**ROAD SAFE ALERT SYSTEM: ENSURING VEHICLE SAFETY THROUGH DIGITAL NOTIFICATIONS DURING MANHOLE AND ROAD CONSTRUCTION DIVERSION**

*Submitted to Jawaharlal Nehru Technological University, Kakinada*

*in partial fulfillment of requirement for the degree of*

#### Bachelor of Technology

in

### ELECTRONICS AND COMMUNICATION ENGINEERING

*Submitted by*

### G. SRINIVASULU

(20KN1A0442)

|  |  |  |
| --- | --- | --- |
| **B. SAMITHA** | **E. GOVINDRAJU** | **K. MADHUBABU** |
| (20KN1A0424) | (20KN1A0440) | (20KN1A0459) |

Under the guidance of

#### N. MALATHI,M. Tech

Assistant Professor



**DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING**

### NRI INSTITUTE OF TECHNOLOGY

### AUTONOMOUS

**Approved by AICTE, New Delhi, Permanently Affiliated to JNTUK, Kakinada**

**Accredited by NBA (ECE, CSE & EEE), Accredited by NAAC with ‘A’ Grade**

**ISO 9001: 2015 Certified Institution**

**Pothavarappadu (V), Agiripalli (M), Eluru (Dt)-521212**

**2020-2024**

# Department of

**ELECTRONICS & COMMUNICATION ENGINEERING**



**CERTIFICATE**

This is to certify that the project report entitled **Road safety alert system: Ensuring vehicle safety through digital notifications during manhole and road construction diversion** that is being submitted by

|  |  |
| --- | --- |
| G. Srinivasulu | -20KN1A0442 |
| B. Samitha | -20KN1A0424 |
| E. Govind Raju | -20KN1A0440 |
| K. Madhu Babu | -20KN1A0459 |

in partial fulfillment for the award of **Bachelor of Technology** in **Electronics &Communication Engineering** to the **Jawaharlal Nehru Technological University, Kakinada** is a record of Bonafide work carried out by them under my guidance and supervision. The results embodied in this project report have not been submitted to any other University or Institute for the award of any degree or diploma.

|  |  |
| --- | --- |
| **N. MALATHI** | **DR. R. SUNITHA** |
| **Assistant Professor** | **Professor & Head** |

**External Examiner**

# Acknowledgement

We are very much grateful to **N. Malathi, M. Tech, Assistant** **Professor**, Electronics & Communication Engineering, for his inspiring guidance and for providing the background knowledge to deal with the problem at every phase of our project in a systematic manner.

We have immense pleasure in expressing our thanks and deep sense of gratitude to **Dr.**

**R. Sunitha M. Tech, Ph. D**, Head of the Department, Electronics & Communication Engineering, for providing necessary facilities for the completion of project.

We are very much indebted to **Dr. V. Srinivasarao, M. Tech, Ph. D, Dean Academics of ECE Department** for his critical analysis during the reviews and useful suggestions towards successful completion of the project work.

We are grateful to our beloved **PRINCIPAL, Dr. NAGABHASKAR, M. Tech., Ph.D., FIE, CE, MISTE, MCMSI** for his consistent encouragement and guidance.

We express our heartfelt gratitude to Dr.R.VENKATARAO, Chairman and Sri K.SREEDHAR, Secretary, of NRI INSTITUTE OF TECHNOLOGY for their patronage and parental hood in achieving excellence in our academic pursuit.

We would like to express our sincere and heartfelt thanks to the faculty of our department for their continuous cooperation, which has given us the ultimate confidence in successful completion of this project.

**Project Associates**

|  |  |
| --- | --- |
| **G. SRINIVASULU** | - **20KN1A0442** |
| **B. SAMITHA** | - **20KN1A0424** |
| **E. GOVIND RAJU** | - **20KN1A0440** |
| **K. MADHU BABU** | - **20KN1A0459** |

# DECLARATION

We the members of the project " **Road safety alert system: Ensuring vehicle safety through digital notifications during manhole and road construction diversion”** hereby declare that the matter embodied in this project is the genuine work done by us and has not been submitted either to this University or to any other University/Institute for the fulfillment of the requirement of any other course of study.

**Project Associate**

|  |  |
| --- | --- |
| **G. SRINIVASULU** | - **20KN1A0442** |
| **B. SAMITHA** | - **20KN1A0424** |
| **E. GOVIND RAJU** | - **20KN1A0440** |
| **K. MADHU BABU** | - **20KN1A0459** |

# TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  | **ABSTRACT** | **i** |
|  | **LIST OF FIGURES** | **ii** |
|  | **LIST OF TABLES** | **v** |
|  | **LIST OF ACRONYMS & ABBREVATIONS** | **vi** |
| **CHAPTER 1** | **INTRODUCTION** | **1** |
|  | 1.1 INTRODUCTION | 1 |
|  | 1.2 EMBEDDED SYSTEM IMPLEMENTATION | 3 |
|  | 1.2.1 EMBEDDED SYSTEM HARDWARE | 5 |
|  | 1.2.2 EMBEDDED SYSTEM SOFTWARE | 6 |
|  | 1.2.3 APPLICATIONS | 8 |
|  | 1.2.4 IMPLEMENTATION FLOW | 9 |
| **CHAPTER 2** | **LITERATURE SURVEY** | **10** |
|  | 2.1 LITERATURE SURVEY | 10 |
|  | 2.2 EXISTING SYSTEM | 11 |
|  | 2.3 DRAWBACKS IN EXISTING SYSTEM | 11 |
| **CHAPTER 3** | **PROPOSED SYSTEM & OVERVIEW** | **12** |
|  | 3.1 PROPOSED SYSTEM | 12 |
|  | 3.1.1 NODE MCU | 13 |
|  | 3.1.2 INFRARED SENSOR | 13 |
|  | 3.1.3 ARDUINO CONTROLLER | 13 |

|  |  |  |
| --- | --- | --- |
|  | 3.1.4 MOTOR DRIVER | 13 |
|  | 3.1.5 ULTRASONIC SENSOR | 14 |
|  | 3.1.6 LED | 14 |
|  | 3.1.7 POWER SUPPLY  3.1.8 LCD | 14  14 |
| **CHAPTER 4** | **HARDWARE REQUIREMENTS** | **15** |
|  | 4.1 ARDUINO | 15 |
|  | 4.1.1 INTRODUCTION TO ARDUINO | 17 |
|  | 4.1.2 FEATURES OF ARDUINO | 18 |
|  | 4.1.3 ARDUINO PINOUT | 20 |
|  | 4.1.4 PIN DESCRIPTION | 21 |
|  | 4.1.5 ARDUINO UNO TECHNICAL SPECIFICATIONS | 23 |
|  | 4.1.6 COMMUNICATION & PROGRAMMING | 24 |
|  | 4.1.7 APPLICATIONS | 25 |
|  | 4.2 LCD  4.2.1 DATA/EXECUTION OF LCD  4.2.2 REGISTER SELECT  4.2.3 COMMAND REGISTER  4.2.4 DATA REGISTER  4.2.4 PIN DESCRIPTION  4.2.5 LCD COMMANDS  4.2.6 BLOCK DIAGRAM OF LCD DISPLAY  4.2.7 CONTROL AND DISPLAY COMMANDS  4.2.8 4-BIT AND 8-BIT MODE OF LCD  4.3 NODE MCU  4.4 FEATURES  4.5 DC MOTOR  4.5.1 WHAT IS A DC MOTOR  4.5.2 HOW DC MOTORS WORK  4.5.3 BUT HOW DO DC MOTORS WORK  4.5.4 WHO INVENTED THE DC MOTOR  4.6 LED  4.6.1 LED MATERIALS  4.6.2 LED STRUCTURE  4.6.3 APPLICATIONS  4.6.4 ADVANTAGES OF USING LEDS  4.7 MOTOR DRIVER  4.7.1 WORKING OF L2930  4.7.2 L2930 LOGIC TABLE  4.7.3 CIRCUIT DIAGRAM FOR L293D MOTOR DRIVER  4.7.4 VOLTAGE SPECIFICATIONS | 26  27  28  28  29  29  30  31  32  32  33  34  35  35  35  35  36  37  38  39  40  40  41  43  43  44  44 |
|  | 4.8 POWER SUPPLY | 45 |
|  | 4.8.1 RECTIFIER | 46 |
|  | 4.8.2 CAPACITOR | 47 |
|  | 4.8.3 VOLTAGE REGULATORS | 48 |
|  | 4.8.4 IR SENSOR | 49 |
|  | 4.8.5 BUZZER | 50 |
|  | 4.8.5.1 BUZZER PIN CONFIGURATION | 50 |
|  | 4.8.5.2 BUZZER FEATURES AND SPECIFICATIONS | 51 |

|  |  |  |
| --- | --- | --- |
|  | 4.8.5.3 HOW TO USE A BUZZER | 51 |
|  | 4.8.5.4 APPLICATIONS OF BUZZER  4.8.6 ULTRASONIC SENSOR | 51  52 |
| **CHAPTER 5** | **SOFTWARE REQUIREMENTS** | **53** |
|  | 5.1 ARDUINO IDE | 53 |
|  | 5.2 INTRODUCTION TO ARDUINO IDE | 53 |
|  | 5.2.1 HOW TO INSTALL ARDUINO IDE | 54 |
|  | 5.3 LIBRARIES | 60 |
|  | 5.4 MAKING PINS INPUT & OUTPUT | 61 |
|  | 5.4.1 HOW TO SELECT THE BOARD | 61 |
|  | 5.4.2 BOOT LOADER |  |
| **CHAPTER 6** | **CODE & RESULTS** | 60 |
|  | 6.1 CODE  6.1.1 CODE FOR ARDUINO UNO  6.1.2 CODE FOR WEB SERVER | 60 |
|  | 6.2 RESULTS  6.2.1 CASE1  6.2.2 CASE2  6.2.3 CASE3 | 63 |
| **CHAPTER 7** | **ADVANTAGES AND APPLICATIONS** | **67** |
|  | 7.1 ADVANTAGES | 67 |
|  | 7.2 APPLICATIONS | 67 |
| **CHAPTER 8** | **CONCLUSION** | **76** |
|  | **REFERENCES** | **77** |

**ROAD SAFE ALERT SYSTEM: ENSURING VEHICLE SAFETY THROUGH DIGITAL NOTIFICATIONS DURING MANHOLE AND ROAD CONSTRUCTION DIVERSION**

**Abstract:**

The project titled "Road Safe Alert System" presents a groundbreaking solution to enhance road safety by providing real-time digital notifications to drivers during manhole and road construction diversions. This innovative system leverages cutting-edge technology, including IR sensors, servo motors, LEDs, an LCD display, and cloud-based communication via the Thing Speak server, to ensure vehicle safety and reduce road hazards.

At its core, the system employs IR sensors to detect the presence of open manholes on the road. When an open manhole is detected, a red LED indicator is activated to warn approaching vehicles, significantly reducing the risk of accidents caused by manhole hazards. Furthermore, the system utilizes a servo motor to indicate road diversions, ensuring that drivers are aware of construction or roadwork zones ahead. The system uploads the detected manhole and diversion information to the Thing Speak server, which serves as a digital notification platform. This information is displayed on an LCD screen within the vehicle, providing drivers with real-time updates about potential road hazards and diversions. The "Road Safe Alert System" not only prioritizes road safety but also demonstrates the power of technology in creating a safer and more informed driving experience for all road users.

Top of Form

Top of Form

Top of Form

**Keywords: Arduino, LCD, Red LED, Manhole, Road diversion**

# LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **FIG.NO** | **NAME OF THE FIGURE** | **PAGE NO** |
| 1.2.1 | Overview of Embedded system | 2 |
| 1.2.2 | Block diagram of Embedded system | 3 |
| 1.2.2.1 | Flow of burning source code to processor | 6 |
| 3.1.1 | Block Diagram of Proposed system | 12 |
| 4.1 | Types of Arduinos | 15 |
| 4.2 | Arduino Board | 16 |
| 4.3 | Micro controller | 16 |
| 4.1.1.1 | Arduino UNO | 17 |
| 4.1.2.1 | Atmega328 Microcontroller | 18 |
| 4.1.3.1 | Arduino Uno Pinout | 20 |
| 4.1.6.1 | Personal Computer (PC) | 24 |
| 4.2.1.1 | LCD -Back view | 28 |
| 4.2.1.2 | LCD-Front View | 28 |
| 4.2.1.3 | Pin Diagram | 28 |
| 4.2.6.1 | Block Diagram of LCD Display | 31 |
| 4.4.1 | Node MCU | 34 |
| 4.5.1 | Motor | 36 |
| 4.6.1 | LED | 37 |

|  |  |  |
| --- | --- | --- |
| 4.6.2 | Working of LED | 38 |
| 4.6.3 | Schematic | 38 |
| 4.6.2.1 | Optocoupler schematic showing LED & phototransistor | 40 |
| 4.7.1 | L293D | 42 |
| 4.7.4.1 | Circuit Diagram | 44 |
| 4.8.1 | Circuit diagram for power supply | 45 |
| 4.8.1.1 | Circuit of rectifier | 46 |
| 4.8.1.2 | Rectifier | 46 |
| 4.8.2.1 | Capacitor | 47 |
| 4.8.3.1 | 7805 Voltage Regulator with pinout | 48 |
| 4.8.3.2 | 7812 Voltage Regulator with pinout | 48 |
| 4.8.4.1 | IR sensor | 49 |
| 4.8.5.1 | Buzzer | 50 |
| 4.8.6.1 | Ultrasonic sensor | 52 |
| 5.2.1.1 | Arduino IDE | 54 |
| 5.2.1.2 | File option in Arduino IDE | 55 |
| 5.2.1.3 | Preferences in Arduino IDE | 56 |

5.2.1.4 Hex file 56

5.2.1.5 Buttons of Menu tab in Arduino IDE 57

5.2.1.6 Output in Serial monitor 58

5.2.1.7 Text editor 59

5.2.1.8 Output pane 59

5.2.1.9 Sketch button 60

5.2.1.10 Board section in Tooth option 61

5.2.1.11 Port section in tools 62

5.2.1.12 Boot loader 63

6.2.1 Photograph of working of road safety alert system 71

6.2.1.2 LCD Display 1 72

6.2.1.3 Output 1 72

6.2.2.1 Distance calculation 2 72

6.2.2.2 Output 2 73

6.2.3.1 LCD Display 3 73

6.2.3.2 Output 3 73

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **NAME OF THE TABLE** | **PAGE NO** |
| 4.1.5.1  4.2.4.1  4.2.5.1  4.2.7.1  4.7.2  4.8.5.1 | Arduino UNO technical specifications  Pin Functions  Hex code &LCD commands  Control and Display commands  Pin Description  Buzzer Pin Description | 23  29  31  32  43  50 |

**LIST OF ACRONYMS**

|  |  |  |
| --- | --- | --- |
| 1 | LCD | –Liquid Crystal Display |
| 2 | MCU | – Micro Controller Unit |
| 3 | GSM | – Global System for Mobile Communications |
| 4 | ESP | – Electronic Stability Program |
| 5 | IR | –Infrared Sensor |
| 6 | PWM | – Pulse Width Modulation |
| 7 | FTDI USB | –Future Technology Devices International Universal Serial Bus |
| 8 | IOREF | – Input – Output voltage Reference |
| 9 | AREF | – Analog Reference |
| 10 | SRAM | – Static Random Access Memory |
| 11 | EEPROM | – Electrically Erasable Programmable Read only memory |
| 12 | ADC | – Analog-to-Digital Converter |
| 13 | LED | – Light Emitting Diode |
| 14 | SPI | – Serial Peripheral Interface |
| 15 | TWI | –Two Wire Interface |
| 16 | IDE | – Integrated Development Environment |
| 17 | AAM | – Active Appearance Model |
| 18 | IMEI | – International Mobile Equipment Identity Number |

|  |  |  |
| --- | --- | --- |
| 19 | SIM | –Subscriber Identity Module |
| 20 | AT | – Attention |
| 21 | SMS | – Short Message Service |
| 22 | CR | – Carriage Return |
| 23 | LF | – Line Feed |
| 24 | MODEM | – Modulator -Demodulator |
| 25 | PC | – Personal Computer |

**CHAPTER -1**

**INTRODUCTION**

**Introduction**

The "Road Safe Alert System" is an innovative project designed to enhance road safety by providing real-time digital notifications to drivers during situations like manhole placements and road construction diversions. With increasing urbanization and road infrastructure development, it's crucial to mitigate potential hazards and accidents caused by sudden road disruptions. This system aims to address these challenges by leveraging modern technology and communication networks.

Traditional methods of warning drivers about road hazards often rely on static signs or manual flagging, which may not always be effective, especially in dynamic situations like temporary road diversions or sudden manhole placements. This project proposes a smarter approach by integrating IoT (Internet of Things) devices and mobile networks to deliver instant notifications directly to drivers' smartphones or vehicle consoles. By providing timely alerts, drivers can make informed decisions and adjust their routes or driving accordingly, reducing the risk of accidents and traffic congestion.

The system operates by deploying IoT sensors strategically along roadsides or construction sites to detect specific events such as manhole placements or road closures. These sensors are equipped with connectivity modules that transmit data to a centralized control system. When an event is detected, such as a manhole cover being opened or road construction activities beginning, the system automatically sends notifications to nearby vehicles equipped with compatible receivers. This proactive approach to road safety not only improves driver awareness but also contributes to overall traffic management and urban planning efforts.

#### Embedded system implementation

###### Introduction:

An embedded system is one kind of a computer system mainly designed to perform several tasks like to access, process, and store and also control the data in various electronics-based systems. Embedded systems are a combination of hardware and software where software is usually known as firmware that is embedded into the hardware. One of its most important characteristics of these systems is, it gives the o/p within the time limits. Embedded systems support to make the work more perfect and convenient. So, we frequently use embedded systems in simple and complex devices too. The applications of embedded systems mainly involve in our real life for several devices like microwave, calculators, TV remote control, home security and neighborhood traffic control systems, etc.

User interface

Embedded system

Hardware

Software

Inputs

Link to other systems

**Fig 1.2.1: Overview of embedded system**

#### Embedded system:

Embedded system includes mainly two sections, they are

1. Hardware

2. Software

Input devices interfacing

and driver circuits

Power supply and oscillator circuits

Timers

Memory

Output devices interfacing

###### Fig 1.2.2 Block diagram of embedded system

Interrupt controller

Parallel ports

Serial communication ports

Processor

Application specific circuits

* + 1. **Embedded System Hardware:**

As with any electronic system, an embedded system requires a hardware platform on which it performs the operation. Embedded system hardware is built with a microprocessor or microcontroller. The embedded system hardware has elements like input output (I/O) interfaces, user interface, memory and the display. Usually, an embedded system consists of:

* + - * Power Supply
      * Processor
      * Memory
      * Timers
      * Serial communication ports
      * Output/Output circuits
      * System application specific circuits

Embedded systems use different processors for its desired operation. Some of the processors used are

1. Microprocessor
2. Microcontroller
3. Digital signal processor **Microprocessor vs. Microcontroller Microprocessor**
   * **CPU** on a chip.
   * We can attach required amount of ROM, RAM and I/O ports.
   * Expensive due to external peripherals.
   * Large in size
   * general-purpose

**Microcontroller**

* + **Computer** on a chip
  + fixed amount of on-chip ROM, RAM, I/O ports
  + Low cost.
  + Compact in size.
  + Specific –purpose
    1. **Embedded System Software:**

The embedded system software is written to perform a specific function. It is typically written in a high-level format and then compiled down to provide code that can be lodged within a non- volatile memory within the hardware. An embedded system software is designed to keep in view of the three limits:

* + - * Availability of system memory
      * Availability of processor’s speed
      * When the system runs continuously, there is a need to limit power dissipation for events like stop, run and wake up.

**Bringing software and hardware together for embedded system:**

To make software to work with embedded systems we need to bring software and hardware together, for this purpose we need to burn our source code into microprocessor or microcontroller which is a hardware component and which takes care of all operations to be done by embedded system according to our code.

Generally, we write source codes for embedded systems in assembly language, but the processors run only executable files.

The process of converting the source code representation of your embedded software into an executable binary image involves three distinct steps:

* Each of the source files must be compiled or assembled into an object file
* All of the object files that result from the first step must be linked together to produce a single object file, called the re-locatable program.
* Physical memory addresses must be assigned to the relative offsets within the re-locatable program in a process called relocation.

The result of the final step is a file containing an executable binary image that is ready to run on the embedded system.

Source code

Assembler

Locator

Executable file

Linker

Processor

**Fig 1.2.2.1 Flow of burning source code to processor**

#### Applications:

Embedded systems have different applications. A few select [applications of embedded](https://www.elprocus.com/embedded-systems-real-time-applications/) [systems](https://www.elprocus.com/embedded-systems-real-time-applications/) are smart cards, telecommunications, satellites, missiles, digital consumer electronics, computer networking, etc.

[Embedded Systems in Automobiles](http://www.edgefx.in/importance-of-embedded-systems-in-automobiles-with-applications/)

* + - * Motor Control System
      * Engine or Body Safety
      * [Robotics](http://www.edgefx.in/top-list-robotics-projects-for-engineering-beginners/) in Assembly Line
      * Mobile and E-Com Access Embedded systems in Telecommunications
      * Mobile computing
      * Networking
      * [Wireless Communications](http://www.edgefx.in/multiple-input-and-multiple-output-mimo-wireless-communications/) Embedded Systems in Smart Cards
      * Banking
      * Telephone
      * [Security Systems](http://www.edgefx.in/microcontroller-based-projects-on-car-security-systems-using-gsm/)

#### Implementation flow:

###### Stage 1:

Considering the problems of existing methods and giving solution to that problem by considering the basic requirements for our proposed system

###### Stage 2:

Considering the hardware requirement for the proposed system For this we need to select the below components:

1. Microcontroller
2. Inputs for the proposed system (ex: sensors, drivers etc...,)
3. Outputs (ex: relays, loads)

###### Stage 3:

After considering hardware requirements, now we need to check out the software requirements. Based on the microcontroller we select there exists different software for coding, compiling, debugging. we need to write source code for that proposed system based on our requirements and compile, debug the code in that software.

After completing all the requirements of software and hardware we need to bring both together to work our system. For this we need to burn our source code into microcontroller, after burning our source code to microcontroller then connect all input and output modules as per our requirement.

#### CHAPTER-2

#### LITERATURE SURVEY

#### Literature Survey

**Bychkovsky, V., Chen, K., Goraczko, H., Hu, H., Hull, B., Miu, A., Shih, E., Zhang, Y., Madden S., and Balakrishnan, H; The cartel: a distributed mobile sensor computing system. In: 4th international conference on Embedded networked sensor systems, SenSys06, pp. 125-138. ACM, Boulder, Colorado, USA (November 2006).**

###### This paper presents "The Cartel," a distributed mobile sensor computing system. It likely discusses the design, implementation, and applications of the system, highlighting its contributions to the field of mobile sensor networks.

**Eriksson, J., Girod, L., Hull, B., Newton, R., Madden, S., and Balakrishnan H.; The pothole patrol: Using a mobile sensor network for road surface monitoring. In: Sixth Annual International conference on Mobile Systems, Applications and Services (MobiSys 2008). IEEE, Breckenridge, U.S.A. (June 2008).**

This paper focuses on using a mobile sensor network for monitoring road surfaces, with a specific emphasis on detecting potholes. It likely discusses the system architecture, data collection methods, and results related to road surface monitoring.

**Chen, K., Lu, M., Fan, X., Wei, M. and Wu, J.; Road Condition Monitoring Using On-board Three axis Accelerometer and GPS Sensor. In: sixth International ICST Conference on Communications and Networking. China (2011).**

This paper explores road condition monitoring techniques using on-board sensors, specifically three-axis accelerometers and GPS sensors. It may cover the methodology, data analysis, and findings related to road condition monitoring.

**Mednis, A., Strazdin, G., Zviedris, R., Kanonirs, G., and Selavo, L.; Real time pothole detection using android smartphone with accelerometers. In: International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS). IEEE (June 2011).**

This paper focuses on real-time pothole detection using Android smartphones equipped with accelerometers. It likely discusses the algorithm, implementation, and performance evaluation of the pothole detection system.

**Bhoraskar, R., Vankadhara, N., Raman, B., Kulkarni P.; Wolverine: Traffic and Road Condition estimation using Smartphone Sensors. In: Fourth International Conference on Communication Systems and Networks (COMSNETS). IEEE (January 2012).**

The paper introduces Wolverine, a system for estimating traffic and road conditions using smartphone sensors. It likely covers the system design, data processing techniques, and applications in traffic estimation.

**Singh, P., Juneja, N. and Kapoor, S.; Using Mobile Phone Sensors to Detect Driving Behavior. In: Third ACM Symposium on Computing for Development, Article No. 53. ACM, Bangalore, India (2013).**

This paper explores the use of mobile phone sensors to detect driving behavior. It may discuss the types of sensors used, feature extraction methods, and applications of detecting driving behavior in different contexts**.**

### EXISTING SYSTEM

The existing methods for ensuring road safety during manhole and road construction diversions typically rely on static road signs and manual flagging by traffic personnel. Road signs indicating diversions and potential hazards, such as open manholes, are placed at strategic locations. However, these static signs are not always effective in alerting drivers in real time, especially in adverse weather conditions or low visibility situations. Moreover, the presence of open manholes may not always be immediately visible to drivers, leading to potential road hazards and accidents. The dependence on manual flagging is labor-intensive and may not provide consistent and timely warnings to drivers. There is a clear need for a more advanced and automated solution, such as the proposed "Road Safe Alert System," to enhance road safety by providing real-time digital notifications and warnings to drivers, minimizing the risk of accidents and improving overall road safety measures.

#### Drawbacks in existing system:

* Dependency on Power Supply
* Limited Coverage Area
* Sensitivity to Environmental Conditions
* Initial Implementation Costs
* Maintenance and Calibration Requirements

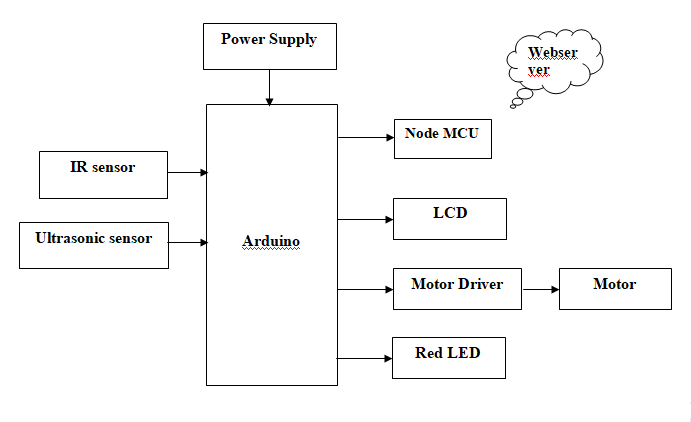
#### CHAPTER-3

#### PROPOSED SYSTEM

#### Proposed system

The proposed "Road Safe Alert System" introduces an advanced solution to significantly enhance road safety by providing real-time digital notifications during manhole hazards and road construction diversions. This system utilizes state-of-the-art technology, including IR sensors for manhole detection and servo motors for road diversion representation. When an open manhole is detected, the system activates a red LED indicator, instantly alerting approaching vehicles to the potential hazard. Simultaneously, the system employs a servo motor to indicate road diversions, ensuring drivers are informed of construction or roadwork zones ahead. Crucially, all detected manhole and road diversion information is uploaded to the Thing Speak cloud-based server, which acts as a digital notification platform. This information is instantly displayed on an LCD screen within the vehicle, providing drivers with real-time updates about potential road hazards and diversions. This comprehensive approach not only prioritizes road safety but also showcases the potential of technology to create a safer and more informed driving environment.

#### Block Diagram:

****

**Fig 3.1.1 Block diagram of proposed system.**

##### **Node MCU:**

##### The ESP8266 Node MCU CP2102 board has ESP8266 which is a highly integrated chip designed for the needs of a new connected world. It offers a complete and self-contained Wi-Fi networking solution, allowing it to either host the application or to offload all Wi-Fi networking functions from another application processor

##### **Infrared sensor:**

IR sensor is an electronic device, that emits the light in order to sense some object of the surroundings. An IR sensor can measure the heat of an object as well as detects the motion. Usually, in the infrared spectrum, all the objects radiate some form of thermal radiation. These types of radiations are invisible to our eyes, but infrared sensor can detect these radiations.

##### **Arduino Controller:**

##### Responsible for coordinating the operation of various components and executing the authentication process. The Arduino controller receives signals from the infrared sensor, processes the data, and transmits serial information to the PC when an intrusion is detected.

**3.1.1.5 Motor Driver:**

A motor driver is an integrated circuit chip which is usually used to control motors in autonomous robots. Motor driver act as an interface between Arduino and the motors. The most commonly used motor driver IC’s are from the L293 series such as L293D, L293NE, etc. These ICs are designed to control 2 DC motors simultaneously. L293D consist of two H-bridge. H-bridge is the simplest circuit for controlling a low current rated motor. We will be referring the motor driver IC as L293D only. L293D has 16 pins.

##### **Ultrasonic sensor:**

It emits an ultrasound at 40 000 Hz which travels through the air and if there is an object or obstacle on its path It will bounce back to the module. Considering the travel time and the speed of the sound you can calculate the distance.

##### **LED:**

The light emitting diode simply, we know as a diode. When the diode is forward biased, then the electrons & holes are moving fast across the junction and they are combining constantly, removing one another out. Soon after the electrons are moving from the n-type to the p-type silicon, it combines with the holes, then it disappears. Hence it makes the complete atom & more stable and it gives the little burst of energy in the form of a tiny packet or photon of light.

##### **Power supply:**

##### The power supply serves as the lifeline, providing the necessary electrical energy to all components for their proper functioning. It ensures continuous and stable power distribution to components such as the Infrared sensor, Arduino microcontroller, GSM module, Camera, without a reliable power supply, the system would be rendered ineffective, unable to perform authentication processes, trigger alert mechanisms, or transmit data to the central server. Therefore, a robust and dependable power supply is essential to maintain the integrity and functionality of the security system, ensuring uninterrupted operation and effective protection against unauthorized access or intruder detection.

##### **LCD:**

LCD (Liquid Crystal Display) is the innovation utilized in scratch pad shows and other littler PCs. Like innovation for light-producing diode (LED) and gas-plasma, LCDs permit presentations to be a lot more lender than innovation for cathode beam tube (CRT). LCDs expend considerably less power than LED shows and gas shows since they work as opposed to emanating it on the guideline of blocking light.

# CHAPTER-4

# HARDWAREREQUIREMENTS

#### Arduino:

Arduino Uno is a very valuable addition in the electronics that consists of USB interface, 14 digital I/O pins, 6 analog pins, and Atmega328 microcontroller. It also supports serial communication using Tx and Rx pins.

There are many versions of Arduino boards introduced in the market like Arduino Uno, Arduino Due, Arduino Leonardo, Arduino Mega, however, most common versions are Arduino Uno and Arduino Mega. If you are planning to create a project relating to digital electronics, embedded system, robotics, or IoT, then using Arduino Uno would be the best, easy and most economical option.

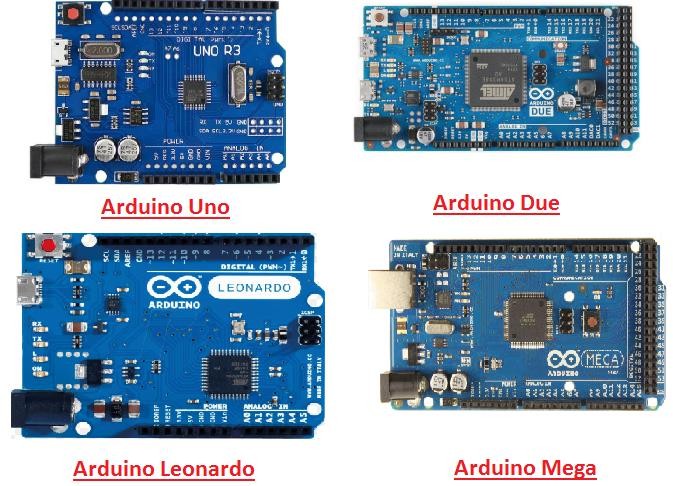


Fig 4.1 Types of Arduinos

It is an open-source platform, means the boards and software are readily available and anyone can modify and optimize the boards for better functionality.

The software used for Arduino devices is called IDE (Integrated Development Environment) which is free to use and required some basic skills to learn it. It can be programmed using C and C++ language.

Some people get confused between **Microcontroller and Arduino**. While former is just an on system 40 pin chip that comes with a built-in microprocessor and later is a board that comes with the microcontroller in the base of the board, bootloader and allows easy access to input-output pins and makes uploading or burning of the program very easy.

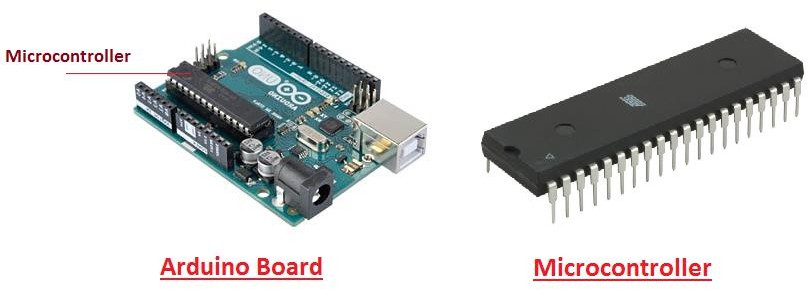
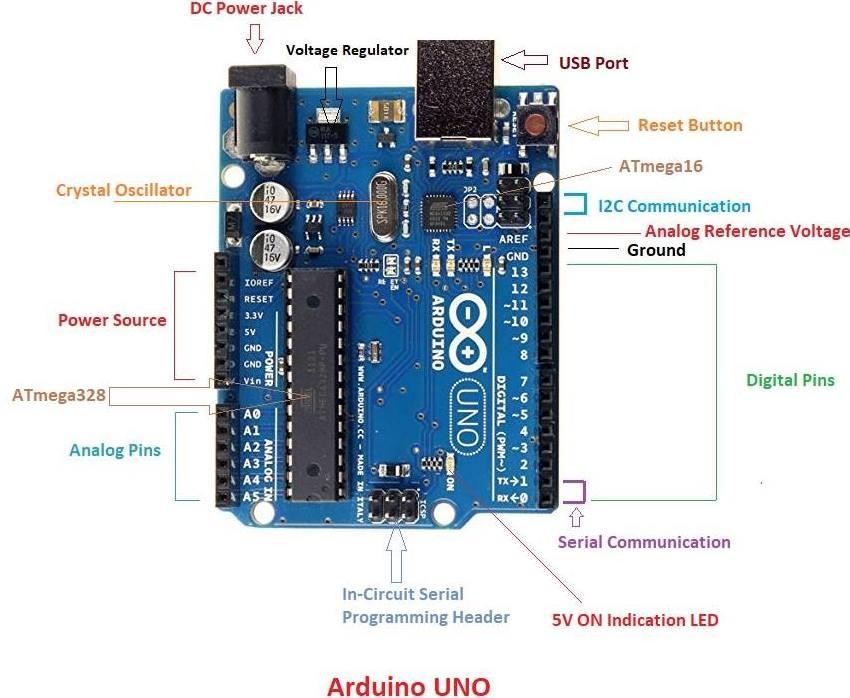


Fig 4.2 Arduino Board Fig 4.3 Microcontroller

While learning microcontroller requires some expertise and skills. Nevertheless, we can say every Arduino is basically a [microcontroller](https://www.theengineeringprojects.com/2018/03/introduction-to-microcontrollers.html) but not every microcontroller is an Arduino.

#### Introduction to Arduino

* + - * **Arduino Uno** is a microcontroller board developed by Arduino.cc which is an open-source electronics platform mainly based on AVR microcontroller Atmega328.
      * First Arduino project was started in Interaction Design Institute Ivrea in 2003 by David Cuartillas and Massimo Banzi with the intention of providing a cheap and flexible way to students and professional for controlling a number of devices in the real world.
      * The current version of Arduino Uno comes with USB interface, 6 analog input pins, 14 I/O digital ports that are used to connect with external electronic circuits. Out of 14 I/O ports, 6 pins can be used for PWM output.
      * It allows the designers to control and sense the external electronic devices in the real world



###### Fig 4.1.1.1 Arduino UNO

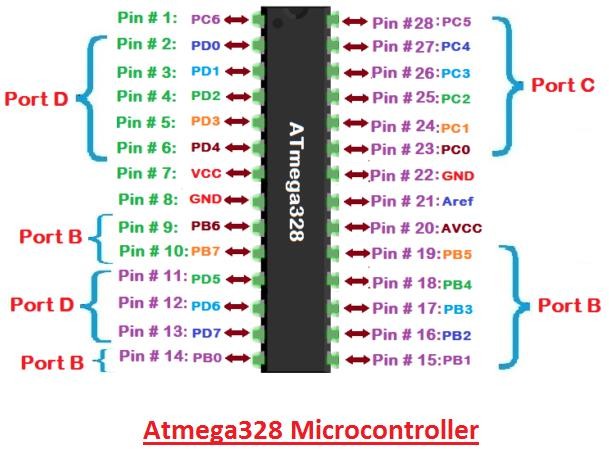
* + - * This board comes with all the features required to run the controller and can be directly connected to the computer through USB cable that is used to transfer the code to the controller

using IDE (Integrated Development Environment) software, mainly developed to program Arduino. IDE is equally compatible with Windows, MAC or Linux Systems; however, Windows is preferable to use. Programming languages like C and C++ are used in IDE.

* + - * Apart from USB, battery or AC to DC adopter can also be used to power the board.
      * Arduino Uno boards are quite similar to other boards in Arduino family in terms of use and functionality, however, Uno boards don’t come with FTDI USB to Serial driver chip.
      * There are many versions of Uno boards available, however, Arduino Nano V3 and Arduino Uno are the most official versions that come with Atmega328 8-bit AVR Atmel microcontroller where RAM memory is 32KB.
      * When nature and functionality of the task go complex, Mirco SD card can be added in the boards to make them store more information.

#### Features of Arduino

* + - * Arduino Uno comes with USB interface i.e. USB port is added on the board to develop serial communication with the computer.
      * [Atmega328](https://www.theengineeringprojects.com/2017/08/introduction-to-atmega328.html) microcontroller is placed on the board that comes with a number of features like timers, counters, interrupts, PWM, CPU, I/O pins and based on a 16MHz clock that helps in producing more frequency and number of instructions per cycle.

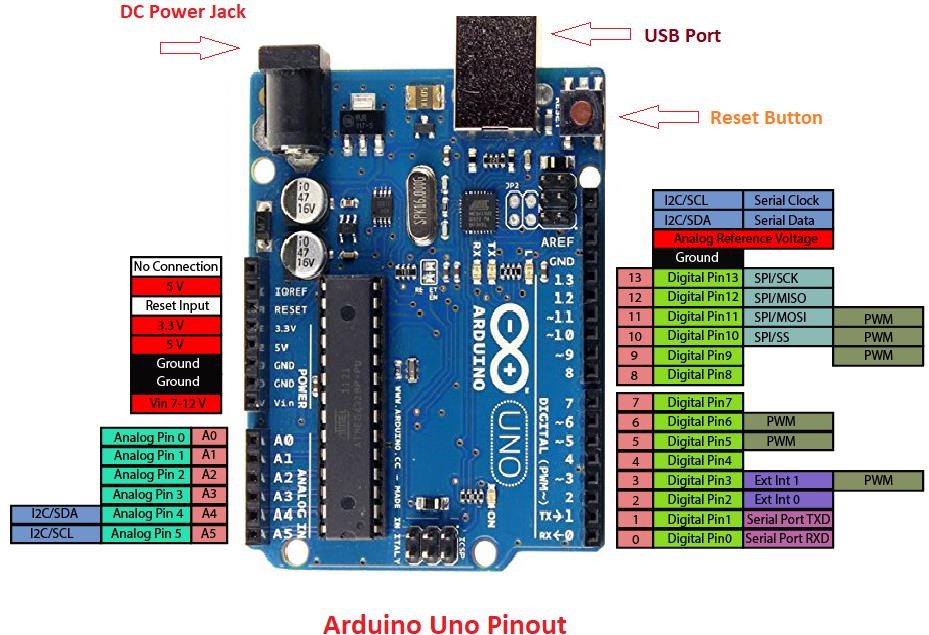


###### Fig 4.1.2.1 Atmega328 Microcontroller

* + - * It is an open-source platform where anyone can modify and optimize the board based on the number of instructions and task they want to achieve.
      * This board comes with a built-in regulation feature which keeps the voltage under control when the device is connected to the external device.
      * Reset pin is added in the board that reset the whole board and takes the running program in the initial stage. This pin is useful when board hangs up in the middle of the running program; pushing this pin will clear everything up in the program and starts the program right from the beginning.
      * There are 14 I/O digital and 6 analog pins incorporated in the board that allows the external connection with any circuit with the board. These pins provide the flexibility and ease of use to the external devices that can be connected through these pins. There is no hard and fast interface required to connect the devices to the board. Simply plug the external device into the pins of the board that are laid out on the board in the form of the header.
      * The 6 analog pins are marked as A0 to A5 and come with a resolution of 10bits. These pins measure from 0 to 5V, however, they can be configured to the high-range using analog Reference () function and AREF pin.
      * 13KB of flash memory is used to store number of instructions in form of code.
      * Only 5 V is required to turn the board on, which can be achieved directly using USB port or external adopter, however, it can support external power source up to 12 V which can be regulated and limit to 5 V or 3.3 V based on the requirement of the project.

#### Arduino Pinout

* + - * Arduino Uno is based on AVR microcontroller called Atmega328. This controller comes with 2KB SRAM, 32KB of flash memory, 1KB of EEPROM. Arduino Board comes with 14 digital pins and 6 analog pins. ON-chip ADC is used to sample these pins. A 16 MHz frequency crystal oscillator is equipped on the board. Following figure shows the pinout of the Arduino Uno Board



**Fig 4.1.3.1 Arduino Uno Pinout**

#### Pin Description:

There are several I/O digital and analog pins placed on the board which operates at 5V. These pins come with standard operating ratings ranging between 20mA to 40mA. Internal pull-up resistors are used in the board that limits the current exceeding from the given operating conditions. However, too much increase in current makes these resisters useless and damages the device.

**LED.** Arduino Uno comes with built-in LED which is connected through pin 13. Providing HIGH value to the pin will turn it ON and LOW will turn it OFF.

**Vin.** It is the input voltage provided to the Arduino Board. It is different than 5 V supplied through a USB port. This pin is used to supply voltage. If a voltage is provided through power jack, it can be accessed through this pin.

**5V.** This board comes with the ability to provide voltage regulation. 5V pin is used to provide output regulated voltage. The board is powered up using three ways i.e. USB, Vin pin of the board or DC power jack.

USB supports voltage around 5V while Vin and Power Jack support a voltage range between 7V to 20V. It is recommended to operate the board on 5V. It is important to note that, if a voltage is supplied through 5V or 3.3V pins, they result in bypassing the voltage regulation that can damage the board if voltage surpasses from its limit.

**GND.** These are ground pins. More than one ground pins are provided on the board which can be used as per requirement.

**Reset.** This pin is incorporated on the board which resets the program running on the board. Instead of physical reset on the board, IDE comes with a feature of resetting the board through programming.

**IOREF.** This pin is very useful for providing voltage reference to the board. A shield is used to read the voltage across this pin which then select the proper power source.

**PWM.** PWM is provided by 3, 5, 6,9,10, 11pins. These pins are configured to provide 8-bit output PWM.

**PI.** It is known as Serial Peripheral Interface. Four pins 10(SS), 11(MOSI), 12(MISO), 13(SCK) provide SPI communication with the help of SPI library.

**AREF.** It is called Analog Reference. This pin is used for providing a reference voltage to the analog inputs.

**TWI.** It is called Two-wire Interface. TWI communication is accessed through Wire Library. A4 and A5 pins are used for this purpose.

**Serial Communication.** Serial communication is carried out through two pins called Pin 0 (Rx) and Pin 1 (Tx).

Rx pin is used to receive data while Tx pin is used to transmit data.

**External Interrupts.** Pin 2 and 3 are used for providing external interrupts. An interrupt is called by providing LOW or changing value

* + 1. **Arduino Uno Technical Specifications**

|  |  |
| --- | --- |
| Microcontroller | [ATmega328P](https://components101.com/microcontrollers/atmega328p-pinout-features-datasheet) – 8 Bit AVR family microcontroller |
| Operating Voltage | 5V |
| Recommended Input Voltage | 7-12V |
| Input Voltage Limits | 6-20V |
| Analog Input Pins | 6 (A0 – A5) |
| Digital I/O Pins | 14 (Out of which 6 provide PWM output) |
| DC Current on I/O Pins | 40 mA |
| DC Current on 3.3V Pin | 50 mA |
| Flash Memory | 32 KB (0.5 KB is used for Bootloader) |
| SRAM | 2 KB |
| EEPROM | 1 KB |
| Frequency (Clock Speed) | 16 MHz |

**Table 4.1.5.1 Arduino UNO technical specifications**

#### Communication and Programming:

Arduino Uno comes with an ability of interfacing with other other Arduino boards, microcontrollers and computer. The Atmega328 placed on the board provides serial communication using pins like Rx and Tx.

The Atmega16U2 incorporated on the board provides a pathway for serial communication using USB com drivers. Serial monitor is provided on the IDE software which is used to send or receive text data from the board. If LEDs placed on the Rx and Tx pins will flash, they indicate the transmission of data.

Arduino Uno is programmed using Arduino Software which a cross-platform application called IDE is written in Java. The AVR microcontroller Atmega328 laid out on the base comes with built- in boot loader that sets you free from using a separate burner to upload the program on the board.



**Fig 4.1.6.1 Personal computer**

#### Applications:

Arduino Uno comes with a wide range of applications. A larger number of people are using Arduino boards for developing sensors and instruments that are used in scientific research. Following are some main applications of the board.

* + - * [Embedded System](https://www.theengineeringprojects.com/2016/10/what-is-embedded-systems.html)
      * Security and Defense System
      * Digital Electronics and Robotics
      * Parking Lot Counter
      * Weighing Machines
      * Traffic Light Count Down Timer
      * Medical Instrument
      * Emergency Light for Railways
      * Home Automation
      * Industrial Automation

There are a lot of other microcontrollers available in the market that are more powerful and cheap as compared to Arduino board. So, why you prefer Arduino Uno?

Actually, Arduino comes with a big community that is developing and sharing the knowledge with a wide range of audience. Quick support is available pertaining to technical aspects of any electronic project. When you decide Arduino board over other controllers, you don’t need to arrange extra peripherals and devices as most of the functions are readily available on the board that makes your project economical in nature and free from a lot of technical experiments.

###### LCD

LCD (Liquid Crystal Display) is the innovation utilized in scratch pad shows and other littler PCs. Like innovation for light-producing diode (LED) and gas-plasma, LCDs permit presentations to be a lot more slender than innovation for cathode beam tube (CRT). LCDs expend considerably less power than LED shows and gas shows since they work as opposed to emanating it on the guideline of blocking light.

A LCD is either made with a uninvolved lattice or a showcase network for dynamic framework show. Likewise alluded to as a meager film transistor (TFT) show is the dynamic framework LCD. The uninvolved LCD lattice has a matrix of conductors at every crossing point of the network with pixels. Two conductors on the lattice send a current to control the light for any pixel. A functioning framework has a transistor situated at every pixel crossing point, requiring less current to control the luminance of a pixel.

Some aloof network LCD's have double filtering, which implies they examine the matrix twice with current in the meantime as the first innovation took one sweep. Dynamic lattice, be that as it may, is as yet a higher innovation.

A 16x2 LCD show is an essential module that is generally utilized in various gadgets and circuits. These modules more than seven sections and other multi fragment LEDs are liked. The reasons being: LCDs are affordable; effectively programmable; have no restriction of showing exceptional and even custom characters (not at all like in seven fragments), movements, etc.

A 16x2 LCD implies 16 characters can be shown per line and 2 such lines exist. Each character is shown in a lattice of 5x7 pixels in this LCD. There are two registers in this LCD, in particular Command and Data.

The directions given to the LCD are put away by the order register. An order is a direction given to LCD to play out a predefined assignment, for example, introducing it, clearing its screen, setting the situation of the cursor, controlling presentation, and so forth. The information register will store the information that will be shown on the LCD. The information is the character's ASCII incentive to show on the LCD.

**4.2.1 Data/Signals/Execution of LCD**

Now that was all about the signals and the hardware. Let us come to data, signals and execution.

Two types of signals are accepted by LCD, one is data and one is control. The LCD module recognizes these signals from the RS pin status. By pulling the R / W pin high, data can now also be read from the LCD display. Once the E pin has been pulsed, the LCD display reads and executes data at the falling edge of the pulse, the same for the transmission case.

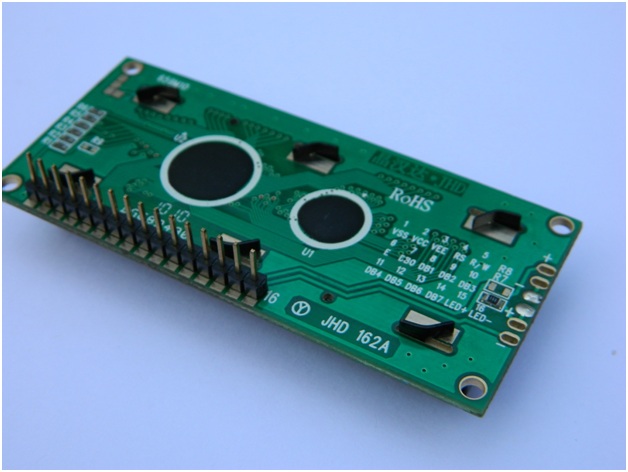
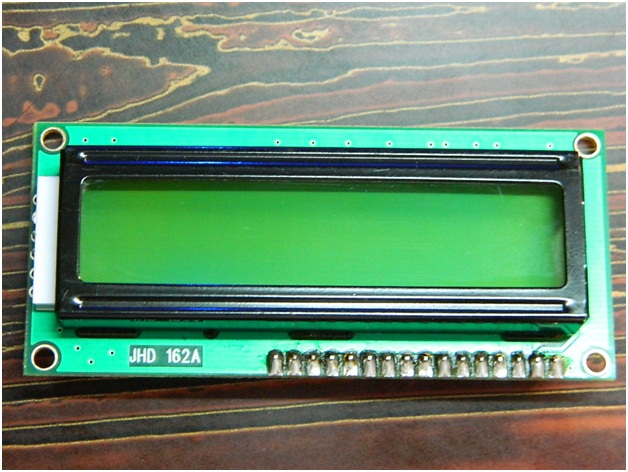
It takes 39-43μS for the LCD display to place a character or execute a command. It takes 1.53ms to 1.64ms except for clearing display and searching for cursor to the home position.

Any attempt to send data before this interval may result in failure in some devices to read data or execute the current data. Some devices compensate for the speed by storing some temporary registers with incoming data.

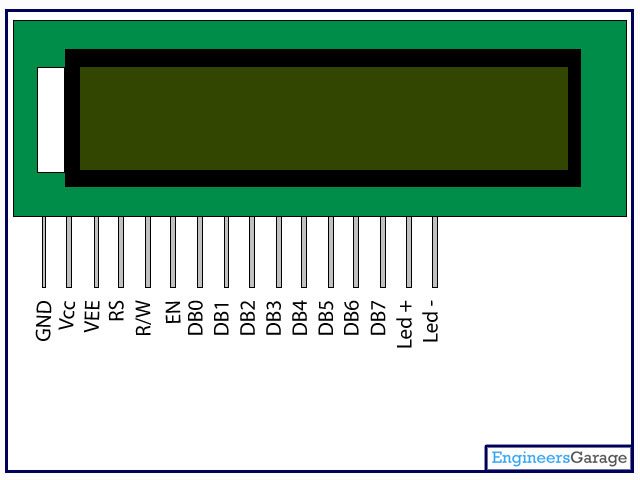
There are two RAMs for LCD displays, namely DDRAM and CGRAM. DDRAM registers the position in which the character would be displayed in the ASCII chart. Each DDRAM byte represents every single position on the display of the LCD.

The DDRAM information is read by the LCD controller and displayed on the LCD screen. CGRAM enables users to define their personalized characters. Address space is reserved for users for the first 16 ASCII characters.

Users can easily display their custom characters on the LCD screen after CGRAM has been set up to display characters.

[](http://www.circuitstoday.com/wp-content/uploads/2012/02/lcd-display-back-side.jpg)[](http://www.circuitstoday.com/wp-content/uploads/2012/02/LCD-Display-Front-Side.jpg)

**Fig 4.2.1.1 LCD – Back View Fig 4.2.1.2** **LCD – Front View**



**Fig 4.2.1.3** **Pin Diagram**

**4.2.2 RS (Register select)**

A 16X2 LCD has two order and information registers. The determination of the register is utilized to change starting with one register then onto the next. RS=0 for the register of directions, while RS=1 for the register of information.

**4.2.3 Command Register**

The guidelines given to the LCD are put away by the direction register. An order is a direction given to LCD to play out a predefined assignment, for example, instating it, clearing its screen, setting the situation of the cursor, controlling showcase, and so on. Order preparing happens in the direction register.

**4.2.3 Data Register:**

The information register will store the information that will be shown on the LCD. The information is the character's ASCII incentive to show on the LCD. It goes to the information register and is prepared there when we send information to the LCD. While choosing RS=1, the information register

**4.2.4 Pin Description:**

|  |  |  |
| --- | --- | --- |
| Pin No | Function | Name |
| 1 | Ground (0V) | Ground |
| 2 | Supply voltage; 5V (4.7V – 5.3V) | Vcc |
| 3 | Contrast adjustment; through a variable resistor | VEE |
| 4 | Selects command register when low; and data register when high | Register Select |
| 5 | Low to write to the register; High to read from the register | Read/write |
| 6 | Sends data to data pins when a high to low pulse is given | Enable |
| 7 | 8-bit data pins | DB0 |
| 8 | DB1 |
| 9 | DB2 |
| 10 | DB3 |
| 11 | DB4 |
| 12 | DB5 |
| 13 | DB6 |
| 14 | DB7 |
| 15 | Backlight VCC (5V) | Led+ |
| 16 | Backlight Ground (0V) | Led- |

**Table 4.2.4.1 Pin Functions**

**4.2.5 LCD Commands:**

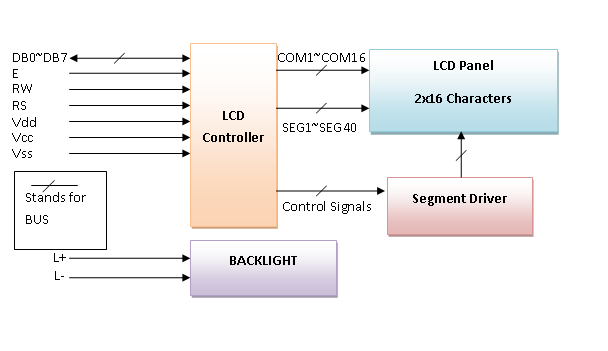
There are some preset commands in the LCD that we need to send to the LCD via some microcontroller. The following are some important command instructions:

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Hex Code** | **Command to LCD instruction Register** |
| 1 | 01 | Clear display screen |
| 2 | 02 | Return home |
| 3 | 04 | Decrement cursor (shift cursor to left) |
| 4 | 06 | Increment cursor (shift cursor to right) |
| 5 | 05 | Shift display right |
| 6 | 07 | Shift display left |
| 7 | 08 | Display off, cursor off |
| 8 | 0A | Display off, cursor on |
| 9 | 0C | Display on, cursor off |
| 10 | 0E | Display on, cursor blinking |
| 11 | 0F | Display on, cursor blinking |
| 12 | 10 | Shift cursor position to left |
| 13 | 14 | Shift cursor position to right |
| 14 | 18 | Shift the entire display to the left |
| 15 | 1C | Shift the entire display to the right |
| 16 | 80 | Force cursor to beginning ( 1st line) |
| 17 | C0 | Force cursor to beginning ( 2nd line) |
| 18 | 38 | 2 lines and 5×7 matrix |

## Command codes for LCD

**Table 4.2.5.1 Hex code and LCD Commands**

**4.2.6 Block Diagram of LCD Display:**

**[](http://www.circuitstoday.com/wp-content/uploads/2012/02/LCD-Display-Block-Diagram.png)**

#### Fig 4.2.6.1 Block Diagram of LCD Display

**4.2.7 Control and display commands**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Instruction** | **Instruction Code** | | | | | | | | | | **Instruction Code Description** | **Execution time** |
| RS | R/W | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 |
| Read Data From RAM | 1 | 1 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | Read data from internal RAM | 1.53-1.64ms |
| Write data to RAM | 1 | 0 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | Write data into internal RAM (DDRAM/CGRAM) | 1.53-1.64ms |
| Busy flag & Address | 0 | 1 | BF | AC6 | AC5 | AC4 | AC3 | AC2 | AC1 | AC0 | Busy flag (BF: 1→ LCD Busy) and contents of address counter in bits AC6-AC0. | 39 µs |

#### Fig 4.2.7.1 Control & Display commands

**4.2.8 4-bit and 8-bit Mode of LCD:**

#### The LCD can work in two striking modes, the 4-bit mode and the 8-bit mode. We send the information snack through snack in 4 bit mode, first upper chomp, by then lower snack. For those of you who don't have the foggiest idea what a goody is: a chomp is a four-piece gathering, so a byte's lower four bits (D0-D3) are the lower snack, while a byte's upper four bits (D4-D7) are the higher snack. This enables us to send 8bit data. This connects with us to send 8bit data.

**4.3 Node MCU:**

The ESP8266 Node-MCU CP2102 board has ESP8266 which is a highly integrated chip designed for the needs of a new connected world. It offers a complete and self-contained Wi-Fi networking solution, allowing it to either host the application or to offload all Wi-Fi networking functions from another application processor.

ESP8266 has powerful on-board processing and storage capabilities that allow it to be integrated with the sensors and other application-specific devices through its GPIOs with minimal development up-front and minimal loading during runtime. Its high degree of on-chip integration allows for minimal external circuitry, and the entire solution, including the front-end module, is designed to occupy minimal PCB area.

The ESP8266 Node-MCU development board – a true plug-and-play solution for inexpensive projects using Wi-Fi. The module arrives pre-flashed with Node-MCU firmware so they’re ready to go – just install your USB driver (below). ESP-12 Lua Node-MCU WIFI Dev Board Internet Of Things board contains a full ESP8266 Wi-Fi module with all the GPIO broken out, a full USB-serial interface, and a power supply all on the one breadboard-friendly package.

This board is pre-flashed with Node-MCU – a Lua-based firmware for the ESP8266 which allows easy control via a neat scripting language – Lua – so you’re ready to go in just a few minutes.

The ESP-12 Lua Node-MCU WIFI Dev Board Internet Of Things with ESP8266 is an all-in-one microcontroller + Wi-Fi platform that is very easy to use to create projects with Wi-Fi and IoT (Internet of Things) applications.

The board is based on the highly popular ESP8266 Wi-Fi Module chip with the ESP-12 SMD footprint. This Wi-Fi development board already embeds in its board all the necessary components for the ESP8266 (ESP-12E) to program and upload code. It has a built-in USB to serial chip upload codes, 3.3V regulator, and logic level converter circuit so you can immediately upload codes and connect your circuits.

**4.4 Features:**

* Open-source, Interactive, Programmable, Low cost, Simple, Smart, WI-FI enabled
* Arduino-like hardware IO
* Integrated TR switch, balun, LNA, power amplifier and matching network
* Integrated PLL, regulators, DCXO and power management units
* Onboard USB to serial chip to easily program and upload codes from the Arduino IDE
* Embeds logic level converter circuits
* Has onboard 3.3V regulator to ensure enough power to function as your go-to Wi-Fi chip!
* Easy access to the GPIO pins for easy prototyping
* ESP-12E Processor
* Easy to use breadboard friendly form factor



#### Fig 4.4.1 Node MCU

## **4.5 DC MOTOR:**

## **4.5.1What is a DC Motor?**

## A direct current (DC) motor is a type of electric machine that converts electrical energy into mechanical energy. DC motors take electrical power through direct current, and convert this energy into mechanical rotation.

## DC motors use magnetic fields that occur from the electrical currents generated, which powers the movement of a rotor fixed within the output shaft. The output torque and speed depends upon both the electrical input and the design of the motor.

## **4.5.2 How DC motors work**

## The term ‘DC motor’ is used to refer to any rotary electrical machine that converts direct current electrical energy into mechanical energy. DC motors can vary in size and power from small motors in toys and appliances to large mechanisms that power vehicles, pull elevators and hoists, and drive steel rolling mills.

## **4.5.3 But how do DC motors work**?

## DC motors include two key components: a **stator** and an **armature**. The stator is the stationary part of a motor, while the armature rotates. In a DC motor, the stator provides a rotating magnetic field that drives the armature to rotate.

## A simple DC motor uses a stationary set of magnets in the stator, and a coil of wire with a current running through it to generate an electromagnetic field aligned with the center of the coil. One or more windings of insulated wire are wrapped around the core of the motor to concentrate the magnetic field.

## The windings of insulated wire are connected to a commutator (a rotary electrical switch), that applies an electrical current to the windings. The commutator allows each armature coil to be energised in turn, creating a steady rotating force (known as torque).

## When the coils are turned on and off in sequence, a rotating magnetic field is created that interacts with the differing fields of the stationary magnets in the stator to create torque, which causes it to rotate. These key operating principles of DC motors allow them to convert the electrical energy from direct current into mechanical energy through the rotating movement, which can then be used for the propulsion of objects.

## **4.5.4 Who invented the DC motor?**

## This amazing piece of electrical equipment has revolutionized our lives in many ways, but who invented the DC motor? As with all major innovations, there are many people who had a role to play through the development of similar mechanisms.

## In the US, Thomas Davenport is widely celebrated as the inventor of the first electric motor, and un-doubted he was the first to patent a useable electric motor in 1837. Davenport, however, was not the first person to build an electric motor, with various inventors in Europe having already developed more powerful versions by the time Davenport filed his patent.

## In 1834, Moritz Jacobi had presented a motor that was three times as powerful as the one Davenport would later patent, while Sibrandus Stratingh and Christopher Becker were the first to demonstrate a practical application for an electric motor, by running a small model car in 1835.

#### C:\Users\khajamoddint\Downloads\dc motor.jpg

#### Fig 4.5.1 Motor

**4.6 Light Emitting Diodes (LEDs):**

The Light emitting diode is a two-lead semiconductor light source. In 1962, Nick Holonyak has come up with an idea of light emitting diode, and he was working for the general electric company. The LED is a special type of diode and they have similar electrical characteristics of a PN junction diode. Hence the LED allows the flow of current in the forward direction and blocks the current in the reverse direction. The LED occupies the small area which is less than the **1 mm2**. [The applications of LEDs](https://www.elprocus.com/future-of-led-lighting-and-leds/) used to make various electrical and electronic projects. In this article, we will discuss the working principle of the LED and its applications.

## **4.6.1 What is a Light Emitting Diode?**

The lighting emitting diode is a [p-n junction diode](https://www.elprocus.com/p-n-junction-diode-theory-and-working/). It is a specially doped diode and made up of a special type of semiconductors. When the light emits in the forward biased, then it is called as a light emitting diode.



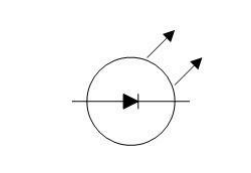
## **Fig 4.6.1 LED**

## **4.6.2 How does the Light Emitting Diode work?**

The light emitting diode simply, we know as a diode. When the diode is forward biased, then the electrons & holes are moving fast across the junction and they are combining constantly, removing one another out. Soon after the electrons are moving from the n-type to the p-type silicon, it combines with the holes, then it disappears. Hence it makes the complete atom & more stable and it gives the little burst of energy in the form of a tiny packet or photon of light.

#### 

#### Fig 4.6.2 Working of LED

****

#### Fig 4.6.3 Schematic

* A Light emitting diode (LED) is essentially a pn junction diode. When carriers are injected across a forward-biased junction, it emits incoherent light.
* Most of the commercial LEDs are realized using a highly doped n and a p Junction.

**4.6.1 LED Materials:**

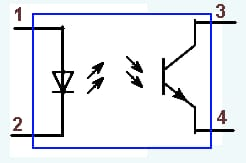
A critical class of business LEDs that cover the unmistakable range. Ternary composites in view of alloying GaAs and GaP which are signified by GaAs1-yPy. InGaAlP is a case of a quarternary (four component) III-V compound with an immediate band crevice. The LEDs acknowledged utilizing two diversely doped semiconductors that are the same material is known as a homojunction. When they are acknowledged utilizing diverse bandgap materials they are known as a heterostructure gadget heterostructure LED is brighter than a homo-Junction LED.

**4.6.2 LED Structure:**

The LED structure assumes a vital part in radiating light from the LED surface. The LEDs are organized to guarantee the majority of the re-combinations happens at first glance by the accompanying two ways. • By expanding the doping grouping of the substrate, so that extra free minority charge transporters electrons move to the top, recombine and emanate light at the surface. • By expanding the dissemination length L = √ Dτ, where D is the dispersion coefficient and τ is the bearer life time. In any case, when expanded past a basic length there is a possibility of re-retention of the photons into the gadget. The LED must be organized so that the photons produced from the gadget are transmitted without being reabsorbed. One arrangement is to make the p layer on the top, sufficiently flimsy to make an exhaustion layer. Taking after picture demonstrates the layered structure. There are diverse approaches to structure the vault for proficient transmitting.

LEDs are normally based on a n-sort substrate, with a terminal joined to the p-sort layer saved on its surface. P-sort substrates, while less basic, happen too. Numerous business LEDs, particularly GaN/InGaN, likewise utilize sapphire substrate. Driven productivity: A vital metric of a LED is the outside quantum proficiency next. It measures the efficiency of the transformation of electrical vitality into transmitted optical vitality. It is characterized as the light yield isolated by the electrical information power.

It is likewise characterized as the result of Internal radiative proficiency and Extraction productivity. ηext = Pout(optical)/IV For backhanded band gap semiconductors next is for the most part under 1%, where concerning an immediate band hole material it could be generous. ηint = rate of radiation recombination/Total recombination. The interior effectiveness is an element of the nature of the material.



**Fig 4.6.2.1 Optocoupler schematic showing LED and phototransistor**

**4.6.3 Applications:**

LED have a lot of applications. Following are few examples.

* + - * + Devices, medical applications, clothing, toys
        + Remote Controls (TVs, VCRs)
        + Lighting
        + Indicators and signs
        + Optoisolators and optocouplers
        + Swimming pool lighting

**4.6.4 Advantages of using LEDs:**

LEDs deliver more light per watt than radiant knobs; this is valuable in battery fueled or vitality sparing gadgets.

LEDs can radiate light of a planned shading without the utilization of shading channels that customary lighting techniques require. This is more effective and can bring down introductory expenses.

The strong bundle of the LED can be intended to center its light. Glowing and fluorescent sources frequently require an outside reflector to gather light and direct it in a usable way.

When utilized as a part of utilizations where darkening is required, LEDs don't change their shading tint as the present going through them is brought down, not at all like radiant lights, which turn yellow.

LEDs are perfect for use in applications that are liable to visit on-off cycling, not at all like fluorescent lights that copy out all the more immediately when cycled much of the time, or High Intensity Discharge (HID) lights that require quite a while before restarting.

LEDs, being strong state segments, are hard to harm with outer stun. Fluorescent and radiant globules are effectively broken if dropped on the ground.

LEDs can have a moderately long valuable life. A Philips LUXEON k2 LED has an existence time of around 50,000 hours, though Fluorescent tubes commonly are appraised at around 30,000 hours, and brilliant lights at 1,000–2,000 hours.

LEDs generally fall flat by darkening after some time, as opposed to the unexpected wear out of brilliant globules.

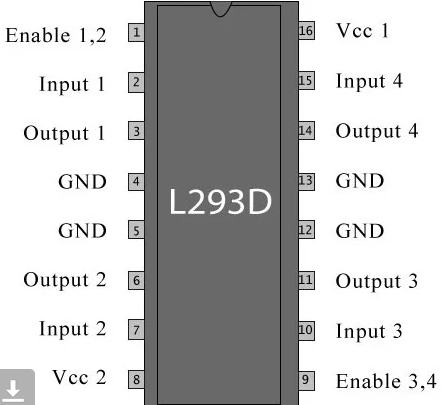
LEDs illuminate rapidly. A run of the mill red marker LED will accomplish full brilliance in microseconds; Philips Lumileds specialized datasheet DS23 for the Luxeon Star states "under 100ns." LEDs utilized as a part of specialized gadgets can have significantly quicker reaction times.

LEDs can be little and are effortlessly populated onto printed circuit sheets.

LEDs don't contain mercury, not at all like conservative fluorescent lights.

**4.7 Motor Driver:**

A motor driver is an integrated circuit chip which is usually used to control motors in autonomous robots. Motor driver act as an interface between Arduino and the motors. The most commonly used motor driver IC’s are from the L293 series such as L293D, L293NE, etc. These ICs are designed to control 2 DC motors simultaneously. L293D consist of two H-bridge. H-bridge is the simplest circuit for controlling a low current rated motor. We will be referring the motor driver IC as L293D only. L293D has 16 pins.



**Fig 4.7.1 L293D**

|  |  |  |
| --- | --- | --- |
| **Pin Number** | **Pin Name** | **Description** |
| 1 | Enable 1,2 | This pin enables the input pin Input 1(2) and Input 2(7) |
| 2 | Input 1 | Directly controls the Output 1 pin. Controlled by digital circuits |
| 3 | Output 1 | Connected to one end of  Motor 1 |
| 4 | Ground | Ground pins are connected to ground of circuit (0V) |
| 5 | Ground | Ground pins are connected to ground of circuit (0V) |
| 6 | Output 2 | Connected to another end of  Motor 1 |
| 7 | Input 2 | Directly controls the Output 2 pin. Controlled by digital circuits |
| 8 | Vcc2 (Vs) | Connected to Voltage pin for running motors (4.5V to 36V) |
| 9 | Enable 3,4 | This pin enables the input pin Input 3(10) and Input 4(15) |
| 10 | Input 3 | Directly controls the Output 3 pin. Controlled by digital circuits |
| 11 | Output 3 | Connected to one end of Motor 2 |
| 12 | Ground | Ground pins are connected to ground of circuit (0V) |
| 13 | Ground | Ground pins are connected to ground of circuit (0V) |
| 14 | Output 4 | Connected to another end of Motor 2 |
| 15 | Input 4 | Directly controls the Output 4 pin. Controlled by digital circuits |
| 16 | Vcc2 (Vss) | Connected to +5V to enable IC function |

The L293D is a 16 pin IC, with eight pins, on each side, dedicated to the controlling of a motor. There are 2 INPUT pins, 2 OUTPUT pins and 1 ENABLE pin for each motor. L293D consist of two H-bridge. H-bridge is the simplest circuit for controlling a low current rated motor.

**Table 4.7.2 Pin Description**

**4.7.1 Working of L293D:**

There are 4 input pins for l293d, pin 2, 7 on the left and pin 15, 10 on the right as shown on the pin diagram. Left input pins will regulate the rotation of motor connected across left side and right input for motor on the right hand side. The motors are rotated on the basis of the inputs provided across the input pins as LOGIC 0 or LOGIC 1.

In simple you need to provide Logic 0 or 1 across the input pins for rotating the motor.

**4.7.2 L293D Logic Table:**

Let’s consider a Motor connected on left side output pins (pin 3, 6). For rotating the motor in clockwise direction the input pins has to be provided with Logic 1 and Logic 0.

Pin 2 = Logic 1 and Pin 7 = Logic 0 | Clockwise Direction

Pin 2 = Logic 0 and Pin 7 = Logic 1 | Anticlockwise Direction

**Pin 2 = Logic 0**and**Pin 7 = Logic 0** | Idle [No rotation] [Hi-Impedance state]

**Pin 2 = Logic 1**and**Pin 7 = Logic 1** | Idle [No rotation]

**4.7.4 Circuit Diagram for l293d motor driver IC controller**

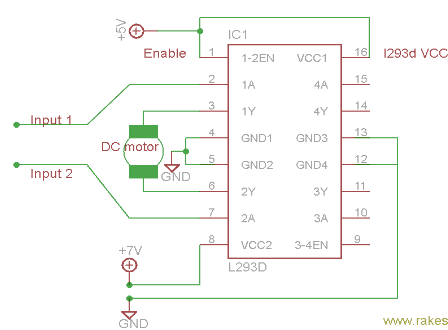


Fig 4.7.4.1 Circuit diagram

## **4.7.5 Voltage Specification:**

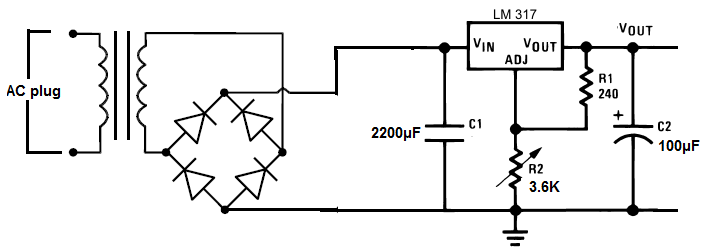
VCC is the voltage that it needs for its own internal operation 5v; L293D will not use this voltage for driving the motor. For driving the motors it has a separate provision to provide motor supply VSS (V supply).  L293d will use this to drive the motor. It means if you want to operate a motor at 9V then you need to provide a Supply of 9V across VSS Motor supply.

The maximum voltage for VSS motor supply is 36V. It can supply a max current of 600mA per channel.Since it can drive motors Up to 36v hence you can drive pretty big motors with this l293d.VCC pin 16 is the voltage for its own internal Operation. The maximum voltage ranges from 5v and up to 36v.

#### 4.8 Power supply:

A power supply is a component that provides at least one electrical charge with power. It typically converts one type of electrical power to another, but it can also convert a different Energy form in electrical energy, such as solar, mechanical, or chemical.

A power supply provides electrical power to components. Usually, the term refers to devices built into the powered component. Computer power supplies, for example, convert AC current to DC current and are generally located along with at least one fan at the back of the computer case.



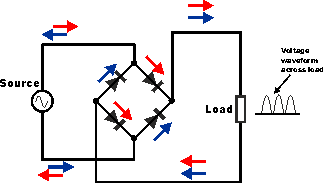
**Fig 4.8.1 Circuit daigram for power supply**

Some basic components used in the supply of power:

#### Rectifier:

A **rectifier** is an electrical device that [converts](https://en.wikipedia.org/wiki/Electric_power_conversion) [alternating current](https://en.wikipedia.org/wiki/Alternating_current) (AC), which periodically reverses direction, to [direct current](https://en.wikipedia.org/wiki/Direct_current) (DC), which flows in only one direction. The process is known as rectification, since it "straightens" the direction of current.

Rectifiers have many uses, but are often found to serve as components of DC power supplies and direct power transmission systems with high voltage. Rectification can be used in roles other than direct current generation for use as a power source.



**Fig 4.8.1.1 Circuit of rectifier**



**Fig 4.8.1.2 Rectifier**

#### 4.8.2Capacitors:

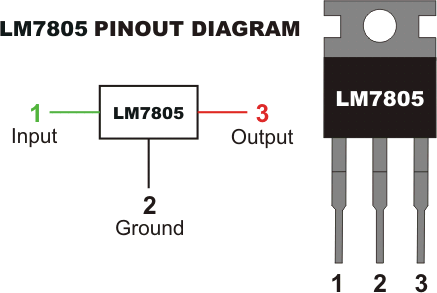
Capacitors are used to attain from the connector the immaculate and smoothest DC voltage in which the rectifier is used to obtain throbbing DC voltage which is used as part of the light of the present identity. Capacitors are used to acquire square DC from the current AC experience of the current channels so that they can be used as a touch of parallel yield.



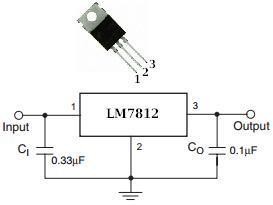
**Fig 4.8.2.1 Capacitor**

#### 4.8.3Voltage regulators:

The 78XX voltage controller is mainly used for voltage controllers as a whole. The XX speaks to the voltage delivered to the specific gadget by the voltage controller as the yield. 7805 will supply and control 5v yield voltage and 12v yield voltage will be created by 7812.The voltage controllers are that their yield voltage as information requires no less than 2 volts. For example, 7805 as sources of information will require no less than 7V, and 7812, no less than 14 volts. This voltage is called Dropout Voltage, which should be given to voltage controllers.



**Fig 4.8.3.1 7805 voltage regulator with pinout**



**Fig 4.8.3.2 7812 voltage regulator with pinout**

#### IR sensor:

IR sensor is an electronic device, that emits the light in order to sense some object of the surroundings. An IR sensor can measure the heat of an object as well as detects the motion. Usually, in the infrared spectrum, all the objects radiate some form of thermal radiation. These types of radiations are invisible to our eyes, but infrared sensor can detect these radiations.



###### Fig 4.8.4.1 IR sensor

The emitter is simply an IR LED (Light Emitting Diode) and the detector is simply an IR photo Diode. Photodiode is sensitive to IR light of the same wavelength which is emitted by the IR LED. When IR light falls on the photodiode, the resistances and the output voltages will change in proportion to the magnitude of the IR light received.

There are five basic elements used in a typical infrared detection system: an infrared source, a transmission medium, optical component, infrared detectors or receivers and signal processing. Infrared lasers and Infrared LED’ s of specific wavelength used as infrared sources.

The three main types of media used for infrared transmission are vacuum, atmosphere and optical fibers. Optical components are used to focus the infrared radiation or to limit the spectral response

**4.8.5 Buzzer:**

A buzzer or beeper is an audio signaling device, which may be mechanical, electromechanical, or piezoelectric. Typical uses of buzzers and beepers include alarm devices, timers and confirmation of user input such as a mouse click or keystroke. Buzzer is an integrated structure of electronic transducers, DC power supply, widely used in computers, printers, copiers, alarms, electronic toys, automotive electronic equipment, telephones, timers and other electronic products for sound devices. Active buzzer 5V Rated power can be directly connected to a continuous sound, this section dedicated sensor expansion module and the board in combination, can complete a simple circuit design, to "plug and play.



#### Fig 4.8.5.1 Buzzer

**4.8.5.1 Buzzer Pin Configuration**

|  |  |  |
| --- | --- | --- |
| **Pin Number** | **Pin Name** | **Description** |
| 1 | Positive | Identified by (+) symbol or longer terminal lead. Can be powered by 5V DC |
| 2 | Negative | Identified by short terminal lead. Typically connected to the ground of the circuit |

#### Table 4.8.5.2 Buzzer Pin Description

**4.8.5.3 Buzzer Features and Specifications**

* Rated Voltage: 6V DC
* Operating Voltage: 4-8V DC
* Rated current: <30mA
* Sound Type: Continuous Beep
* Resonant Frequency: ~2300 Hz
* Small and neat sealed package
* Breadboard and Perf board friendly

**4.8.5.4** **How to use a Buzzer**

A **buzzer**is a small yet efficient component to add sound features to our project/system. It is very small and compact 2-pin structure hence can be easily used on [breadboard](https://components101.com/misc/breadboard-connections-uses-guide), Perf Board and even on PCBs which makes this a widely used component in most electronic applications.

There are two types are buzzers that are commonly available. The one shown here is a simple buzzer which when powered will make a Continuous Beeeeeeppp.... sound, the other type is called a readymade buzzer which will look bulkier than this and will produce a Beep. Beep. Beep. Sound due to the internal oscillating circuit present inside it. But, the one shown here is most widely used because it can be customized with help of other circuits to fit easily in our application.

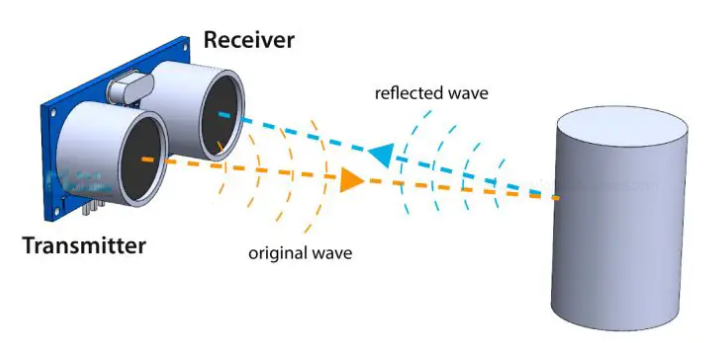
This buzzer can be used by simply powering it using a DC power supply ranging from 4V to 9V. A simple 9V battery can also be used, but it is recommended to use a regulated +5V or +6V DC supply. The buzzer is normally associated with a switching circuit to turn ON or turn OFF the buzzer at required time and require interval.

**4.8.5.5 Applications of Buzzer**

* Alarming Circuits, where the user has to be alarmed about something
* Communication equipment’s
* Automobile electronics
* Portable equipment’s, due to its compact size

**4.8.6 Ultrasonic sensor:**

It emits an ultrasound at 40 000 Hz which travels through the air and if there is an object or obstacle on its path It will bounce back to the module. Considering the travel time and the speed of the sound you can calculate the distance.



**Fig4.8.6.1Ultrasonicsensor**

In order to generate the ultrasound we need to set the Trig pin on a High State for 10 µs. That will send out an 8 cycle ultrasonic burst which will travel at the speed of sound. The Echo pins goes high right away after that 8 cycle ultrasonic burst is sent, and it starts listening or waiting for that wave to be reflected from an object.

If there is no object or reflected pulse, the Echo pin will time-out after 38ms and get back to low state.

If we receive a reflected pulse, the Echo pin will go down sooner than those 38ms. According to the amount of time the Echo pin was HIGH, we can determine the distance the sound wave traveled, thus the distance from the sensor to the object.

For that purpose we are using the following basic formula for calculating distance

Distance = Speed x Time

**CHAPTER-5**

**SOFTWARE REQUIREMENTS**

**5.1 Arduino IDE:**

**Arduino IDE**where IDE stands for Integrated Development Environment – An official software introduced by Arduino.cc, that is mainly used for writing, compiling and uploading the code in the Arduino Device. Almost all Arduino modules are compatible with this software that is an open source and is readily available to install and start compiling the code on the go.

**5.2 Introduction to Arduino IDE:**

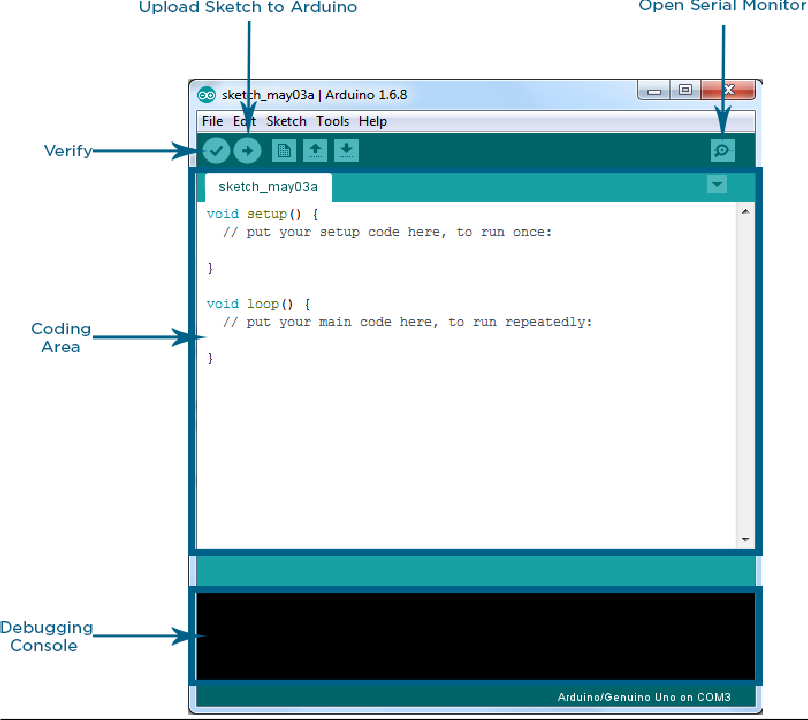
* Arduino IDE is an open-source software that is mainly used for writing and compiling the code into the Arduino Module.
* It is an official Arduino software, making code compilation too easy that even a common person with no prior technical knowledge can get their feet wet with the learning process.
* It is easily available for operating systems like MAC, Windows, and Linux and runs on the Java Platform that comes with inbuilt functions and commands that play a vital role for debugging, editing and compiling the code in the environment.
* A range of Arduino modules available including Arduino Uno, Arduino Mega, Arduino Leonardo, [Arduino Micro](https://www.theengineeringprojects.com/2018/09/introduction-to-arduino-micro.html) and many more.
* Each of them contains a microcontroller on the board that is actually programmed and accepts the information in the form of code.
* The main code, also known as a sketch, created on the IDE platform will ultimately generate a Hex File which is then transferred and uploaded in the controller on the board.
* The IDE environment mainly contains two basic parts: Editor and Compiler where former is used for writing the required code and later is used for compiling and uploading the code into the given Arduino Module.
* This environment supports both C and C++ languages.

#### 5.2.1 How to install Arduino IDE:

You can download the Software from [Arduino](https://www.arduino.cc/en/Main/Software) main website. As I said earlier, the software is available for common operating systems like Linux, Windows, and MAX, so make sure you are downloading the correct software version that is easily compatible with your operating system.

* + - * If you aim to download Windows app version, make sure you have Windows 8.1 or Windows 10, as app version is not compatible with Windows 7 or older version of this operating system.
* The IDE environment is mainly distributed into three sections
  + - * **1. Menu Bar**
      * **2. Text Editor**
      * **3. Output Pane**

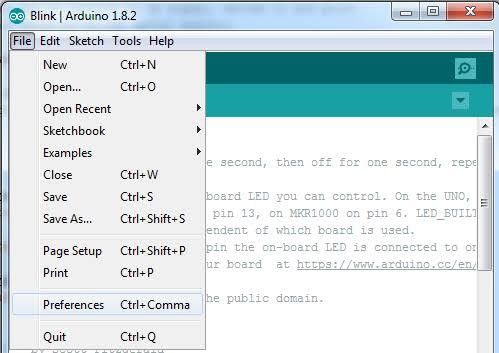
As you download and open the IDE software, it will appear like an image below.



###### Fig 5.2.1.1 Arduino IDE

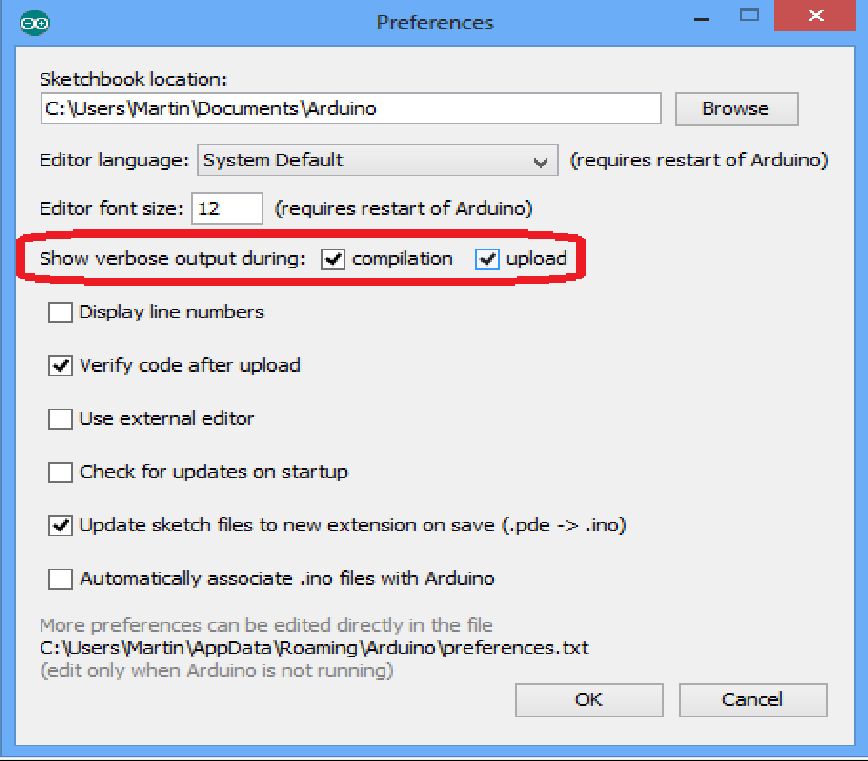
The bar appearing on the top is called **Menu Bar** that comes with five different options as follow

* + - * **File** – You can open a new window for writing the code or open an existing one. Following table shows the number of further subdivisions the file option is categorized into.



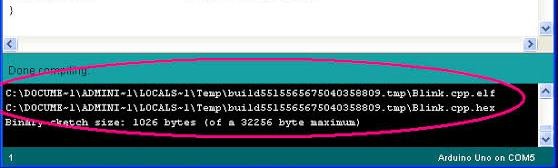
###### Fig 5.2.1.2 File option in Arduino IDE

As you go to the preference section and check the compilation section, the Output Pane will show the code compilation as you click the upload button.



###### Fig 5.2.1.3 Preferences in Arduino IDE

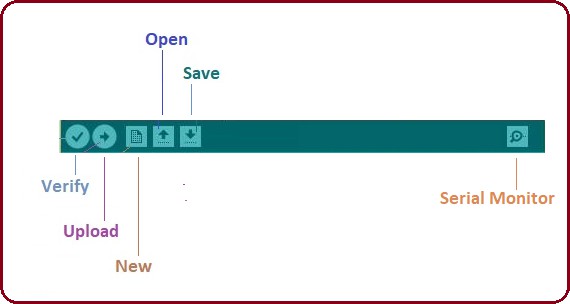
And at the end of compilation, it will show you the hex file it has generated for the recent sketch that will send to the Arduino Board for the specific task you aim to achieve.



###### Fig 5.2.1.4 Hex file

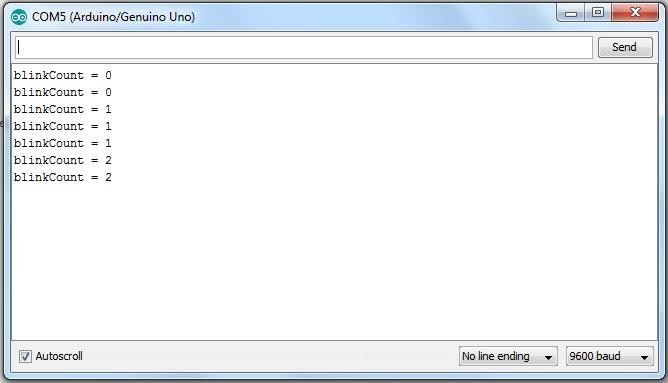
* + - * **Edit** – Used for copying and pasting the code with further modification for font
      * **Sketch** – For compiling and programming
      * **Tools** – Mainly used for testing projects. The Programmer section in this panel is used for burning a bootloader to the new microcontroller.
      * **Help** – In case you are feeling skeptical about software, complete help is available from getting started to troubleshooting.

The **Six Buttons** appearing under the Menu tab are connected with the running program as follow.



###### Fig 5.2.1.5 Buttons of Menu tab in Arduino IDE

* + - * The check mark appearing in the circular button is used to verify the code. Click this once you have written your code.
      * The arrow key will upload and transfer the required code to the Arduino board.
      * The dotted paper is used for creating a new file.
      * The upward arrow is reserved for opening an existing Arduino project.
      * The downward arrow is used to save the current running code.
      * The button appearing on the top right corner is a **Serial Monitor** – A separate pop-up window that acts as an independent terminal and plays a vital role for sending and receiving the Serial Data. You can also go to the Tools panel and select Serial Monitor, or pressing Ctrl+ Shift+ M all at once will open it instantly. The Serial Monitor will actually help to debug the written Sketches where you can get a hold of how your program is operating. Your Arduino Module should be connected to your computer by USB cable in order to activate the Serial Monitor.
      * You need to select the baud rate of the Arduino Board you are using right now. For my Arduino Uno Baud Rate is 9600, as you write the following code and click the Serial Monitor, the output will show as the image below.



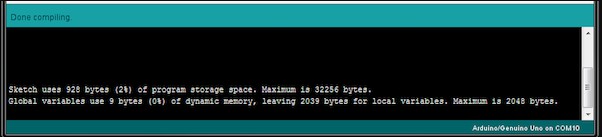
###### Fig 5.2.1.6 Output in serial monitor

The main screen below the Menu bard is known as a simple text editor used for writing the required code.



###### Fig 5.2.1.7 Text editor

The bottom of the main screen is described as an Output Pane that mainly highlights the compilation status of the running code: the memory used by the code, and errors occurred in the program. You need to fix those errors before you intend to upload the hex file into your Arduino Module.

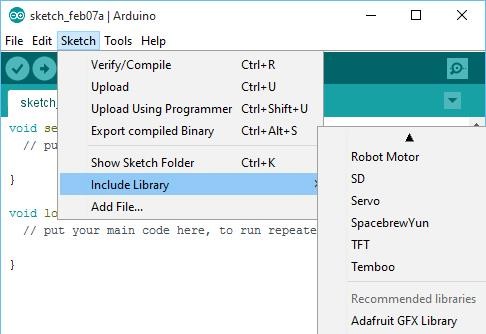


###### Fig 5.2.1.8 Output Pane

More or less, Arduino C language works similar to the regular C language used for any embedded system microcontroller, however, there are some dedicated libraries used for calling and executing specific functions on the board.

#### Libraries:

Libraries are very useful for adding the extra functionality into the Arduino Module. There is a list of libraries you can add by clicking the Sketch button in the menu bar and going to Include Library.



###### Fig 5.2.1.9 Sketch button

As you click the Include Library and Add the respective library it will on the top of the sketch with a #include sign. Suppose, I Include the EEPROM library, it will appear on the text editor as

#include <EEPROM. h>.

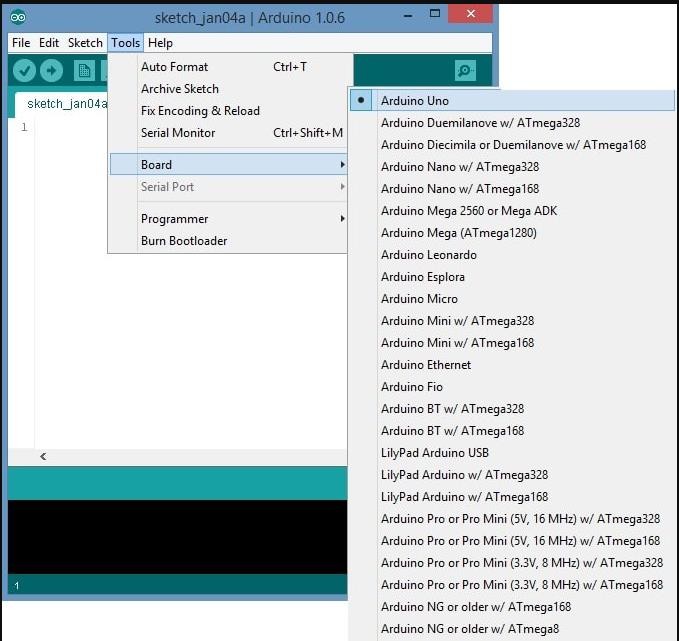
Most of the libraries are preinstalled and come with the Arduino software. However, you can also download them from the external sources.

#### Making pins Input and output:

The digital Read and [digital Write](https://www.theengineeringprojects.com/2018/09/how-to-use-digitalwrite-arduino-command.html) commands are used for addressing and making the Arduino pins as an input and output respectively.

These commands are text sensitive i.e. you need to write them down the exact way they are given like digital Write starting with small “d” and write with capital “W”. Writing it down with Digital write or digital write won’t be calling or addressing any function.

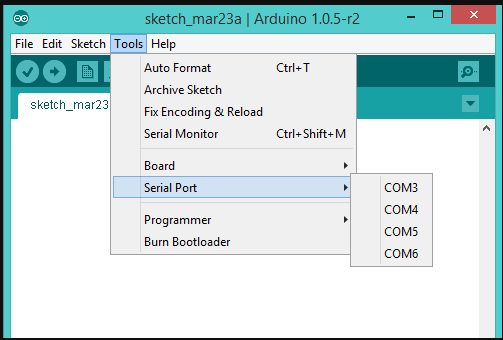
#### How to select the board:

 In order to upload the sketch, you need to select the relevant board you are using and the ports for that operating system. As you click the Tools on the Menu, it will open like the figure below.

###### Fig 5.2.1.10 Board section in Tools options

Just go to the “Board” section and select the board you aim to work on. Similarly, COM1, COM2, COM4, COM5, COM7 or higher are reserved for the serial and USB board. You can look for the USB serial device in the ports section of the Windows Device Manager.

Following figure shows the COM4 that I have used for my project, indicating the Arduino Uno with COM4 port at the right bottom corner of the screen.

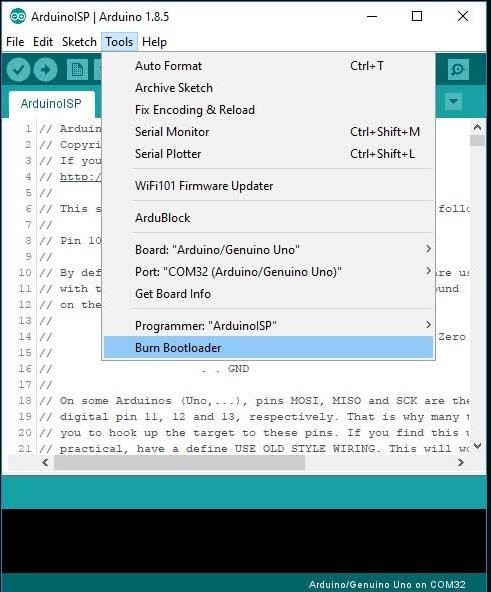


###### Fig 5.2.1.11 Port section in Tools

* + - * The sketch is written in the text editor and is then saved with the file extensionion. It is important to note that the recent Arduino Modules will reset automatically as you compile and press the upload button the IDE software, however, older version may require the physical reset on the board.

**Note**: The port selection criteria mentioned above is dedicated for Windows operating system only, you can check this [Guide](https://www.arduino.cc/en/Guide/Environment) if you are using MAC or Linux.

#### Boot Loader:

 As you go to the Tools section, you will find a boot loader at the end. It is very helpful to burn the code directly into the controller, setting you free from buying the external burner to burn the required code.

###### Fig 5.2.1.12 Boot loader

When you buy the new Arduino Module, the bootloader is already installed inside the controller. However, if you intend to buy a controller and put in the Arduino module, you need to burn the bootloader again inside the controller by going to the Tools section and selecting the burn bootloader.

# CHAPTER-6

# CODE & RESULTS

#### Code

* + 1. **Code for Arduino UNO**

#include <SoftwareSerial.h>

SoftwareSerial espSerial(9,10);

#include<LiquidCrystal.h>

#include <Servo.h>

LiquidCrystal lcd(A5,A4,A3,A2,A1,A0);

int mot1=6;

int mot2=7;

String str;

const int trigPin = 3;

const int echoPin = 4;

int ir=2;

int led=5;

int manhole;

// defines variables

long duration;

int distance;

int i=0;

int j=0;

void setup()

{

pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output

pinMode(echoPin, INPUT); // Sets the echoPin as an Input

pinMode(ir, INPUT);

pinMode(led, OUTPUT);

pinMode(mot1,OUTPUT);

pinMode(mot2,OUTPUT);

digitalWrite(led,LOW);

digitalWrite(mot1,LOW);

digitalWrite(mot2,LOW);

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

espSerial.begin(9600);

lcd.begin(16,2);

lcd.clear();

lcd.setCursor(0,0);

lcd.print("MAN HOLE AND");

lcd.setCursor(0,1);

lcd.print("ROAD DIVERSION");

}

void loop()

{

// Clears the trigPin

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;

delay(500);

manhole=digitalRead(ir);

lcd.clear();

lcd.setCursor(0,0);

lcd.print("DISTANCE: ");

lcd.setCursor(14,0);

lcd.print(distance);

lcd.setCursor(0,1);

lcd.print("MAN HOLE: ");

lcd.setCursor(14,1);

lcd.print(manhole);

delay(2000);

if(manhole==1&&i==0)

{

digitalWrite(led,HIGH);

digitalWrite(mot1,LOW);

digitalWrite(mot2,HIGH);

delay(100);

digitalWrite(mot1,LOW);

digitalWrite(mot2,LOW);

lcd.clear();

lcd.setCursor(0,0);

lcd.print("MANHOLE OPENED");

lcd.setCursor(0,1);

lcd.print("DIVERSION AHEAD");

delay(2000);

i=i+1;

if(distance<20)

{

lcd.clear();

lcd.setCursor(0,0);

lcd.print("TAKE CARE");

lcd.setCursor(0,1);

lcd.print("WORK IN PROGRESS");

delay(2000);

}

}

else if(manhole==0&&i==1&&j==0)

{

digitalWrite(led,LOW);

digitalWrite(mot1,HIGH);

digitalWrite(mot2,LOW);

delay(100);

digitalWrite(mot1,LOW);

digitalWrite(mot2,LOW);

lcd.clear();

lcd.setCursor(0,0);

lcd.print("MANHOLE CLOSED");

lcd.setCursor(0,1);

lcd.print("NO DIVERSIONS");

delay(2000);

i=0;

}

str =String(manhole)+String("@")+String(distance);

espSerial.println(str);

delay(1000);

}

* + 1. **Code for Webserver**

#include <ESP8266WiFi.h>

#include "ThingSpeak.h"

const char\* ssid = "project"; // your network SSID (name)

const char\* password = "1234567890"; // your network password

WiFiClient client;

unsigned long myChannelNumber = 1365737;

const char \* myWriteAPIKey = "Q7VI0258PMWZRG99";

// Timer variables

unsigned long lastTime = 0;

unsigned long timerDelay = 30000;

String String\_main;

String String\_1;

String String\_2;

void setup() {

Serial.begin(9600);

WiFi.mode(WIFI\_STA);

ThingSpeak.begin(client);

while (!Serial) {

; // wait for serial port to connect. Needed for native USB port only

}

}

void loop()

{

if ((millis() - lastTime) > timerDelay) {

// Connect or reconnect to WiFi

if(WiFi.status() != WL\_CONNECTED){

Serial.print("Attempting to connect");

while(WiFi.status() != WL\_CONNECTED){

WiFi.begin(ssid, password);

delay(5000);

}

Serial.println("\nConnected.");

}

if (Serial.available())

{

String\_main=Serial.readString();

Serial.println(String\_main);

String\_1=String\_main.substring(0,1);

Serial.print(String\_1);

delay(500);

String\_2=String\_main.substring(2,4);

Serial.print(String\_2);

delay(500);

ThingSpeak.setField(1,String\_1);

ThingSpeak.setField(2,String\_2);

int x = ThingSpeak.writeFields(myChannelNumber,myWriteAPIKey);

if(x == 200){

Serial.println("Channel update successful.");

}

else{

Serial.println("Problem updating channel. HTTP error code " + String(x));

}

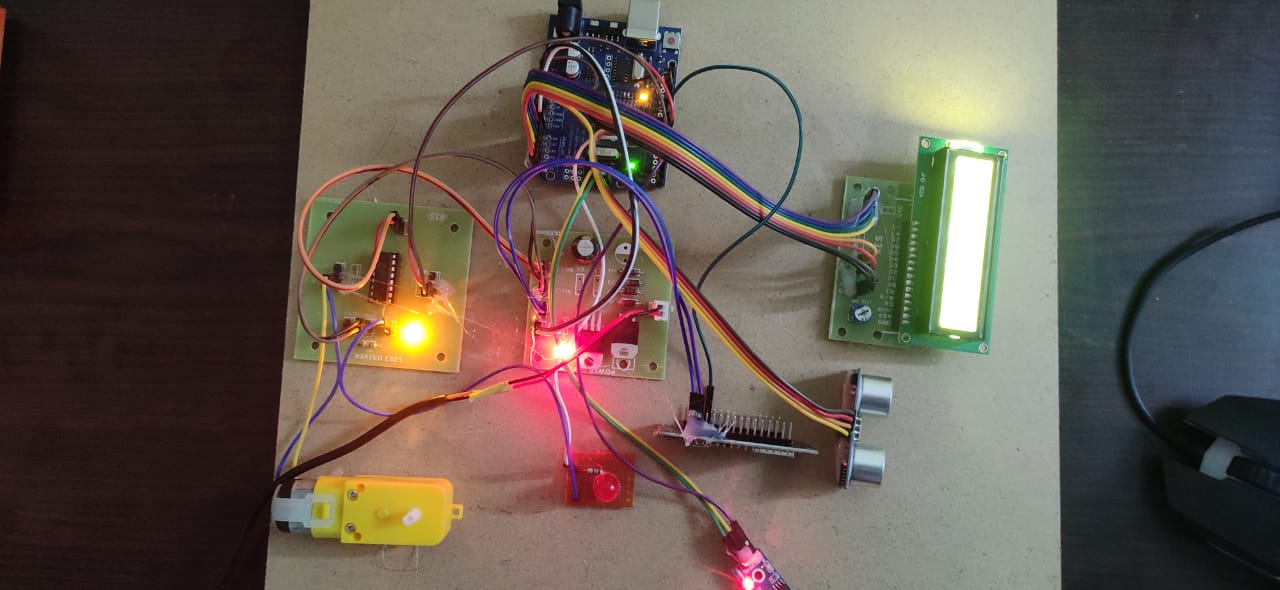
lastTime = millis();

}

}

}

**6.2 Results:**

****

**Fig 6.2.1 Photograph of working of road safety alert system**

##### The above figure shows the working of road safety alert system include components like Arduino UNO, Ultrasonic sensor, IR sensor, Motor drive, LED, LCD display and motor. Microcontroller is the heart of the project and it is brain of the device. Ultrasonic sensor measures the distance between vehicle and manhole by using ultrasonic waves.IR sensor detects the either manhole is open or not. LED is used for notifications for output purpose. LCD display displays the distance and comments related to manhole. Motor rotates both clockwise and anti-clockwise directions. In this project we have three cases:

##### **6.2.1 Case 1**:

##### Initially the manhole is closed.IR sensor detects manhole is close. Ultra sonic sensor measures the distance between vehicle and manhole. Then led is turned off. So motor rotates clockwise direction. LCD displays “MANHOLE IS CLOSED, NO DIVERSIONS”.

##### 

##### **Fig 6.2.1.2 LCD Display 1 Fig 6.2.1.3 Output 1**

**6.2.2 Case 2:**

In this case, IR sensor detects manhole is opened. Then led is turned on, motor rotates anti-clockwise direction. So, it displays “MANHOLE OPENED AND DIVERSION AHEAD”.



**Fig6.2.2.1 Distance calculation 2**

****

**Fig 6.2.2.2 Output 2**

**6.2.3 Case 3:**

At this final case, manhole is opened. But vehicle moving towards the manhole, then LED glows ON. Motor rotates clockwise direction from motor drive. It displays “TAKE CARE AND WORK IN PROGRESS”.

****

**Fig 6.2.3.1 LCD display 3**

****

**Fig 6.2.3.2 Output 3**

**CHAPTER-7**

**ADVANTAGES & APPLICATIONS**

**7.1 ADVANTAGES**

* Real-Time Hazard Alerts
* Comprehensive Hazard Coverage
* Autonomous Operation
* Enhanced Weather Resistance
* Reduced Reliance on Manual Intervention

**APPLICATIONS**

**7.2 APPLICATIONS**

* **Enhanced Road Safety:** The primary application of the Road Safe Alert System is to improve road safety by providing timely notifications to drivers about potential hazards such as manhole placements or road diversions. By alerting drivers in real-time, the system helps prevent accidents and reduces the risk of injuries or fatalities on the road.
* **Traffic Management Optimization:** With the ability to inform drivers about upcoming road disruptions, the system contributes to better traffic management by allowing drivers to plan alternative routes or adjust their driving behavior accordingly. This helps minimize traffic congestion and improves overall traffic flow in urban areas.
* **Construction Site Safety:** For construction workers operating near roads or highways, the Road Safe Alert System offers an additional layer of safety by notifying drivers about ongoing construction activities. This reduces the likelihood of accidents involving construction vehicles and improves the safety of workers on-site.
* **Emergency Response Coordination:** In the event of emergencies such as accidents or natural disasters, the system can be used to disseminate critical information to drivers and emergency response teams. By providing timely updates on road conditions and potential hazards, emergency services can coordinate their efforts more effectively.
* **Public Transportation Efficiency:** Public transportation services can benefit from the Road Safe Alert System by integrating the notifications into their scheduling and route planning systems. Bus and taxi drivers can receive alerts about road conditions in advance, allowing them to adjust their routes and schedules to minimize delays for passengers.
* **Pedestrian Safety:** The system can also include features to enhance pedestrian safety by alerting drivers to the presence of pedestrians near construction zones or busy intersections. By raising awareness among drivers, the system helps reduce the risk of accidents involving pedestrians.
* **Environmental Impact Reduction:** By reducing traffic congestion and the associated idling of vehicles, the Road Safe Alert System contributes to lower emissions and improved air quality. This environmental benefit aligns with sustainability goals and initiatives aimed at reducing the carbon footprint of transportation systems.
* **Smart City Integration:** As part of a broader smart city infrastructure, the system can integrate with other urban management systems to optimize resource allocation and improve overall city livability. By providing real-time data on road conditions, the system supports data-driven decision-making for city planners and policymakers.
* **Tourism and Navigation:** Tourists and visitors to urban areas can benefit from the Road Safe Alert System by receiving notifications about road closures or construction activities that may affect their travel plans. This improves the overall experience for tourists and helps them navigate unfamiliar city streets more safely.
* **Education and Awareness Campaigns:** The system can be used as a platform for education and awareness campaigns aimed at promoting safe driving practices and increasing public awareness of road safety issues. By delivering **Top of Form**

**CHAPTER-8**

**CONCLUSION**

In conclusion, the Road Safe Alert System represents a significant advancement in road safety technology, offering a comprehensive solution to mitigate potential hazards and improve overall traffic management. Through its real-time notifications and alerts, the system empowers drivers with crucial information about road conditions, construction activities, and other potential obstacles, allowing them to make informed decisions and adapt their driving behaviour accordingly. By enhancing situational awareness and providing early warnings, the system contributes to reducing the frequency and severity of accidents, ultimately saving lives and preventing injuries on the road.

Furthermore, the Road Safe Alert System serves as a valuable tool for urban planners, transportation authorities, and emergency response teams, enabling better coordination and resource allocation during various scenarios, including construction projects, emergency situations, and special events. By leveraging technology to facilitate communication and data sharing among stakeholders, the system supports more efficient traffic management, emergency response coordination, and public safety initiatives. Additionally, the system's integration with smart city infrastructure and public transportation networks further enhances its utility and effectiveness in promoting safer and more sustainable urban mobility.

Looking ahead, the Road Safe Alert System holds immense potential for future enhancements and expansions, including the integration of additional sensors, the development of predictive analytics capabilities, and the incorporation of advanced communication technologies such as vehicle-to-infrastructure (V2I) and vehicle-to-everything (V2X) connectivity. By embracing innovation and collaboration, stakeholders can continue to leverage the power of technology to address emerging challenges and opportunities in road safety and urban mobility, ultimately creating safer, more resilient, and more liveable cities for all.

# REFERENCES

[1] Bychkovsky, V., Chen, K., Goraczko, H., Hu, H., Hull, B., Miu, A., Shih, E., Zhang, Y., Madden S., and Balakrishnan, H; The cartel: a distributed mobile sensor computing system. In: 4th international conference on Embedded networked sensor systems, SenSys06, pp. 125-138. ACM, Boulder, Colorado, USA (November 2006).

[2] Eriksson, J., Girod, L., Hull, B., Newton, R., Madden, S., and Balakrishnan H.; The pothole patrol: Using a mobile sensor network for road surface monitoring. In: Sixth Annual International conference on Mobile Systems, Applications and Services (MobiSys 2008). IEEE, Breckenridge, U.S.A. (June 2008).

[3] Chen, K., Lu, M., Fan, X., Wei, M. and Wu, J.; Road Condition Monitoring Using On-board Threeaxis Accelerometer and GPS Sensor. In: sixth International ICST Conference on Communications and Networking. China (2011).

[4] Mohan, P., Padmanabhan, V. N. and Ramjee R.; Nericell: rich monitoring of road and traffic conditions using mobile smartphones. In: 6th ACM conference on Embedded network sensor systems, SenSys 08, pp. 323336. ACM, New York, NY, USA (2008).

[5] Mednis, A., Strazdin, G., Zviedris, R., Kanonirs, G., and Selavo, L.; Real time pothole detection using android smartphone with accelerometers. In: International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS). IEEE (June 2011).

[6] Bhoraskar, R., Vankadhara, N., Raman, B., Kulkarni P.; Wolverine: Traffic and Road Condition estimation using Smartphone Sensors. In: Fourth International Conference on Communication Systems and Networks (COMSNETS). IEEE (January 2012).

[7] Singh, P., Juneja, N. and Kapoor, S.; Using Mobile Phone Sensors to Detect Driving Behavior. In: Third ACM Symposium on Computing for Development, Article No. 53. ACM, Bangalore, India (2013).