

LINE FOLLOWER REMOTE CONTROL

Submitted in partial fulfilment of the requirements for the award of

Bachelor of Engineering in

Electronics and Communication Engineering

By Batch 107

AVULA GOPINADH (38130022)

A RAM MANOJ (38130005)



SCHOOL OF ELECTRICAL AND ELECTRONICS ENGINEERING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A” by NAAC

**JEPPIAAR NAGAR, RAJIV GANDHI
SALAICHENNAI – 600119**

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION
ENGINEERING**



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

BONAFIDE CERTIFICATE

This is to certify that Project Report is the bonafide work of "**Batch 107**" who carried out the Project entitled "**LINE FOLLOWING AND REMOTE CONTROLLER**".

Under the guidance of

[Dr R Narmada](#)

[M.E., Ph.D., during the period December 10 2021 to march 9 2022.](#)

INTERNAL GUIDE

Dr. R Narmada M.E PHD

HEAD OF THE DEPARTMENT

Dr. T. RAVI, M.E, PHD

Submitted for Viva Voce held on

Internal Examiner

External Examiner

DECLARATION

We Mr. AVULA GOPINADH (38130022)

And Mr A RAM MANOJ (38130005)

Hereby declare that the Project report entitled **LINE FOLLOWING REMOTE CONTROL** is done under the guidance of **DR R Narmada M.E., Ph.D.**, is submitted in partial fulfilment of the

requirements for the award of Bachelor of Engineering degree in **ELECTRONICS AND COMMUNICATION ENGINEERING.**

DATE: 05-05-2022

PLACE: CHENNAI

Signature of
(AVULA GOPINADH)
(A RAM MANOJ)

ACKNOWLEDGEMENT

I pleased to acknowledge my sincere thanks to board management of SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY for their kind

encouragement in doing this PROJECT and for completing it successfully. I am grateful to them.

I convey my thanks to Dr. Rekha

I convey my thanks to **Dr. Rekha Chakravarthy, M.E., Ph.D.** school of Electrical and Electronics engineering and **Dr. T. RAVI, M.E., Ph.D.**, Head of the Department, Electronics and Communication engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr R Narmada M.E., Ph.D.**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my Project.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of Electronics and Communication Engineering who were helpful in many ways for the completion of the project.

ABSTRACT

The Internet of Things (IoT):

The Internet of Things is the ability to have devices communicate with one another via the internet or the other networks, remotely tracking to provide feedback to assist with decision making for commercial, industrial and residential purpose. This is commonly done using sensors connecting to a back-to-base system.

The Internet of Things (IoT) is the network of physical objects or "things" embedded with electronics, software, sensors, and network connectivity, which enables these objects to collect and exchange data.

"Things," in the IoT sense, can refer to a wide variety of devices such as heart monitoring implants, biochip transponders on farm animals, electric clams in coastal waters, automobiles with built-in sensors, DNA analysis devices for environmental/food/pathogen monitoring or field operation devices that assist fire-fighters in search and rescue operations.

As the name suggests, the line follower robot is an automated vehicle that follows a visual line embedded on the surface. This visual line is a path on which the line follower robot runs. Generally, it uses a black line on a white surface, or you can adjust it as a white line on a black surface

In industries, giant line follower robots are used for assisting the automated production process. They are also used in military applications, human assistance purposes, delivery services, etc.

CONTENTS:

1.INTRODUCTION

1.1 Introduction

1.2 Pins

1.3 Pin diagram

2. Literature Survey

2.1 LITERATURE SURVEY

3.PROPOSED METHODOLOGY

3.1 EXISTING SYSTEM

3.2 PROPESED SYSTEM

3.3 BLOCK DIAGRAM

3.4 HARDWARE REQUIREMENTS

3.5 SOFTWARE REQUIREMENTS

4.HARDWARE DETAILS

4.1 ARDUINO UNO

4.2 IR SENSORS

4.3	BLUETOOTH MODULE
5.	SOFTWARE DETAILS
5.1	EMBEDDED C++
5.2	ARDUINO IDE
5.3	ADA FRUIT LIBRARY
6.	CODE
7.	RESULT AND ANALYSIS
8.	CONCLUSION
9.	REFERENCES

1.INTRODUCTION:

Embedded System

. An embedded system can be thought of as a computer hardware system having software embedded in it. An embedded system can be an independent system or it can be a part of a large

system. An embedded system is a microcontroller or microprocessor based system which is

designed to perform a specific task. For example, a fire alarm is an embedded system; it will

sense only smoke.

An embedded system has three components –

- ☐ It has hardware.
- ☐ It has application software.

□ It has Real Time Operating system (RTOS) that supervises the application software and provide mechanism to let the processor run a process as per scheduling by following a plan to control the latencies. RTOS defines the way the system works. It sets the rules during the execution of application program. A smallscale embedded system may not have RTOS.

So we can define an embedded system as a Microcontroller based, software driven, reliable, real-time control system.

Arduino/Genuine Uno : is a **microcontroller board based on the ATmega328P** (datasheet). It has 14 digital input/output pins (of which 6 can be

used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button.

Microcontroller: ATmega328P

Digital I/O Pins: 14 (of which 6 provide PWM o...

Analog Input Pins: 6

Input Voltage (limit): 6-20V

Pins:

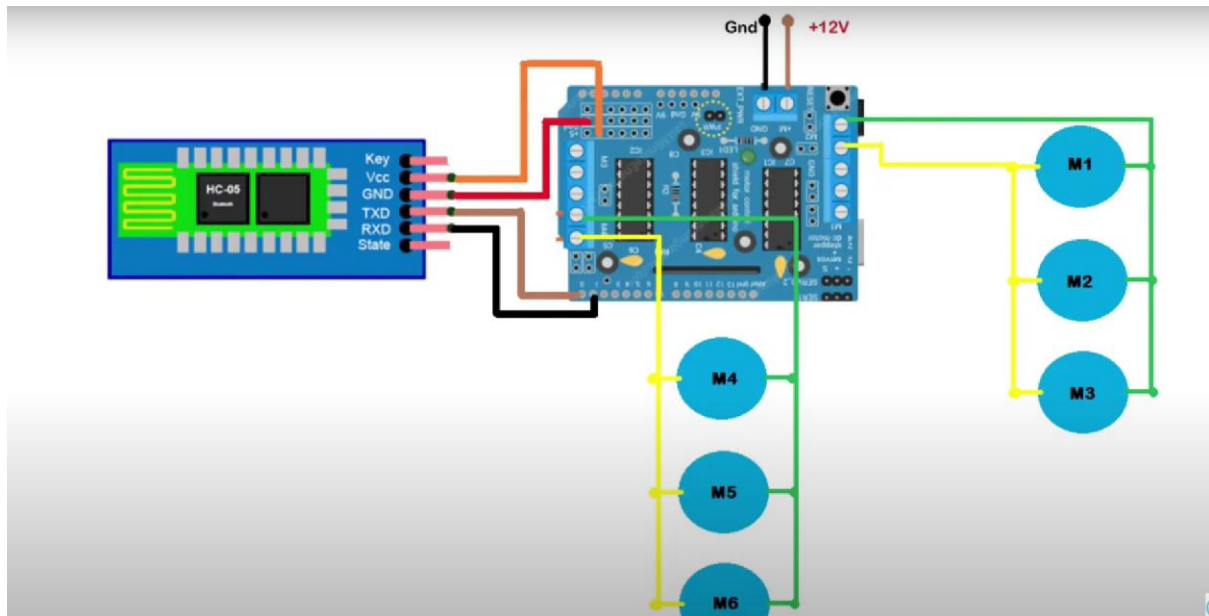
NodeMCU provides access to the GPIO (General Purpose Input/Output) and a pin mapping table is part of the API documentation.

I/O index	ESP8266 pin
0 [*]	GPIO16
1	GPIO5
2	GPIO4
3	GPIO0
4	GPIO2
5	GPIO14
6	GPIO12
7	GPIO13
8	GPIO15
9	GPIO3
10	GPIO1
11	GPIO9

12	GPIO10
----	--------

[*] D0 (GPIO16) can only be used for GPIO read/write. It does not support open-drain/interrupt/PWM/I²C or 1-Wire.

Pin Diagram:



LITERATURE SURVEY

TITLE/AUTHOR	YEAR OF PUBLICATION	METHODOLOGY	INFERENCE
1. O Gumus	2020	computer controlled line follower robots in public transport	Fuzzy Systems and Soft Computing
2. M. B. Bhuiya	2019	Controlling line follower robot with the remote web server”	Proceedings of SAI Intelligent Systems Conference
3. M. Engin and D. Engin	2021	Path planning of line follower robot” Proceedings	European DSP Education and Research Conference
4. K. M. Hasan, A. AI-Nahid	2018	Implementation of autonomous line follower robot	International Conference on Informatics, Electronics & Vision

EXISTING TECHNOLOGY

- computer controlled line follower robots in public transport
- Controlling line follower robot with the remote web server”
- Path planning of line follower robot” Proceedings

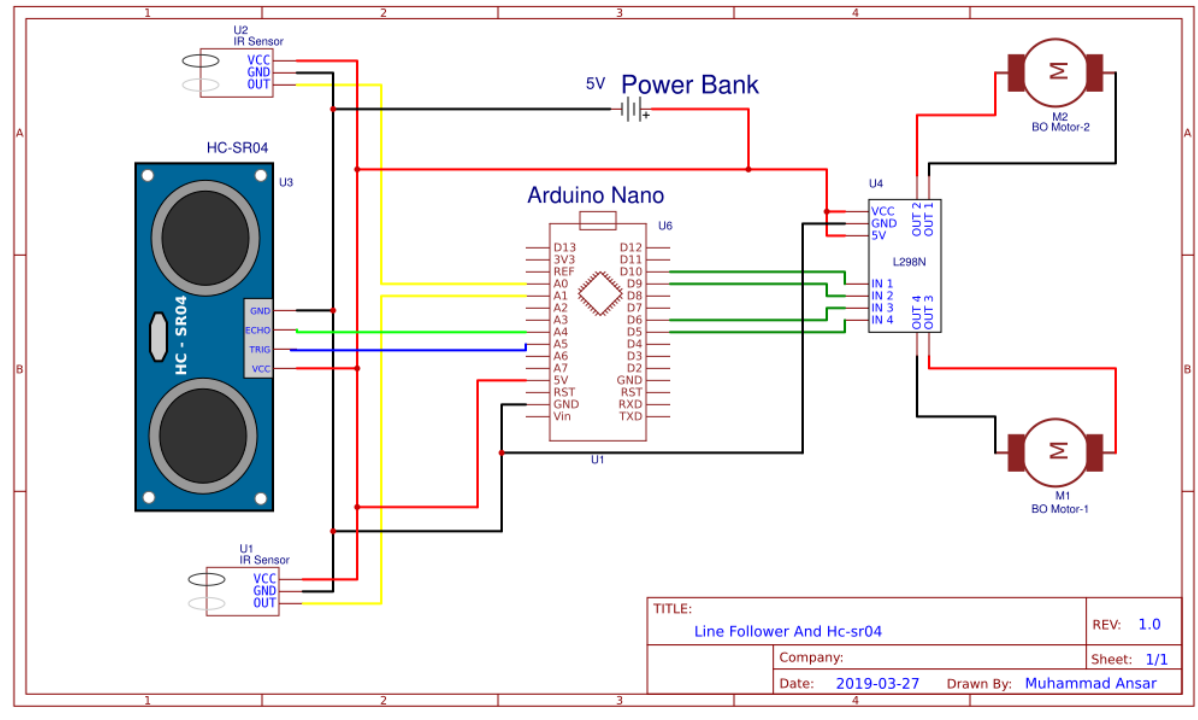
- Implementation of autonomous line follower **robot**

WORK METHODOLOGY

Fuzzy Systems and Soft Computing

- Based on computing some of them created and implemented
- Proceedings of SAI Intelligent Systems Conference
- European DSP Education and Research Conference

BLOCK DIAGRAM:

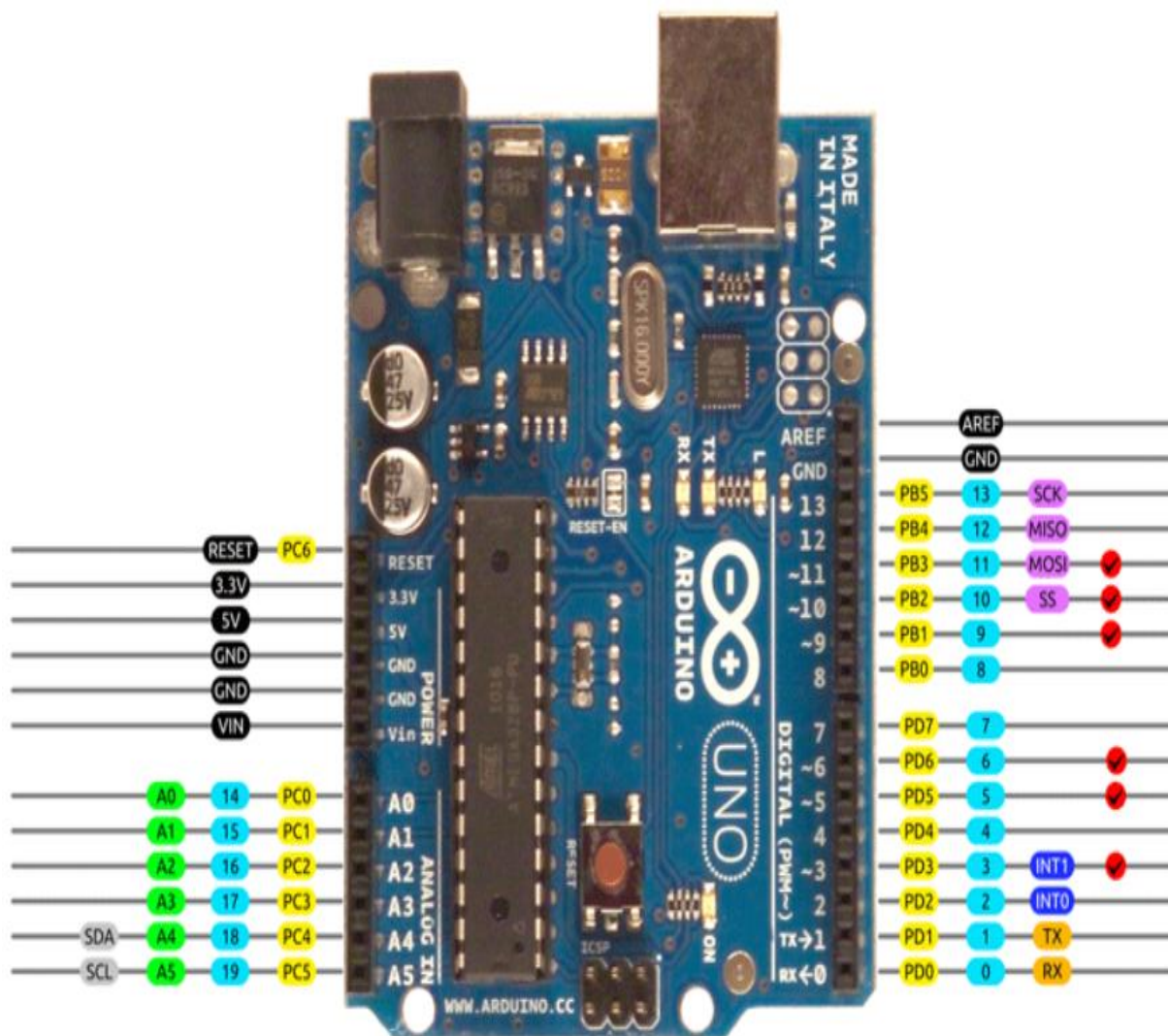


HARD WARE COMPONENTS:

1.ARDUINO UNO

2.BLUETOOTH MODULE

3.IR SENSORS



AVR DIGITAL ANALOG POWER SERIAL SPI I2C PWM INTERRUPT

Software:

Download the Arduino IDE

ARDUINO 1.8.10

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

This software can be used with any Arduino board.

Refer to the [Getting Started](#) page for Installation instructions.

[Windows Installer, for Windows XP and up](#)
[Windows ZIP file for non admin install](#)

[Windows app Requires Win 8.1 or](#)



[10](#)
[Mac OS X 10.8 Mountain Lion or newer](#)

[Linux 32 bits](#)

[Linux 64 bits](#)

[Linux ARM 32 bits](#)

[Linux ARM 64 bits](#)

[Release Notes](#)

[Source Code](#)

[Checksums \(sha512\)](#)

Basic Functions of Arduino Technology: -

- Digital write pin is used to write the digital value of the given pin.
- Pin mode pin is used to set the pin to I/O mode.
- Analog read pin reads and returns the value.
- Analog write pin writes the value of the pin.Serial.
- Begins pin sets the beginning of serial communication by setting the rate of bit.

Advantages of Arduino Technology:

- It is cheap.
- It comes with an open supply hardware feature that permits users to develop their own kit.

- The software of the Arduino is well-suited with all kinds of in operation systems like Linux, Windows, and Macintosh, etc.
- It also comes with open supply software system feature that permits tough software system developers to use the Arduino code to merge with the prevailing programming language libraries and may be extended and changed.
- For beginners, it is very simple to use.

Disadvantages:

1- Structure:

Yes, the structure of Arduino is its disadvantage as well. During building a project you have to make its size as small as possible. But with the big structures of

Arduino we have to stick with big sized PCB's. If you are working on a small micro-controller like ATmega8 you can easily make your PCB as small as possible.

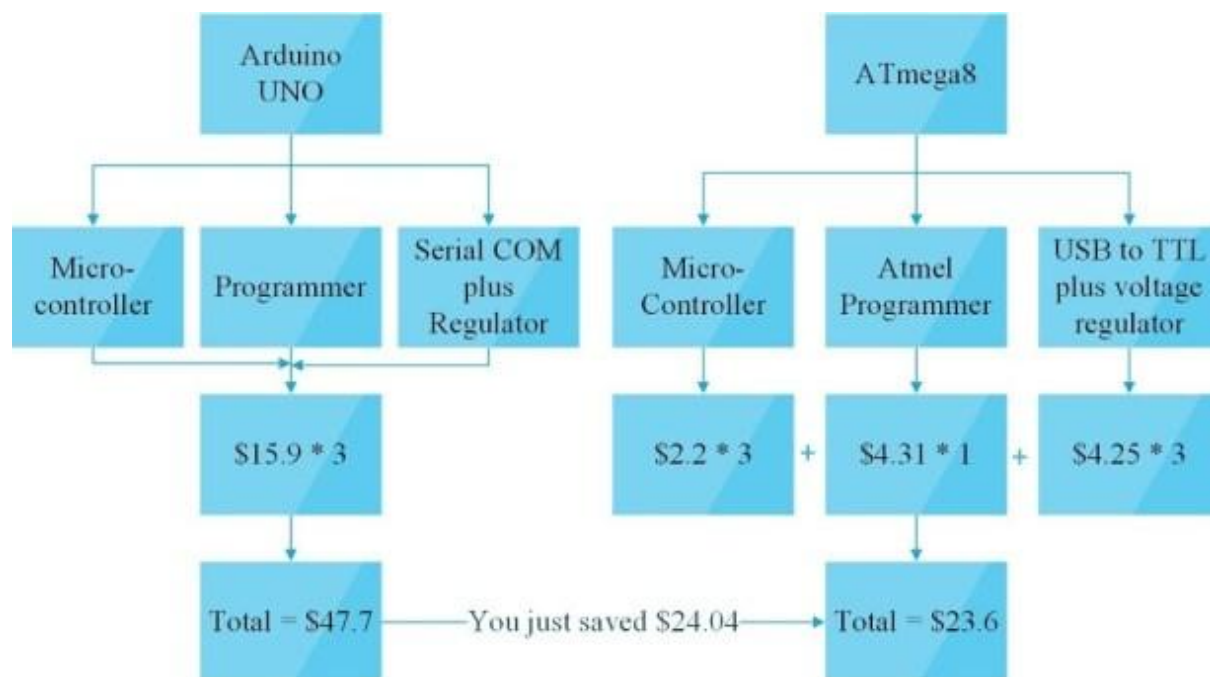


Figure 2: Cost Comparison of Arduino and ATmega

The thing must be noted that I multiplied Atmel Programmer with 1 because we don't need many programmers for all the micro-controllers. Only one programmer is enough.

The difference between the costs is mainly due to this programmer reason. Still if you need one package then the cost difference will be as less as nearly \$5 and it will rise when you have to use many packages.

3- Easy to use:

In my opinion, if you started your journey of micro-controllers with Arduino then it will be very difficult for you to make the complex intelligent circuitries in future. The easy to use hardware/software of Arduino unable a person to learn the basics of many things likes Serial communication, ADC, I2C etc.

On the Arduino Website:

- [Tutorials](#): code examples and circuits for performing many tasks.
- [Reference](#): documentation of the Arduino programming language and functions.
- [Hardware](#): descriptions of the various Arduino boards and other hardware, with schematics, PCB layout files, assembly instructions, etc.
- [Language comparison](#): compares the Arduino/Wiring language (based on C/C++) with Processing (based on Java).
- [FAQ](#): frequently asked questions about Arduino.
- [Discussion forums](#): for questions about anything Arduino related.
- [Playground](#): a publicly editable wiki collecting community documentation ([register here](#)).

•

- **External Resources on Arduino:**
- [Video lectures](#): Tom Igoe introduces Arduino. (thanks to Pollie Barden).
- [Course guides](#): Longer documents introducing Arduino: [class 1 \(getting started\)](#), [class 2 \(input and sensors\)](#), [class 3 \(communication, servos, and pwm\)](#), [class 4 \(piezo sound & sensors, arduino+processing, stand-alone operation\)](#)).

Electronics and Physical Computing:

- [Wiring electronics reference](#): circuit diagrams for connecting a variety of basic electronic components.
- [Schematics to circuits](#): from Wiring, a guide to transforming circuit diagrams into physical circuits.

- [Open Circuits](#): a public wiki collecting electronic circuits, components, and techniques.
- [Tom Igoe's Physical Computing Site](#): lots of information on electronics, microcontrollers, sensors, actuators, books, etc.

Related Sites:

- [Instant Soup](#) is an introduction to electronics through a series of beautifully-documented fun projects.
- [Make magazine](#) has some great links in its [electronics archive](#).
- [hack a day](#) has links to interesting hacks and how-to articles on various topics.

The text of the Arduino getting started guide is licensed under a [Creative Commons](#)

[Attribution-ShareAlike 3.0 License](#). Code samples in the guide are released into the public domain.

Line Follower And Remote Controller :-

INTRODUCTION:

This project is based upon a microcontroller more specifically Arduino. But it is also possible to develop a line follower robot without a microcontroller. But there will be limited functionality and the stopped will be lower than any microcontroller-based vehicles.

Because the data processing power of the Microcontrollers is greater than simple Op-Amp ICs. This will be a basic line follower car project report ppt.

In this short descriptive article, I will show you how to make a Line Follower Car using Arduino UNO and 2 IR Sensor Modules. The name Line follower name suggests a Fully Automated Car. That follows a visual line (Most likely white or Black) that is embedded on any floor or any surface.

As the name suggests, the line follower car is an automated vehicle that follows a visual line embedded on the surface. This visual line is a path on which the line follower robot runs. Generally, it uses a black line on a white surface, or you can adjust it as a white line on a black surface.

Usually, beginners and students would get their first experience with this type of robot. In this project-based article, we have informed

about how to make a line follower robot using Arduino.

In industries, giant line follower robots are used for assisting the automated production process. They are also used in military applications, human assistance purposes, delivery services, etc.

How does the Line Follower and Remote Controller?

This is a simple Microcontroller-based car. Let me tell you in detail how the thing works. The car consists of 2 IR sensors.

This sensor senses the line colour is reflective or not. Here reflection means the surface sends back the light. In this case, the reflective surface is the White surface. (The non-reflective surface is the black lines only.) It means that those black tracks don't reflect any light back.

The working of the sensor is that it senses if there is any reflective light coming back or not. Then only the sensor will give the output (Output two types. I will discuss that later). And if there is no reflection then the IR sensor will not give any output or in other words, the output will be low.

Based on this High and Low Output results the microcontroller can control the car.

The microcontroller gives the direction to the car where to go and then when to take turns (Left or Right) and where to stop (I created the code in such a way if the 2 sensors get Black Surface, then it will tell the microcontroller to stop the car.).

Things Needed for Arduino Line Follower And Remote Controller

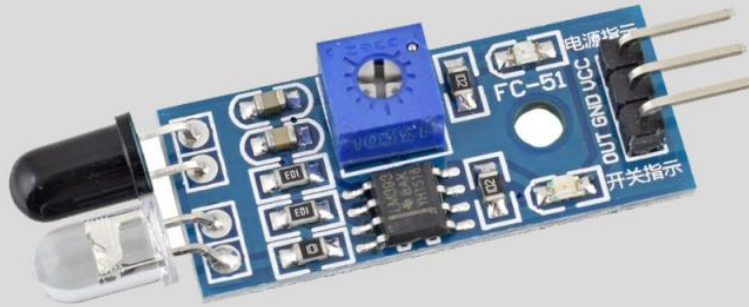
- . IR Sensor
- . Arduino UNO R3
- . L293D Motor Driver
- . Kit 4 curious (Rocker Bhogie)

- Jumper Wire
- 4v x 1.5amp (3 batteries)
- Soldering Iron
- Glue Gun

IR SENSOR:-

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.



IR Sensor

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.

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, etc. It has 14 digital I/O pins(6 pins can be used as PWM pins). It has six separate analog input pins, a USB connection, a Power barrel jack, an ICSP header, and a reset button.



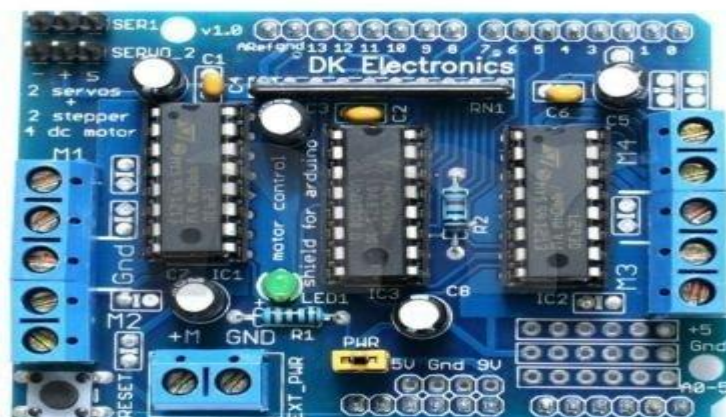
This board is programmable with the Arduino IDE (Integrated Development Environment) platform via a type B USB cable. This board can be powered by a USB cable or an external voltage between 7 to 20 volts. To know more about an Arduino Uno, refer to the article, what is Arduino Uno?

L293D Motor Driver :

L298N is one of the easiest and chipset way to control DC motors. It is the two-channel motor driver that can control the speed and spinning direction of DC motors.

This L298N Motor Driver is a high-power motor driver module. It is used for driving DC and Stepper Motors. This motor driver consists of an L298N motor driver IC and a 78M05 5V voltage regulator, resistors, capacitor, power LED, 5V jumper in an integrated circuit.

When the jumper is placed, it enables the 78M05 Voltage regulator. When the power

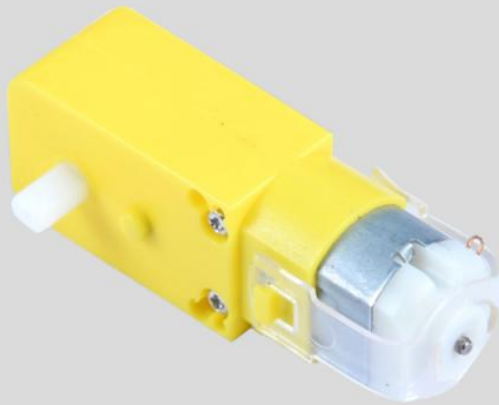


supply is less than or equal to 12V, the voltage regulator will power on the internal circuitry. When the power supply is more than 12v, then the jumper should not place and should give a separate 5v to power the internal circuitry.

Here, ENA & ENB pins are speed control pins for Motor A, and Motor B. IN1& IN2 and IN3 & IN4 are direction control pins for Motor A and Motor B.

BO Motors :-

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.

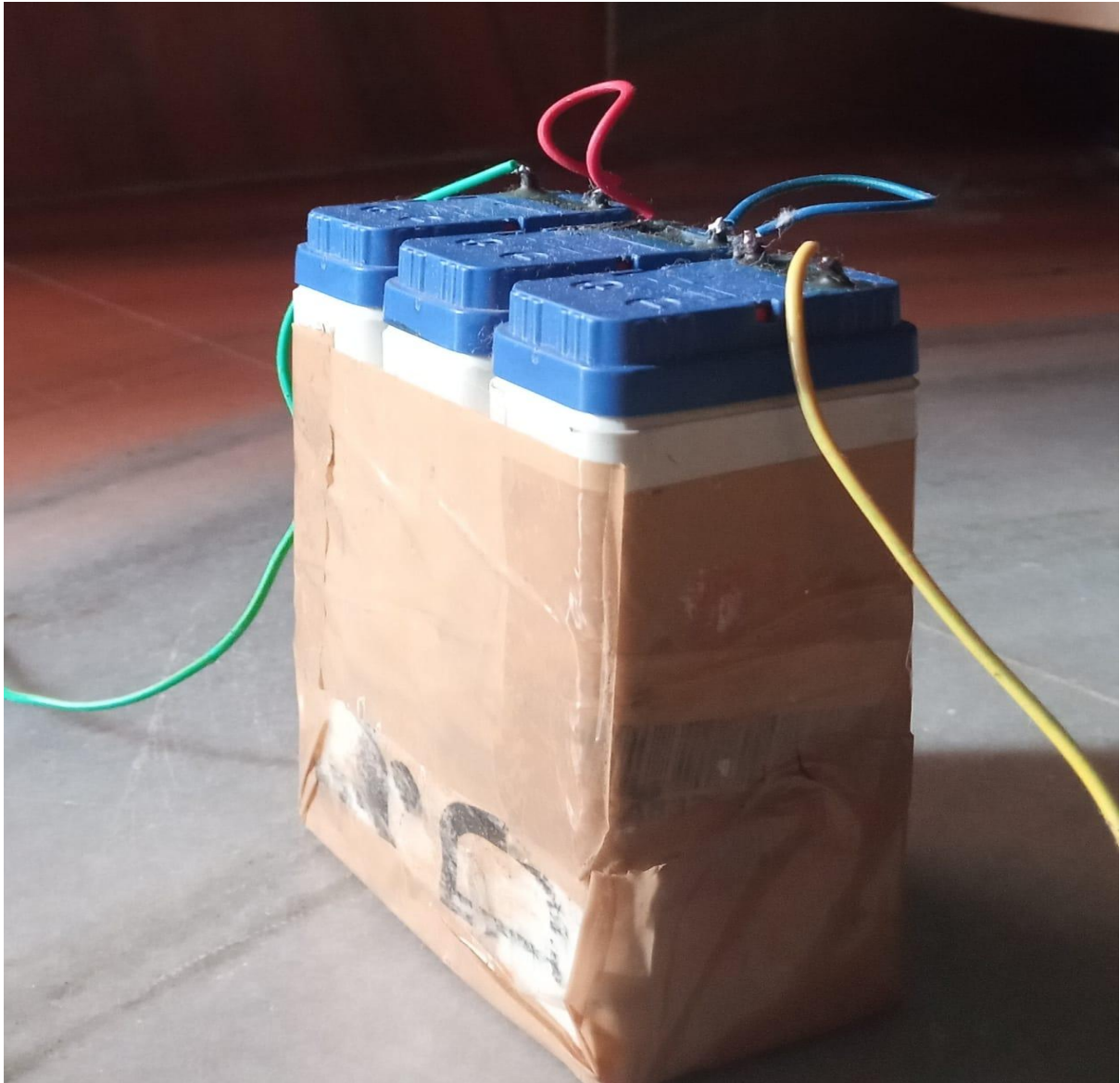


BO Motor

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.

Lithium-ion Battery :

A lithium-ion battery is a rechargeable battery. It is commonly used in portable devices such as mobiles, laptops, electronics, and electric vehicles. Also, they are growing in popularity for military and aerospace applications.



Lithium-ion battery provides 3.7V at storage mode and 4.2V at full charge mode. In this project, we have connected two lithium-ion batteries in series so that the total battery voltage will be 8.4V at full charge. To know more about the lithium-ion battery, refer to the article, working of lithium-ion battery.

Popular Categories:

- 1.Circuits
- 2.Arduino Projects
- 3.DIY Project

Arduino Line Follower And Remote Controller Code :

//ARDUINO LINE FOLLOWER CAR Based On
Arduino Motor Driver Shield

//You have to install the AFMotor library
before uploading the sketch

//First download the AFMotor Zip file then
open Arduino IDE

//go to sketch >> Include library >> ADD .Zip
file >> Select the downloaded AFMotor Zip
File >> open

// Created by DIY Builder

// You Can modify the sketch according to
your need.

#include<AFMotor.h> //Include Adafruit
Motor Driver Shield Library

//create motor objects

AF_DCMotor motor1(1, MOTOR12_1KHZ);

AF_DCMotor motor2(2, MOTOR12_1KHZ);

AF_DCMotor motor3(3, MOTOR34_1KHZ);

AF_DCMotor motor4(4, MOTOR34_1KHZ);

#define LEFT_SENSOR A0 // connect the Left
sensor with analog pin A0

```
#define RIGHT_SENSOR A1 // connect the  
Right sensor with analog pin A1
```

```
void setup() {
```

```
pinMode(RIGHT_SENSOR, INPUT); // initialize  
Right sensor as an input
```

```
pinMode(LEFT_SENSOR, INPUT); // initialize  
Left sensor as input
```

```
}
```

```
void loop() {
```

```
if(analogRead(RIGHT_SENSOR)<=35 &&  
analogRead(LEFT_SENSOR)<=35) //compare  
both sensor value to set the directionc
```

```
{
```

```
motor1.run(FORWARD); // run motor1  
clockwise
```

```
    motor1.setSpeed(120); // set motor1 speed  
    50 percent
```

```
    motor2.run(FORWARD); // run motor2  
    clockwise
```

```
    motor2.setSpeed(120); // set motor2 speed  
    50 percent
```

```
    motor3.run(FORWARD); // run motor3  
    clockwise
```

```
    motor3.setSpeed(120); // set motor3 speed  
    50 percent
```

```
    motor4.run(FORWARD); // run motor4  
    clockwise
```

```
    motor4.setSpeed(120); // set motor4 speed  
    50 percent
```

```
}
```

```
    else if(!analogRead(RIGHT_SENSOR)<=35 &&  
    analogRead(LEFT_SENSOR)<=35) //compare  
    both sensor value to set the direction
```

```
{
```

```
    motor1.run(FORWARD); // run motor1
    clockwise

    motor1.setSpeed(255); // set motor1 speed
    100 percent

    motor2.run(FORWARD); // run motor2
    clockwise

    motor2.setSpeed(255); // set motor2 speed
    100 percent

    motor3.run(BACKWARD); // run motor3
    anti-clockwise

    motor3.setSpeed(255); // set motor3 speed
    100 percent

    motor4.run(BACKWARD); // run motor4
    anti-clockwise

    motor4.setSpeed(255); // set motor4 speed
    100 percent

}
```

else if(analogRead(RIGHT_SENSOR)<=35 &&
!analogRead(LEFT_SENSOR)<=35) //compare
both sensor value to set the direction

{

motor1.run(BACKWARD); // run motor1
anti-clockwise

motor1.setSpeed(255); // set motor1 speed
100 percent

motor2.run(BACKWARD); // run motor2
anti-clockwise

motor2.setSpeed(255); // set motor2 speed
100 percent

motor3.run(FORWARD); // run motor3
clockwise

motor3.setSpeed(255); // set motor3 speed
100 percent

motor4.run(FORWARD); // run motor4
clockwise

```
    motor4.setSpeed(255); // set motor4 speed
    100 percent
}
else if(!analogRead(RIGHT_SENSOR)<=35 &&
!analogRead(LEFT_SENSOR)<=35) //compare
both sensor value to set the direction
{
    //stop all the motors
    motor1.run(RELEASE);
    motor2.run(RELEASE);
    motor3.run(RELEASE);
    motor4.run(RELEASE);
}
}
```

RESULT AND ANALYSYS:

```
sketch_oct30a

#include <AFMotor.h> //Include Adafruit Motor Driver Shield Library
//create motor objects
AF_DCMotor motor1(1, MOTOR12_1KHZ);
AF_DCMotor motor2(2, MOTOR12_1KHZ);
AF_DCMotor motor3(3, MOTOR34_1KHZ);
AF_DCMotor motor4(4, MOTOR34_1KHZ);

#define LEFT_SENSOR A0 // connect the Left sensor with analog pin A0
#define RIGHT_SENSOR A1 // connect the Right sensor with analog pin A1

void setup() {

  pinMode(RIGHT_SENSOR, INPUT); // initialize Right sensor as an input
  pinMode(LEFT_SENSOR, INPUT); // initialize Left sensor as an input

}

void loop() {
  if(analogRead(RIGHT_SENSOR)<=35 && analogRead(LEFT_SENSOR)<=35) //compare both sensor value to set the direction
  {
    motor1.run(FORWARD); // run motor1 clockwise
    motor1.setSpeed(120); // set motor1 speed 50 percent
    motor2.run(FORWARD); // run motor2 clockwise
    motor2.setSpeed(120); // set motor2 speed 50 percent
    motor3.run(FORWARD); // run motor3 clockwise
    motor3.setSpeed(120); // set motor3 speed 50 percent
    motor4.run(FORWARD); // run motor4 clockwise
    motor4.setSpeed(120); // set motor4 speed 50 percent
  }
  else if(!analogRead(RIGHT_SENSOR)<=35 && !analogRead(LEFT_SENSOR)<=35) //compare both sensor value to set the direction
  {
    motor1.run(FORWARD); // run motor1 clockwise
    motor1.setSpeed(255); // set motor1 speed 100 percent
    motor2.run(FORWARD); // run motor2 clockwise
    motor2.setSpeed(255); // set motor2 speed 100 percent
  }
}
```

Please select a port to obtain board info.

```
sketch_oct30a

motor4.run(FORWARD); // run motor4 clockwise
motor4.setSpeed(120); // set motor4 speed 50 percent
}
else if(!analogRead(RIGHT_SENSOR)<=35 && analogRead(LEFT_SENSOR)<=35) //compare both sensor value to set the direction
{
  motor1.run(FORWARD); // run motor1 clockwise
  motor1.setSpeed(255); // set motor1 speed 100 percent
  motor2.run(FORWARD); // run motor2 clockwise
  motor2.setSpeed(255); // set motor2 speed 100 percent
  motor3.run(BACKWARD); // run motor3 anti-clockwise
  motor3.setSpeed(255); // set motor3 speed 100 percent
  motor4.run(BACKWARD); // run motor4 anti-clockwise
  motor4.setSpeed(255); // set motor4 speed 100 percent
}

else if(analogRead(RIGHT_SENSOR)<=35 && !analogRead(LEFT_SENSOR)<=35) //compare both sensor value to set the direction
{
  motor1.run(BACKWARD); // run motor1 anti-clockwise
  motor1.setSpeed(255); // set motor1 speed 100 percent
  motor2.run(BACKWARD); // run motor2 anti-clockwise
  motor2.setSpeed(255); // set motor2 speed 100 percent
  motor3.run(FORWARD); // run motor3 clockwise
  motor3.setSpeed(255); // set motor3 speed 100 percent
  motor4.run(FORWARD); // run motor4 clockwise
  motor4.setSpeed(255); // set motor4 speed 100 percent
}
else if(!analogRead(RIGHT_SENSOR)<=35 && !analogRead(LEFT_SENSOR)<=35) //compare both sensor value to set the direction
{
  //stop all the motors
  motor1.run(RELEASE);
  motor2.run(RELEASE);
  motor3.run(RELEASE);
  motor4.run(RELEASE);
}
}
```

Conclusion:

In conclusion, the Arduino code uploaded to the Arduino micro controller which is then connected to the L293D motor driver with help of this the wheels are able to rotate. Not only for the motor wheels but also for the IR sensor which is connected to the motor driver. For working of this entire setup, it is powered by Three 4v batteries. The car follows the path of the black lanes, which is kept in a steady and uniform path.

Finally, the car followed the desired path without any alterations, this prototype can be used in many places mainly where it is dangerous for humans for example in mines. Also, they can be used as an alternative for the conventional methods.

- [1] Razvan Solea, Adrian Filipescu and Grigore Stamatescu” Sliding-mode real-time mobile platform control in the presence of uncertainties ”,Decision and Control(2019) 32 16-18
- [2] T. Palleja,M. Tresanchez,M. Teixido,J. Palacin" Modeling floor-cleaning coverage performances of some domestic mobile robots in a reduced scenario", Robotics and Autonomous Systems(2020) 58 37-45.
- [3] M.R.B. Bahara, A.R. Ghiasib, H.B. Bahara, "Grid roadmap based ANN corridor search for collision free, path planning ",Scientia Iranica (2012) 19 1850-1855.
- [4] Ayoub Bahmanikashkoolia , Majid Zareb, Bahman Safarpourc, Mostafa Safarpourd" Application of Particle Swarm Optimization Algorithm for Computing Critical Depth of Horseshoe Cross Section

Tunnel "APCBEE Procedia(2018)9 207–211

- [5] Spyros G. Tzafestas"9 – Mobile Robot Control V: Vision-Based Methods",Introduction to Mobile Robot Control(2014) 319–384
- [6] Spyros G. Tzafestas"11 – Mobile Robot Path, Motion, and Task Planning”, Introduction to Mobile Robot Control (2014) 429–478
- [7] Masoud Nosrati , Ronak Karimi , Hojat Allah Hasanvand “Investigation of the * (Star) Search Algorithms: Characteristics, Methods and Approaches” Applied Programming(2012) 2 251-256
- [8] Dr. R.Anbuselvi “PATH FINDING SOLUTIONS FOR GRID BASED GRAPH” Advanced Computing(2013) 4