1. INTRODUCTION

1.1 Problem Definition

Drunk Driving:

Drunk driving refers to the act of operating a motor vehicle while under the influence of alcohol or other substances that impair a person's ability to drive safely. It is a serious safety concern as alcohol can significantly impair a driver's coordination, reaction time, judgment, and overall cognitive functions. This impairment increases the risk of accidents, injuries, and fatalities for both the driver and others on the road.

Problem Definition: The problem of drunk driving involves individuals choosing to operate a vehicle despite being intoxicated. This behavior can lead to a higher likelihood of accidents and harm to themselves and others on the road.

Falling Asleep While Driving:

Falling asleep while driving is a dangerous situation where a driver becomes so fatigued that they unintentionally fall asleep at the wheel. This can result in loss of control over the vehicle, leading to accidents, injuries, and fatalities. Factors such as sleep deprivation, drowsiness, long hours of driving, and untreated sleep disorders contribute to this problem.

Problem Definition: Falling asleep while driving poses a significant risk to road safety. Drivers who are excessively tired or drowsy might lose focus, reaction time, and the ability to make timely decisions, putting themselves and others in danger.

It's important to address both of these issues through various strategies, including public awareness campaigns, stricter enforcement of laws and regulations, education on the risks of impaired driving, and encouraging responsible driving behaviors. These efforts aim to reduce the incidence of accidents and improve road safety for everyone.

1.2 Objective

- The Alcohol Detection with Engine Locking system helps to reduce accidents which are occurring due to drunk driving.
- The primary objective is to ensure the safety of both the driver and other road users by preventing intoxicated individuals from operating a vehicle.
- The chance of Loss of Life and Property due to drunken driving minimized. Less accidents, more safety.
- The main objective of this project is to develop a cost-effective device which integrates drowsy detection, alcohol detection while driving to prevent accident rate.
- The primary objective is to detect the early signs of driver drowsiness or sleepiness. By continuously monitoring the touch patterns of the driver.

1.3 Abstract

- The combination of alcohol and driver sleep detection with vehicle engine locking system would provide a powerful way to prevent drunk driving and drowsy driving accidents. This system would save lives and make the roads safer for everyone.
- This system prevents the drunk driving and drowsy driving accidents. This system would work by using an alcohol sensor to measure the level of alcohol in the driver's breath. If the alcohol level is above a certain threshold, the system would lock the engine and prevent the vehicle from starting.
- Drowsy driving is another major cause of accidents, and it can be just as dangerous as drunk driving. The system could use the sensors.

1.4 Methodology

- To implement the above goals, the following methodology needs to be followed:
- Specifying the Application and various components of the Architecture.
- Build connection between each module and sensors.
- Specifying each connection with microcontroller.
- Upload code throw microcontroller.

2.SCOPE

2.1 Existing System

- In those days, they used a Breathalyzer device. However, they depend on individual monitoring and are not conducted for every driver on the road.
- The authors propose using a smart helmet to avoid accidents. i.e., only two wheelers, which is not a feasible idea while driving, especially for short distances.
- The authors discuss complex health monitoring systems and IR (Infrared Sensors)9 to detect the presence of alcohol. A major drawback of this system is the possibility of a false alarm. The system is designed in such a manner that even a slight change in some condition can result in false alarms ringing even though everything is normal.
- A major drawback of this system is that they have used a PIC16F877A (Programmable Integrated Circuit) microcontroller, which is not as useful as an Arduino Uno microcontroller.
- The drawback of their system is that they are using an MQ2 alcohol sensor, which is not accurate and is not specifically sensitive to alcohol.
- The author has used the P89V51RD2 microcontroller, which is expensive when compared to the Arduino Uno. Also, this system has a limited scope of usage because it can only work with 2-wheelers and not with any other part of vehicle.
- Sleep detection systems typically depend on specific sensors, such as cameras, steering sensors, or infrared sensors, to monitor the driver's behavior and physiological indicators. However, these sensors may have limitations or dependencies that can impact the system's effectiveness.
- For example, poor lighting conditions can affect camera-based systems, and incorrect positioning of sensors may compromise the accuracy of the detection.
- To detect the driver's drowsiness, they use low-resolution cameras to capture the driver's facial image and facial expression, which cannot be detected at night and it requires extra money to implement.

2.2 Proposed System

- All the components are embedded in the Arduino UNO. The alcohol sensor, MQ-3 (Metal Oxide Quality-3), is the input to the Arduino UNO because it detects the presence of alcohol in the driver’s breath and sends data to the Arduino UNO for further action.
- When the system is started, the alcohol sensor is activated by the car engine. It will measure the presence of alcohol in the driver’s breath and display his or her condition on the LCD (Liquid Crystal Display).
- Next, the buzzer rings continuously to alert another vehicle user on the road. Then next, the parking light will be on. Then, the location of the vehicle is tracked down by using the GPS (Global Positioning System) module, and the details are sent to the concerned person via SMS (Short Message Service) notification.
- This module tracks the position of the vehicle and sends the details back to the Arduino. From there, the GSM (Global System for Mobile Communication) module is activated.
- A SMS notification and the vehicle's current location will be sent to related persons of the vehicle user. (Those are registered numbers that are saved on the SIM card of the GSM module.)
- If the driver is in a sleepy condition, the touch sensor is used to detect the driver's sleepiness.
- The touch sensor would need to be sensitive enough to detect even the slightest movement of the driver's head.
- The touch sensor would need to be able to distinguish between the driver's head and other objects that might be in contact with the sensor.
- The touch sensor would need to be able to alert the driver or another passenger if it detected that the driver was asleep. This could be done by sounding an alarm and vibrating the seat.
- It reduces the speed of the vehicle to 0 km/h, and the parking light will be on to alert other vehicles and other road users.

3.IOT (INTERNET OF THINGS)

3.1 Introduction

The Internet of Things (IoT) is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction.

In other words, IoT is a network of physical objects (things) that are embedded with sensors, software, and network connectivity to collect and exchange data. This allows these objects to be monitored and controlled remotely across the internet.

The IoT is a rapidly growing technology with the potential to revolutionize many industries. As the IoT continues to develop, we can expect to see even more innovative and useful devices that make our lives easier, more efficient, and safer.

IoT devices can be found in a wide variety of industries, including:

- **Manufacturing:** IoT devices are used to monitor and control industrial equipment, track inventory, and optimize production processes.
- **Healthcare:** IoT devices are used to monitor patient health, track medication, and manage medical devices.
- **Retail:** IoT devices are used to track inventory, manage customer flow, and personalize shopping experiences.
- **Transportation**: IoT devices are used to monitor traffic conditions, track vehicles, and improve safety.
- **Smart homes**: IoT devices are used to control home appliances, monitor energy usage, and provide security.

Here are some examples of IoT devices:

- **Smart thermostats**: These devices can be programmed to adjust the temperature of your home based on your schedule and preferences.
- **Smart light bulbs:** These bulbs can be turned on and off remotely, and they can even change color.

- **Smart security cameras:** These cameras can be monitored remotely, and they can send alerts if they detect motion.
- **Smart refrigerators:** These refrigerators can track the expiration dates of your food and suggest recipes based on what you have on hand.
- **Self-driving cars:** These cars use sensors and software to navigate the road without human input.

These are just a few examples of the many IoT devices that are available today. As the IoT continues to develop, we can expect to see even more innovative and useful devices that make our lives easier and more connected.

The IoT is still in its early stages of development, but it has the potential to revolutionize many industries. By connecting physical objects to the internet, IoT can provide businesses and consumers with new insights, improve efficiency, and make our lives more convenient and enjoyable.

3.2 Benefits of IoT

- **Improved efficiency:**

  IoT devices can collect and analyze data in real time, which can help businesses to improve their efficiency. For example, IoT devices can be used to monitor the performance of industrial equipment and identify potential problems before they cause downtime.

- **Increased productivity:**

  IoT devices can automate tasks that were previously performed manually, which can free up employees to focus on more strategic work. For example, IoT devices can be used to order supplies automatically when they are running low, or to schedule maintenance on equipment before it breaks down.

- **Better customer service:**

  IoT devices can collect data about customer preferences and behavior, which can be used to personalize the customer experience. For example, IoT devices can be used to recommend products to customers based on their past purchases, or to send alerts when a product is in stock that a customer has been waiting for.

- **Enhanced safety:**

  IoT devices can be used to monitor and control hazardous environments, which can help to prevent accidents. For example, IoT devices can be used to monitor the temperature of a chemical plant or to track the location of workers in a construction zone.

- **Reduced costs:**

  IoT devices can help businesses to reduce costs by automating tasks, improving efficiency, and preventing accidents. For example, IoT devices can be used to monitor energy usage in a building and automatically adjust the thermostat to save energy.

3.3 Sensors

- A **sensor** is a device that measures a physical quantity and converts it into a'signal' which can be read by an observer or by an instrument.
- A sensor is a device that detects or measures physical properties such as temperature, pressure, light, motion, or other environmental conditions.
- It converts these measurements into electrical signals or data that can beinterpreted and used for various purposes, such as monitoring, control, or analysis.
- Sensors are commonly used in a wide range of applications, from smartphonesand automobiles to industrial machinery and scientific instruments.

4. FEASIBILITY STUDY

4.1 Technical Feasibility

The technical feasibility in the proposed system deals with the technology used in the system. The concept of alcohol and driver sleep detection vehicle engine locking system has gained attentionin recent years as a potential technological solution to enhance their mobility and independence. It deals with the hardware and software used in the system whetherthey are of latest technology or not. It happens that after a system is prepared a newtechnology arises and the user wants the system based on that technology. Thus, ALCOHOL AND DRIVER SLEEP DETECTION VEHICLE ENGINE LOCKING SYSTEM is technically feasible.

4.2 Economical Feasibility

Economic analysis is the most frequently used method for evaluating the effectiveness of a new system. More commonly known as cost/benefit analysis.

4.3 Operational Feasibility

The project has been developed in such a way that it becomes very easy even fora person with little technical knowledge to operate it. This system is very user friendly and does not require any specialized person to operate. Thus, the project iseven operationally feasible.

5. SOFTWARE AND HARDWARE REQUIREMENTS:

5.1 Software Requirement:

1. IDE: Arduino UNO IDE
2. Operating System: Windows 11
3. Language: Embedded C

5.2 Hardware Requirement:

1. Arduino UNO R3
2. MQ-3sensor (Metal Oxide Quality-3)
3. Buzzer
4. LED (Light Emitting Diode)
5. LCD Display (Liquid Crystal Display)
6. GSM SIM800A (Global System for Mobile Communication)
7. DC Motor (Direct Current)
8. GPS Module (Global Positioning System)
9. Touch Sensor
10. Vibrator
11. Relay
12. Toggle Switch
13. Demo car

6. LANGUAGES USED

6.1 Embedded C:

Embedded C refers to a specialized version of the C programming language that is specifically designed for programming embedded systems. Embedded systems are computer systems that are integrated into other devices or machines and are responsible for performing specific tasks. These systems have constraints on resources like memory, processing power, and energy consumption, which require a different programming approach compared to traditional desktop or server applications.

Here are some key aspects and concepts related to Embedded C:

1. **Resource Constraints:**

   Embedded systems often have limited resources, such as memory and processing power. Embedded C programming focuses on optimizing code to make efficient use of these resources.

2. **Memory Management:**

   Efficient memory usage is crucial in embedded systems due to limited RAM and storage. Programmers need to carefully manage memory allocation and deallocation to prevent memory leaks and fragmentation.

3. **I/O Operations:**

   Embedded systems often interact with external hardware devices, sensors, and actuators. Embedded C provides ways to control these peripherals using hardware-specific registers and memory-mapped I/O.

4. **Real-Time Requirements:**

   Many embedded systems have real-time constraints, meaning they must respond to external events within a specific time frame. Embedded C allows programmers to write code that meets these timing requirements.

5. **Low-Level Programming:**

   Embedded C involves working at a low level, closer to the hardware, which requires understanding of processor architecture, instruction sets, and hardware interfaces.

6. **Interrupt Handling:**

Embedded systems often rely on interrupts to handle external events and ensure timely responses. Embedded C provides mechanisms to write interrupt service routines (ISRs) that execute in response to hardware interrupts.

7. **Cross-Compilation:**

In many cases, the code is developed on a host computer and then cross-compiled for the target embedded platform. This requires setting up a development environment andtoolchain specific to the target architecture.

8. **Portability:**

While Embedded C involves writing hardware-specific code, efforts are made to keep the code as portable as possible to facilitate reuse across different platforms.

9.Bit-Level Manipulation:

Embedded C allows manipulation of individual bits and binary operations, which isoften necessary when working with hardware registers and flags.

10. Code Optimization:

Embedded C programming involves optimizing code for performance, size, and energy consumption. Techniques like code size reduction, loop unrolling, and use of inline assembly may be employed.

11. Power Management:

Embedded systems often run on battery power or have strict power requirements. Embedded C programmers need to consider power-efficient programming techniques to extend battery life.

It's important to note that Embedded C might have variations or extensions depending on the target hardware and compiler being used. Additionally, familiarity withthe specific microcontroller or microprocessor architecture is crucial when working on embedded systems.

Overall, Embedded C is a specialized skill set that requires a deep understanding ofboth C programming concepts and the unique challenges posed by embedded systems.

7. IDE SPECIFICATION

7.1 Arduino IDE

Arduino - Installation

In this section, we will learn in easy steps how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1 − First you must have your Arduino board and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable.

In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image.

Step 2 − Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.

Step 3 − Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4 − Launch Arduino IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

Step 5 − Open your first project.

Once the software starts, you have two options −

1. Create a new project.
2. Open an existing project example.

To create a new project, select File → New.

To open an existing project example, select File → Example → Basics → Blink.

To open an existing project example, select File → Example → Basics → Blink. Here, we are selecting just one of the examples with the name Blink. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 6 − Select your Arduino board.

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

Go to Tools → Board and select your board.

Step 7 − Select your serial port.

Select the serial device of the Arduino board. Go to Tools → Serial Port menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be on the Arduino board. Reconnect the board and select that serial port.

Step 8 − Upload the program to your board.

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.

A − Used to check if there is any compilation error.

B − Used to upload a program to the Arduino board.

C − Shortcut used to create a new sketch.

D − Used to directly open one of the example sketches.

E − Used to save your sketch.

F − Serial monitor used to receive serial data from the board and send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

Arduino - Program Structure

In this chapter, we will study in depth the Arduino program structure and we will learn more new terminologies used in the Arduino world. The Arduino software is open-source. The source code for the Java environment is released under the GPL and the C/C++ microcontroller libraries are under the LGPL.

Sketch − The first new terminology is the Arduino program called “sketch”.

Structure

Arduino programs can be divided in three main parts: Structure, Values (variables and constants), and Functions. In this tutorial, we will learn about the Arduino software program, step by step, and how we can write the program without any syntax or compilation error.

Let us start with the Structure. Software structure consist of two main functions −

1. Setup( ) function
2. Loop( ) function

```
            void setup()
      {
      }
```

PURPOSE − The setup() function is called when a sketch starts. Use it to initialize the variables, pin modes, start using libraries, etc. The setup function will only run once, after each power up or reset of the Arduino board.

8. INPUT & OUTPUT REPRESENTETION

Input design is a part of overall system design. The main objectiveduring the input design is as given below:

- To produce a cost-effective method of input.
- To achieve the highest possible level of accuracy.
- To ensure that the input is acceptable and understood by the microcontroller.

8.1 Input Stages

The main input stages can be listed below:

- Data recording
- Data transcription
- Data conversion
- Data verification
- Data control
- Data transmission
- Data validation
- Data correction

8.2 Input Types

It is necessary to determine the various types of inputs.Inputs can be categorized as follows:

- External inputs, which are prime inputs for the system.
- Internal inputs, which are user communications with the system.

8.3 Input Media

At this stage choice has to be made about the input media.

To conclude about the input media consideration has to be given.

- Type of input

- Flexibility of format
- Speed
- Accuracy
- Verification methods
- Rejection rates
- Ease of correction
- Security
- Easy to use
- Portability

Keeping in view the above description of the input types and input media, it canbe said that most of the inputs are of the form of internal and interactive.

As Input data is to be the directly keyed in by the user, the keyboard can beconsidered to be the most suitable input device.

8.4 Output Definition

The outputs should be defined in terms of the following points:

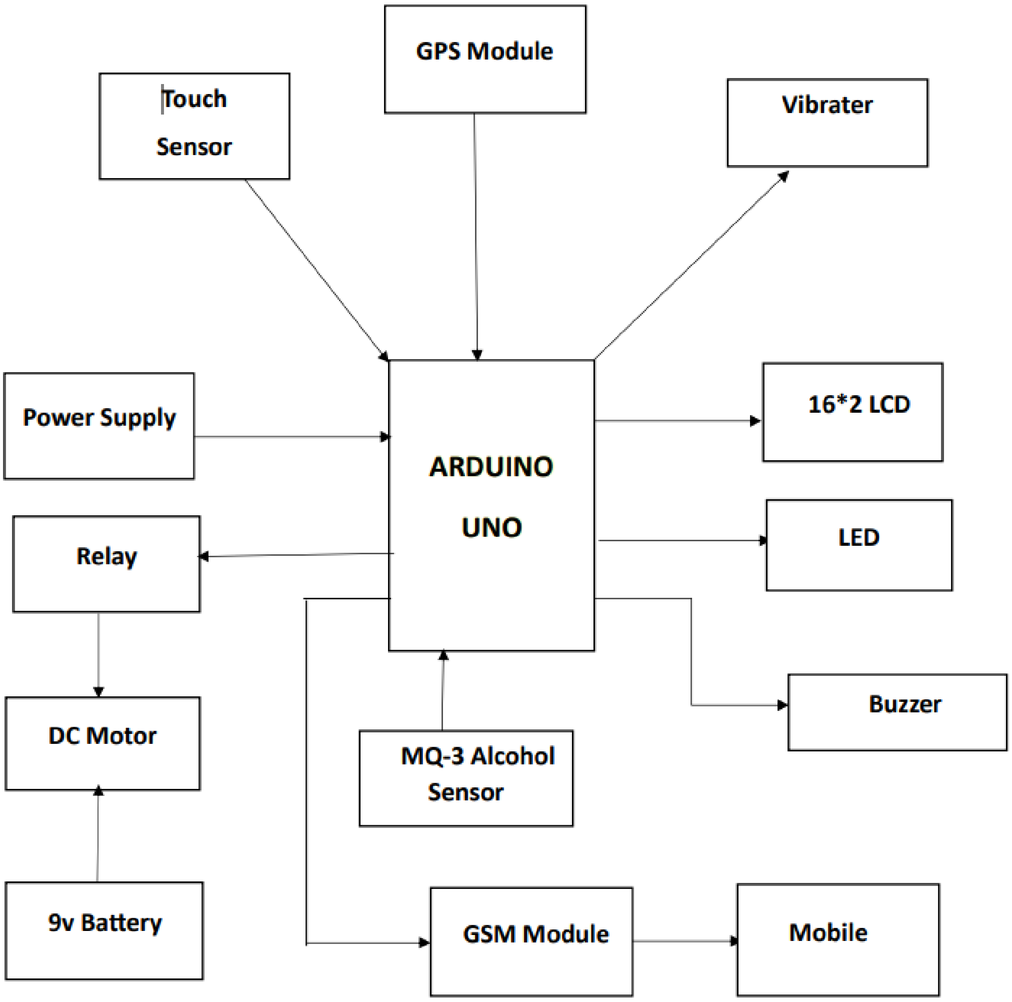
- Type of the output
- Content of the output
- Format of the output
- Location of the output
- Frequency of the output
- Volume of the output
- Sequence of the output

8.5 Output Media

- Buzzer
- Vibrator
- Mobile phone

9.SYSTEM DESIGN

9.1Circuit Diagram



9.2 Port Specification

Port 2: touch pin

Port 3: vibrator pin

Port 4: led1 pin

Port 5: led2 pin

Port 8: buzzer pin

Port 9: motor pin

Port 10: RX for GSM

Port 11: TX for GSM

Port 12: RX for GPS

Port 13: TX for GPS

9.2.1 MQ-3 Port Specification

Data pin: port A0

Power pin

Ground pin

9.2.2 Motor port Specification

Data pin: port 9

Power pin

Ground pin

9.2.3 Touch Port Specification

Data pin: port 2

Power pin

Ground pin

9.2.4 Lcd Port Specification

Data pin: port A4

Data pin: port A5

Power pin

Ground pin

9.2.5 Buzzer Port Specification

Data pin: port 8

Ground pin

9.2.6 Vibrator Port Specification

Data pin: port 3

Ground pin

9.2.7 GSM Port Specification

TX: port 10

RX: port 11

Power pin

Ground pin

9.2.8 GPS Port Specification

TX: port 12

RX: port 13

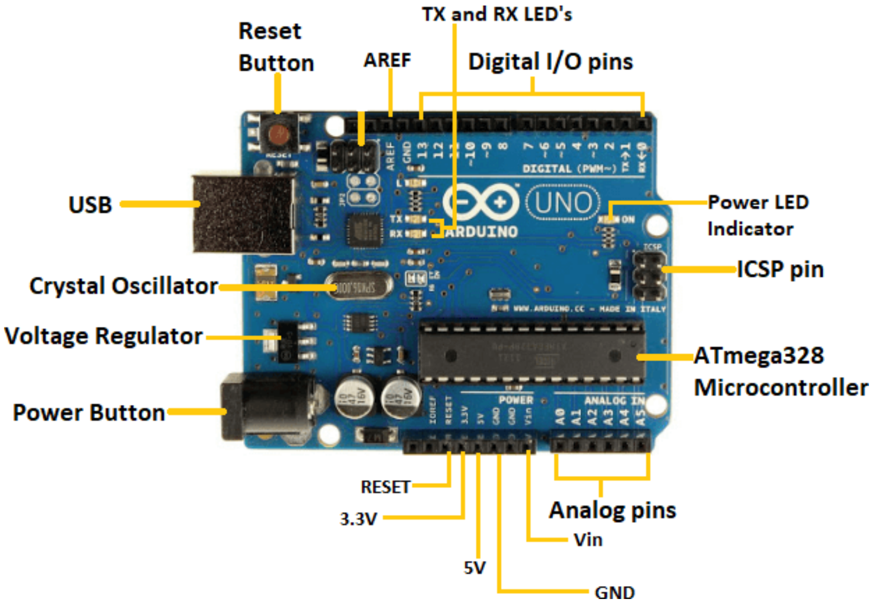
Power pin

Ground pin

9.3 Hardware Specification

9.3.1 Arduino UNO Board

The Arduino UNO is a standard board of Arduino. It is based on an ATmega328P microcontroller. The Arduino UNO includes 6 analogue pin inputs, 14 digital pins, a USB connector, a power jack, and an ICSP (In-Circuit Serial Programming) header. It is programmed based on IDE, which stands for Integrated Development Environment. It can run on both online and offline platforms.



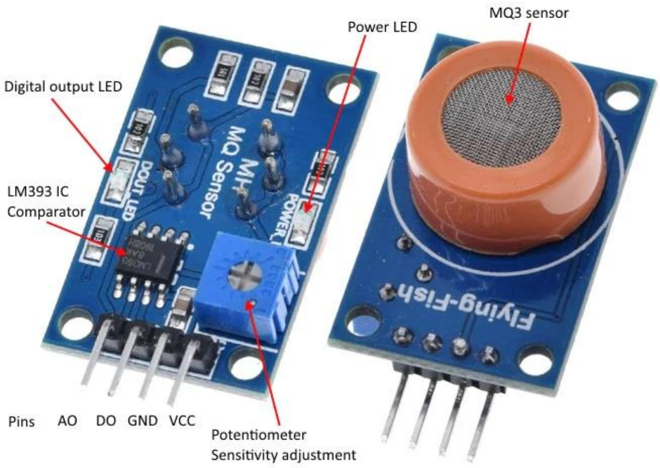
The Arduino Uno R3 board includes the following specifications.

- It is an ATmega328P based Microcontroller
- The Operating Voltage of the Arduino is 5V
- The recommended input voltage ranges from 7V to 12V
- The i/p voltage (limit) is 6V to 20V
- Digital input and output pins-14
- Digital input & output pins (PWM)-6
- Analog i/p pins are 6
- DC Current for each I/O Pin is 20 mA
- DC Current used for 3.3V Pin is 50 mA
- Flash Memory -32 KB, and 0.5 KB memory is used by the boot loader

- SRAM is 2 KB
- EEPROM is 1 KB
- The speed of the CLK is 16 MHz
- In Built LED
- Length and width of the Arduino are 68.6 mm X 53.4 mm

9.3.2 MQ-3 Sensor (Metal Oxide Quality-3)

The MQ3 alcohol gas sensor is a semiconductor sensor that uses the change in resistance of a tin dioxide (SnO2) layer to detect the presence of alcohol gas. The sensor has the following specifications:



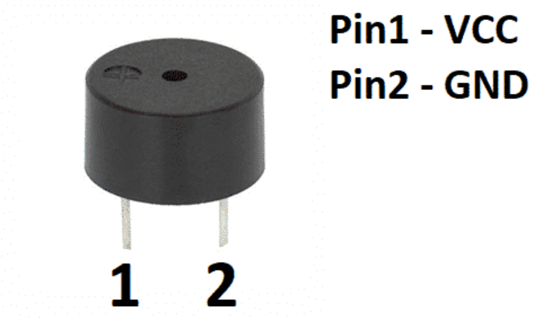
The MQ-3 includes the following specifications

- Detection gas: Alcohol gas
- Concentration range: 0.04 mg/L to 4 mg/L
- Supply voltage: 5 V
- Heater voltage: 5 V
- Heater resistance: 31 Ω
- Load resistance: Adjustable
- Sensitivity: ≥5
- Sensing resistance: 1MΩ to 8MΩ @ 0.4mg/L alcohol
- Operating temperature: -10°C to 50°C

- Storage temperature: -20°C to 70°C
- Humidity: <95%RH
- Response time: 15 seconds
- Preheating duration: 20 seconds

9.3.3 Buzzer

A Buzzer is an audio signaling device. There are many types of buzzer and here 5V passive Buzzer is used, which is used to create the sound.



Buzzer Specifications

- Operation Voltage: 5V DC.
- Frequency: 3,300Hz.
- Current: <15mA.
- SPL: 85dBA/10cm.
- Color: Black.
- Operating Temperature: - 20° to +60°C.
- Polarity: Positive Pin marked on the surface.
- Number of pins: 2.

9.3.4 Light Emitting Diode (LED)

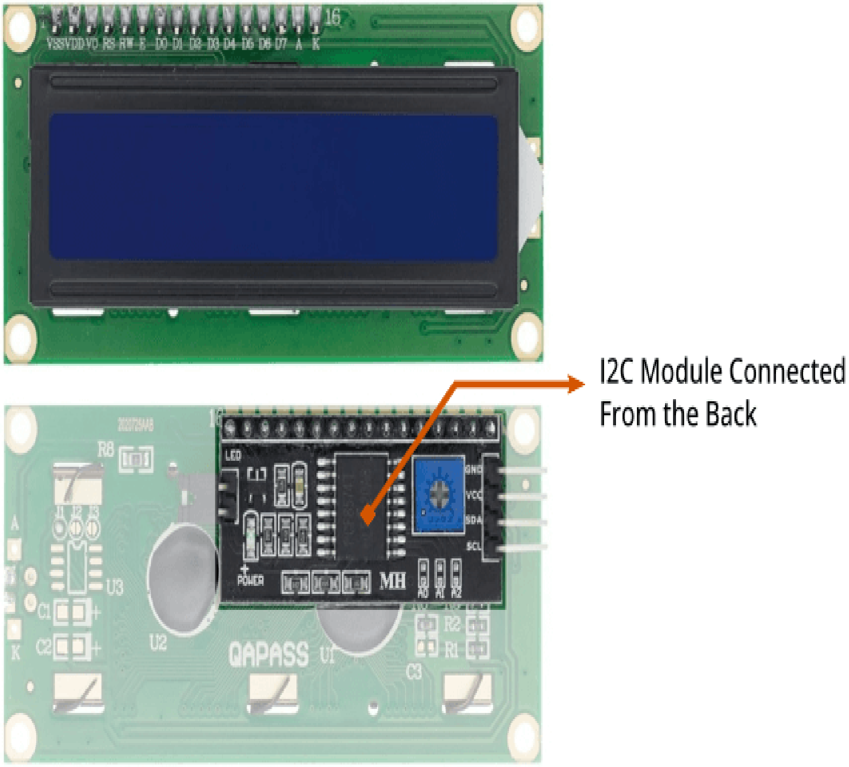
A few LED lights are used in our project; they are used to indicate the power indication, the presence of alcohol, and also for parking lights. If alcohol is present, then a red LED will glow, indicating the presence of alcohol, and if no alcohol is detected, then a green LED will glow.

LED Specifications

- Long Life: LEDs can last over 100,000 hours (10+ years) if used at rated specifications
- No annoying flicker like from fluorescent lamps
- LEDs are impervious to heat, cold, shock and vibration
- LEDs do not contain breakable glass
- Solid-State, shock and vibration resistant
- Extremely fast turn On/Off times
- Low power consumption puts less load on the electrical systems increasing battery life

9.3.5 LCD Display (Liquid Crystal Display)

16X2 LCD is a device used to display messages in the form of text and numbers. It is easy to program and can be used with various microcontrollers. The 16X2 LCD has two registers command and data. The command registers store command instructions given to the LCD. Commands are instructions given to the LCD to perform predefined tasks such as initialization, clearing the screen, setting the cursor position, controlling the display, etc. Data registers store the data displayed on the LCD. The data is the ASCII value of the characters displayed on the LCD. In our project, the LCD plays a very important role in displaying information about the status of the system.



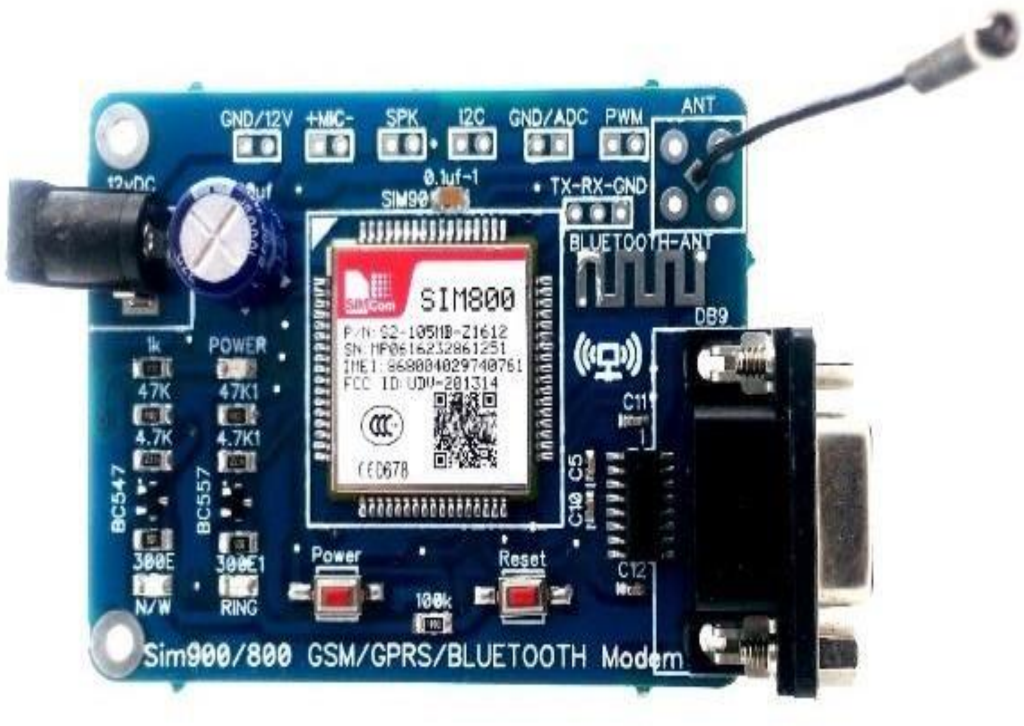
Lcd Specification

- Display: 16 characters x 2 lines
- Color: White characters on blue background
- Backlight: Adjustable brightness
- Interface: I2C
- Voltage: 5V
- Current: 2 mA @ 5 VDC
- Dimensions: 80 x 36 mm (3.1" x 1.4")
- Address: 0x27 (default)

9.3.6 GSM (Global System for Mobile Communications)

GSM is developed by the European Telecommunications Standards Institute (ETSI).

A GSM module or a GPRS module is a chip or circuit that use to establish communication between a mobile device or a computing machine and a GSM or GPRS system.



GSM Specifications

- Industry: telecommunication
- Founded: December 1991
- Successor:2G
- Products: digital cellular networks

9.3.7 DC Motor (Direct Current)

A DC, or direct current, motor works on the principle that a current-carrying conductor is placed in a magnetic field. In our project, a DC motor is used as an engine starter, which would be connected to the engine. The speed of a DC motor is directly proportional to the supply voltage, so if we reduce the supply voltage, the motor will run at half speed. The speed controller works by varying the average voltage sent to the motor. This voltage depends on the alcohol sensor (mq3). That means when the alcohol sensor senses an alcohol percentage less than 40%, the motor will run. But if the sensor senses an alcohol percentage above 40%, the motor will stop.

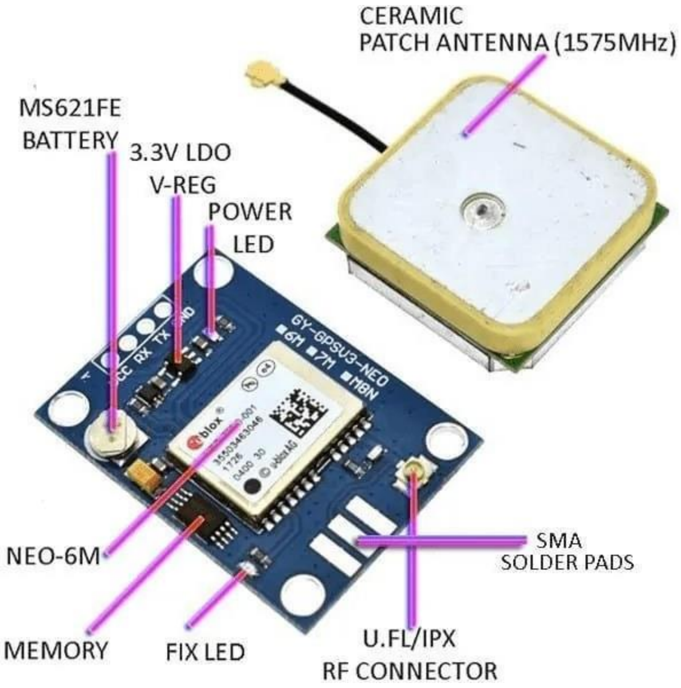
DC Motor Specifications

- Output Voltage Range: 0 to 30 V
- Output Current: 0 to 5 amp.
- Load Regulation : +/- 0.05 % (From No Load to Full Load)
- Line Regulation: +/- 0.05 %
- Ripple :+/- 0.05 % rms
- Digital Meter: (Separate Meter for DC Voltage & current)
- Accuracy For Voltage: 0.5% +/- 2 dgt For Current: 0.5% +/- 2 dgt
- Protection Overload & Short circuit
- Input Power Requirement 230 V or 110 V +/- 10 %, 50 / 60 Hz

9.3.8 GPS Module (Global Positioning System)

A GPS module is used to locate and acquire farmland information, including yield monitoring, soil sample collection and etc.

The computer system determines the management measures of farmland plotsby analyzing and processing the data, and loads the yield and soil status informationinto the GPS equipment.

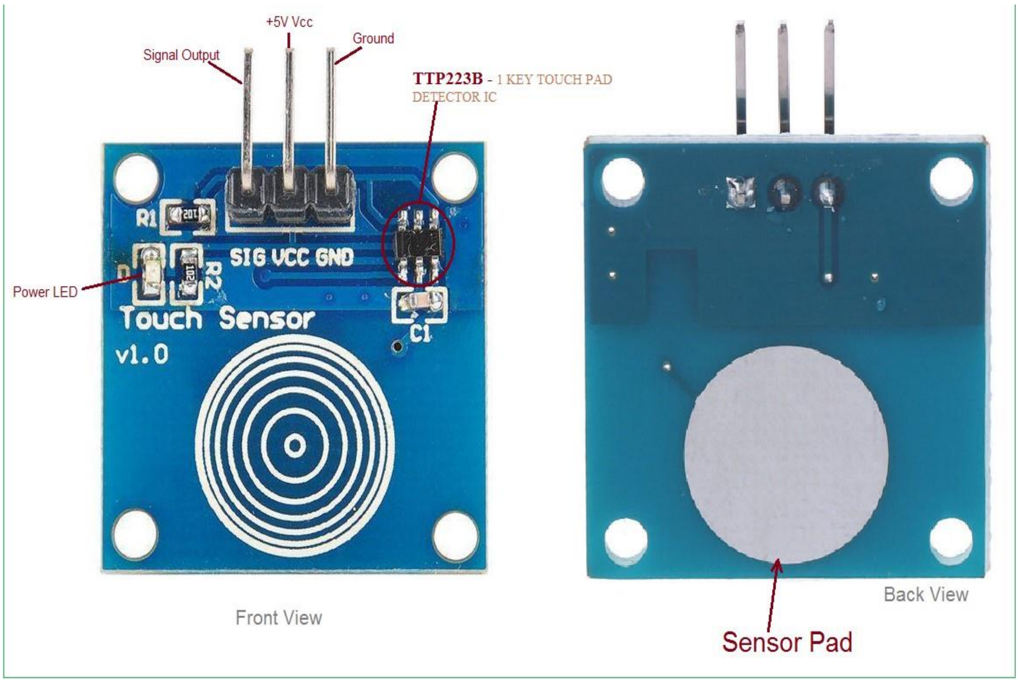


GPS Module Specification

- The GPS chipset is Media Tek MT 3318, 51 Channel.
- Frequency is L1, 1575.42 MHz; C/A Code.
- Protocol: NMEA 0183 v3.01, MTK NMEA Compound.
- UART interface.
- Baud Rate of 9600bps (default).
- Working Temperature: -40 °C to 85 °C.

9.3.9 Touch sensor

Touch sensors can be used to adjust the seat position. Touch sensors can also be used to detect head posture. Drowsy drivers often tilt their heads forward or to the side. Touch sensors can be placed on the driver's head to measure the angle of the head and identify changes in head posture that may indicate drowsiness.



Touch Sensor Module Specification

- Operating voltage 3V to 5V
- Operating current 1mA to 10mA
- Response time 10ms to 100ms
- Sensitivity 1mm to 10mm
- Resolution 1mm to 10mm
- Number of channels 1, 2, or 4
- Environmental conditions -20°C to 70°C, 5% to 95% RH

9.3.10 Vibrator

A vibrator is an electromechanical device that takes a DC electrical supply and converts it into pulses that can be fed into a transferor. It is similar in purpose (although greatly different in operation) to the solid-state power inverter.

Vibrator specification

- Voltage Range: DC 2.5-4V.
- Motor Diameter: 10 mm.
- Motor Thickness: 3.4 mm.
- Min. Rated Speed: 9000RPM
- Max. Rated Current: 90mA

9.3.11 Relay

A relay is an electrically operated switch. It consists of a coil, which receives an electric signal and converts it to a mechanical action, and contacts that open and close the electric circuit.

Relay specification

- Switched Power 10 VA 60 VA

- Breakdown Voltage 300 V peak 1000 V peak
- Carry Current 1.0 Amps 2.5 Amps
- Switch Speed 1 ms 2 ms

9.3.12 Toggle Switch

A toggle switch is a type of electromechanical switch that uses a lever or baton as an actuator. The lever is typically moved up and down or left and right to switch an electrical circuit. Toggle switches are available in many sizes and configurations and offer a wide range of uses.

Toggel Switch specification

- Mechanical life: 40,000 make and break cycles.
- Maximum contact resistance: 10MOhms.
- Initial voltage: 2 to 4V dc.
- Current: 10mA for both silver and gold plated contacts.
- Minimum Insulation resistance: 1,000MOhms.
- Dielectric strength: 1,000Vrms at sea level.
- Operating temperature: -30 Degree Celsius to 85 Degree Celsius.

9.3.13 Demo Car

In the alcohol and driver sleep detection and engine locking system project, a demo car is typically equipped with sensors that can detect the presence of alcohol in the driver's breath. These sensors are integrated into the car's interior, often near the steering wheel. When a driver exhales into the sensor, it measures the alcohol content in their breath.

If the sensor detects alcohol levels above a certain threshold, the engine locking system is activated. This prevents the car's engine from starting or continuing to run.

The demo car show cases how this technology works in a controlled environment. Which could potentially be integrated into real vehicles to reduce the risk of accidents caused by drunk driving.

10. DATAFLOW DIAGRAM

A data flow diagram is graphical tool used to describe and analyze movement of data through a system. These are the central tool and the basis from which the other components are developed. The transformation of data from input to output, through processed, may be described logically and independently of physical componentsassociated with the system. These are known as the logical data flow diagrams. The physical data flow diagrams show the actual implements and movement of data betweenpeople, departments and workstations. A full description of a system actually consists of a set of data flow diagrams. Using two familiar notations Yourdon, Gene and Sarsen notation develops the data flow diagrams. Each component in a DFD is labeled with a descriptive name. Process is further identified with a number that will be used for identification purpose. The development of DFD'S is done in several levels. Each process in lower-level diagrams can be broken down into a more detailed DFD in the nextlevel. The lop-level diagram is often called context diagram. It consists a single processbit, which plays vital role in studying the current system. The process in the context leveldiagram is exploded into other process at the first level DFD.

The idea behind the explosion of a process into more process is that understanding at one level of detail is exploded into greater detail at the next level. Thisis done until further explosion is necessary and an adequate amount of details describedfor analyst to understand the process.

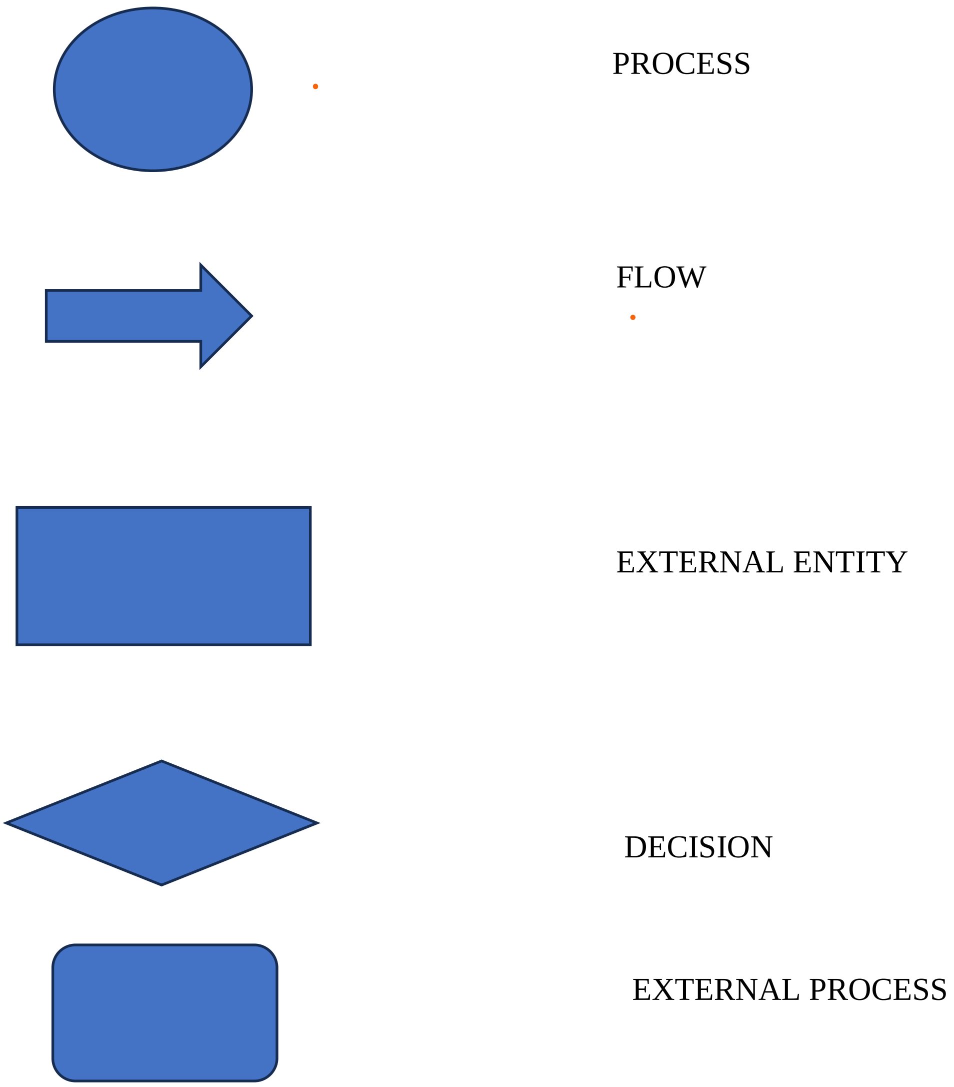
Larry Constantine first developed the DFD as a way of expressing system requirements in a graphical from, this lead to the modular design.

A DFD is also known as a "bubble Chart" has the purpose of clarifying system requirements and identifying major transformations that will become programs in systemdesign. So, it is the starting point of the design to the lowest level of detail. A DFD consists of a series of bubbles joined by data flows in the system.

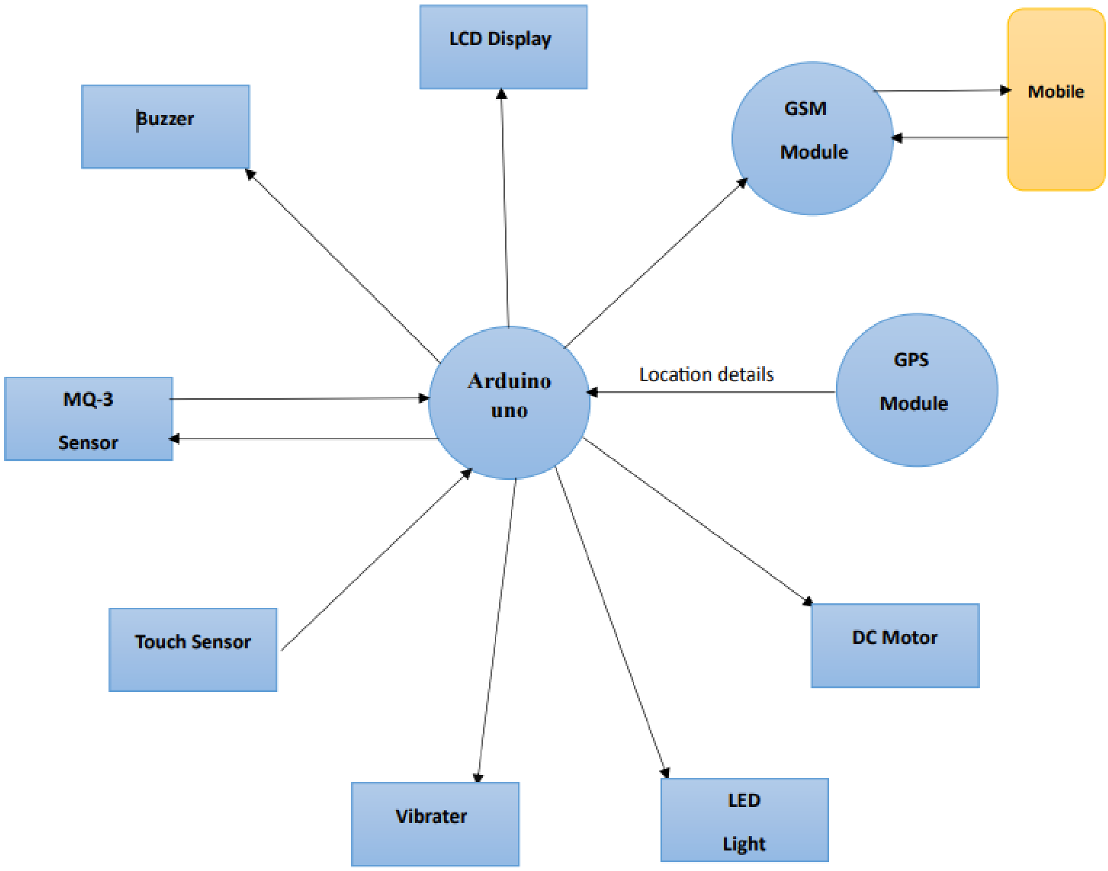
10.1 DFD Symbols:

In the DFD, there are three symbols.

1. A rectangle defines the external entity.
2. An arrow identifies data flow. It is the pipeline through which the informationflows.
3. A circle represents a process that transforms incoming data flow into outgoing data flows.

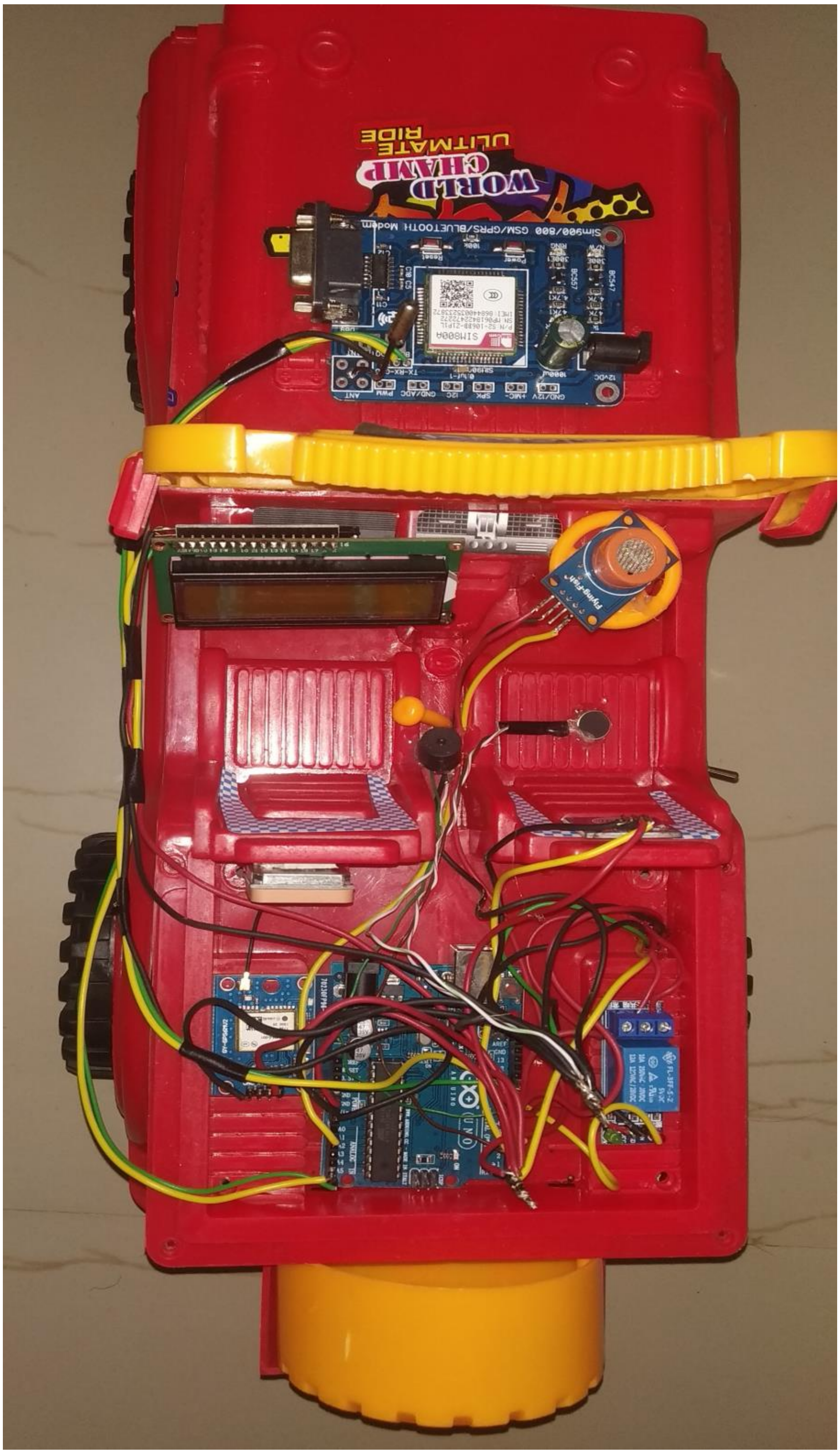


10.2 Data Flow Diagram of Alcohol and Driver Sleep Detection Vehicle Engine Locking System

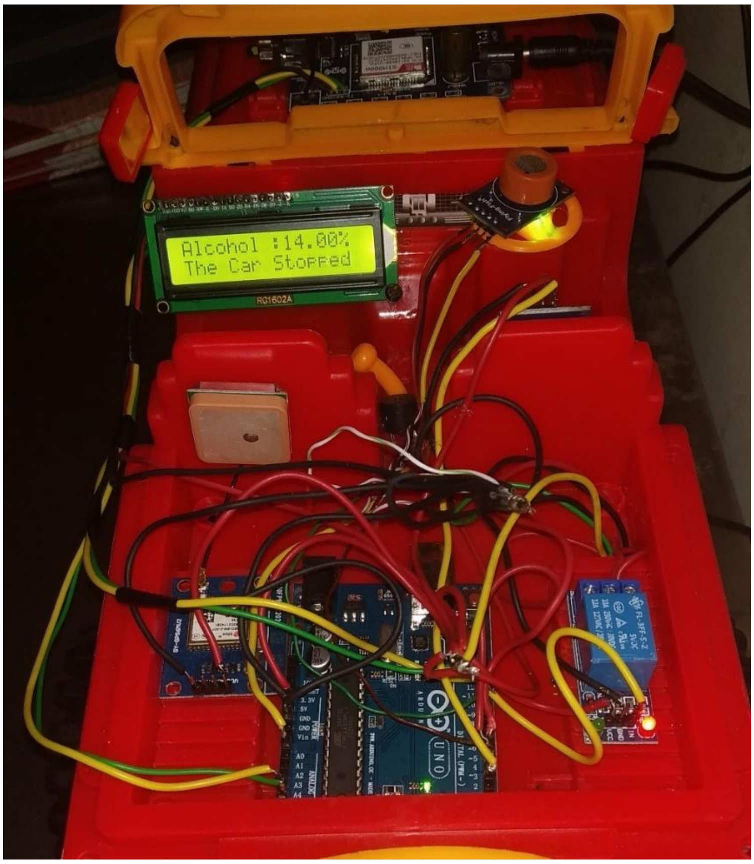


11.PROJECT SNAPSHOTS

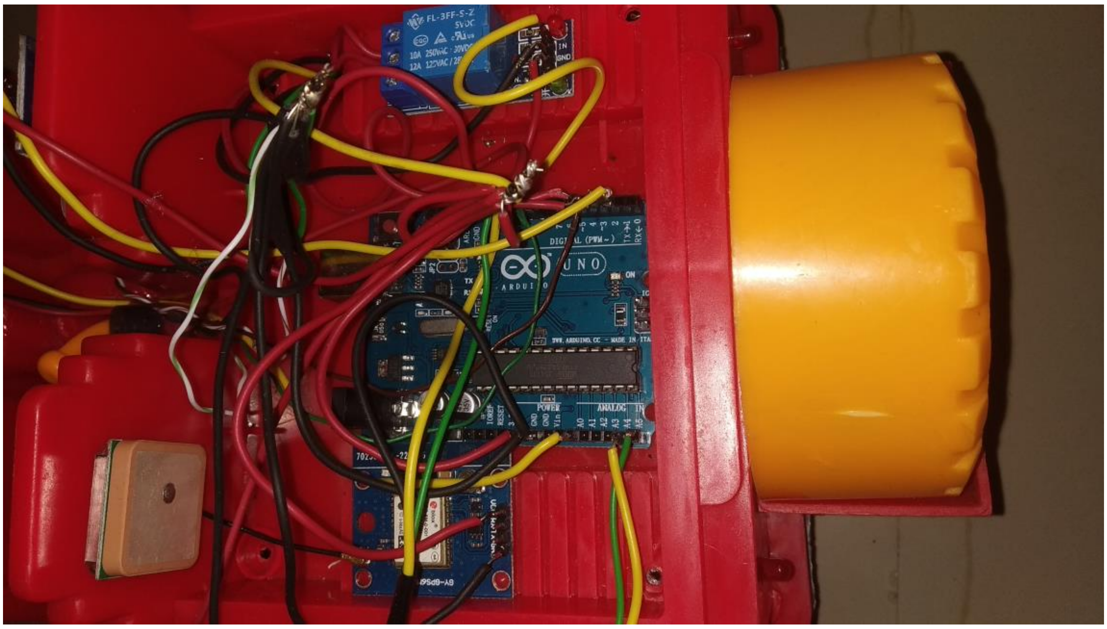
11.1 Model View



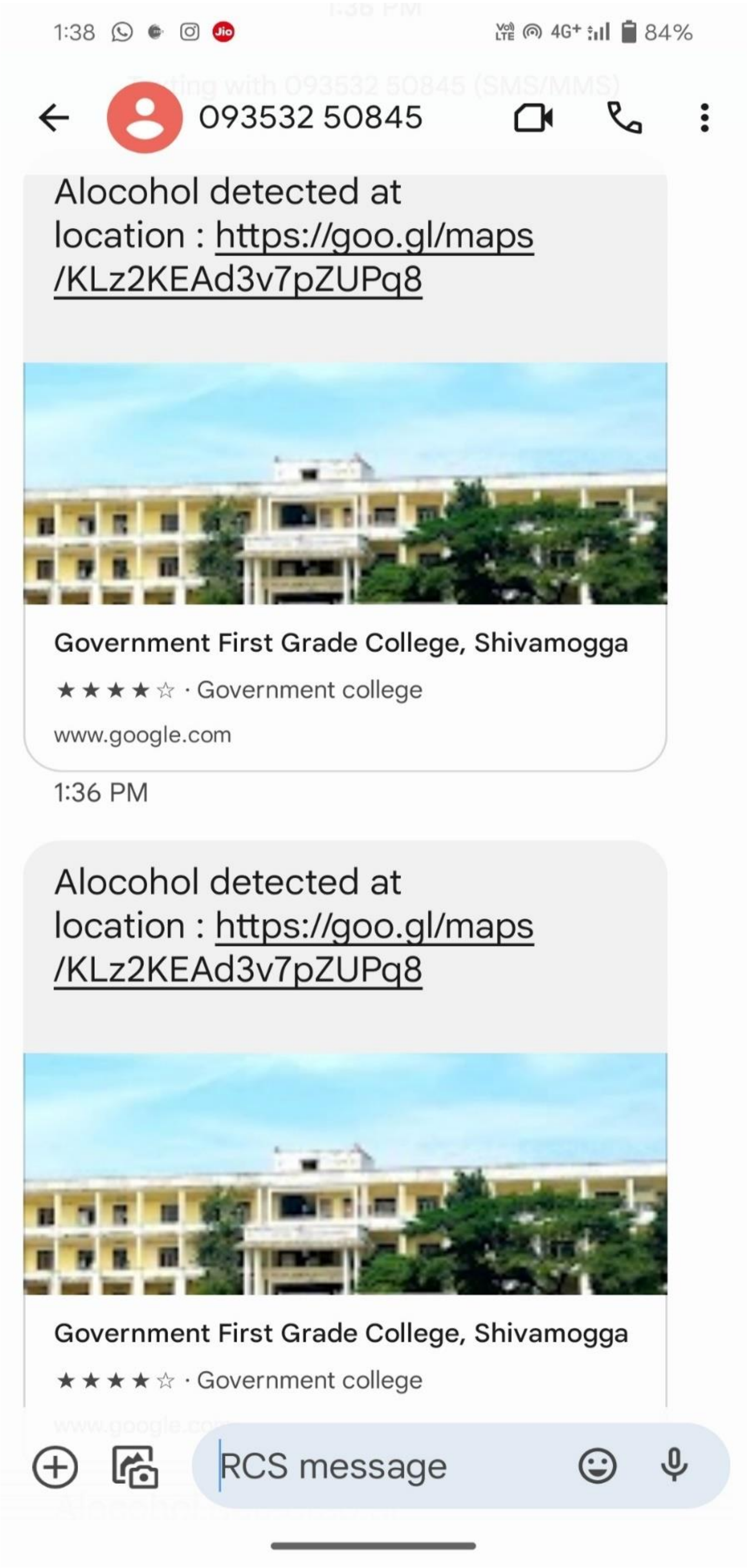
System Started



Circuit Connection to Arduino UNO



11.2 Location Tracking Snapshots



Alocohol detected at location : https://goo.gl/maps/KLz2KEAd3v7pZUPq8

Government First Grade College, Shivamogga

★★★★☆ · Government college

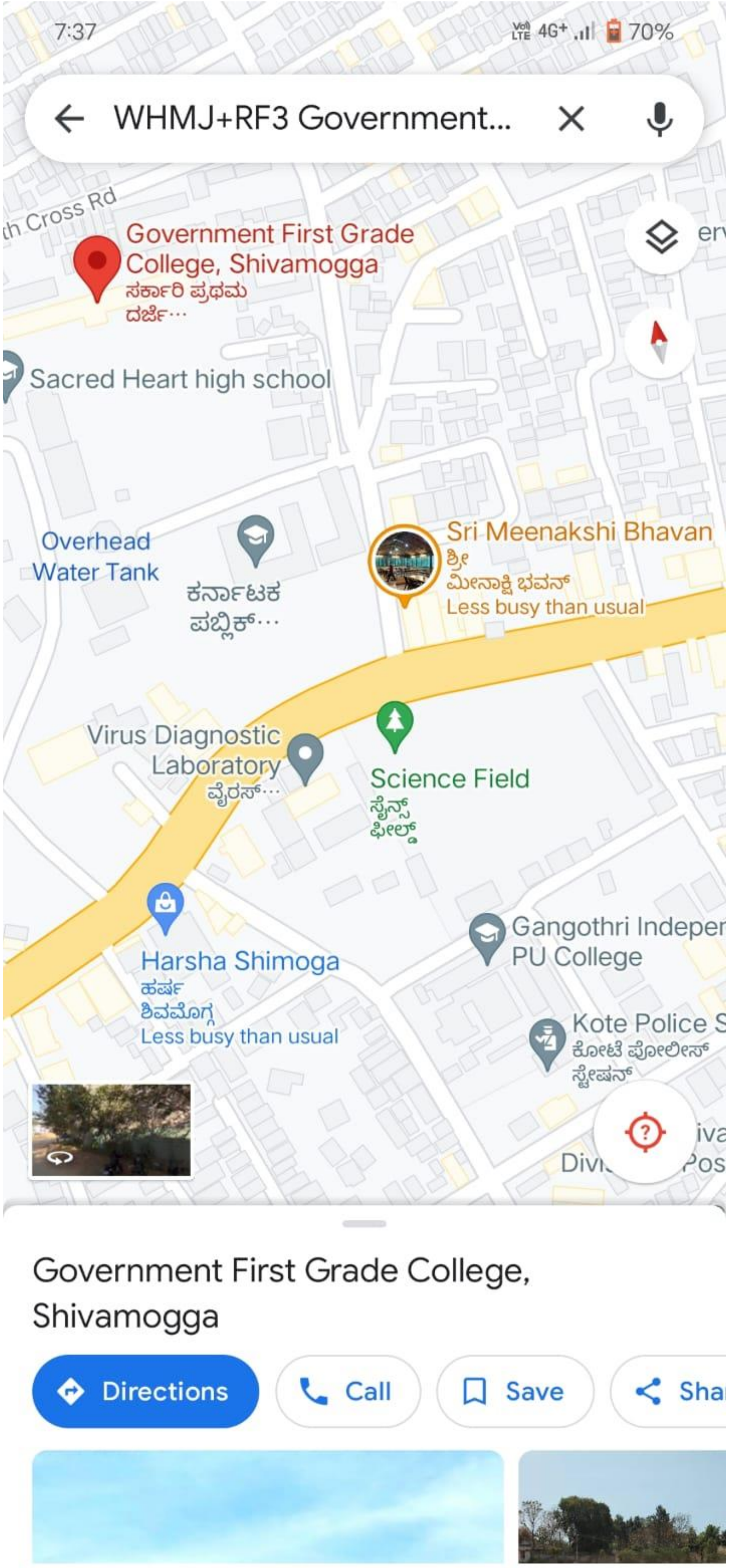
www.google.com

1:36 PM

Alocohol detected at location : https://goo.gl/maps/KLz2KEAd3v7pZUPq8

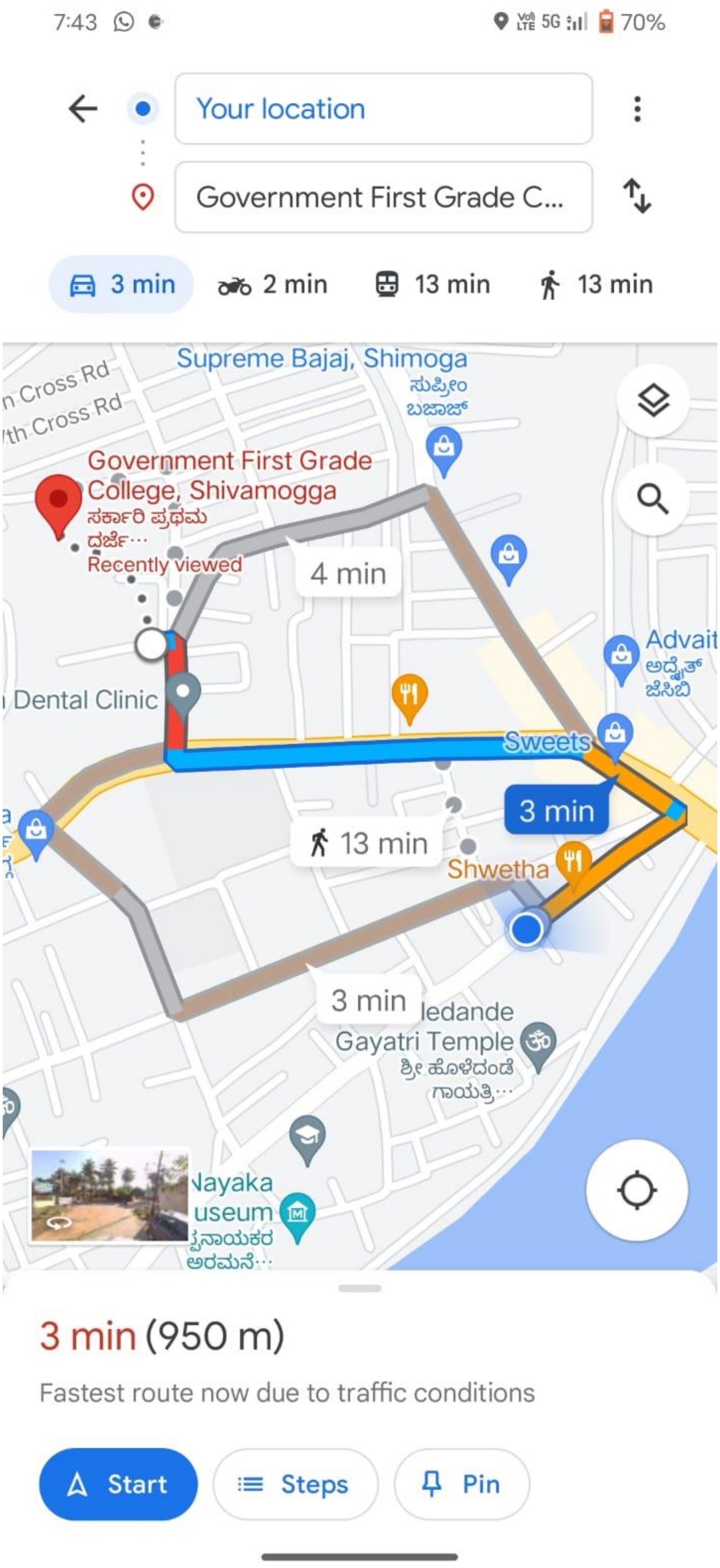
Government First Grade College, Shivamogga

★★★★☆ · Government college



WHMJ+RF3 Government...

Government First Grade College, Shivamogga



Your location

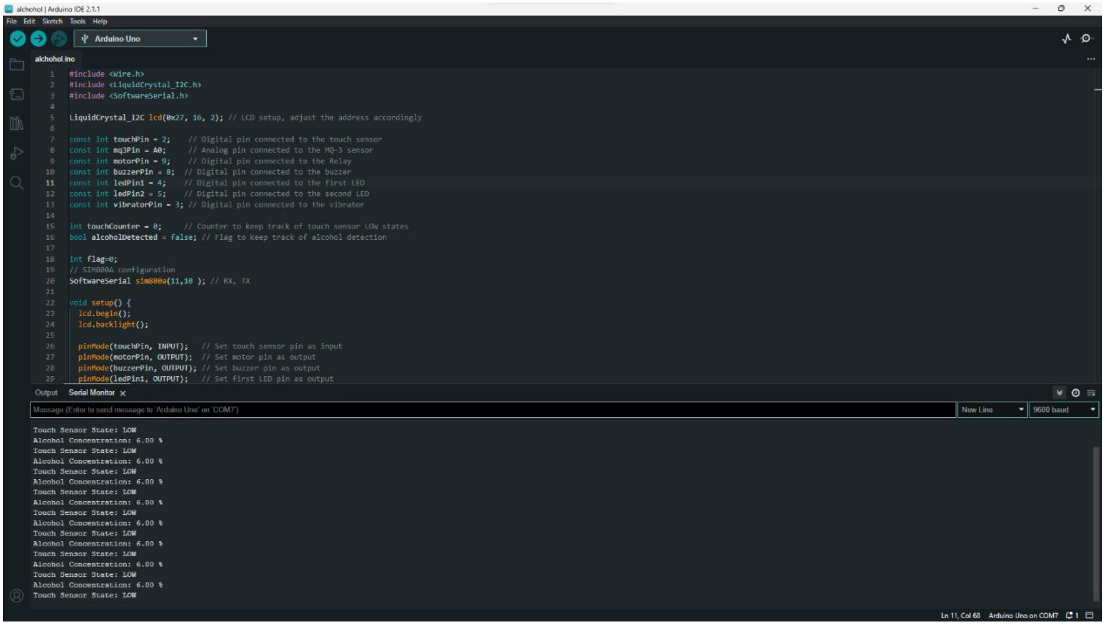
Government First Grade C...

3 min 2 min 13 min 13 min

3 min (950 m)

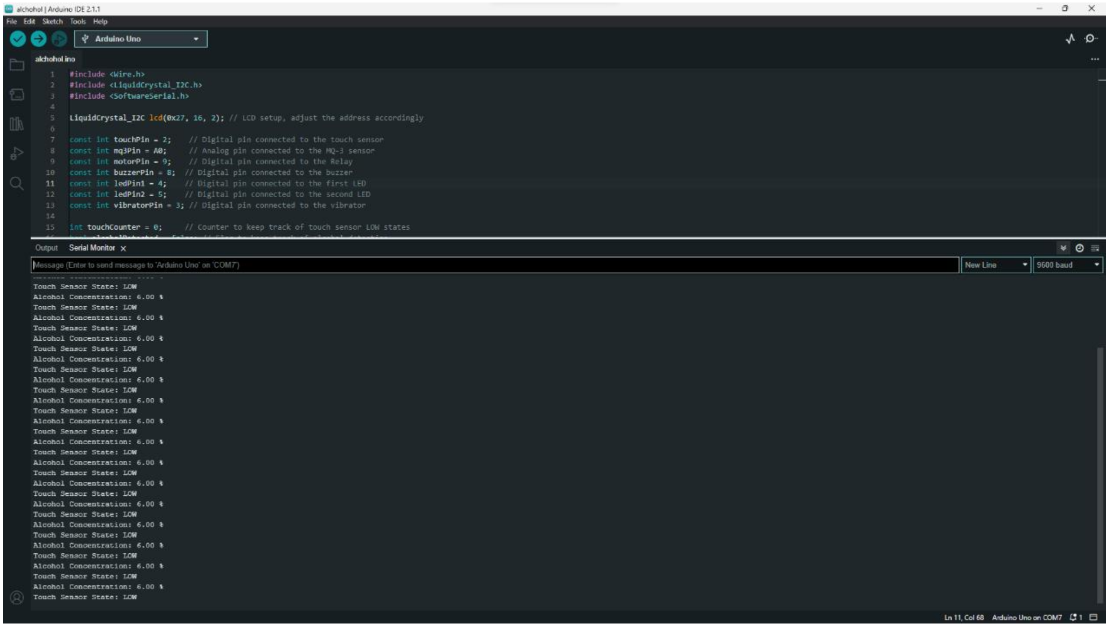
Fastest route now due to traffic conditions

11.3 Model Code Testing Snapshots


alchohol | Arduino IDE 2.1.1
File Edit Sketch Tools Help
Arduino Uno
alchohol.ino
Output
Serial Monitor
Message (Enter to send message to 'Arduino Uno' on 'COM7')
New Line
9600 baud
Ln 11, Col 68 Arduino Uno on COM7

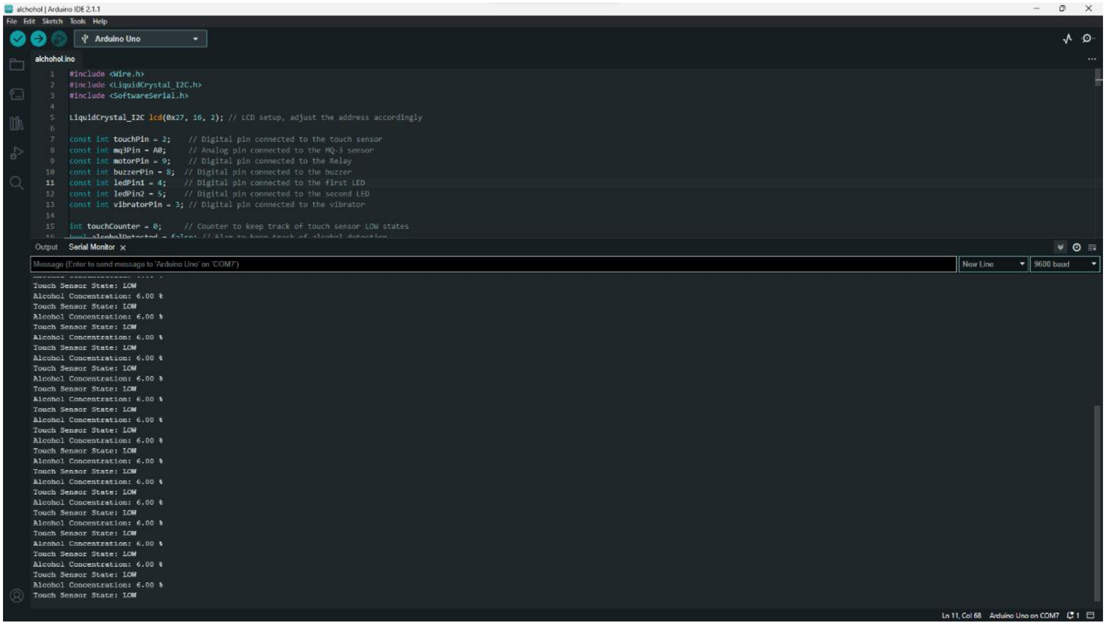
```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <SoftwareSerial.h>

LiquidCrystal_I2C lcd(0x27, 16, 2); // LCD setup, adjust the address accordingly

const int touchPin = 2;    // Digital pin connected to the touch sensor
const int mq3Pin = A0;     // Analog pin connected to the MQ-3 sensor
const int motorPin = 9;    // Digital pin connected to the Relay
const int buzzerPin = 8;  // Digital pin connected to the buzzer
const int ledPin1 = 4;    // Digital pin connected to the first LED
const int ledPin2 = 5;    // Digital pin connected to the second LED
const int vibratorPin = 3; // Digital pin connected to the vibrator

int touchCounter = 0;     // Counter to keep track of touch sensor LOW states
bool alcoholDetected = false; // Flag to keep track of alcohol detection

int flag=0;
// SIM800A configuration
SoftwareSerial sim800a(11,10 ); // RX, TX

void setup() {
  lcd.begin();
  lcd.backlight();

  pinMode(touchPin, INPUT);   // Set touch sensor pin as input
  pinMode(motorPin, OUTPUT);  // Set motor pin as output
  pinMode(buzzerPin, OUTPUT); // Set buzzer pin as output
  pinMode(ledPin1, OUTPUT);   // Set first LED pin as output
```

```
Touch Sensor State: LOW
Alcohol Concentration: 6.00 %
Touch Sensor State: LOW
Alcohol Concentration: 6.00 %
Touch Sensor State: LOW
Alcohol Concentration: 6.00 %
Touch Sensor State: LOW
Alcohol Concentration: 6.00 %
Touch Sensor State: LOW
Alcohol Concentration: 6.00 %
Touch Sensor State: LOW
Alcohol Concentration: 6.00 %
Touch Sensor State: LOW
Alcohol Concentration: 6.00 %
Touch Sensor State: LOW
Alcohol Concentration: 6.00 %
Touch Sensor State: LOW
```


alchohol | Arduino IDE 2.1.1
File Edit Sketch Tools Help
Arduino Uno
alchohol.ino
Output
Serial Monitor
Message (Enter to send message to 'Arduino Uno' on 'COM7')
New Line
9600 baud
Ln 11, Col 68 Arduino Uno on COM7

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

LiquidCrystal_I2C lcd(0x27, 16, 2); // LCD setup, adjust the address accordingly

const int touchPin = 2;    // Digital pin connected to the touch sensor
const int mq3Pin = A0;     // Analog pin connected to the MQ-3 sensor
const int motorPin = 9;    // Digital pin connected to the Relay
const int buzzerPin = 8;  // Digital pin connected to the buzzer
const int ledPin1 = 4;    // Digital pin connected to the first LED
const int ledPin2 = 5;    // Digital pin connected to the second LED
const int vibratorPin = 3; // Digital pin connected to the vibrator

int touchCounter = 0;     // Counter to keep track of touch sensor LOW states
```

```
Touch Sensor State: LOW
Alcohol Concentration: 6.00 %
Touch Sensor State: LOW
Alcohol Concentration: 6.00 %
Touch Sensor State: LOW
Alcohol Concentration: 6.00 %
Touch Sensor State: LOW
Alcohol Concentration: 6.00 %
Touch Sensor State: LOW
Alcohol Concentration: 6.00 %
Touch Sensor State: LOW
Alcohol Concentration: 6.00 %
Touch Sensor State: LOW
Alcohol Concentration: 6.00 %
Touch Sensor State: LOW
Alcohol Concentration: 6.00 %
Touch Sensor State: LOW
Alcohol Concentration: 6.00 %
Touch Sensor State: LOW
Alcohol Concentration: 6.00 %
Touch Sensor State: LOW
Alcohol Concentration: 6.00 %
Touch Sensor State: LOW
Alcohol Concentration: 6.00 %
Touch Sensor State: LOW
Alcohol Concentration: 6.00 %
Touch Sensor State: LOW
Alcohol Concentration: 6.00 %
Touch Sensor State: LOW
Alcohol Concentration: 6.00 %
Touch Sensor State: LOW
```


alchohol | Arduino IDE 2.1.1
File Edit Sketch Tools Help
Arduino Uno
alchohol.ino
Output
Serial Monitor
Message (Enter to send message to 'Arduino Uno' on 'COM7')
New Line
9600 baud
Ln 11, Col 68 Arduino Uno on COM7

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

LiquidCrystal_I2C lcd(0x27, 16, 2); // LCD setup, adjust the address accordingly

const int touchPin = 2;    // Digital pin connected to the touch sensor
const int mq3Pin = A0;     // Analog pin connected to the MQ-3 sensor
const int motorPin = 9;    // Digital pin connected to the Relay
const int buzzerPin = 8;  // Digital pin connected to the buzzer
const int ledPin1 = 4;    // Digital pin connected to the first LED
const int ledPin2 = 5;    // Digital pin connected to the second LED
const int vibratorPin = 3; // Digital pin connected to the vibrator

int touchCounter = 0;     // Counter to keep track of touch sensor LOW states
bool alcoholDetected = false; // Flag to keep track of alcohol detection
```

```
Touch Sensor State: LOW
Alcohol Concentration: 6.00 %
Touch Sensor State: LOW
Alcohol Concentration: 6.00 %
Touch Sensor State: LOW
Alcohol Concentration: 6.00 %
Touch Sensor State: LOW
Alcohol Concentration: 6.00 %
Touch Sensor State: LOW
Alcohol Concentration: 6.00 %
Touch Sensor State: LOW
Alcohol Concentration: 6.00 %
Touch Sensor State: LOW
Alcohol Concentration: 6.00 %
Touch Sensor State: LOW
Alcohol Concentration: 6.00 %
Touch Sensor State: LOW
Alcohol Concentration: 6.00 %
Touch Sensor State: LOW
Alcohol Concentration: 6.00 %
Touch Sensor State: LOW
Alcohol Concentration: 6.00 %
Touch Sensor State: LOW
Alcohol Concentration: 6.00 %
Touch Sensor State: LOW
Alcohol Concentration: 6.00 %
Touch Sensor State: LOW
Alcohol Concentration: 6.00 %
Touch Sensor State: LOW
Alcohol Concentration: 6.00 %
Touch Sensor State: LOW
```

12. CODING

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <SoftwareSerial.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);  // LCD setup, adjust the address accordingly
const int touchPin = 2;    // Digital pin connected to the touch sensor
const int mq3Pin = A0;     // Analog pin connected to the MQ-3 sensor
const int motorPin = 9;    // Digital pin connected to the Relay
const int buzzerPin = 8;  // Digital pin connected to the buzzer
const int ledPin1 = 4;    // Digital pin connected to the first LED
const int ledPin2 = 5;    // Digital pin connected to the second LED
const int vibratorPin = 3; // Digital pin connected to the vibrator
int touchCounter = 0;     // Counter to keep track of touch sensor LOW states
bool alcoholDetected = false; // Flag to keep track of alcohol detection
// Declare functions
void sendSMS(String message);
int flag=0;
String url;
// SIM800A configuration
SoftwareSerial sim800a(11,10 ); // RX, TX connect TX, RX
SoftwareSerial gpsSerial(12, 13);  // RX, TX connect TX, RX
void setup() {
lcd.begin();
lcd.backlight();
pinMode(touchPin, INPUT);   // Set touch sensor pin as input
pinMode(motorPin, OUTPUT);  // Set motor pin as output
pinMode(buzzerPin, OUTPUT); // Set buzzer pin as output
pinMode(ledPin1, OUTPUT);   // Set first LED pin as output
pinMode(ledPin2, OUTPUT);   // Set second LED pin as output
pinMode(vibratorPin, OUTPUT); // Set vibrator pin as output
Serial.begin(9600); // Initialize serial communication with PC
sim800a.begin(9600); // Initialize SIM800A communication
Serial.begin(9600); // Serial monitor baud rate
gpsSerial.begin(9600); // GPS module baud rate
lcd.print("System Started");
delay(3000);
}
void loop() {
float alcoholConcentration = analogRead(mq3Pin);
alcoholConcentration=map(alcoholConcentration, 0, 1023, 0, 100);
// Print the alcohol concentration
Serial.print("Alcohol Concentration: ");
Serial.print(alcoholConcentration);
Serial.println(" %");
// Display sensor values on the LCD
lcd.clear();
lcd.print("Alcohol :");
lcd.print(alcoholConcentration);
lcd.print("%");
```

```
// Read the state of the touch sensor
int touchState = digitalRead(touchPin);
// Print the state of the touch sensor
Serial.print("Touch Sensor State: ");
if (touchState == LOW) {
Serial.println("LOW");
} else {
Serial.println("HIGH");
}
// Check if any sensor is triggered or alcohol concentration is more than 20
if (touchState == LOW ) {
// Increment the touch counter
touchCounter++;
lcd.setCursor(0, 1);
lcd.print("Sleep Detected:");
lcd.print(touchCounter);
// Check if the touch sensor count is 3 or alcohol concentration is above the threshold, make the vibrator vibrate
if (touchCounter > 3 ) {
digitalWrite(vibratorPin, HIGH);
delay(100);
digitalWrite(vibratorPin, LOW);
delay(100);
}
```

```
// Check if the touch sensor count is more than 10 and alcohol detected, stop the motor and turn on the lights and buzzer
if (touchCounter > 5 ) {
digitalWrite(ledPin1, HIGH);
digitalWrite(ledPin2, HIGH);
digitalWrite(buzzerPin, HIGH);
// Send SMS indicating the emergency status
//sendSMS("Emergency: Motor Stopped and Lights On");
lcd.setCursor(0, 1);
lcd.print("The Car Stopped  ");
}
} else {
// Reset the touch counter and alcohol detection flag when the touch sensor is not LOW
touchCounter = 0;
digitalWrite(vibratorPin, LOW); // Turn off the vibrator
digitalWrite(buzzerPin, LOW); // Turn off the buzzer
digitalWrite(ledPin1, LOW); // Turn off the first LED
digitalWrite(ledPin2, LOW); // Turn off the second LED
}
if(alcoholConcentration>30 && flag<5)
{
sendSMS("Alocohol detected at location : https://goo.gl/maps/KLz2KEAd3v7pZUPq8");
flag++;
lcd.setCursor(0, 1);
lcd.print("Message sent");
}
if(gpsSerial.available()) {
String sentence = gpsSerial.readStringUntil('\n');
if (sentence.startsWith("$GPGGA")) {
// Split the sentence into individual fields
String fields[15];
int fieldCount = 0;
int startIndex = 0;
for (int i = 0; i < sentence.length(); i++) {
if (sentence.charAt(i) == ',') {
fields[fieldCount] = sentence.substring(startIndex, i);
fieldCount++;
startIndex = i + 1;
}
}
// Extract latitude and longitude
String latitude = fields[2];
String longitude = fields[4];
// Print latitude and longitude
float lon = atof(longitude.c_str()) / 100;
float latu = atof(latitude.c_str()) / 100;
// Concatenate the string values and store them in the 'url' variable
url = "https://www.google.com/maps/@";
url += String(latu, 7);
url += ",";
url += String(lon, 7);
url += ",15z";
// Print the GPS location to the serial monitor
Serial.println(url);
}
}
if(alcoholConcentration>30 || touchCounter > 5)
{
digitalWrite(motorPin, HIGH);
}
else
{
digitalWrite(motorPin, LOW); // Spin the motor
}
delay(1000);
}
void sendSMS(String message) {
sim800a.println("AT+CMGF=1"); // Set the GSM module to SMS mode
delay(100);
sim800a.println("AT+CMGS=\"+917411326909\""); // Replace +1234567890 with the recipient's phone number
delay(100);
sim800a.println(message); // The content of the message
delay(100);
sim800a.println((char)26); // ASCII code of CTRL+Z to send the message
delay(1000);
```

```
sendSMS1(message);
}
void sendSMS1(String message) {
sim800a.println("AT+CMGF=1"); // Set the GSM module to SMS mode
delay(100);
sim800a.println("AT+CMGS=\"+919353250845\""); // Replace +1234567890
with the recipient's phone number
delay(100);
sim800a.println(message); // The content of the message
delay(100);
sim800a.println((char)26); // ASCII code of CTRL+Z to send the message
delay(1000);
}
// Function to display GPS location
void displayGPSLocation() {
// Print the GPS location to the serial monitor
Serial.println(url);
}
```

13. CONCLUSION

In this project, we have developed a real-time model that can automatically lock the engine when a drunken driver tries to drive a car. Now a days, car accidents are mostly seen. By fitting this alcohol sensor into the car, we can save the lives of the driver and the remaining passengers. It is a very simple application. The life of the project is long. It has low or zero maintenance costs and, of course, low power consumption. By implementing this design, a safe car is possible, decreasing the accident rate due to drinking. By implementing this design, drunken drivers can be controlled, as can accidents due to drunken driving. The government must enforce laws to install such circuits in every car and must regulate all car companies to preinstall such mechanisms while manufacturing the car itself. If this is achieved, the number of deaths due to drunken drivers can be brought to a minimum.

The integration of touch sensors into the driver's seat provides a novel approach to detecting driver sleep and preventing accidents caused by drowsy driving. While this method alone may not capture the complete picture of driver drowsiness, it can significantly contribute to existing driver monitoring systems and enhance their overall effectiveness in ensuring road safety.

14. FUTURE ENHANCEMENT

- Currently, with the developed system, we can implement Heart Rate pulse variability to accurately identify the driving behavior of drivers and assist them.
- The MQ3-based system for detecting alcohol content in blood is also subject to failure and system crashes. In the future, we will have to implement it by using advanced sensors.
- If the driver wears the mask, it cannot be sensed. This work can be further expended by using the tough sensors based on the driver's figure on the steering.
- in the future. However, it is more likely that other technologies, such as eye tracking, yawning detection, and facial recognition, will be used for detecting driver drowsiness

.

15. BIBLIOGRAPHY

1. Paul Baskett, Yi Shang, Michael V. Patterson, Timothy Trull, Towards A System for Body-Area Sensing and Detection of Alcohol Craving and Mood Dvsregulation, Â© 2013 IEEE.
2. Accidental Deaths And suicides in India 2015 – NCRB
3. M.H. Mohamad, Mohd AminBin Hasanuddin, Mohd Hafizzie Bin Ramli "Vehicle Accident Prevention System Embedded with Alcohol Detector",
4. Karthik Kumar Vaigandla, Dr.N. Venu, " Survey on Massive MIMO: Technology, Challenges, Opportunities and Benefits," YMER, VOLUME 20: ISSUE 11 (Nov) - 2021, Page No:271-282.
5. https://www.rcciit.org/students_projects/projects/aeie/2019/GR9.pdf
6. https://www.irjet.net/archives/V9/i5/IRJET-V9I5332.pdf
7. https://www.researchgate.net/publication/341876799_Automatic_Engine_Locking_System_Through_Alcohol_Detection