DESIGN & IMPLEMENTATION OF AUTOMOBILE SAFETY DISTANCE

A Thesis
By

Md. Mizanur Rahman
Shojib Alam
Md. Shibly Sadiq
Md. Kohim Uddin Mondol
Satirtho Roy



Department of Mechanical Engineering

SONARGAON UNIVERSITY (SU)

January, 2025

DESIGN & IMPLEMENTATION OF AUTOMOBILE SAFETY DISTANCE

A Thesis

By

Md Mizanur Rahman	: ME2102024001
Shojib Alam	: BME1701011503
Md. Shibly Sadiq	: ME2102024086
Md. Kohim Uddin Mondol	: ME2102024358
Satirtho Roy	: ME2101023412



Supervisor : M. A. Faisal Junaeat Imrul

Designation : Lecturer & Asst. Course Coordinator

Department of Mechanical Engineering

Submitted to the

Department of Mechanical Engineering (ME)

SONARGAON UNIVERSITY (SU)

In partial fulfillment of the requirement for award of the degree Of Bachelor of Science in Mechanical Engineering

January,2025

DECLARATION

This dissertation has been submitted as thesis entitled "Design & Implementation of Automobile Safety Distance Application" under the guidance of **M. A. Faisal Junaeat Imrul** to the Sonargaon University (SU) in partial fulfillment to the requirements for the degree of Bachelor of Science in Mechanical Engineering, Sonargaon University Dhaka, Bangladesh. The work presented in this thesis has not been submitted previously for any degree or diploma, and to the best of our knowledge, it does not contain any material previously published or written by another person, except where due acknowledgment has been made. We also declare that all the contributions of other researchers, including my supervisor, have been acknowledged appropriately.

Authors

Md. Mizanur Rahman
ME2102024001

Shojib Alam
BME1701011503

Md. Shibly Sadiq
ME2102024086

Md. Kohim Uddin Mondol
ME2102024358

Satirtho Roy
ME2101023412

Approval

This is to certify that the Project and Thesis Work (ME-400) Submitted by Md. Mizanur Rahman (Student ID: ME2102024001), Shojib Alam (Student ID: BME1701011503), Md. Shibly Sadiq (Student ID: ME2102024086), Md. Kohim Uddin Mondol (Student ID: ME2102024358), Satirtho Roy (Student ID: ME2101023412), entitled " Design & Implementation of Automobile Safety Distance Application " has been approved by the examiners for the partial fulfillment of the requirements for the degree of Bachelor of Science in Mechanical Engineering. Sonargaon University, Green Road, Dhaka.

Supervised by

M. A. Faisal Junaeat Imrul

Lecturer & Asst. Course Coordinator

Department of Mechanical Engineering

Sonargaon University (SU)

Dhaka , Bangladesh

ACKNOWLEDGEMENT

The report titled as on “Design & Implementation of Automobile Safety Distance Application” has been prepared to fulfill the requirement of our practicum program. In the process of doing and preparing our practicum report, we would like to pay our gratitude to some persons for their enormous help and vast co-operation. At first, we would like to show, our gratitude to the University authority to permit us to do our practicum. Specially, we would like to thank to our honorable supervisor **M. A. Faisal Junarat Imral,** Lecturer & Asst. Course Coordinator, Department of, Mechanical Engineering, SU- Sonargaon University Dhaka, for his valuable and patient advice, sympathetic assistance, co-operation, contribution of new idea. Deep Theoretical and hardware knowledge & keen interest of our supervisor in this field influenced us to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stage have made it possible to complete this project.

Finally, we would like to thank again to the respected Vice-Chancellor of SU, also thanks to Head of Department of SU, Mechanical Engineering, because they are designated such an environment for learning through which we got the opportunity to acquire knowledge under B.Sc. in ME program, and that will be very helpful for our prospective career.

We are, indeed, grateful to all those from whom we got sincere cooperation and help for the preparation of this report

ABSTRACT

Automobile safety distance is a critical factor in ensuring road safety and reducing collision risks. It refers to the minimum distance a vehicle should maintain from the one ahead, allowing sufficient reaction time to avoid accidents. The safety distance depends on factors such as vehicle speed, road conditions, weather, and driver reaction time. Advanced technologies like adaptive cruise control (ACC) and collision avoidance systems (CAS) use sensors and real-time data to automatically adjust the safety distance, enhancing driver assistance and safety. Additionally, public awareness and adherence to traffic rules play a vital role in maintaining appropriate spacing on the road. By integrating innovative technologies and promoting safe driving habits, the concept of safety distance significantly contributes to minimizing road fatalities, optimizing traffic flow, and improving overall transportation safety. This paper explores the impact of maintaining safety distance and the role of technology in fostering safer road environments.

TABLE OF CONTENTS

Letter of Transmittal		ii
Declaration		iii
Acknowledgement		iv
Abstract		v
Table of Contents		vi
Table of Figure		viii
List of Table		ix
CHAPTER-1	**INTRODUCTION**	
1.1	Background	01
1.2	Background Study	02
1.3	Problem Statement	04
1.4	Motivation	04
1.5	Objective	05
1.6	Research Outline	05
CHAPTER 2	**LITERATURE REVIEW**	
2.1	Introduction	06
2.2	Literature Review	06
2.3	Summary	08
CHAPTER-3	**METHODOLOGY**	
3.1	System Design	09
3.2	Methodology	09
3.3	Block Diagram	09
3.4	Schematic Diagram	10
3.5	Working Principle	11
3.6	Component List	11
3.7	Arduino Nano	11
3.8	Ultrasonic Sensor	15
3.9	LCD Display	17
3.10	Battery	18

3.11	Buck Converter	20
3.12	Gear Motor	21
3.13	Mototr Driver	22
3.14	Bluetooth Module	26
3.15	Arduino IDE	29
3.16	Easy EDA	33
CHAPTER-4	**RESULT & DISCUSSION**	
4.1	Result Analysis	35
4.2	Advantage	36
4.3	Application	37
4.4	Limitation	37
4.5	Discussion	37
4.6	Cost Analysis	38
CHAPTER-5	**CONCLUSION**	
5.1	Conclusion	39
5.2	Future Scope	39
Reference		**40**
Appendix		**43**

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO
1.1	Car Safety Distance	01
3.1	Block Diagram of Automatic Safety Distance	10
3.2	Schematic Diagram of Automobile Safety Distance	10
3.3	Arduino Nano	12
3.4	Arduino Schematic Diagram	12
3.5	How Arduino Nano Looks Like	13
3.6	Micro-controller IC AT mega 328P	14
3.7	Ultrasonic Sensor Working Procedure	15
3.8	Ultrasonic Sensor	16
3.9	16*2 LCD display	17
3.10	Pin Out of Ultrasonic Sensor	17
3.11	16*2 LCD Display Interfacing with Arduino	18
3.12	3.7V Battery	19
3.13	DC-DC Buck Converter	20
3.14	Gear Motor	21
3.15	Motor Driver IC L293D	23
3.16	L293D Circuit Diagram	25
3.17	Bluetooth Module HC-05	26
3.18	HC-05 Bluetooth Module	27
3.19	Arduino Software Interface IDE	30
3.20	Easy EDA Software Interface	34
4.01	Complete Project prototype	35
4.2	Project progress Image	36

LIST OF TABLE

TABLE NO	TABLE NAME	PAGE NO
1.1	Pin Configuration of Ultrasonic Sensor	16
2.1	Battery Specification	19
3.1	Motor Driver Pinout	24
4.1	Pin Configuration of Bluetooth Module	28
5.1	Cost Of this Project	38

Chapter – 1
Introduction

1.1 Introduction

Maintaining an appropriate safety distance between vehicles is fundamental to ensuring road safety and preventing accidents. Safety distance refers to the minimum space a driver should maintain from the vehicle ahead, providing sufficient time to react to sudden stops or changes in traffic conditions. This distance is influenced by various factors, including vehicle speed, road surface, weather conditions, and driver attentiveness.

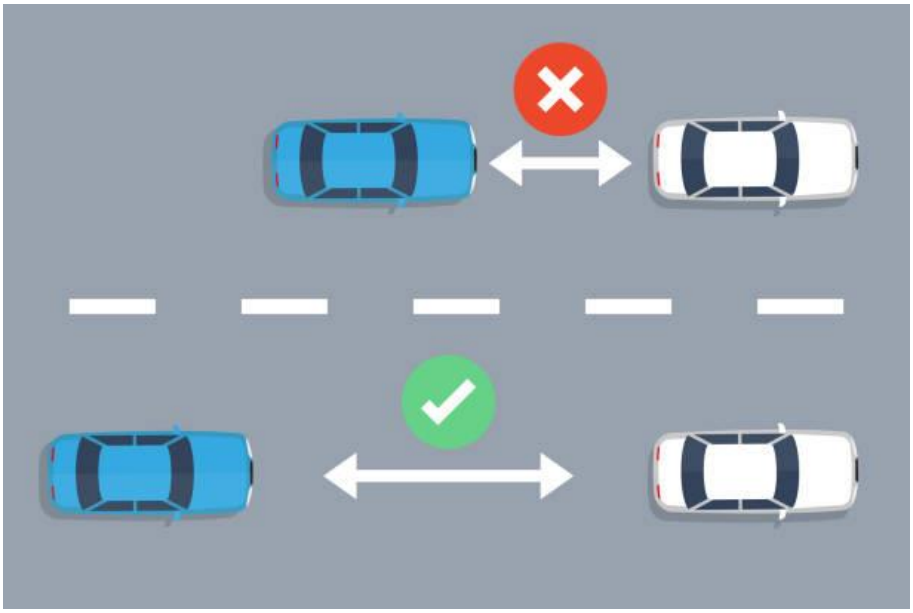


Figure 1.1 Car Safety Distance (19)

Modern advancements in automobile technology have introduced systems such as adaptive cruise control (ACC) and collision avoidance systems (CAS), which help drivers maintain safe distances automatically. These systems utilize sensors, and real-time data to monitor the vehicle's surroundings and adjust its speed accordingly.

Despite technological innovations, driver awareness and compliance with traffic regulations remain critical in preventing tailgating and rear-end collisions. Promoting the concept of safety distance is essential for creating safer roads, reducing traffic congestion, and enhancing overall driving experiences.

1.2 Background Study

Every year all over the world, traffic accidents occur due to vehicle crashes. While different factors contribute to vehicle crashes, such as vehicle mechanical problems and bad weather and driver behavior. Driver behavior is considered to be the leading cause of more than 90 percent of all accidents [1]. Vehicle distance is one of the most important indexes of traffic safety [2]. Indeed, accidents can be reduced if more time is provided for the driver to react with the situation by increasing or reducing the host vehicle speed. Several automakers have developed electronic warning sound devices to alert the driver about the near vehicles. But most of these solutions are high cost, computerized uses DSP or special microcontroller units and consequently having complicated systems.

Today's systems use signals from a wide variety of sensors around the vehicle to support the driver. Modern cars have a large variety of advanced driver assistance systems that help the driver in driving and parking situations and can even take over some other tasks completely.

Indeed, the drivers can decide whether they want to have safety alarming system in their vehicles or not. Such a system may use different types of sensors that supply the vehicle with data and information about its surroundings, for example, Radar, cameras and ultrasound sensors. Recently, in modern vehicle safety systems design, radar, cameras, infrared and ultrasound sensors can be linked intelligently and simultaneously by sensor fusion to achieve the same function [3] [4]. Wireless Communication Protocols are proposed in to enhance traffic safety parameter at highway between Vehicle-to-Vehicle to reduce the rate of dangerous car accidents. In [5], vehicle-to-vehicle (V2V) communication is proposed for traffic safety application through periodical broadcasts for information messages including location and velocity of the around vehicle's stringent reliability requirement of highway traffics.

The development of pre-crash safety system with pedestrian collision avoidance assist has been presented in [6] by Toyota Motor Corporation Team. This system is an example of the systems developed by Motor big companies that is capable of detecting both vehicles and pedestrians and helps the driver to avoid a collision by automatically braking the vehicle.

Pedestrian detection is enabled by a sensing system that combines millimeter wave radar and a stereo camera. This system is capable of stable object detection regardless of day or night. At night, the system uses near infrared projectors to enhance the detection performance of the camera.

In [7], the authors proposed a system that monitors and displays the distance between two cars in a timely manner. Then, it predicts rear collision through the traffic safety distance algorithm. If there exists the risk of rear collision, the system will remind the driver of taking measures to avoid collision by means of screen display and brake lights. The work in [8] presents safety distance system based on ultrasonic distance sensor to keep 10 m distance between host vehicle and front vehicle, the electronic design in done in is based on Arduino Uno microcontroller. In the above studies, the introduced solutions are presented based special integrated controlling unit. while other studies for different wired/wireless design objectives are proposed based on discrete components in order to avoid system complexity, and to achieve high response . For the above-mentioned reasons, the proposed solution in this paper adopts discrete electronic components.

In our study, vehicle distance alarming system is proposed based on discrete components in order to alert the driver for three different states, firstly, when the distance between the host vehicle and the front vehicle is less than certain configured minimum distance limit**.** Secondly, the system works on alarming driver when the vehicle is running with low speed and the distance is longer than a certain maximum distance limit. The third state, the system is triggered to indicate driver that the host vehicle is in safe distance range condition. The important benefit of the proposed system will be to allow the driver to react with the situation before an accident may take place while keeping safe distance range between the vehicles during driver’s trips

1.3 Problem Statement

Inadequate maintenance of automobile safety distance is a significant contributor to road accidents and traffic congestion worldwide. Many drivers fail to adhere to the recommended safety distance due to factors such as lack of awareness, overconfidence, distracted driving, or aggressive behavior. This negligence leads to rear-end collisions, which are among the most common types of accidents, especially in high-speed or congested traffic conditions.

The problem is exacerbated by variables such as poor weather conditions, sudden braking by vehicles, and road infrastructure limitations. While technological advancements like adaptive cruise control and collision warning systems have been developed, their adoption remains limited, particularly in low-cost vehicles. Furthermore, these systems may not always account for all real-world complexities, such as unpredictable human behavior.

Addressing this issue requires a combination of public education, stricter traffic enforcement, and the widespread implementation of intelligent vehicle systems to ensure safer and more reliable driving practices.

1.4 Motivation

Ensuring adequate automobile safety distance is vital for reducing road accidents and enhancing traffic flow. The rising number of rear-end collisions, often caused by tailgating or distracted driving, highlights the need for improved safety measures. Advances in technology, such as adaptive cruise control and collision avoidance systems, provide opportunities to automate and optimize safety distances. Additionally, increasing public awareness of safe driving practices can further minimize risks. This project is driven by the goal of leveraging innovative solutions and promoting responsible driving behavior to create safer roads, prevent accidents, and foster a more efficient and reliable transportation system for all.

1.5 Objective

The main goal of this project is to design and develop such a device by which we can measure bottle filling machine using Arduino. The main objectives are as follows:

- To study the **Automobile Safety Distance.**
- To design and construct to find the **Automobile Safety Distance.**
- To ensure safer roads by maintaining optimal safety distances to prevent collisions
- To take necessary notes from the project for future improvements.

1.6 Research Outline

This project book consists of five chapter. The first chapter contains the statement of the introduction, Background Study, Problem Statement for the project, objectives of the study, Scope of The Study, Significance of the Study and the Research Outline. Chapter two contains system Literature Review in Details. Chapter three describes System Overview , Block Diagram, Circuit Diagram with component details and the software which we have used for our work. Chapter four deals with the Implementation & result discussion, advantage, application and shows the complete prototype of the project that we have built. In the final chapter, we have discussed the conclusion and future scope of the project and also about some aspects we had to overcome while doing the project and lastly, we gave the conclusion of the book.

Chapter – 2
Literature Review

2.1 Introduction

In this chapter we discuss our system overview. In this paper we are also work about Automobile Safety Distance . In below section we will describe some previous literature review. From previous literature we gather more knowledge to build this project and we successfully made it .

2.2 Literature Review

Dynamic security highlights, similar to electronically monitored slowing mechanisms, footing control and electronic brake dissemination, are intended to keep away from a mishap. Inactive security highlights, as in assembled disintegrate zones in the monolog body shell which distort in a head-on impact to ingest the vitality of the approaching vehicle, are being to ensure the driver and travelers inside the vehicle when a mishap happens notwithstanding the working of the different dynamic wellbeing frameworks. Airbags, side effect pillars in the entryways and collapsible guiding segments are different cases of inactive security frameworks. [9]

The work done in this paper is an endeavor to outline a propel vehicle security framework that utilizations GPS and GSM framework to anticipate burglary and to decide the correct area of vehicle. Today robbery is going on the stopping or in some uncertain place. The wellbeing of the vehicle is astoundingly fundamental. The propel vehicle security framework is planned utilizing GPS and GSM innovation. The framework contains GPS module, GSM modem, Infrared sensors, 8051 microcontroller, transfer switch, paint shower and high voltage work. GPS frameworks track the present area of vehicle, there are two kinds of following utilized one is web based following and other is disconnected following. GSM framework is additionally introduced in the vehicle for sending the data to the client since GPS framework can just get the vehicle area data from satellites.[10]

This paper provides the study of Integrated Safety Systems (ISS) like pre-crash frameworks with impact cautioning or independent vehicle activities. The Survey venture is specifically reacting to this by creating test and appraisal techniques for the assessment of pre-crash frameworks. Techniques are being produced for driver behavioral viewpoints, pre-crash detecting execution and crash execution under conditions impacted by pre-crash driver and vehicle activities. [11]

Smart vehicle associations with enhance Activity security is one of the most encouraging advantages of Vanet. The principle highlight of VANET is Correspondence between two vehicles for message trade by means of Vehicle to Vehicle (V2V) and Vehicle to Foundation (V2I) correspondence convention. VANETs are sorted by high versatility, quickly evolving convention, transient and one-time cooperation. The Goal of this paper is vehicles can openly collaborate for bringing advantages of more prominent wellbeing and effectiveness. This correspondence can be utilized to evasion of vehicles crash, transmit data about activity. [12]

Pramodshelke et al. [13] had proposed that a review on Arduino Based embedded System in Car for Road Safety using RFID .In this work when the RFID tag is detected by the RFID reader the speed of the motor reduces in specified areas and it avoids accidents.

Ch . Sonali Shankar et al., [14] had proposed that a review on Arduino based Embedded System in Passenger Car for Road Safety .In this work the sound of the horn stops and speed is also reduced when the RFID tag is detected by the RFID reader.

Deepa et al., [15] had proposed that a review on Embedded System in Passenger Car for Road Safety.in this work the proposed features are automatic collision notification that gives notification to the victim's relative, red light traffic control makes sure the vehicle doesn't break signal, speed control alters speed in different zones, horn control prevents honking in horn prohibited zone, alcohol detection detects drunk driving and vehicle security is used to prevent theft Suhas Katkar et al., [16] had proposed that a review on An Embedded System In passenger Car For Road Safety. This project is designed to inform about an accident or theft that has occurred to the vehicle, to the family members of the travelling persons and

concerned authorities. Anto bennet et al., had proposed that a review on An Embedded System in Passenger Car for Road Safety using GPS and GSM. This work is designed to inform about an accident that is occurred to a vehicle to the family members of the travelling persons.

Anto bennet et al., [17] had proposed that a review on an Embedded System in Passenger Car for Road Safety using GPS and GSM.GPS Receiver gets the location information from satellite in the form of latitude and longitude. The system can be interconnected with the alcohol detection and alert the owner on his mobile phone.

Anto bennet et al., [18] had proposed that a review on Alcohol Detection and Accident Avoidance Using locking With Tracking. This system will detect drunker driver by alcohol sensor through driver breathe fitted on steering in front of driver ,the message is sent to the police through GSM system and provides GPS base system to track those cars.

2.3 Summary

This literature is help us to a brief concept of Automobile Safety Distance. Many people are trying to make this project. We also try this project. We are added four different sources which will be very helpful for user.

Chapter – 3
Methodology

3.1 System Design

In this project, we have used a Battery section to produce power in the system. Here we use micro-controller. All kind of controlling system work under Arduino Nano . Here we also use Ultrasonic Sensor, Bluetooth Module, LCD display, Buck Converter, Motor Driver, DC Gear Motor. All of this equipment's are connected like circuit design and it full our desire outcome.

3.2 Methodology

Our methodology for the project:

- Creating an idea for design and construction of a **Design and Implementation of Automobile Safety Distance** with the combination of Arduino micro controller.
- And designing a block diagram & circuit diagram to know which components we need to construct it. Collecting all the components and programming the micro controller to control the whole system.
- Setting up all the components in a PCB board & then soldering. Lastly, assembling all the blocks in a board and to run the system & for checking purposes.

3.3 Block Diagram

Here is the block diagram of the **Design and Implementation of Automobile Safety Distance** with all the essential components. All of the components are shown in below as a block in this diagram. **Here we use Arduino Nano for the main micro-controller which automatically operates the system and provides online information of our project by using Bluetooth module.** Other equipment's are attached with this micro-controller and work together for perform as our desire outcome.

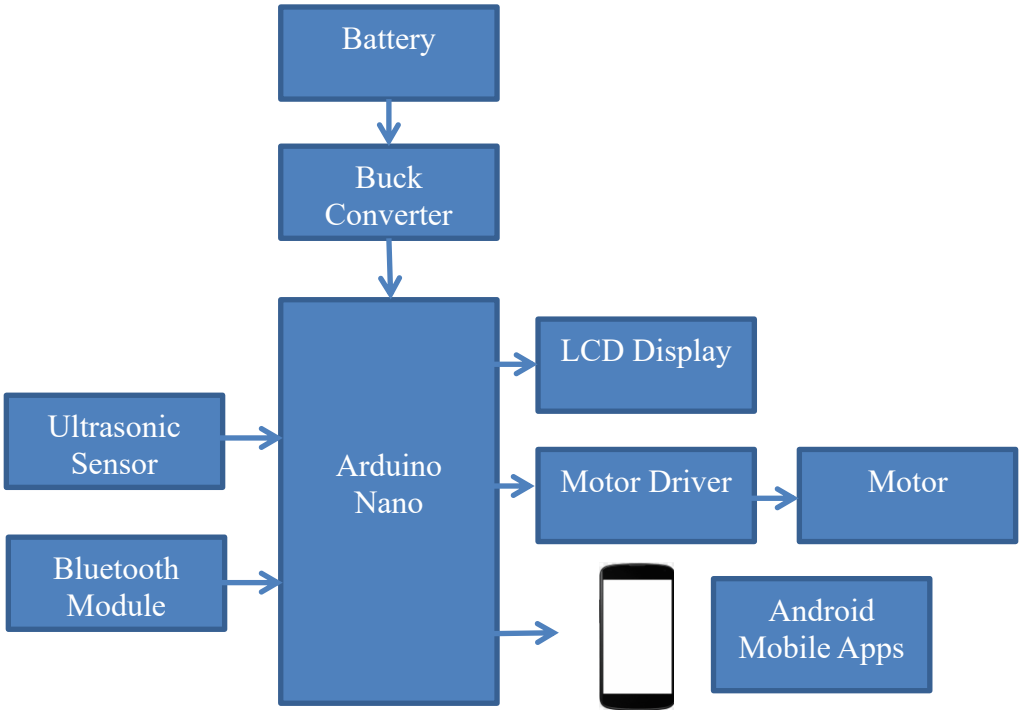


Figure 3.1: Block Diagram of Automatic safety Distance

3.4 Schematic Diagram

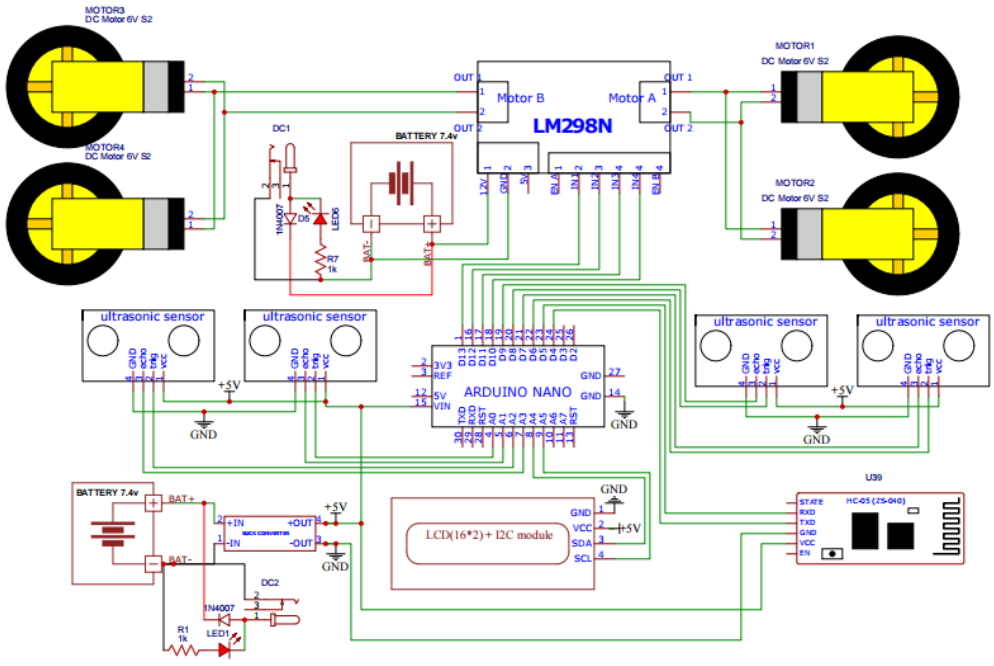


Figure 3.2: Schematic Diagram of Automobile Safety System

3.5 Working Principle

The automobile safety system uses an Arduino Nano as the central controller to manage inputs and outputs. An ultrasonic sensor detects obstacles by measuring distance, and the data is displayed on an LCD for real-time feedback. A Bluetooth module enables wireless communication, allowing users to monitor or control the system via a smartphone. The gear motor, controlled by a motor driver, drives safety mechanisms like obstacle avoidance or automated responses. A battery powers the system, with a buck converter regulating voltage for stable operation. This integrated system enhances vehicle safety by providing timely alerts and responsive collision prevention.

3.6 Component List

3.6.1 Hardware

- Arduino Nano
- Ultrasonic Sensor
- LCD Display
- Battery
- Buck Converter
- Gear Motor
- Motor Driver
- Bluetooth Module

3.6.2 Software

- Arduino IDE
- Easy EDA

3.7 Arduino Nano

Arduino is open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It's intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments. Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling Lights, motors, and other actuators.



Figure 3.3: Arduino Nano

The micro-controller on the board is programmed using the Arduino programming language (based on Wiring) and the Arduino development environment (based on Processing). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, Maxims'). **Arduino Nano is a surface mount breadboard embedded version with integrated USB. It i**s a small, complete, and breadboard friendly component. It has everything that Decimal/ Duemilanove has (electrically) with more analog input pins and onboard +5V AREF jumper. **Physically, it is missing power jack. The Nano can automatically sense and switch to the higher potential source of power**.

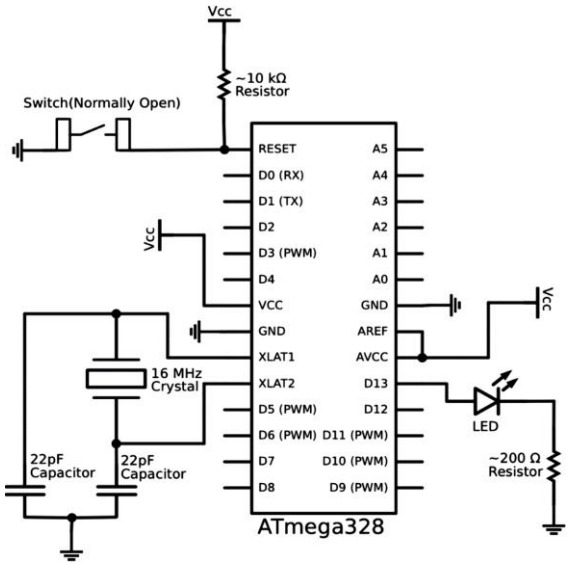


Figure 3.4: Arduino Nano Schematic Diagram [20]

Nano's got the breadboard-ability of the Boarding and the Minibus with smaller footprint than either, so users have more breadboard space. It's got a pin layout that works well with the Mini or the Basic Stamp (TX, RX, ATN, and GND on one top, power and ground on the other). This new version 3.0 comes with ATMEGA328 which offer more programming and data memory space. It has two layers. That make it easier to hack and more affordable. One of the best features of Arduino Nano is, it's easy to use, compact and also small.

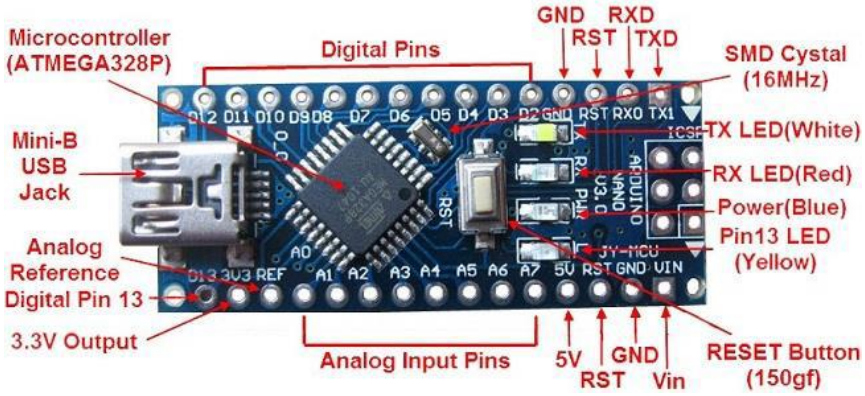


Figure 3.5: How Arduino Nano looks like [20]

3.7.1 Specifications:

- Micro-controller: Atmel ATmega328
- **Operating Voltage (logic level): 5 V**
- **Input Voltage (recommended): 7-12 V**
- **Input Voltage (limits): 6-20 V**
- Digital I/O Pins: 14 (of which 6 provide PWM output)
- Analog Input Pins: 8
- DC Current per I/O Pin: 40 mA
- Flash Memory: 32 KB (of which 2KB used by boot loader)
- SRAM : 2 KB
- EEPROM: 1 KB
- Clock Speed: 16 MHz
- Dimensions: 0.70" x 1.70"

3.7.2 Features:

- Automatic reset during program download
- Power OK blue LED
- Green (TX), red (RX) and orange (L) LED
- Auto sensing/switching power input
- Small mini-B USB for programming and serial monitor
- ICSP header for direct program download
- Manual reset switch

Micro-controller IC ATmega328p



Figure 3.6: Micro-controller IC AT mega 328p [20]

The high-performance Microchip Pico Power 8-bit AVR RISC-based micro-controller combines 32KB ISP flash memory with read-while-write capabilities, 1024B EEPROM, 2KB SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible timer/counters with compare modes, internal and external interrupts, serial programmable USART, a byte-oriented 2-wire serial interface, SPI serial port, a 6-channel 10-bit A/D converter (8-channels in TQFP and QFN/MLF packages), programmable watchdog timer with internal oscillator, and five software selectable power saving modes. The device operates between 1.8-5.5 volts. By executing powerful instructions in a single clock cycle, the device achieves throughput's approaching 1 MIPS per MHz, balancing power consumption and processing speed.

3.8 Ultrasonic Sensor

An ultrasonic sensor is an electronic device that measures the distance of a target object by emitting ultrasonic sound waves and converts the reflected sound into an electrical signal. Ultrasonic waves travel faster than the speed of audible sound.

3.8.1 HC-SR04 Ultrasonic Sensor - Working

As shown above the **HC-SR04 Ultrasonic (US) sensor** is a 4 pin module, whose pin names are Vcc, Trigger, Echo and Ground respectively. This sensor is a very popular sensor used in many applications where measuring distance or sensing objects are required. The module has two eyes like projects in the front which forms the Ultrasonic transmitter and Receiver. The sensor works with the simple high school formula that –

Distance = Speed × Time

The Ultrasonic transmitter transmits an ultrasonic wave, this wave travels in air and when it gets objected by any material it gets reflected back toward the sensor this reflected wave is observed by the Ultrasonic receiver module as shown in the picture below



Figure 3.7: Ultrasonic Sensor Working Procedure [21]

Now, to calculate the distance using the above formulae, we should know the Speed and time. Since we are using the Ultrasonic wave we know the universal speed of US wave at room conditions which is 330m/s. The circuitry inbuilt on the module will calculate the time taken for the US wave to come back and turns on the echo pin high for that same particular

amount of time, this way we can also know the time taken. Now simply calculate the distance using a micro-controller or microprocessor.

3.8.2 HC-SR04 Sensor Features

- Operating voltage: +5V
- Theoretical Measuring Distance: 2cm to 450cm
- Practical Measuring Distance: 2cm to 80cm
- Accuracy: 3mm
- Measuring angle covered: <15°
- Operating Current: <15mA
- Operating Frequency: 40Hz

3.8.3 Ultrasonic Sensor Pin Configuration

Table 1.1 : Pin Configuration of Ultrasonic Sensor

Pin Number	Pin Name	Description
1	Vcc	The Vcc pin powers the sensor, typically with +5V
2	Trigger	Trigger pin is an Input pin. This pin has to be kept high for 10us to initialize measurement by sending US wave.
3	Echo	Echo pin is an Output pin. This pin goes high for a period of time which will be equal to the time taken for the US wave to return back to the sensor.
4	Ground	This pin is connected to the Ground of the system.

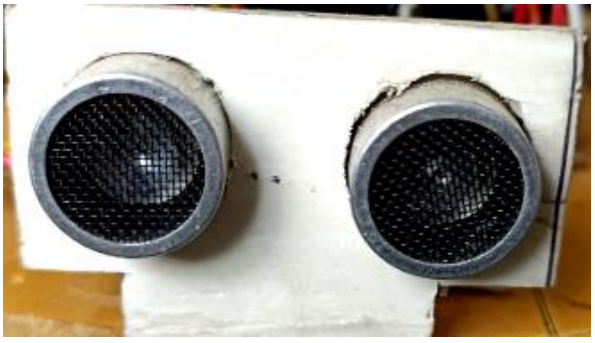


Figure 3.8: Ultrasonic Sensor

3.9 LCD Display

The LCD (liquid crystal display) screen is an electronic display module and looks for various applications. A 16x2 LCD display is a very basic module and it is very commonly used in various devices and circuits. These modules are preferred over seven sections and many other segments.



Figure 3.9: 16*2 LCD Display

The reasons for having LCD are economic; Easily programmable, special and even custom characters (different in seven sections), there are no restrictions on displaying animations. A 16x2 LCD means it can display 16 characters per line and contains 2 lines. Each character on this LCD is displayed in a 5x7 pixel matrix.

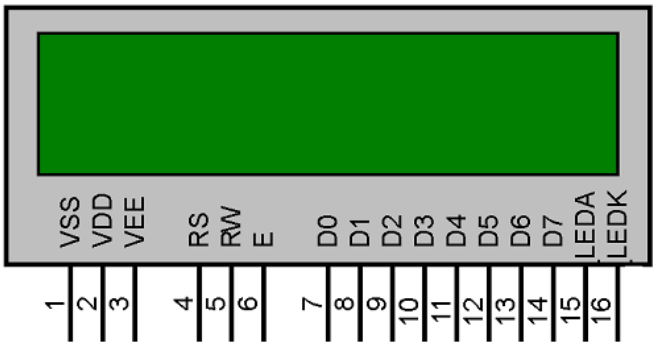


Figure 3.10: Pin out of 16*2 LCD Display [22]

This LCD contains two articles called Command and Data. The command register stores the command instructions on the LCD. The command is a command given to the LCD to perform a predefined task such as starting it, clearing its screen, locating the cursor, controlling the display, etc.

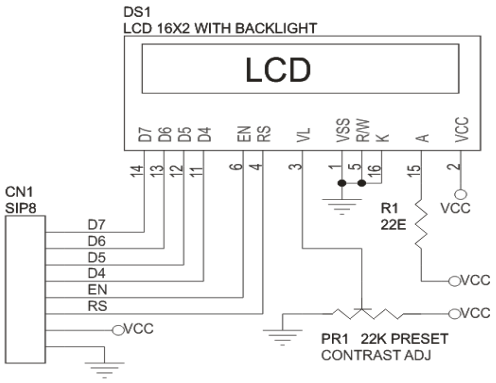


Figure 3.11: 16*2 LCD Display interfacing with Arduino [22]

3.9.1 Feature

The features of this LCD mainly include the following.

- The operating voltage of this LCD is 4.7V-5.3V
- It includes two rows where each row can produce 16-characters.
- The utilization of current is 1mA with no backlight
- Every character can be built with a 5×8 pixel box
- The alphanumeric LCDs alphabets & numbers
- Is display can work on two modes like 4-bit & 8-bit
- These are obtainable in Blue & Green Backlight
- It displays a few custom generated characters

3.10 Battery

Lithium batteries are primary batteries that have metallic lithium as an anode. These types of **batteries are also referred to as lithium-metal batteries**. They stand apart from other batteries in their high charge density and high cost per unit. Depending on the design and chemical compounds used, lithium cells can produce voltages from 1.5 V (comparable to a zinc–carbon or alkaline battery) to about 3.7 V.

Disposable primary lithium batteries must be distinguished from secondary lithium-ion or a lithium-polymer,[1] which are rechargeable batteries. Lithium is especially useful, because its ions can be arranged to move between the anode and the cathode, using

an intercalated lithium compound as the cathode material but without using lithium metal as the anode material. Pure lithium will instantly react with water, or even moisture in the air; the lithium in lithium ion batteries is in a less reactive compound.

Lithium batteries are widely used in portable consumer electronic devices. The term "lithium battery" refers to a family of different lithium-metal chemistry, comprising many types of cathodes and electrolytes but all with metallic lithium as the anode. The battery requires from 0.15 to 0.3 kg of lithium per kWh. As designed these primary systems use a charged cathode, that being an electro-active material with crystallographic vacancies that are filled gradually during discharge.



Figure 3.12: 3.7V Battery [23]

3.10.1 Product Specification

Table 2.1 : Battery Specification

Voltage	3.7 V
Product Type	Lithium-Ion
Battery Capacity	2200mAh
Weight	45 g
Model Number	ICR 18650

3.11 Buck Converter

A buck converter (step-down converter) is a DC-to-DC power converter which steps down voltage (while drawing less average current) from its input (supply) to its output (load). It is a class of switched-mode power supply (SMPS) typically containing at least two semiconductors (a diode and a transistor, although modern buck converters frequently replace the diode with a second transistor used for synchronous rectification) and at least one energy storage element, a capacitor, inductor, or the two in combination. To reduce voltage ripple, filters made of capacitors (sometimes in combination with inductors) are normally added to such a converter's output (load-side filter) and input (supply-side filter). It is called a buck converter because the voltage across the inductor "bucks" or opposes the supply voltage.



Figure 3.13: DC -DC Buck Converter

DC-DC Buck Converter Step Down Module LM2596 Power Supply is a step-down(buck) switching regulator, capable of driving a 3-A load with excellent line and load regulation. These devices are available in fixed output voltages of 3.3 V, 5 V, 12 V, and an adjustable output version. The LM2596 series operates at a switching frequency of 150kHz, thus allowing smaller sized filter components than what would be required with lower frequency switching regulators.

Specifications of DC-DC Buck Converter Step Down Module LM2596 Power Supply :

- Conversion efficiency: 92%(highest)
- Switching frequency: 150KHz
- Output ripple: 30mA9maximum)

- Load Regulation: ± 0.5%
- Voltage Regulation: ± 0.5%
- Dynamic Response speed: 5% 200uS
- Input voltage:4.75-35V
- Output voltage:1.25-26V(Adjustable)
- Output current: Rated current is 2A, maximum 3A (Additional heat sink is required)
- Conversion Efficiency: Up to 92% (output voltage higher, the higher the efficiency)
- Switching Frequency: 150KHz
- Rectifier: Non-Synchronous Rectification
- Module Properties: Non-isolated step-down module (buck)
- Short Circuit Protection: Current limiting, since the recovery
- Operating Temperature: Industrial grade (-40 to +85) (output power 10W or less)

3.12 Gear Motor

Gear motors are mechanisms that adjust the speed of electric motors, leading them to operate at a certain speed. They are composed of a series of gears that make up a kinematic chain, working on a set of rotary parts. **Their main purpose is to allow the reduction from an initial high speed to a lower one without negatively affecting the mechanism**. In addition to this adjustment, a gear motor is in charge of adjusting the mechanical power of a system.



Figure 3.14: Gear Motor

Mechanical components that complement gear motors

Gear motors are composed of an electric motor and gears, which form the kinematic chain – the fundamental component of the gear ratio.

3.12.1 Kinematic chain

A motor's speed reducer is composed of a speed reducer and its gears. This speed reducer is basically a variable speed drive that allows for the speed to be reduced and increased at the output shaft.

3.12.2 Gears

Gears are toothed wheels made of metal or plastic (and new materials with each passing day) that transmit motion when meshing with each other. They are defined by their number of teeth and their size. In addition, they may have straight-cut or helical teeth.

3.12.3 Motors

The five types of motors that see the most use in gear motors are:

- Brushed motors, with brushes normally made out of carbon. They are bidirectional and may be used with DC or AC. They have a service life of about 3000 hours.
- Asynchronous motors, which are brush less single-direction motors. They are highly limited.
- Synchronous brush less motors, which may be single-direction or bidirectional. They have a constant speed if the frequency of the power source is stable.
- Brush less DC motors that use a driver and can attain high speeds.
- Stepper DC brush less motors. They can be positioned with an average precision of 7.5º.

3.13 Motor Driver

The L293D is a popular 16-Pin Motor Driver IC. **As the name suggests it is mainly used to drive motors.** A single L293D IC is capable of running two DC motors at the same time; also the direction of these two motors can be controlled independently.

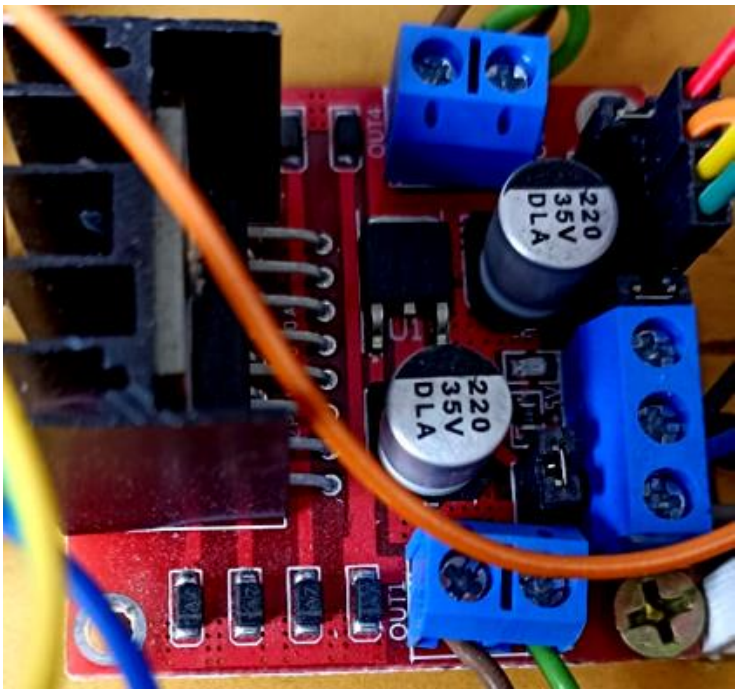


Figure 3.15: Motor driver IC L293D

3.13.1 Working Process

L293D IC is a typical Motor Driver IC which allows the DC motor to drive on any direction. This IC consists of 16-pins which are used to control a set of two DC motors instantaneously in any direction. It means, by using a L293D IC we can control two DC motors.

3.13.2 Features

- Can be used to run Two DC motors with the same IC.
- Speed and Direction control is possible
- Motor voltage Vcc2 (Vs): 4.5V to 36V
- Maximum Peak motor current: 1.2A
- Maximum Continuous Motor Current: 600mA
- Supply Voltage to Vcc1(vss): 4.5V to 7V
- Transition time: 300ns (at 5Vand 24V)
- Automatic Thermal shutdown is available

3.13.3 L293D Pin Configuration

Table 3.1: Motor Driver Pinout

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

3.13.4 U**se of a L293D Motor Driver IC:**

Using this L293D motor driver IC is very simple. The IC works on the principle of Half H-Bridge, let us not go too deep into what H-Bridge means, but for now just know that H bridge is a set up which is used to run motors both in clock wise and anti clockwise direction. As said earlier this IC is capable of running two motors at the any direction at the same time, the circuit to achieve the same is shown below.

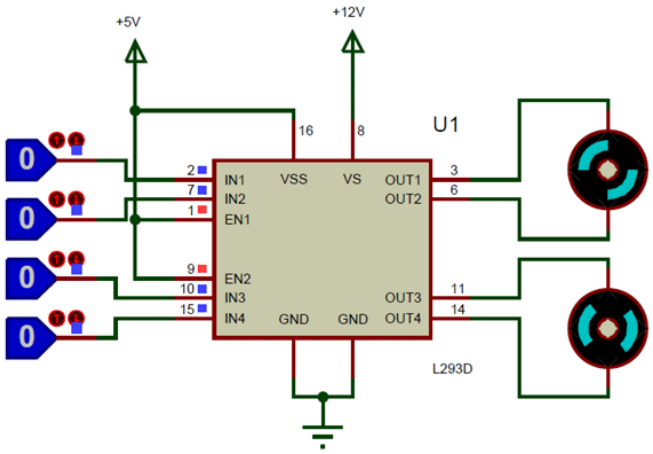


Figure 3.16: L293D circuit Diagram [24]

All the Ground pins should be grounded. There are two power pins for this IC, one is the Vss(Vcc1) which provides the voltage for the IC to work, this must be connected to +5V. The other is Vs(Vcc2) which provides voltage for the motors to run, based on the specification of your motor you can connect this pin to anywhere between 4.5V to 36V, here I have connected to +12V.

The Enable pins (Enable 1,2 and Enable 3,4) are used to Enable Input pins for Motor 1 and Motor 2 respectively. Since in most cases we will be using both the motors both the pins are held high by default by connecting to +5V supply. The input pins Input 1,2 are used to control the motor 1 and Input pins 3,4 are used to control the Motor 2. The input pins are connected to the any Digital circuit or micro-controller to control the speed and direction of the motor.

3.13.5 Applications

- Used to drive high current Motors using Digital Circuits
- Can be used to drive Stepper motors
- High current LED's can be driven

3.14 Bluetooth Module

Bluetooth is a wireless technology standard for exchanging data between fixed and mobile devices over short distances using short-wavelength UHF radio waves in the industrial, scientific and medical radio bands, from 2.400 to 2.485 GHz, and building personal area networks.

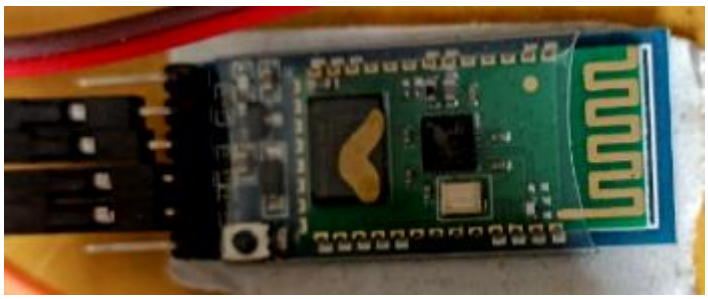


Figure 3.17: Bluetooth Module HC-05

3.14.1 HC-05 Technical Specifications

- Serial Bluetooth module for Arduino and other microcontrollers
- Operating Voltage: 4V to 6V (Typically +5V)
- Operating Current: 30mA
- Range: <100m
- Works with Serial communication (USART) and TTL compatible
- Follows IEEE 802.15.1 standardized protocol
- Uses Frequency-Hopping Spread spectrum (FHSS)
- Can operate in Master, Slave or Master/Slave mode
- Can be easily interfaced with Laptop or Mobile phones with Bluetooth
- Supported baud rate: 9600,19200,38400,57600,115200,230400,460800.

3.14.2 How to Use the HC-05 Bluetooth Module

The HC-05 has two operating modes, one is the data mode which allows it to send and receive data from other Bluetooth devices and the other is the command mode where the default device settings can be changed. We can operate the device in either of these two modes using the PIN key described in the PIN description. Connecting the HC-05 module to microcontrollers is very easy because it operates using the Serial Port Protocol (SPP). Power the module with only + 5V and connect the RX pin of the module to the TX of the MCU and the TX pin of the module to the RX of the MCU as shown in the figure below.

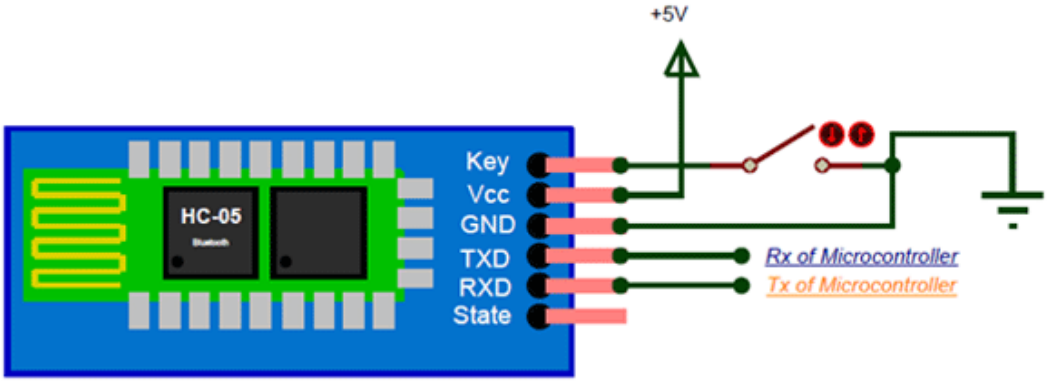


Figure: 3.18 HC-05 Bluetooth Module [25]

During power-up, the key pin can be used to enter the command mode. If this pin is not used, it will enter the data mode by default. After the input module is activated, you should be able to detect the Bluetooth device as "HC-05" and then as the default password. Use 1234 to connect and start communicating with it.

3.14.3 Pin Configuration of HC-05 Bluetooth Module

Table 4.1 : Pin Configration of Bluetooth Module

Pin Number	Pin Name	Description
1	Enable / Key	This pin is used to toggle between Data Mode (set low) and AT command mode (set high). By default it is in Data mode
2	Vcc	Powers the module. Connect to +5V Supply voltage
3	Ground	Ground pin of module, connect to system ground.
4	TX Transmitter	Transmits Serial Data. Everything received via Bluetooth will be given out by this pin as serial data.
5	RX Receiver	Receive Serial Data. Every serial data given to this pin will be broadcasted via Bluetooth
6	State	The state pin is connected to on board LED, it can be used as a feedback to check if Bluetooth is working properly.
7	LED	Indicates the status of Module Blink once in 2 sec: Module has entered Command Mode Repeated Blinking: Waiting for connection in Data Mode Blink twice in 1 sec: Connection successful in Data Mode
8	Button	Used to control the Key/Enable pin to toggle between Data and command Mode

3.15 Arduino IDE

The digital microcontroller unit named as Arduino Nano can be programmed with the Arduino software IDE. There is no any requirement for installing other software rather than Arduino. Firstly, Select "Arduino Nano from the Tools, Board menu (according to the microcontroller on our board). The IC used named as ATmega328 on the Arduino Nano comes pre burned with a boot loader that allows us to upload new code to it without the use of an external hardware programmer.

Communication is using the original STK500 protocol (reference, C header files). We can also bypass the boot loader and programs the microcontroller through the ICSP (In Circuit Serial Programming) header. The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available. The ATmega16U2/8U2 is loaded with a DFU boot loader, which can be activated by:
On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2. On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode.

The Arduino Nano is one of the latest digital microcontroller units and has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL at (5V) with serial communication, which is available on digital pins 0 -(RX) for receive the data and pin no.1 (TX) for transmit the data. An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, an .in file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board.

The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial Communication on pins 0 and 1). A Software Serial library allows for serial communication on any of the Nano's

digital pins. The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus. Arduino programs are written in C or C++ and the program code written for Arduino is called sketch. The Arduino IDE uses the GNU tool chain and AVR Lab to compile programs, and for uploading the programs it uses argued. As the Arduino platform uses Atmel microcontrollers, Atmel's development environment, AVR Studio or the newer Atmel Studio, may also be used to develop software for the Arduino.

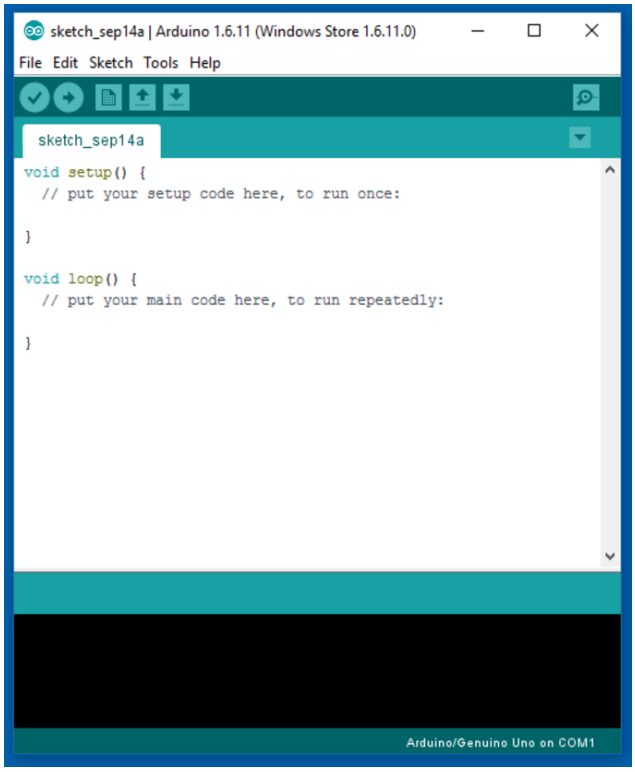


Figure 3.19: Arduino Software Interface IDE [26]

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

3.15.1 Writing Sketches

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

3.15.2 Sketchbook

The Arduino Software (IDE) uses the concept of a sketchbook: a standard place to store your programs (or sketches). The sketches in your sketchbook can be opened from the File > Sketchbook menu or from the Open button on the toolbar. The first time you run the Arduino software, it will automatically create a directory for your sketchbook. You can view or change the location of the sketchbook location from with the Preferences dialog. Beginning with version 1.0, files are saved with a .ino file extension. Previous versions use the .pde extension. You may still open .pde named files in version 1.0 and later, the software will automatically rename the extension to .ino.

3.15.3 Tabs, Multiple Files, and Compilation

Allows you to manage sketches with more than one file (each of which appears in its own tab). These can be normal Arduino code files (no visible extension), C files (.c extension), C++ files (.cpp), or header files (.h).

3.15.4 Uploading

Before uploading your sketch, you need to select the correct items from the Tools > Board and Tools > Port menus. The boards are described below. On the Mac, the serial port is probably something like /dev/tty.usbmodem241 (for an Uno or Mega2560 or Leonardo) or /dev/tty.usbserial-1B1 (for a Duemilanove or earlier USB board), or /dev/tty.USA19QW1b1P1.1 (for a serial board connected with a Keyspan USB-to-Serial

adapter). On Windows, it's probably COM1 or COM2 (for a serial board) or COM4, COM5, COM7, or higher (for a USB board) - to find out, you look for USB serial device in the ports section of the Windows Device Manager. On Linux, it should be /dev/ttyACMx , /dev/ttyUSBx or similar. Once you've selected the correct serial port and board, press the upload button in the toolbar or select the Upload item from the Sketch menu. Current Arduino boards will reset automatically and begin the upload. With older boards (pre-Diecimila) that lack auto-reset, you'll need to press the reset button on the board just before starting the upload. On most boards, you'll see the RX and TX LEDs blink as the sketch is uploaded. The Arduino Software (IDE) will display a message when the upload is complete, or show an error.

When you upload a sketch, you're using the Arduino bootloader, a small program that has been loaded on to the microcontroller on your board. It allows you to upload code without using any additional hardware. The bootloader is active for a few seconds when the board resets; then it starts whichever sketch was most recently uploaded to the microcontroller. The bootloader will blink the on-board (pin 13) LED when it starts (i.e. when the board resets).

3.15.5 Libraries

Libraries provide extra functionality for use in sketches, e.g. working with hardware or manipulating data. To use a library in a sketch, select it from the Sketch > Import Library menu. This will insert one or more #include statements at the top of the sketch and compile the library with your sketch. Because libraries are uploaded to the board with your sketch, they increase the amount of space it takes up. If a sketch no longer needs a library, simply delete its #include statements from the top of your code.

There is a list of libraries in the reference. Some libraries are included with the Arduino software. Others can be downloaded from a variety of sources or through the Library Manager. Starting with version 1.0.5 of the IDE, you do can import a library from a zip file and use it in an open sketch. See these instructions for installing a third-party library.

3.15.6 Third-Party Hardware

Support for third-party hardware can be added to the hardware directory of your sketchbook directory. Platforms installed there may include board definitions (which appear in the board menu), core libraries, bootloaders, and programmer definitions. To install, create the hardware directory, then unzip the third-party platform into its own sub-directory. (Don't use "arduino" as the sub-directory name or you'll override the built-in Arduino platform.) To uninstall, simply delete its directory.For details on creating packages for third-party hardware, see the Arduino IDE 1.5 3rd party Hardware specification.

3.15.7 Serial Monitor

This displays serial sent from the Arduino or Genuino board over USB or serial connector. To send data to the board, enter text and click on the "send" button or press enter. Choose the baud rate from the drop-down menu that matches the rate passed to Serial.begin in your sketch. Note that on Windows, Mac or Linux the board will reset (it will rerun your sketch) when you connect with the serial monitor. Please note that the Serial Monitor does not process control characters; if your sketch needs a complete management of the serial communication with control characters, you can use an external terminal program and connect it to the COM port assigned to your Arduino board.

3.16 Easy EDA

Easy EDA is a web-based EDA tool suite that enables hardware engineers to design, simulate, share-publicly and privately-and discuss schematics, simulations and printed circuit boards. Other features include the creation of a bill of materials, Gerber files and pick and place files and documentary outputs in PDF, PNG and SVG formats. Easy EDA allows the creation and editing of schematic diagrams, SPICE simulation of mixed analogue and digital circuits and the creation and editing of printed circuit board layouts and, optionally, the manufacture of printed circuit boards.

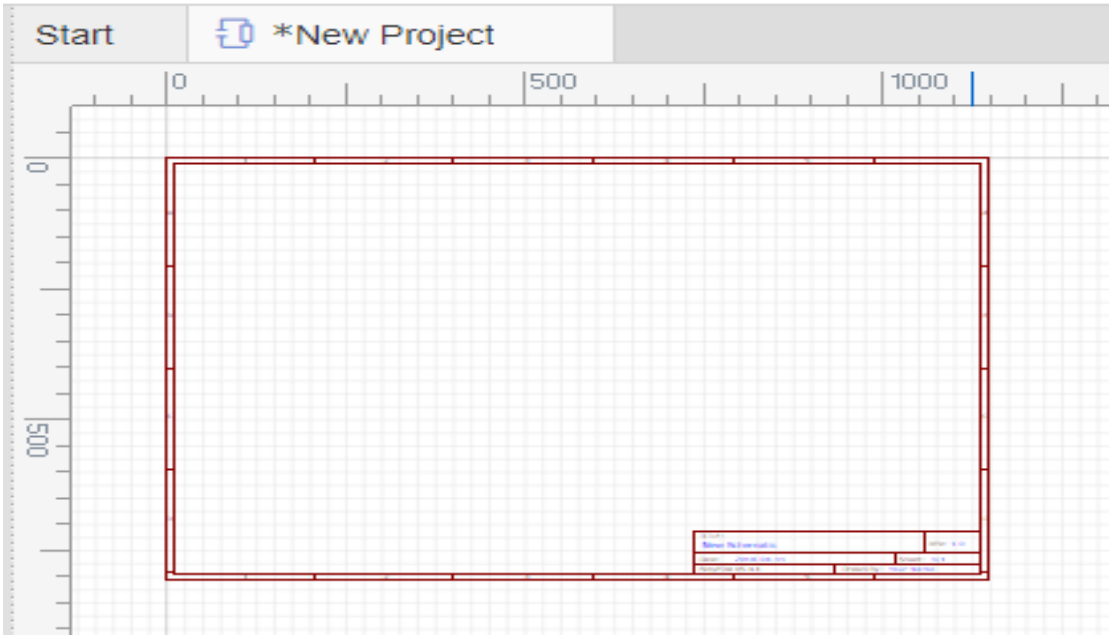


Figure 3.20: Easy EDA Software Interface [27]

Subscription-free membership is offered for public plus a limited number of private projects. The number of private projects can be increased by contributing high quality public projects, schematic symbols, and PCB footprints and/or by paying a monthly subscription. Registered users can download Gerber files from the tool free of charge; but for a fee, Easy EDA offers a PCB fabrication service. This service is also able to accept Gerber file inputs from third party tools.

3.16.1 Data Table : How Far Will Provide the Signal

Side	Distance
Right	10 (cm)
Left	10 (cm)
Front	10 (cm)
Back	10 (cm)

Chapter – 4

Result & Discussion

4.1 Result Analysis

This chapter contains the results obtained and discussion about the full project. Our project is **Automobile Safety Distance.** In our project making we used PVC boards for total hardware making. After finally completing this project, we ran it & we observed the output of this project. We can see that it is working well as expected. After making our project we observe it very careful. It works as we desire. Our project give output perfectly and all equipment are work perfectly. We check how much it works and we get perfect output from this project.

- Finally, have completed our project successfully & check our project its run accurately according to our objective.
- At first, start our system.
- Then it was control by Arduino Nano with the help Bluetooth Module of android apps.
- Then the robot was detect the obstacle and show the robot in-front view in Android app.

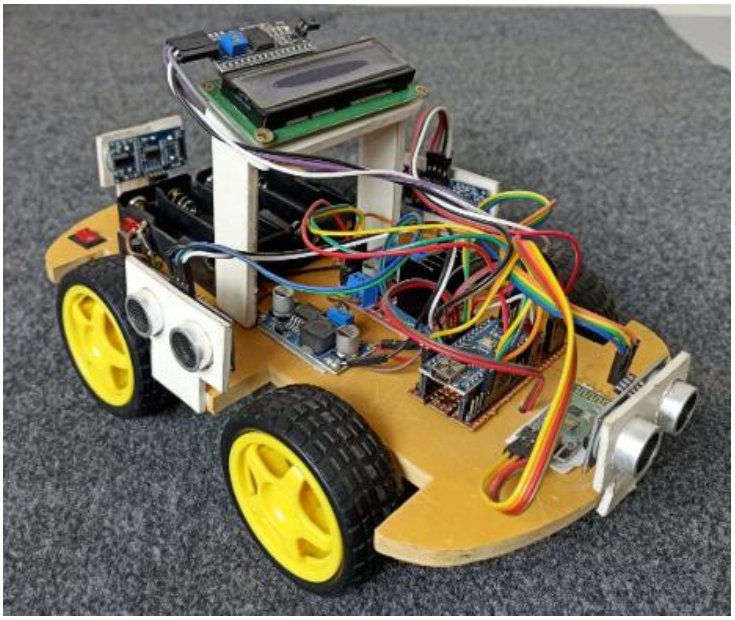
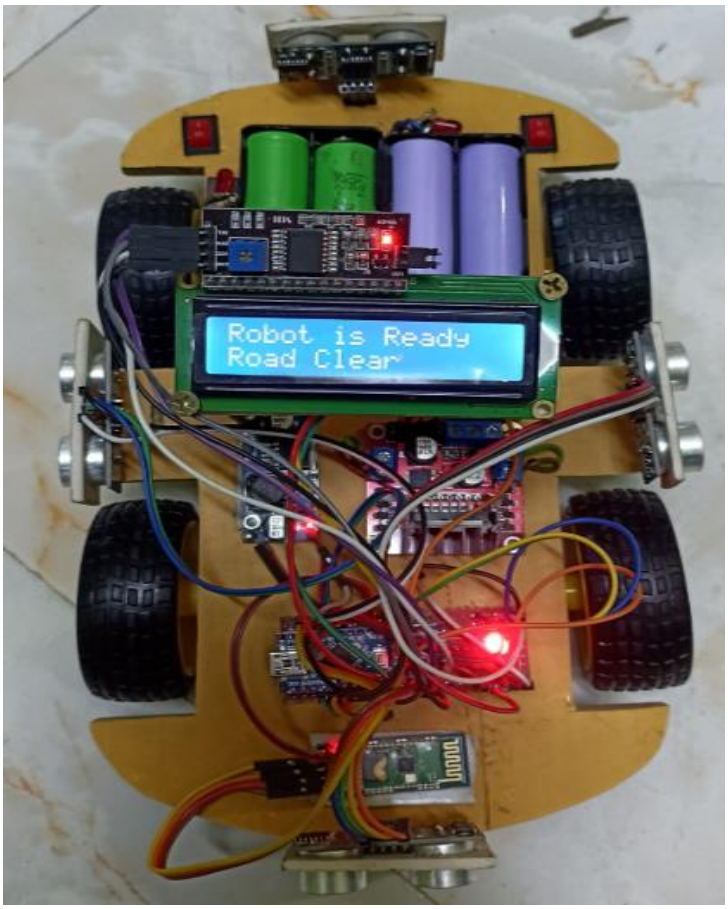


Figure 4.1: Complete Project prototype



4.2 Project progress Image

4.2 Advantages

There are many advantages of the project. Some of these are given below:

- Can detect the obstacle in-all-around of the robot.
- Can control the robot with the help of android apps control.
- This project is easy to use
- Cost effective
- User-friendly
- Automatic and requires less human interaction
- The whole system consumes very little energy.

4.3 Applications

This project has applications in many fields due its necessity. We have selected a few of them and they are given below:

- It can be use in industrial area.
- It can be used in any fire incident to inspection that place.
- It can be used in any industry work monitoring.
- It can be used Home Security

4.4 Limitations

Though this project has many advantages but it has some limitations as well and they are listed below:

- Used cheap Chinese products for the prototype so there's some processing delay present in the circuit
- This project can now be only used for small scale purposes.

4.5 Discussion

While working on our project, we did face some difficulties as it is a very complex system but the end results, we came up with were quite satisfactory. We have put the whole system through several tasks to validate our work and also have taken necessary notes for future improvements. Some future recommendations that we have involves improvement in system design and wiring, adding features for more efficient.

4.6 Cost Analysis

Table 5.1: Cost of this project

Sr. No	Product Name	Specification	Quantity	Unit Price (Taka)	Total Price (Taka)
1	Arduino Nano	ATMega328P	1	520	520
2	Battery	3.7 V	4	100	400
3	Ultrasonic Sensor	HCSR04	4	250	1000
4	Buck Converter	LM2596	1	200	200
5	Motor Driver	L298N	1	400	400
6	DC Gear Motor		4	250	1000
7	LCD Display	16*2	1	300	300
8	I2C Module		1	150	150
9	Wheel		4	100	400
10	Bluetooth Module	HC-06		400	400
11	Others				800
Total					**5570/=**

Chapter – 5

Conclusion and Future Scope

5.1 Conclusion:

Automobile safety systems play a vital role in reducing accidents and enhancing passenger protection, thereby saving lives. Advances in technology have revolutionized these systems, incorporating features like anti-lock braking systems (ABS), airbags, electronic stability control (ESC), and advanced driver assistance systems (ADAS) such as lane departure warnings, adaptive cruise control, and automatic emergency braking. These innovations work collectively to prevent collisions, minimize impact, and protect occupants in hazardous situations.

Furthermore, the integration of IoT and artificial intelligence is paving the way for smarter, autonomous vehicles with enhanced situational awareness and decision-making capabilities. However, ensuring reliability, addressing cybersecurity risks, and improving affordability remain significant challenges.

In conclusion, automobile safety systems are indispensable for modern transportation, fostering safer roads and reducing fatalities. Continuous innovation, combined with stringent safety standards and public awareness, will drive the evolution of safer vehicles, contributing to a sustainable and secure future for mobility.

5.2 Future Scope

The model can be improved by making some changes in the hardware and components. Some suggestions are given below-

- In future development this project can be develop by more sensor & alarm system.
- In future we will add fire extinguishing system.
- In future develop IoT System
- In future we will add GSM system.

References

[1] S. Biswas, R. Tatchikou and F. Dion, "Vehicle-to-Vehicle Wireless Communication Protocols for Enhancing Highway Traffic Safety," IEEE Communications Magazine, (2006), pp. 74-82.

[2] R. B. Chen, N. Li, . H. Feng Xiao and W. Hou, "Research on Safety Vehicle Distance Recognition Based on Stereo Vision and DSP," Applied Mechanics and Materials, Vols. 333-335, (2013), pp. 805-810.

[3] Shereen Ismail, Eman Alkhader and Said Elnaffar, “Object Tracking in Wireless Sensor Networks: Challenges and Solutions”, Journal of Computer Sciences, Vol. 12, No. 4, (2012), pp. 201-212.

[4] Khalid A. Darabkh, Shereen S. Ismail, Mohammad AlShurman, Iyad F. Jafar, Eman Al khader, Mamoun F.AlMistarihi,“Performance evaluation of selective and adaptive heads clustering algorithms over wireless sensor networks”, Journal of Network and Computer Applications, Elsevier, 2012, pp. 1-13, http://dx.doi.org/10.1016/j.jnca.2012.08.008.

[5] Lei Shi and Ki Won Sung, “Spectrum Requirement for Vehicle-to-Vehicle Communication for Traffic Safety”, IEEE Journal on Selected Areas in Communications, Oct. 2007, https://www.metis2020.com/wpcontent/uploads/publications/VTCSpring_2014_Shi _etal_Spe ctrumRequirementForV2VCommunication.pdf

[6] R. B. Chen, N. Li, H. F. Xiao and W. Hou, "DEVELOPMENT OF PRE-CRASH SAFETY SYSTEM WITH PEDESTRIAN COLLISION AVOIDANCE ASSIST," in 23rd International Conference on the Enhanced Safety of Vehicles, Seoul, Republic of Korea, 2013.

[7] X. Yin and M. Wang, "Research on Safety Distance Mathematical Model of Pro-Active Head Restraint in Rearend Collision Avoidance System," International Journal of Security and Its Applications, vol. 9, no. 1, (2015), pp.347- 356

[8] M. Prabha, M. Seema, P. Saraswathi, “Distance based Accident Avoidance System using Arduino”, International Research Journal of Engineering and Technology (IRJET), Vol. 02 Issue 07, (2015), pp. 777-780.

[9] Sivakumara, T. and Krishnarajb, R., 2013. A Study on Application of Advanced Automobile Safety Features and their Implication on Road Traffic Accidents and Road Fatalities.

[10] Singh, P., Sethi, T., Balabantaray, B.K. and Biswal, B.B., 2015, March. Advanced vehicle security system. In Innovations in Information, Embedded and Communication Systems (ICIIECS), 2015 International Conference on (pp. 1-6). IEEE.

[11] Lemmen, P., Fagerlind, H., Unselt, T., Rodarius, C., Infantes, E. and van der Zweep, C., 2012. Assessment of integrated vehicle safety systems for improved vehicle safety. Procedia-Social and Behavioral Sciences, 48, pp.1632-1641.

[12] Thenmozhi, R. and Govindarajan, S., 2016. Safety Related Services Using Smart Vehicle Connections. International Journal of Applied Engineering Research, 11(4), pp.2384- 2387.

[13] Prof.Pramodshelke, Ram Anjankar, SakesheeWanare, Ganesh Surve“A Review on Accident Avoidance by Eye Blink Sensor and Alcohol Sensor Using GSM” E-ISSN:2321-9637, 09th April 2017

[14] Sonali Shankar Chalwad, Snehal Bhimrao Gaikwad, PrashantA.Chougule “Accident Prevention Using Eye Blink Sensor”ISSN:2393- 2835, Volume-4,Issue-1, January-2017

[15] SuhasKatkar,MaheshManikKumbhar, Priti Navanath Kadam “Accident Prevention System Using Eye Blink Sensor” Volume:03, Issue:05, May-2016

[16] Pratik S. Danage, Ajay B.Dongare, SangramsinhS.Dongare, Rahul B.Ghogare, Ganesh N.Khare “Automatic Breaking Sysetm Using Eye Blinking Sensor”Vol.2, Issue 2, pp(166- 170),October2015- March-2016

[17] Deepa K. B, Chaitra M, Ankit Kumar ShrmaSreedar V. S, Prashant Kumar H. R.” Accident prevention by eye binking sensor and alcohol detector” Volume-4 , Issue-7 , pp: 351- 354, July-2015

[18] Dr.AntoBennet, M , Sankaranarayanan S, Ashokram S ,Dinesh Kumar T R,“Testing of Error Containment Capability in can Network”, International Journal of Applied Engineering Research, Volume 9, Number 19 (2014) pp. 6045-6054

[18] https://www.istockphoto.com/illustrations/car-stopping-distance

[19] https://www.istockphoto.com/illustrations/car-stopping-distance

[20] https://en.wikipedia.org/wiki/Ultrasonic_transducer

[21] https://en.wikipedia.org/wiki/Liquid-crystal_display

[22] https://en.wikipedia.org/wiki/Lithium-ion_battery

[23] https://en.wikipedia.org/wiki/Motor_drive

[24] https://en.wikipedia.org/wiki/Bluetooth

[25] https://www.arduino.cc/en/software

[26] https://easyeda.com/

Appendix

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <SoftwareSerial.h>

// Bluetooth pins (connect to RX and TX of the Bluetooth module)
#define BT_RX 10
#define BT_TX 11

// Motor control pins (using A0, A1, A2, A3 for direction)
#define IN1 A0
#define IN2 A1
#define IN3 A2
#define IN4 A3

// Ultrasonic sensor pins
#define TRIG_FRONT 2
#define ECHO_FRONT 3
#define TRIG_BACK 4
#define ECHO_BACK 5
#define TRIG_LEFT 6
#define ECHO_LEFT 7
#define TRIG_RIGHT 8
#define ECHO_RIGHT 9

// Initialize Bluetooth and I2C LCD
SoftwareSerial bluetooth(BT_RX, BT_TX); // RX, TX
LiquidCrystal_I2C lcd(0x27, 16, 2);     // I2C address, columns, rows (update address if needed)

long distanceFront, distanceBack, distanceLeft, distanceRight;
unsigned long lastSensorRead = 0;
unsigned long sensorInterval = 50; // Adjust sensor reading interval in milliseconds
bool lcdNeedsUpdate = true;

void setup() {
  Serial.begin(9600);
  bluetooth.begin(9600);

  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("Robot is Ready");

  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
```

```
  pinMode(TRIG_FRONT, OUTPUT);
  pinMode(ECHO_FRONT, INPUT);
  pinMode(TRIG_BACK, OUTPUT);
  pinMode(ECHO_BACK, INPUT);
  pinMode(TRIG_LEFT, OUTPUT);
  pinMode(ECHO_LEFT, INPUT);
  pinMode(TRIG_RIGHT, OUTPUT);
  pinMode(ECHO_RIGHT, INPUT);

  Serial.println("Bluetooth Initialized...");
  Serial.println("Waiting for RC Commands...");
}

void loop() {
  // Handle Bluetooth commands
  if (bluetooth.available()) {
    char incomingByte = bluetooth.read();
    Serial.print("Received: ");
    Serial.println(incomingByte);

    processBluetoothCommand(incomingByte);
  }

  // Read sensors at intervals
  if (millis() - lastSensorRead >= sensorInterval) {
    lastSensorRead = millis();

    // Read distances one at a time to reduce delay
    static int sensorIndex = 0;

    switch (sensorIndex) {
      case 0: distanceFront = getDistance(TRIG_FRONT, ECHO_FRONT); break;
      case 1: distanceBack = getDistance(TRIG_BACK, ECHO_BACK); break;
      case 2: distanceLeft = getDistance(TRIG_LEFT, ECHO_LEFT); break;
      case 3: distanceRight = getDistance(TRIG_RIGHT, ECHO_RIGHT); break;
    }

    sensorIndex = (sensorIndex + 1) % 4;
    lcdNeedsUpdate = true; // Trigger LCD update
  }

  // Update LCD only when needed
  if (lcdNeedsUpdate) {
    updateLCD();
    lcdNeedsUpdate = false;
  }
}

void processBluetoothCommand(char command) {
  switch (command) {
```

```
    case 'F': forward(); Serial.println("Moving Forward"); break;
    case 'B': backward(); Serial.println("Moving Backward"); break;
    case 'L': left(); Serial.println("Turning Left"); break;
    case 'R': right(); Serial.println("Turning Right"); break;
    case 'S': stopMotors(); Serial.println("Stopping"); break;
    default: Serial.println("Unknown Command"); break;
  }
}

void updateLCD() {
  if (distanceFront < 10) {
    lcd.setCursor(0, 1);
    lcd.print("Obstacle Front   ");
  } else if (distanceBack < 10) {
    lcd.setCursor(0, 1);
    lcd.print("Obstacle Back    ");
  } else if (distanceLeft < 10) {
    lcd.setCursor(0, 1);
    lcd.print("Obstacle Left    ");
  } else if (distanceRight < 10) {
    lcd.setCursor(0, 1);
    lcd.print("Obstacle Right   ");
  } else {
    lcd.setCursor(0, 1);
    lcd.print("Road Clear       ");
  }
}

void left() {
  digitalWrite(IN1, HIGH); digitalWrite(IN2, LOW);
  digitalWrite(IN3, HIGH); digitalWrite(IN4, LOW);
}

void right() {
  digitalWrite(IN1, LOW); digitalWrite(IN2, HIGH);
  digitalWrite(IN3, LOW); digitalWrite(IN4, HIGH);
}

void forward() {
  digitalWrite(IN1, LOW); digitalWrite(IN2, HIGH);
  digitalWrite(IN3, HIGH); digitalWrite(IN4, LOW);
}

void backward() {
  digitalWrite(IN1, HIGH); digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW); digitalWrite(IN4, HIGH);
}

void stopMotors() {
  digitalWrite(IN1, LOW); digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW); digitalWrite(IN4, LOW);
```

```
}

long getDistance(int trig, int echo) {
  digitalWrite(trig, LOW);
  delayMicroseconds(2);
  digitalWrite(trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(trig, LOW);

  long duration = pulseIn(echo, HIGH, 30000); // Timeout to prevent blocking
  return (duration > 0) ? (duration * 0.034 / 2) : -1; // Return -1 if no echo
}
```

_____END_____