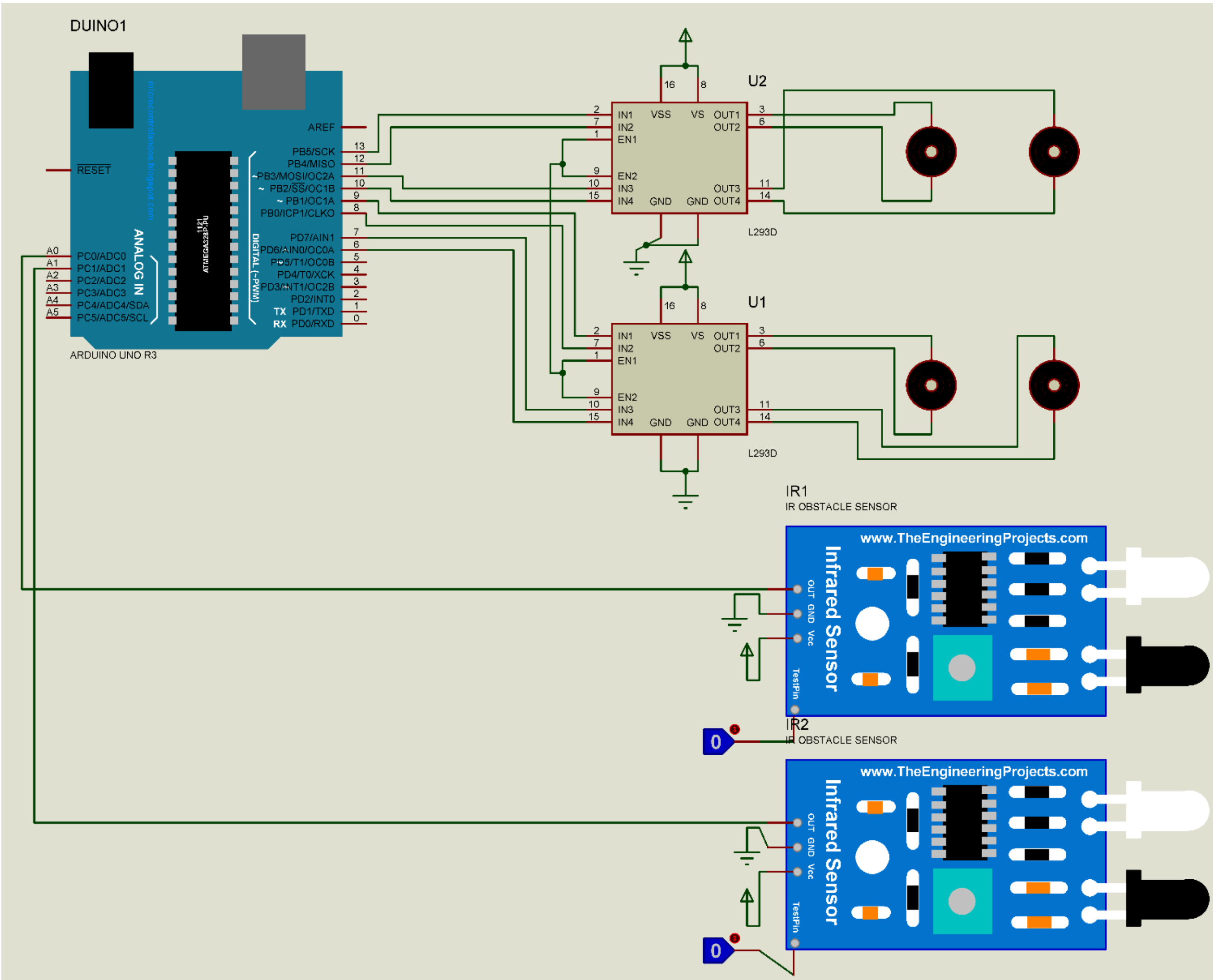

Figure 5: Complete Logic Diagram

2.5 Simulation

<u>Illustration of simple functioning of robot:-</u>

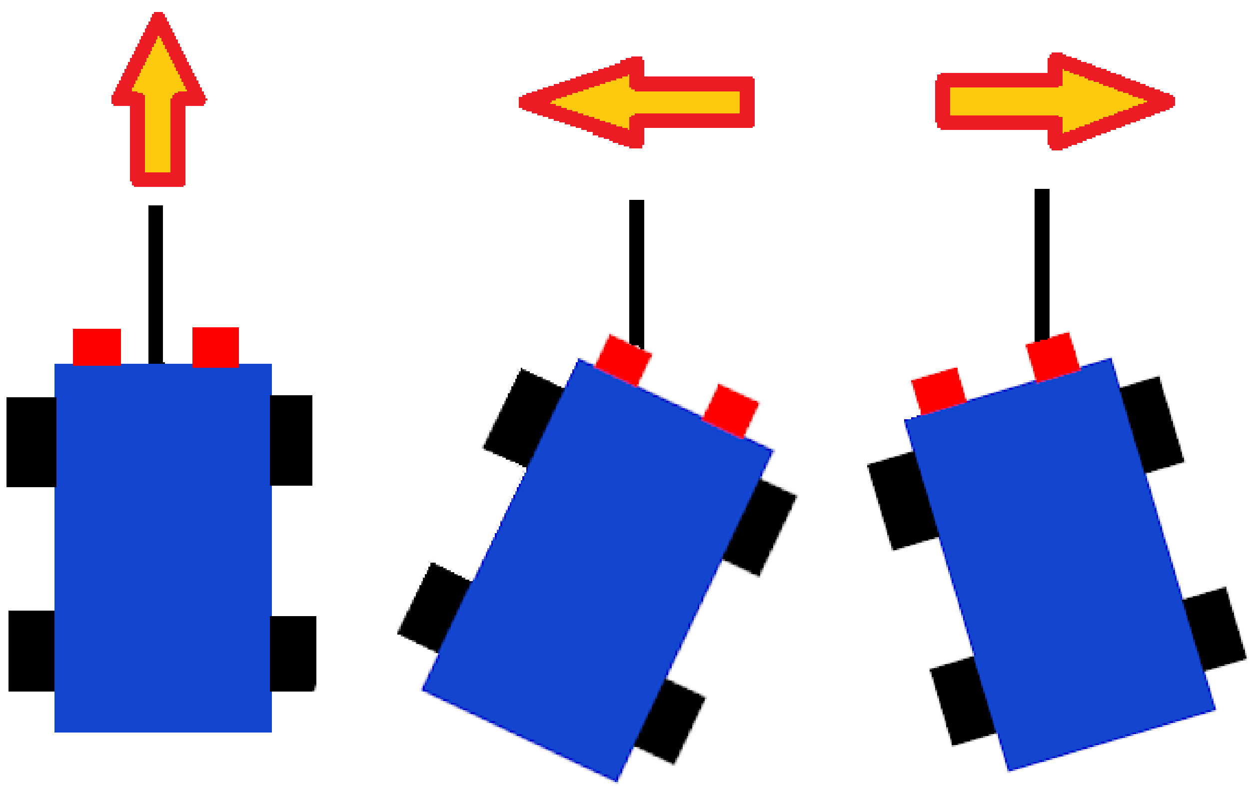

Figure 6: Illustration of Functioning Robot

<u>Key:-</u>

IR Sensors

Line to be Followed.

When both the sensors do not detect the line:-

All the motors run forwards.

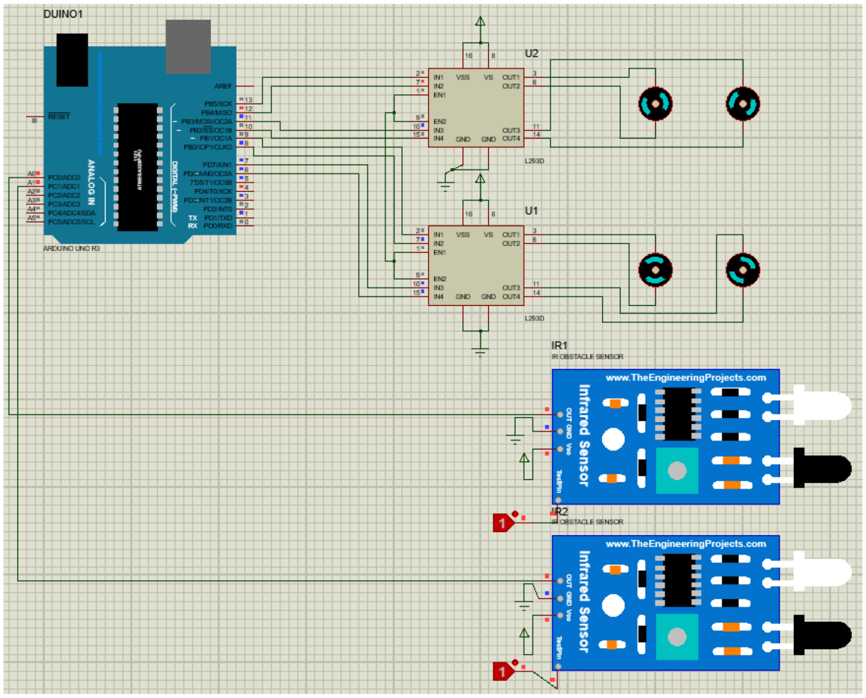

Figure 7.1: When both the sensors do not detect the line

When only Left sensor detects the line:-

Motors on the right run's forwards but the motors on the left runs backwards.

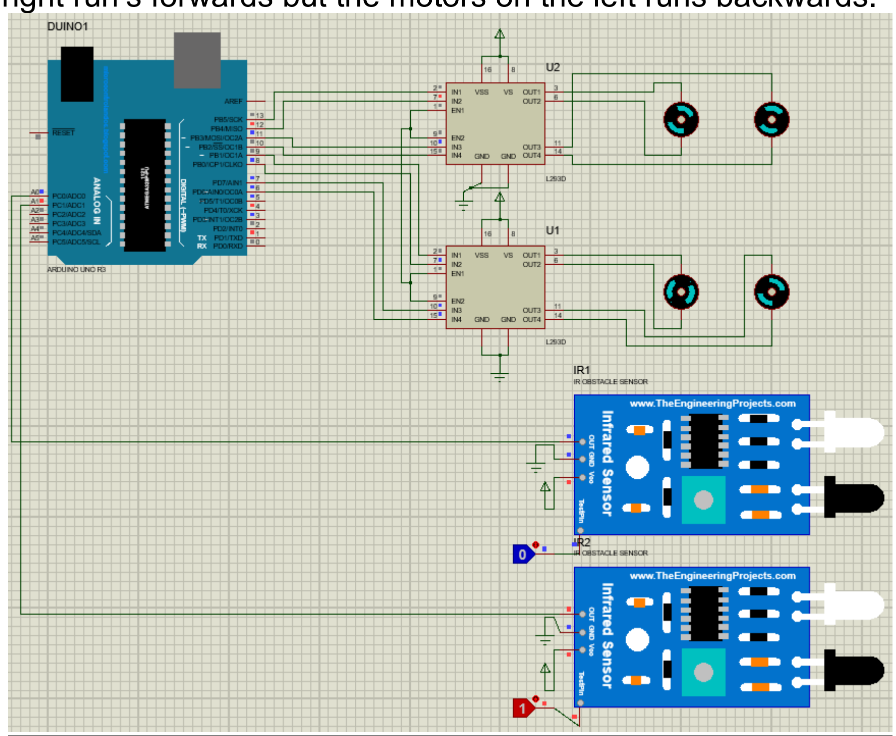

Figure 7.2: When only Left sensor detects the line

When only Right sensor detects the line:-

Motors on the left run’s forwards but the motors on the right runs backwards.

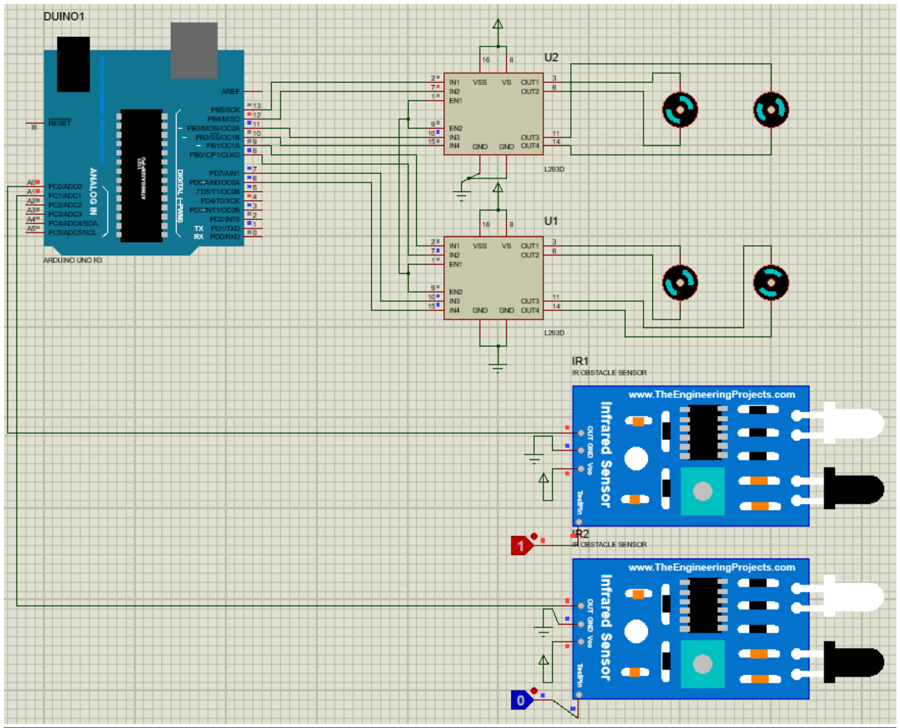

Figure 7.3: When only Right sensor detects the line

When both the sensors detect the line:-

All the motors stop working.

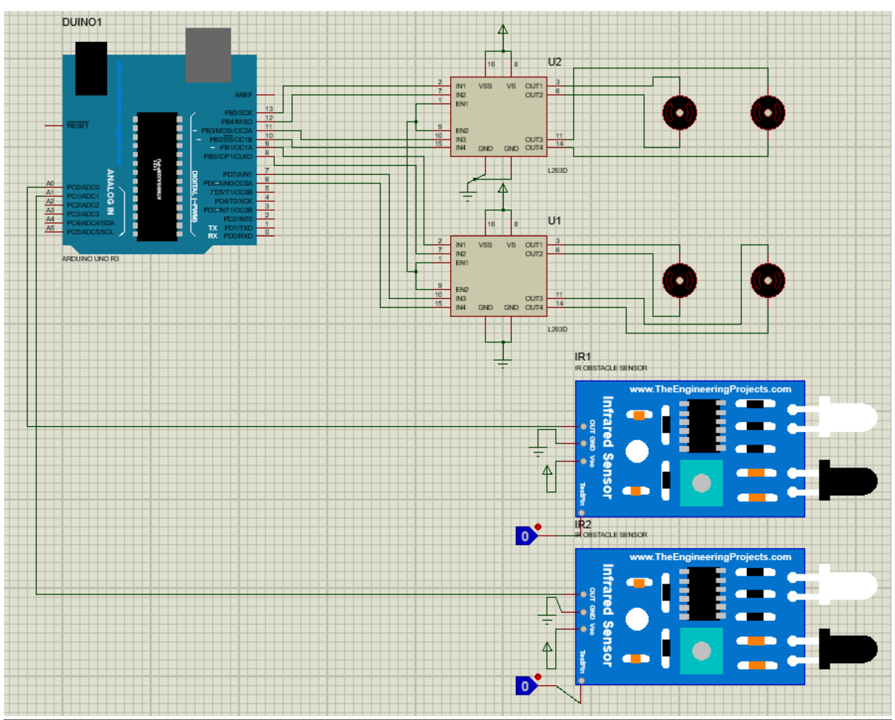

Figure 7.4: When both the sensors detect the line

2.6 Detailed Schematic of Design and its Description

Schematic for Arduino UNO:-

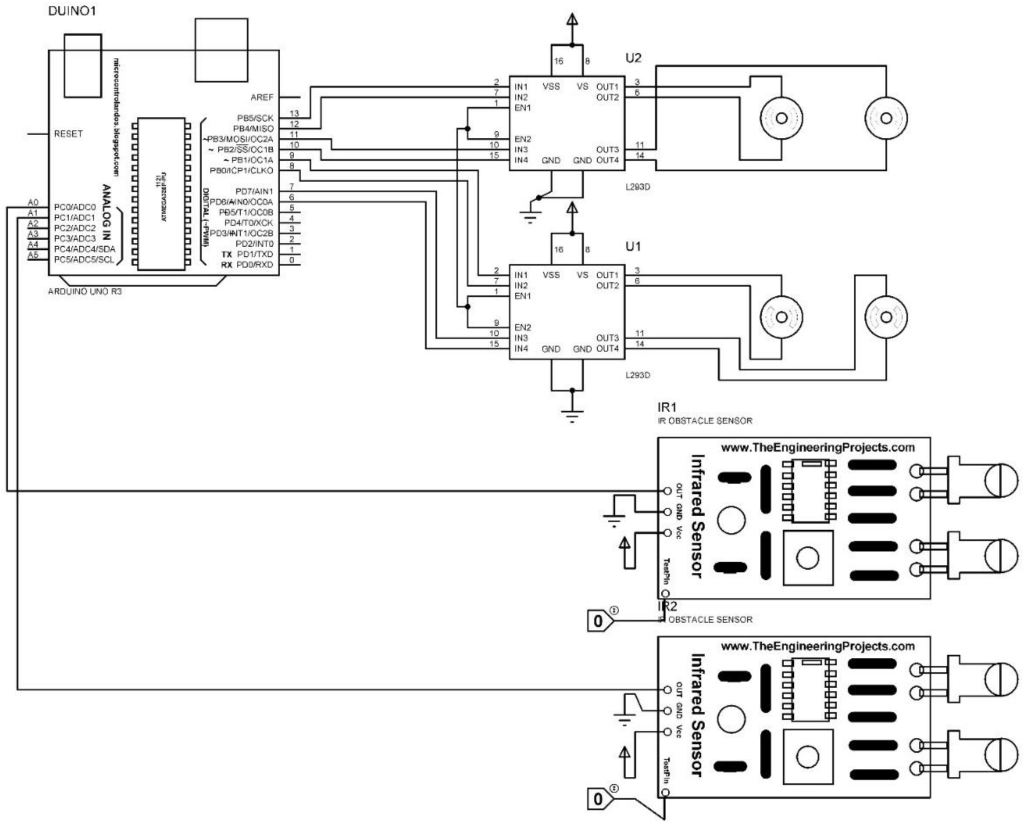

Figure 8: Schematic for Arduino UNO

Schematic for L293D Motor Driver Shield and External Components:-

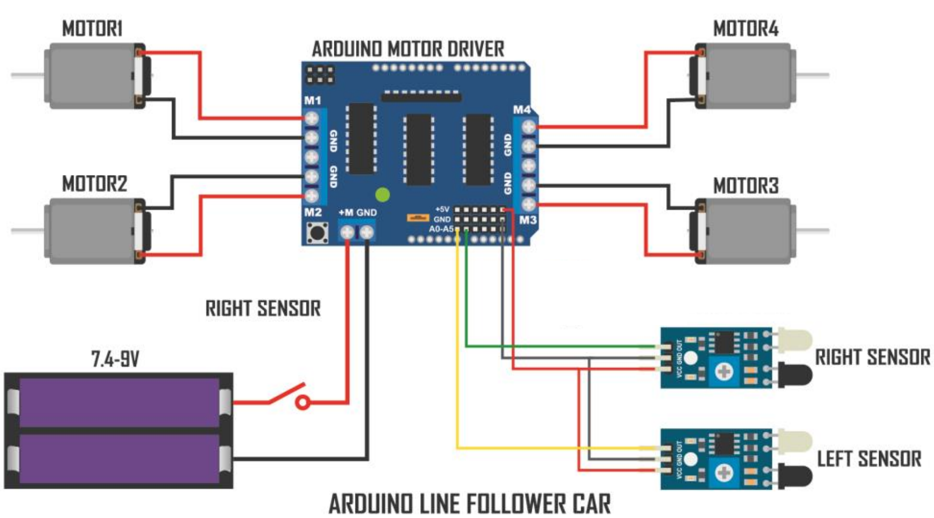

Figure 9: Schematic for L293D Motor Driver Shield and External Components

2.7 Details of components used:

- ### Arduino UNO

Function Table

Pins	Functions
Serial: 0 (RX) and 1 (TX).	Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL serial chip.
External Interrupts: 2 and 3.	Configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
PWM: 3, 5, 6, 9, 10 and 11.	Provide 8-bit PWM output with the analogueWrite() function
SPI:10(SS), 11(MOSI), 12 (MISO), 13 (SCK).	These pins support SPI communication using the SPI library.
LED: 13	There is a built-in LED connected to digital pin13. HIGH pin value - LED on, LOW pin value - LED off.
TWI	A4 or SDA pin and A5 or SCL pin. Support communication using the wire library
AREF	Reference voltage for the analogue inputs. Used with analogueReference().
Reset	To reset the microcontroller

Figure 10: Function Table for Arduino

Schematic

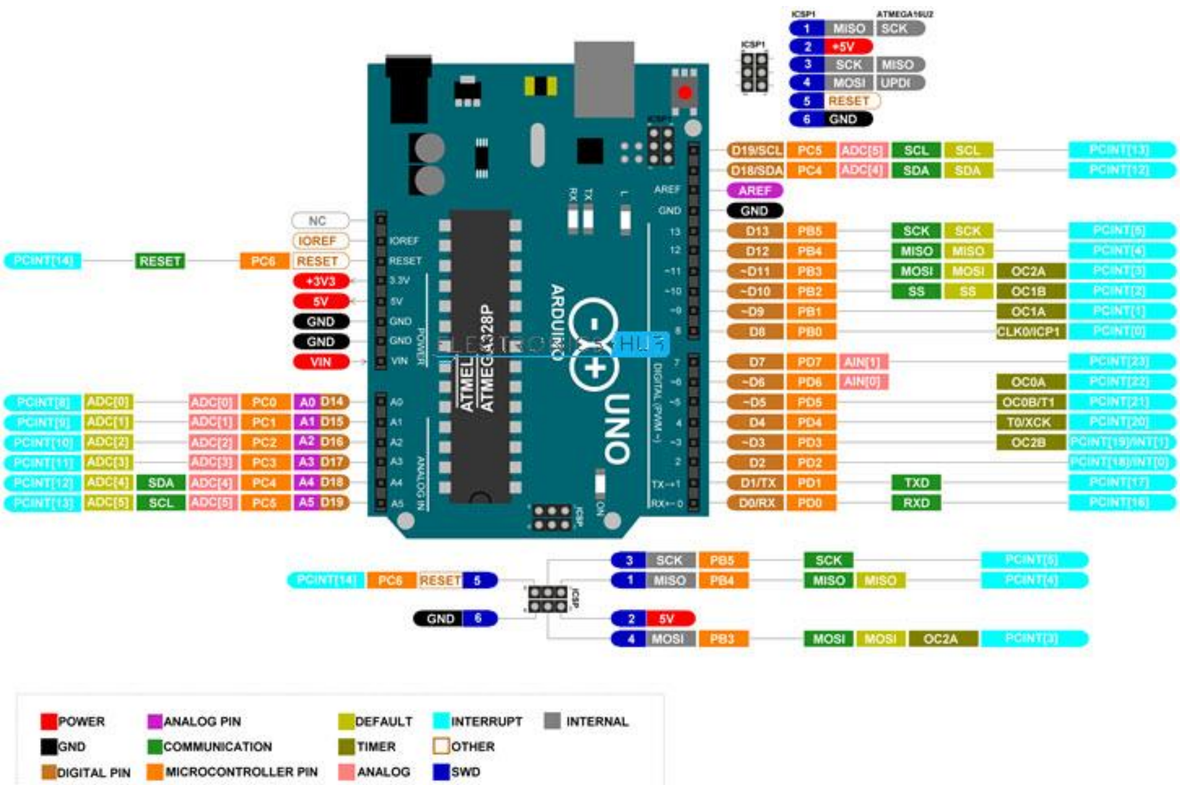

Figure 11: Schematic for Arduino

- **L293D Driver IC**

Function Table

Pin 1	Pin 2	Pin 7	Function
High	High	Low	Turn Anti-clockwise (Reverse)
High	Low	High	Turn clockwise (Forward)
High	High	High	Stop
High	Low	Low	Stop
Low	X	X	Stop

High ~+5V, Low ~0V, X=Either high or low (don't care)

Figure 12: Function Table for L293D Driver IC

Schematic

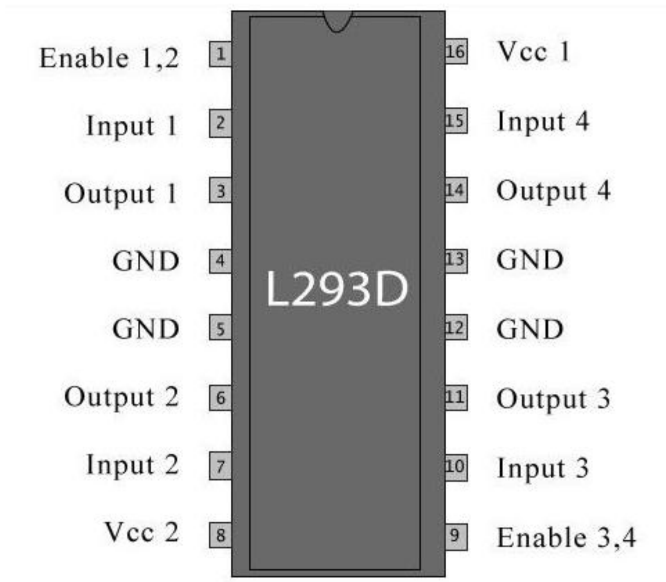

Figure 13: Schematic for L293D Driver IC

2.8 Details of Other Components used like diodes, transistors, resistors etc.

The main features incorporated into the hardware are given below:

- Arduino Uno.
- The IR-LED with IR illuminance modules.
- The H-bridge motor control shield (L293D)
- DC Motors, with coupled reduction gears.
- Connectors to join the different boards to form one functional device.
- Arduino Code, coded in Arduino IDE.

Optical Sensors;-

The robot uses photodiode sensors to sense the line; an array of two IR- LEDs (TX) and Photodiode sensors (Rx), facing the ground used in this setup. An analog signal is obtained in output, depends on the amount of light reflected back, which is provided to the comparator to produce 0s and 1s which are then fed to the Arduino.

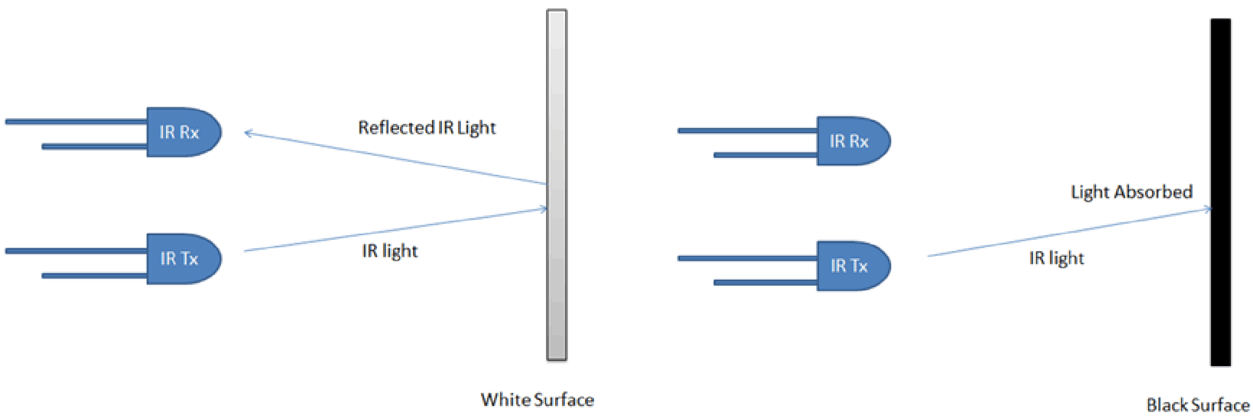

Figure 14: Illustration of working of Optical Sensors

Arduino:-

Arduino is an open-source computer hardware and software company, project and user community that designs and manufactures microcontroller-based kits for building digital devices and interactive objects that can sense and control objects in the physical world. The heart of Arduino is the microcontroller. For Arduino Uno ATmega328 is used. It has specification of 8-bit CPU, 16 MHZ clock speed, 2 KB SRAM 32 KB flash Memory, 1 KB EEPROM [2].

Features:-

- 14 digital input output pins (3,5,6,9,10 and 11 pins are able to generate PWM).
- 6 analog input pins
- Voltage input from the 7 – 12 V

Motor Driver:-

Motor driver is a current enhancing device; it can also be act as Switching Device. Thus, after inserting motor driver among the motor and microcontroller. Motor driver taking the input signals from microcontroller and generate corresponding output for motor.

IC L293D:-

This is a motor driver IC that can drive two motor simultaneously. Supply voltage (Vss) is the voltage at which motor drive. Generally, 6V for dc motor and 6 to 12V for gear motor are used, depending upon the rating of the motor. Logical Supply Voltage deciding what value of input voltage should be considered as high or low .So if the logical supply voltage equals to +5V, then -0.3V to 1.5V will be considered as Input low voltage and 2.3V to 5V is taken into consider as Input High Voltage. The Enable 1 and Enable 2 are the input pin for the PWM led speed control for the motor L293D has 2 Channels. One channel is used for one motor.

DC Motors:-

Motor is a device that converts any form of energy into mechanical energy or imparts motion. In constructing a robot, motor usually plays an important role by giving movement to the robot. In general, motor operating with the effect of conductor with current and the permanent magnetic field. The conductor with current usually producing magnetic field that will react with the magnetic field produces by the permanent magnet to make the motor rotate. There are generally three basic types of motor, DC motor, even servomotor and stepper motor, which are always being used in building a robot. DC motors are most easy for controlling. One DC motor has two signals for its operation. Reversing the polarity of the power supply across it can change the direction required. Speed can be varied by varying the voltage across motor.

Software Required:-

For the simulation of the circuit, Proteus 8.1® software is used. For coding and uploading the sketch, the Arduino 1.8.13 ® is used.

Arduino Sketch Code

```
#include <AFMotor.h>

//defining pins and variables
#define left A0
#define right A1

//defining motors
AF_DCMotor motor1(1, MOTOR12_1KHZ);
AF_DCMotor motor2(2, MOTOR12_1KHZ);
AF_DCMotor motor3(3, MOTOR34_1KHZ);
AF_DCMotor motor4(4, MOTOR34_1KHZ);

void setup() {
  //declaring pin types
  pinMode(left,INPUT);
  pinMode(right,INPUT);

  //begin serial communication
  Serial.begin(9600);

}

void loop(){
  //printing values of the sensors to the serial monitor
  //Serial.println(digitalRead(left));

  //Serial.println(digitalRead(right));

  //No line detected by both sensors
  if(digitalRead(left)==0 && digitalRead(right)==0){
    //Forward
    Serial.print("\nForward");
    motor1.run(FORWARD);
    motor1.setSpeed(150);
    motor2.run(FORWARD);
    motor2.setSpeed(150);
    motor3.run(FORWARD);
    motor3.setSpeed(150);
    motor4.run(FORWARD);
    motor4.setSpeed(150);
  }
```

```
  //line detected by right sensor
  else if(digitalRead(left)==0 && !analogRead(right)==0){
    //turn right
    Serial.print("\nRight");
    motor1.run(FORWARD);
    motor1.setSpeed(200);
    motor2.run(FORWARD);
    motor2.setSpeed(200);
    motor3.run(BACKWARD);
    motor3.setSpeed(200);
    motor4.run(BACKWARD);
    motor4.setSpeed(200);

  }

  //line detected by left sensor
  else if(!digitalRead(left)==0 && digitalRead(right)==0){
    //turn left
    Serial.print("\nLeft");
    motor1.run(BACKWARD);
    motor1.setSpeed(200);
    motor2.run(BACKWARD);
    motor2.setSpeed(200);
    motor3.run(FORWARD);
    motor3.setSpeed(200);
    motor4.run(FORWARD);
    motor4.setSpeed(200);

  }

  //line detected by both sensors
  else if(!digitalRead(left)==0 && !digitalRead(right)==0){
    //stop
    Serial.print("\nStop");
    motor1.run(RELEASE);
    motor1.setSpeed(0);
    motor2.run(RELEASE);
    motor2.setSpeed(0);
    motor3.run(RELEASE);
    motor3.setSpeed(0);
    motor4.run(RELEASE);
    motor4.setSpeed(0);

  }

}
```

2.9 Simulation Issues / Results/ Observations

Results:-
The robot is now fully prepared for its real-world applications. It successfully detects the line and also manages to correct itself autonomously in most situations. The robot successfully follows the path as long as the path is not obstructed or tempered with.

The line following robot project challenged the group to cooperate, communicate, and expand understanding of electronics, mechanical systems, and their integration with programming. The successful completion of every task demonstrated the potential of mechatronic systems and a positive group dynamic.

Observations:-
Following are some important observations that we observed during project development.

- We needed to have a good understanding of the Digital PWM outputs on the Arduino UNO so that we can understand the speed control system on the machine level.
- A lot of code bugs were fixed during prototyping.
- Motors have their own dedicated libraries which needed to be included in the Arduino code.
- Soldering a lot of components required. The hardest soldering task was to solder input, Vcc and Ground pins on to the L293D Motor driver shield.

Issues:-
A lot of the issues were face during project development. Some of them are listed below.

- First problem was that the Arduino UNO board was not working with the Arduino IDE, so we had to install some custom drivers to enable it to function properly.
- A lot of soldering was required on the PCB boards. Soldering is generally not a major issue but soldering on the PCB boards through THS technique (Through Hole Soldering) is a tiresome task as many things can get wrong.
- Another issue we had was that after completing the project all the motors were not working properly and the ICs were becoming so hot that they were close to get fried. Later we found that one of the motor outputs were short and was earthing the whole system. We solved that problem with insulating the shield terminals.
- Another issue we had while the simulation. Proteus does not contain all the necessary libraries for our components. So, we had to use some third-party libraries and some bypasses to properly simulate our project on proteus.

Chapter 3: Project Applications

This project has many applications in our modern world were we. Our world is now very fast paced and because of the rising capitalism and consumerism the demand for products from the Industries are ever growing. For this industrialist tries different methods that can improve their working efficiency while keeping the production line fast paced. In this regard robots are a good alternative for human labor because robots are efficient and generally a less liability for industrial companies.

Other than the industrial applications, it also has vast applications in Medical, Military and even for City management and house management systems. Following are some major applications in various fields.

- **Industrial Applications**: These robots can be used as automated equipment carriers in industries replacing traditional conveyer belts and worker trollies.
- **Automobile applications**: These robots can also be used as automatic cars running on roads with embedded magnets.
- **Domestic applications**: These can also be used at homes for domestic purposes like floor cleaning etc.
- **Guidance applications**: These can be used in public places like shopping malls, museums etc. to provide path guidance.
- **Medical Applications**: It can be used in Hospitals to transport supplies within the hospitals. Robots can increase the efficiency of the hospital by this way.

Chapter 4: Future Recommendations

Following are my future recommendations for improving the project:-

1. We can increase the number of IR Sensors to increase the accuracy of Line Following Function.
2. Right now, the robot is incapable of avoiding obstructions in the path. We can add an Ultrasonic Sensor mounted on the front of the robot so that object avoidance function can be added.
3. Another thing we can do is to use a camera for detecting the line and also object avoidance. For this we need to use Raspberry Pi instead of Arduino because Arduino does not support Camera modules and only Raspberry Pi can.
4. We can add different functions like Race mode, Hauling mode etc. to suit for the given conditions.

References / Bibliography

- Kumar Jha, Sujeet & Karn, Manish & Khan, Ahmed. (2016). PROJECT LINE FOLLOWING ROBOT. 10.13140/RG.2.2.21215.79522.
- Takiul, Md & Sakib, Md Takiul Hasan. (2019). Line Following Robot Using Arduino.
- Punetha, Deepak & Kumar, Neeraj & Mehta, Vartika. (2013). Development and Applications of Line Following Robot Based Health Care Management System. International Journal of Advanced Research in Computer Engineering & Technology (IJARCET_. 2. 2446-2450.
- https://forbiddenbit.com/en/arduino-projects/line-flower-arduino/
- https://create.arduino.cc/projecthub/electropeak/arduino-l293d-motor-driver-shield-tutorial-c1ac9b
- https://lastminuteengineers.com/l293d-motor-driver-shield-arduino-tutorial/
- https://www.arduino.cc/reference/en/language/functions/analog-io/analogread/
- https://www.arduino.cc/reference/en/language/functions/digital-io/digitalread/
- https://www.youtube.com/watch?v=t7k9D1jDEtk
- https://www.youtube.com/watch?v=WSMFLkL-niY
- https://www.youtube.com/watch?v=HW1VTOLAXgE
- https://www.youtube.com/watch?v=Fp37DPZVdRI
- https://www.researchgate.net/publication/277205656_Development_and_Applications_of_Line_Following_Robot_Based_Health_Care_Management_System
- https://www.researchgate.net/publication/327814718_PROJECT_REPORT_LINE_FOLLOWING_ROBOT
- https://www.researchgate.net/publication/337934605_Line_Following_Robot_Using_Arduino