CRASH ALERT PRO : "A Smartphone Application for Bicycle Crash Detection and Reporting"

A PROJECT REPORT

Submitted by,

S GIRISH	**20201CIT0026**
CHANDANA S	**20201CIT0036**
DHANRAJ N	**20201CIT0034**
SHASHANK A	**20201CIT0032**
SHREE VISHNU	**20201CIT0007**

Under the guidance of,

Ms. SRIDEVI.S

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

**COMPUTER SCIENCE AND ENGINEERING
(INTERNET OF THINGS)**

At

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

PRESIDENCY UNIVERSITY

BENGALURU

JANUARY 2024

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE ENGINEERING

CERTIFICATE

This is to certify that the Project report **"CRASH ALERT PRO - A Smartphone Application for Bicycle Crash Detection and Reporting"** being submitted by "S GIRISH, CHANDANA S, DHANRAJ N, SHASHANK A, SHREE VISHNU" bearing roll number(s) "20201CIT0026, 20201CIT0036, 20201CIT0034, 20201CIT0032, 20201CIT0007" in partial fulfilment of requirement for the award of degree of Bachelor of Technology in Computer Science and Engineering(Internet of Things) is a bonafide work carried out under my supervision.

Ms. SRIDEVI.S Assistant Professor School of CSE&IS Presidency University	**Dr. SP.ANANDARAJ** Professor & HoD School of CSE&IS Presidency University

Dr. C. KALAIARASAN Associate Dean School of CSE&IS Presidency University	**Dr. L. SHAKKEERA** Associate Dean School of CSE&IS Presidency University	**Dr. SAMEERUDDIN KHAN** Dean School of CSE&IS Presidency University

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE ENGINEERING

DECLARATION

We hereby declare that the work, which is being presented in the project report entitled **CRASH ALERT PRO – “A SMARTPHONE APPLICATION FOR BICYCLE CRASH DETECTION AND REPORTING**” in partial fulfilment for the award of Degree of **Bachelor of Technology** in **Computer Science and Engineering(Internet of Things)**, is a record of our own investigations carried under the guidance of **Ms**.**SRIDEVI.S, Assistant Professor, School of Computer Science Engineering, Presidency University, Bengaluru.**

We have not submitted the matter presented in this report anywhere for the award of any other Degree.

Name(s)	**Roll No(s)**	**Signature(s)**
S GIRISH	20201CIT0026	
CHANDANA S	20201CIT0036	
DHANRAJ N	20201CIT0034	
SHASHANK A	20201CIT0032	
SHREE VISHNU	20201CIT0007	

ABSTRACT

The integration of an Arduino-based sensor system into vehicles, coupled with IoT technology and GPS, presents an innovative and life-saving solution for enhancing the safety and security of individuals in motion. This compact system is designed to detect sudden changes in the vehicle's orientation or acceleration, signaling a potential fall or accident. Upon detection, the system utilizes IoT connectivity to transmit real-time alerts and pertinent data to a centralized platform. The incorporation of GPS ensures accurate recording and transmission of the incident's location, facilitating swift response and assistance during emergencies.

In addition to fall detection, the system addresses fire hazards by integrating sensors such as a fire sensor, a gas sensor (MQ2), and a DHT11 temperature sensor. These sensors continuously monitor smoke, gas concentrations, and temperature variations within the vehicle environment. In the event of an accident, the Arduino Uno microcontroller processes data from these sensors, triggering an immediate response. The system activates an alarm and sends an alert through the IoT network, including critical information like the vehicle's precise location and relevant data.

The seamless fusion of hardware and wireless connectivity not only ensures prompt notifications but also provides precise geographic context. This approach revolutionizes vehicle safety, enabling quicker response times in the event of accidents, especially in remote or uncharted areas. The system's ability to transmit information rapidly to a central server, accessible to authorized personnel and emergency responders, further enhances its efficacy in promoting safety on the road.

ACKNOWLEDGEMENT

First of all, we indebted to the **GOD ALMIGHTY** for giving me an opportunity to excel in our efforts to complete this project on time.

We express our sincere thanks to our respected dean **Dr. Md. Sameeruddin Khan**, Dean, School of Computer Science Engineering & Information Science, Presidency University for getting us permission to undergo the project.

We record our heartfelt gratitude to our beloved Associate Deans **Dr. Kalaiarasan C and Dr. Shakkeera L,** School of Computer Science Engineering & Information Science, Presidency University and **Dr. S.P. Anandaraj**. Head of the Department, School of Computer Science Engineering, Presidency University for rendering timely help for the successful completion of this project.

We are greatly indebted to our guide **Ms.Sridevi.S, Assistant Professor**, School of Computer Science Engineering & Information Science, Presidency University for her inspirational guidance, and valuable suggestions and for providing us a chance to express our technical capabilities in every respect for the completion of the project work.

We would like to convey our gratitude and heartfelt thanks to the University Project-II Coordinators **Dr. Sanjeev P Kaulgud, Dr. Mrutyunjaya MS** and also the department Project Coordinator **Ms. Manasa C M**.

We thank our family and friends for the strong support and inspiration they have provided us in bringing out this project.

S Girish

Chandana S

Dhanraj N

Shashank A

Shree Vishnu

LIST OF FIGURES

TABLE OF CONTENTS

CHAPTER-1
INTRODUCTION

Securing the safety of passengers and vehicles in vehicular transportation has become a top priority due to the potential threats posed by accidents, collisions, and hazards like fire or gas leaks. In response to this, the Vehicle Fall Detection System (VFDS) utilizes Arduino, IoT, and GPS integration, representing a crucial innovation in automotive safety. This system combines advanced sensor technologies such as gyro, ultrasonic, temperature, gas, and fire sensors with the versatile Arduino Uno microcontroller, creating a robust safety network for vehicles.

The gyro sensor is employed to identify sudden changes in the vehicle's orientation, signaling a potential accident or fall. Simultaneously, the ultrasonic sensor detects obstacles that might compromise vehicle integrity. To address fire-related risks, the system integrates a fire sensor, a gas sensor (MQ2), and a temperature sensor to monitor smoke, gas concentrations, and temperature variations within the vehicle environment. GPS technology further enhances the system by providing precise location tracking.

In the event of an accident or collision, the Arduino Uno microcontroller processes data from these sensors, triggering an immediate response. The IoT unit, represented by a NodeMCU, facilitates seamless communication between the vehicle and a centralized server or cloud platform. This real-time data transmission enables swift emergency responses and empowers remote monitoring of vehicle conditions. The paper comprehensively explores the design, implementation, and functionality of the VFDS, presenting it as a versatile safety solution applicable to various vehicle types, from individual automobiles to comprehensive fleet management systems.

Given the growing importance of vehicular safety, the VFDS with IoT and GPS integration using Arduino stands out as a significant advancement in safeguarding lives and property on the roadways.

CHAPTER-2
LITERATURE SURVEY

2.1 TITLE: Accident Prevention and Identification System For Vehicles Using Arduino
AUTHOR: R. LAVANYA, S. BALAMURUGAN
YEAR: 2022

DESCRIPTION: The global automobile industry is experiencing steady growth, leading to an anticipated rise in car ownership. Despite ongoing efforts to enhance vehicle safety through adaptive architectures, a notable gap exists in post-accident reporting centers. While road accidents remain inevitable, timely data and assistance can significantly impact saving lives. This proposed model seeks to address this issue by providing an alternative solution.

The research not only focuses on post-accident reporting but also introduces a mechanism to prevent accidents caused by reckless driving and excessive speeding. The safety system monitors vehicle speed, issuing an initial alarm to alert the driver about the driving speed. If the driver disregards the warning, the system then sends an overspeed message to relevant parties. With the increasing use of vehicles in proportion to the population, accidents are on the rise, exacerbated by traffic congestion.

This congestion leads to delays in ambulance arrival at accident scenes and transporting victims to hospitals, resulting in potential loss of life. It becomes crucial to expedite the transfer of accident victims to hospitals promptly. In the event of an accident, reporting to investigative agencies is essential. Streamlining this process by transferring reports to the research department can minimize research time, further optimizing the overall response to accidents.

DISADVANTAGE:
The potential limitation of the system's compatibility with diverse vehicle makes and models could pose a challenge. Customizing and adapting the system for different vehicles may necessitate additional time and effort, potentially hindering its accessibility for a broader user base.

2.2 TITLE: Accident Detection and Notification System using GPS and GSM navigation technology

AUTHOR: Dilkhush, Nikhil Pal, Kedar Nath Parida

YEAR:2020

DESCRIPTION: In today's context, the frequency of accidents is on the rise, often attributed to the increased use of vehicles like cars and bikes due to employment. This heightened vehicle usage, particularly when associated with overspeeding, contributes significantly to accidents. The growing demand for superior automobile performance and quality has paralleled a rising need for anti-accident systems.

Accidents are predominantly caused by factors such as high speed, driving under the influence, distraction, stress, and the use of electronic devices. This project specifically addresses accidents resulting from driver negligence. The proposed system introduces an accident detection and notification system designed to assist the driver. In situations where the driver is unable to maintain control, leading to an accident, the system automatically sends information to a pre-registered mobile number.

The project revolves around a GPS and GSM-based accident identification and information system, primarily focusing on accidents occurring during travel. The primary goal is to save lives by promptly reporting accidents to both hospitals and system owners.

DISADVANTAGE:

The system might face constraints in data transmission capacity, potentially limiting the volume of information that can be included in an accident notification. This restriction has the potential to impact the depth of the alerts and the level of detail communicated to emergency services.

2.3 TITLE: Accident Detection And Alerting System

AUTHOR: P.G. KATE, S.T. SHINDE

YEAR: 2019

DESCRIPTION: Transportation is a fundamental requirement in society, significantly contributing to the ease and comfort of human life. However, with the increase in transportation, accidents have also seen a rise, resulting in human casualties and bodily harm. To address this issue, we propose an Accident Detection and Messaging System utilizing GPS and GSM technology. The system incorporates a vibration sensor as an input, with the Arduino analyzing the corresponding responses. When an accident occurs, and the sensor readings surpass a predefined threshold, the system takes prompt action by sending an SMS to relevant authorities, facilitating immediate assistance to those involved in the accident.

This embedded approach has shown promising results in preventing and addressing accidents swiftly. The evolution of transportation systems has propelled human civilization, with automobiles playing a crucial role in our daily lives for commuting, communication, and logistics. Despite its significance, vehicles can also pose risks, leading to disasters and fatalities. Speed emerges as a critical risk factor, impacting crash severity and elevating the likelihood of accidents.

Despite ongoing efforts worldwide through various awareness programs, accidents persist. Efficient automatic accident detection, coupled with automatic notifications to emergency services, is crucial for saving lives. The timely transmission of crash information to emergency services can make a significant difference in mitigating the impact of accidents and, ultimately, preserving human life.

DISADVANTAGE:

The implementation and upkeep expenses for a GPS-based accident detection system may pose a considerable financial burden. This encompasses the costs associated with acquiring GPS hardware, securing data plans for real-time tracking, and covering ongoing maintenance costs.

2.4 TITLE: An Automatic Vehicle Accident Detection and Rescue System

AUTHOR: Tafadzwa Petros Chikaka ,Omowunmi Mary Longe

YEAR: 2021

DESCRIPTION: The loss of human lives due to road accidents is a profound issue, casting negative impacts on the socio-economic development of societies. In many developing countries, fatalities in road accidents are notably higher due to the absence of an efficient and rapid system for reporting accidents to emergency services, hindering immediate rescue efforts. The survival prospects of accident victims hinge significantly on the prompt arrival of emergency medical services and their ability to transport victims quickly to the nearest hospital for treatment.

This paper proposes a robust automatic vehicle accident detection and alert system that utilizes an accelerometer to identify the tilting and crashing of a vehicle. The system then transmits the Global Positioning System (GPS) location of the accident scene to designated security, medical, and family contacts. The innovative design demonstrates a turnaround response that surpasses conventional rescue systems lacking these features. Consequently, this technological advancement aims to save more lives by expediting the emergency response process.

DISADVANTAGE:

The system may entail intricacies that necessitate professional assistance, potentially contributing to an escalation in the overall implementation costs.

2.5 TITLE: Accident Detection And Tracking System Using GSM, GPS And Arduino
AUTHOR: CH. Gowri, B. Raj Kumar
YEAR: 2020

DESCRIPTION: The advent of technology and improved infrastructure has undoubtedly simplified our lives. However, it has also brought about an increase in traffic hazards, leading to frequent road accidents and substantial losses in terms of life and property due to inadequate emergency facilities. This project aims to address these challenges by introducing an automated system designed to detect accidents and promptly alert the nearest emergency services.

The system utilizes various components, including an accelerometer, ultrasonic sensor, vibration sensor, GPS, and GSM module, to ensure effective accident detection and notification. The accelerometer is responsible for identifying sudden changes in the car's axes, while the vibration sensor detects heavy vibrations within the vehicle. The ultrasonic sensor plays a role in reducing the speed of the vehicle when it approaches another vehicle, and the GSM module sends an alert message to mobile devices, providing the accident's location.

To pinpoint the accident location accurately, the system employs GPS technology, and the information is conveyed in the form of a Google Map link derived from latitude and longitude coordinates. Once the accident location is confirmed, necessary actions can be taken, facilitating timely response from rescue services and potentially saving precious human lives.

DISADVANTAGE:
Implementing the system may incur substantial expenses, encompassing the acquisition of hardware, data plans, and installation costs. This financial burden could pose a drawback for individuals or organizations operating within constrained budgets.

2.6 TITLE: Smart Accident Detection and Alert System

AUTHOR: Arnav Chaudhari; Harsh Agrawal

YEAR: 2021

DESCRIPTION: Street car accidents represent a significant public health concern, leading to substantial loss of lives, property, and time. The prompt provision of clinical assistance can play a pivotal role in saving numerous lives. This paper introduces an intelligent accident detection and alert system designed to inform the user's emergency contacts when an accident occurs, sending a message along with the detected location.

In the event of a vehicle accident, the system's sensor promptly identifies the incident and dispatches an SMS to the designated emergency contacts. A reset button is incorporated, allowing users to halt the alarm from being sent if everyone inside the vehicle is confirmed to be safe. The escalating number of vehicles on the roads, surpassing even economic and population growth rates, has become a pressing issue.

India, in particular, has consistently ranked highest in road fatalities since the release of the first Global Status Report on Roads by WHO in 2009. With a vast six-million-kilometer road network, India stands out as one of the most traffic-intensive countries. However, road traffic injuries and fatalities continue to surge at an alarming rate. Government reports indicate that half of India's road traffic fatalities could have been prevented with basic trauma treatment within the crucial first hour after a crash.

DISADVANTAGE:

The system's detection range is constrained, presenting a potential drawback, especially in situations where accidents occur beyond the sensor's coverage. This limitation has the potential to lead to delayed or overlooked accident detection, particularly in expansive areas or along highways.

CHAPTER-3

RESEARCH GAPS OF EXISTING METHODS

3.1 EXISTING SYSTEM

In the existing system is Street car accidents pose a significant public health concern due to their considerable impact on lives, property, and time. The timely provision of clinical assistance is crucial for saving lives. This paper introduces an intelligent accident detection and warning system designed to notify the user's emergency contacts with the detected location when an accident occurs.

In the event of a vehicle accident, the system's sensor promptly identifies the incident, triggering the sending of an SMS to the designated emergency contacts. A reset button is incorporated to prevent the alert from being transmitted to crisis contacts in situations where everyone within the vehicle is confirmed to be safe.

The proposed system employs a specific methodology: a vibration sensor measures the vibrations resulting from the vehicle's crash, and an ultrasonic sensor identifies the presence of individuals inside the vehicle. The entire system is activated only if the sensor detects the presence of an individual. A microcontroller reads the vibration frequencies, configuring the GPS module to obtain the accident's coordinates. These coordinates are then transmitted to the emergency contacts via SMS as a Google Maps URL, facilitated by the GSM module.

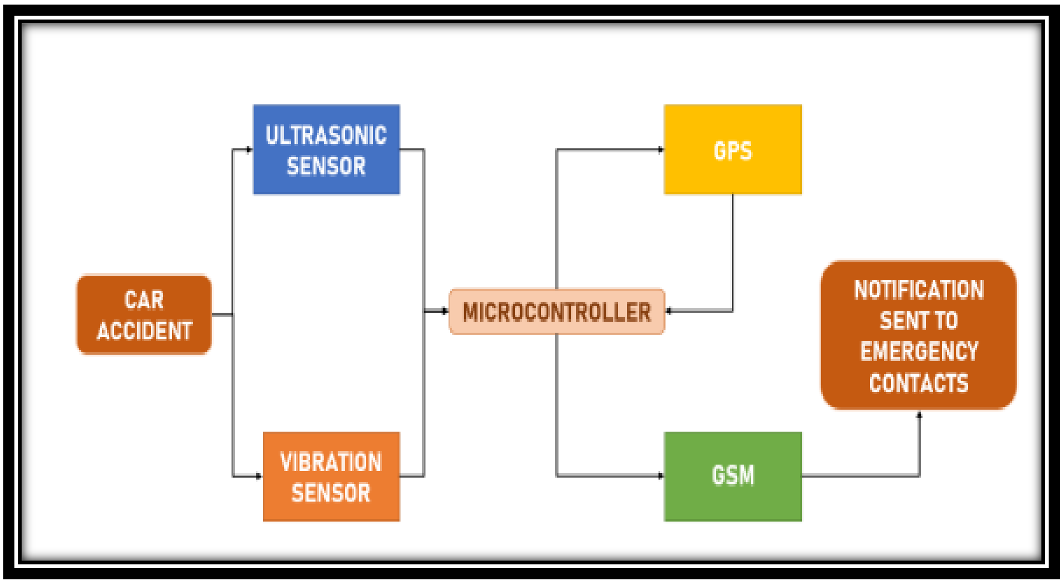


Figure 3.1 Existing System Block Diagram

3.2 DISADVANTAGE

- The system's functionality may be compromised in regions with inadequate or no network coverage, resulting in potential delays in reporting accidents or rendering the system inoperable.
- While these systems excel at promptly alerting emergency services, the efficiency of the response hinges on the accessibility and proximity of emergency services. In remote or rural areas, response times may still be prolonged.

CHAPTER-4
PROPOSED MOTHODOLOGY

4.1 PROPOSED SYSTEM

The proposed Vehicle Fall Detection System (VFDS) leverages advanced technology to elevate road safety and streamline response efforts during vehicular accidents. This innovative system integrates an Arduino-based hardware platform with cutting-edge components such as a Gyro Sensor (ADXL335), ultrasonic sensor, gas sensor, temperature sensor, fire sensor, GPS technology, and the Internet of Things (IoT), providing a holistic safety solution. The Gyro Sensor and ultrasonic sensor collaboratively monitor the vehicle's movements, identifying sudden obstacles and potential accidents in real-time. Strategically positioned within the vehicle, these sensors work cohesively to detect changes in orientation and obstacles, enhancing accident detection capabilities.

The system's design incorporates a gyro sensor for orientation changes and an ultrasonic sensor for obstacle detection, adding an extra layer of accident prevention. To address fire-related risks, the system includes a fire sensor, a gas sensor (MQ2), and a DHT11 temperature sensor, collectively monitoring smoke, gas concentrations, and temperature variations in the vehicle environment. The GPS module ensures precise location data is continually available. The IoT network serves as the system's backbone, enabling seamless transmission of crucial accident information to a central server accessible to relevant authorities and emergency responders.

In the event of a fall or obstacles, the system autonomously triggers alerts containing the vehicle's location and pertinent data, facilitating immediate responses. By integrating these technologies, the vehicle fall detection system aims to substantially improve vehicle safety, mitigate accident severity, and expedite response times. The provision of timely and accurate information supports effective emergency responses and ongoing safety management, ultimately contributing to a safer and more efficient transportation ecosystem.

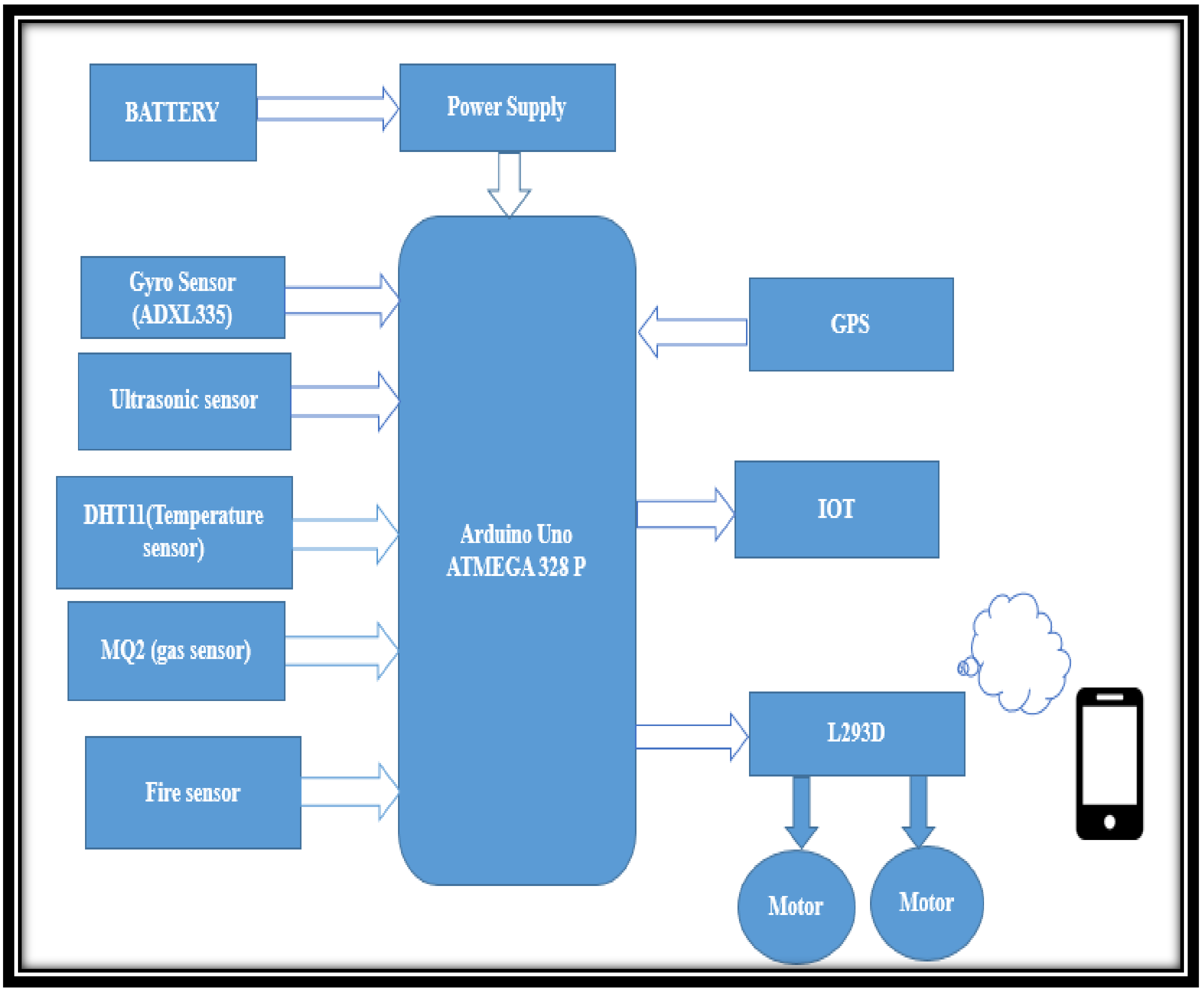


Figure 4.1 Proposed System Block Diagram

4.2 METHODOLOGY

SENSORS UNIT

- **Accelerometer Sensor:** establish connections with the microcontroller's analog pins, ensuring power (VCC), ground (GND), and data (OUT) are appropriately linked.
- **Ultrasonic Sensor:** Connect the ultrasonic sensor to the Adriano Uno, employ two pins (trigger and echo). Utilize a compatible library to measure ultrasonic wave travel time and convert it to distance, indicating object proximity. Connect VCC to Arduino's 5V, GND to Arduino's GND, Trig to a digital pin (e.g., D2), and Echo to another digital pin (e.g., D3)
- **GPS Sensor:** establish connections with the microcontroller through communication interfaces like UART or I2C. Power (VCC), ground (GND), and communication pins (TX, RX) are essential for proper operation.
- **DHT11 temperature sensor:** Connect the DHT11 to Arduino Uno, supplying power (VCC), ground (GND), and selecting a digital pin for data (OUT). Employ the DHT library to retrieve temperature and humidity values from the sensor.
- **Gas sensor:** To integrate the Gas Sensor (MQ2), link it to an analog pin on Arduino Uno. Read analog values to determine gas concentrations and implement logic for triggering alarms in the presence of elevated gas levels.
- **Fire sensor:** connect it to a digital or analog pin on Arduino Uno. Monitor the sensor output for flame detection and implement logic to activate alarms in case of a fire.

MICROCONTROLLER UNIT (ATMEGA328P)

- The central processing unit of the system, responsible for analyzing sensor data and executing decisions.
- Establish power (VCC) and ground (GND) connections for the microcontroller, ensuring compatibility with the power supply.
- For the accelerometer and ultrasonic sensors, establish connections using two pins each: trigger and echo. Assign analog or digital pins based on the specifications of each sensor.
- Link the GPS sensor to the microcontroller's communication pins (TX, RX) to facilitate data exchange.

IOT UNIT

- IoT Module (e.g., ESP8266 or equivalent): Enables seamless communication with the IoT platform.
- Establish a connection between the IoT module and the microcontroller using an appropriate communication protocol (e.g., UART, SPI, etc.).
- Ensure the provision of power (VCC) and ground (GND) to the IoT module.
- Implement any essential level shifting or voltage regulation as needed.

VEHICLE MODEL UNIT

- L293D Driver Board: Used for driving DC motors.
- DC Motor: The actuators that might perform specific actions (e.g., deploy airbags, trigger alarms, etc.).
- Connect the L293D driver board to the microcontroller. Wire the input pins to the microcontroller's output pins, and connect power (VCC) and ground (GND).
- Connect the DC motor to the output pins of the L293D driver board. Ensure proper power supply and polarity.

CHAPTER-5

OBJECTIVES

- **Continuous Safety Monitoring:** Construct a system with the capability to consistently monitor vehicle movement and orientation, enabling the detection of abrupt falls or rollovers for enhanced vehicle occupant safety.
- **Precision in Location Tracking:** Incorporate GPS technology to offer accurate and real-time location data, facilitating prompt responses and assistance in the aftermath of an accident.
- **Established IoT Connectivity:** Create a smooth IoT network to automatically transmit crucial accident information to a centralized server, ensuring swift dissemination of alerts to pertinent authorities and emergency responders.
- **Automated Alert System:** Design an alert system that activates notifications containing information such as vehicle location, incident time, and other pertinent data, providing timely insights to authorities and emergency services when accidents occur.

CHAPTER-6

REQUIREMENT ANALYSIS

6.1 Software Requirement

- Programming Language - Embedded C
- Compiler - ARDUINO IDE 1.8.19
- Simulation - PROTEUS

SOFTWARE DESCRIPTION

6.1.1 ARUDINO IDE SOFTWARE

Figure6.1.1 Arudino Ide Software

The Arduino Integrated Development Environment (IDE) serves as a user-friendly software platform designed to streamline the programming and project development process for Arduino microcontroller boards. Utilized for writing and uploading code onto the physical board, the Arduino IDE's simplicity has played a pivotal role in the widespread adoption of Arduino technology. Compatibility with the Arduino IDE has become a key criterion for new microcontroller board introductions. Over time, the IDE has evolved, incorporating features such as managing third-party libraries and boards, while retaining its programming simplicity.

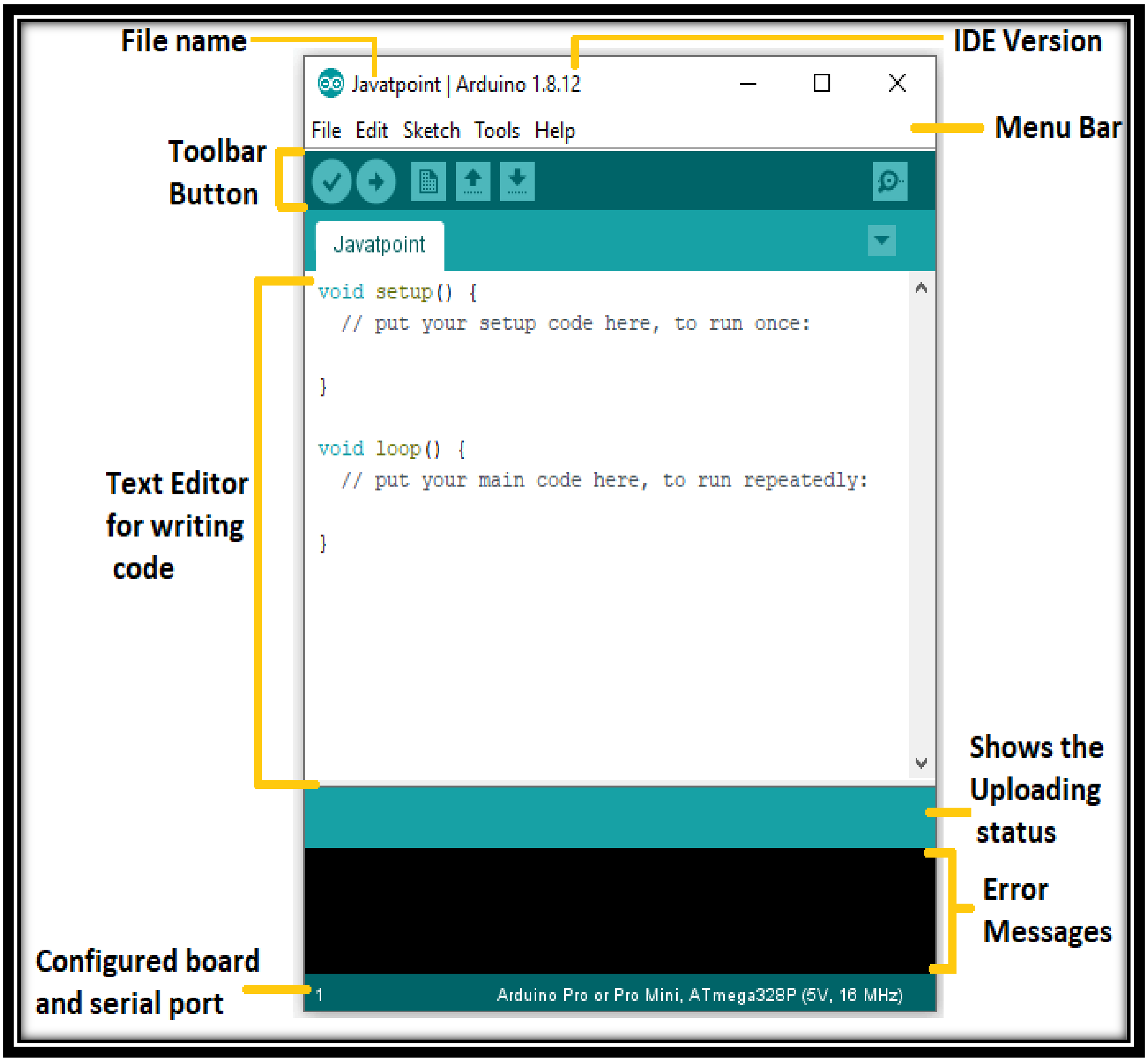


Figure 6.1.2 Arduino IDE Software Screenshot

The main window of the Arduino IDE provides an uncomplicated interface, as depicted in Figure 6.1.2, featuring the straightforward "Blink" example. The IDE's core component is a text editor where users write, edit, and save Arduino code, typically in a simplified variant of the C++ programming language. This simplicity caters to beginners, and the text editor includes features like syntax highlighting, auto-indentation, and code suggestions.

After code creation, the IDE facilitates one-click compilation, checking for errors and generating machine code understandable by the Arduino microcontroller. Debugging features, error messages, and a console aid in identifying and rectifying issues. Upon successful compilation, the IDE manages the USB-based upload process, ensuring proper code transfer and execution.

The Arduino IDE streamlines the use of external libraries, offering a Library Manager for easy searching, installation, and management. This capability extends project functionalities without requiring complex code development. The IDE supports a vast library collection for various components, such as sensors and communication protocols.

Arduinos IDE is compatible with a diverse range of Arduino-compatible boards, allowing easy switching between boards for different projects. The IDE includes a Serial Monitor for debugging and real-time communication between the computer and the Arduino board, making it valuable for displaying sensor readings and debugging messages.

The supportive Arduino community enhances the IDE's utility, with extensive documentation, forums, tutorials, and examples available online. As a crucial component of the Arduino ecosystem, the IDE provides an efficient and user-friendly environment for writing, compiling, and uploading code to Arduino microcontroller boards. Its features, library management system, and community support contribute to its popularity among both beginners and experienced developers in the field of electronics and embedded systems.

6.2 Hardware Requirement

HARDWARE DETAILS

1. Battery
2. Power Supply Unit
3. Atmega328p Microcontroller
4. Gyro Sensor (Adxl335)
5. Ultrasonic sensor
6. GPS
7. Nodemcu
8. DHT11(Temperature sensor)
9. MQ2 (gas sensor)
10. Fire sensor
11. Driver Board
12. Dc Motor

HARDWARE DESCRIPTION

6.2.1 BATTERY

Figure 6.2.1 BATTERY

A 12-volt battery, commonly known as a 12V battery, is a widely used type of lead-acid battery with applications across various industries. Renowned for its ability to deliver a consistent 12-volt voltage, these batteries power a diverse range of electrical and electronic devices. The standard voltage of 12V batteries is particularly prevalent in the automotive sector, where they serve as essential components for starting engines, powering lights, radios, and other vehicle accessories. Outside of the automotive industry, these batteries find utility in uninterruptible power supplies (UPS), recreational vehicles, boats, and as backup power sources for smaller electronic devices.

Typically based on lead-acid chemistry, 12V batteries consist of multiple cells connected in series within a durable plastic casing. Each cell comprises positive and negative lead plates immersed in a sulfuric acid electrolyte. Charging the battery initiates a chemical reaction, converting lead dioxide on the positive plate to lead sulfate and lead on the negative plate to lead sulfate. During discharge, this process reverses, generating electrical energy. The battery is equipped with terminals for connecting to external electrical circuits.

Charging a 12V battery involves applying a voltage higher than its own, typically ranging from 13.6 to 14.4 volts. This higher voltage drives the chemical reactions in reverse, converting lead sulfate back into lead dioxide and lead. Discharging occurs when electrical loads are connected, releasing the stored chemical energy as electrical power and gradually decreasing the battery's voltage.

The capacity of a 12V battery, measured in ampere-hours (Ah) or milliampere-hours (mAh), represents the amount of electrical charge it can deliver over time. The capacity varies widely, with smaller batteries for motorcycles and larger ones for vehicles and backup power systems. Applications range from providing starting power to engines and supplying auxiliary power in campers and boats to serving as backup power during electrical outages.

To ensure the longevity and reliability of 12V batteries, proper maintenance is crucial. This includes checking electrolyte levels, ensuring clean and secure terminal connections, and charging when necessary. The lifespan varies based on factors like usage, temperature, and battery quality. Typically, an automotive 12V battery may last around 3 to 5 years, while deep-cycle 12V batteries used in off-grid and renewable energy systems can last longer with proper maintenance.

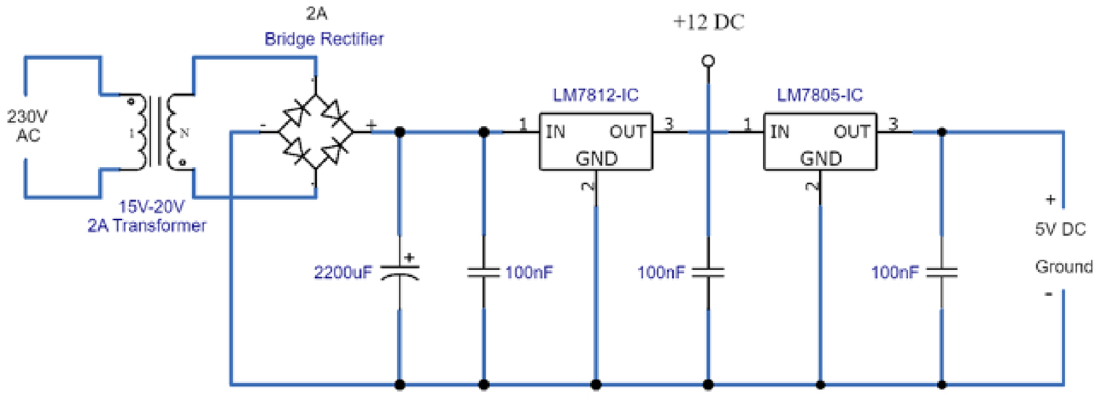
6.2.2 POWER SUPPLY UNIT

Figure 6.2.2 Power Supply Unit

A Power Supply Unit (PSU) is a critical element in electronic devices, essential for providing the required electrical energy to facilitate their operation. The fundamental working process revolves around the conversion of electrical energy from a source, such as a wall outlet or battery, into a stable and regulated form usable by the device. Typically, the PSU incorporates key components like a transformer, rectifier, and voltage regulator.

The process initiates with the reception of incoming AC (alternating current) power from the source, which may exhibit varying voltage levels and frequencies. The transformer serves the initial role of stepping down the voltage to a level suitable for the device's requirements. Subsequently, the rectifier transforms the AC into DC (direct current) by smoothing out the voltage, ensuring a consistent flow of electricity in a single direction. To maintain a steady voltage level, a voltage regulator is employed, adjusting the output voltage as necessary.

To enhance device protection, the PSU includes mechanisms such as over-current and over-voltage protection, aiming to safeguard the connected device from potential electrical surges or fluctuations. The regulated DC output from the PSU is then distributed to the various components within the device, providing them with the requisite power for reliable functionality. In summary, the Power Supply Unit's working process guarantees a stable and standardized power source for electronic devices, enabling efficient operation without the risk of damage due to electrical irregularities.



6.2.2.1 power Supply Unit Circuit Diagram

The primary function of a power supply is to convert electrical current from a source into the appropriate voltage, current, and frequency to power the load. Power supplies are often referred to as electric power converters, and they can exist as standalone devices or be integrated into the appliances they power. In the context of discussing the circuit for a 12V 5V power supply, this system is a dual-power circuit utilizing two different fixed voltage regulator ICs.

The central IC in this circuit is LM7812, responsible for converting the voltage from the transformer and providing a regulated 12V output on pin 1 of the LM7805 IC. Subsequently, the voltage is further converted to 5V DC by the LM7805 IC, resulting in the generation of two distinct voltages simultaneously. It is advisable to use appropriate heatsinks with the ICs for efficient heat dissipation. The transformer can be either a 230V or 110V to 15V to 20V 2A transformer.

The 12V power supply in this context is commonly employed to supply power to components and devices that require higher voltage levels. It finds applications in motors, fans, certain types of lighting, and, in some cases, specific computer components such as hard drives and graphics cards.

Conversely, the 5V power supply delivers a lower voltage level and is widely used to power various digital electronics, microcontrollers, and logic circuits. Additionally, it serves as the standard voltage level for USB ports, which are extensively utilized for charging and connecting various devices.

6.2.3 ARDUINO UNO

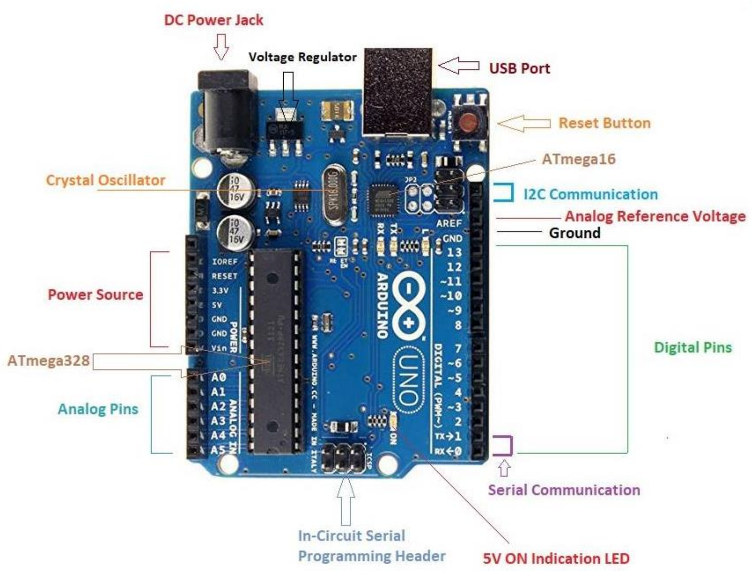


Figure 6.2.3 Arduino uno

The Arduino UNO microcontroller is centered around the ATmega328P, providing simplicity in comparison to other boards like the Arduino Mega. It incorporates various components such as shields, additional circuitry, and digital/analog input/output (I/O) pins.

Key features of the Arduino UNO include six analog pin inputs, fourteen digital pins, a USB port, a power jack, and an ICSP (In-Circuit Serial Programming) header. Programming for the Arduino UNO is done through the Integrated Development Environment (IDE), compatible with both offline and online environments.

Microcontrollers, in contrast to microprocessors, have analog and digital pins that can be easily configured as input or output pins. Analog pins are connected internally to an analog-digital converter (ADC) capable of translating voltage ranges (e.g., 0-3.3V) into numerical values (e.g., 0-1023). Digital pins typically support LOW (0V) and HIGH (3.3V) levels.

PIN CONFIGURATIONS

The ATmega328P microcontroller, an 8-bit chip in the ATmel family, is at the core of the Arduino Uno. It includes various features such as memory (SRAM, EEPROM, and Flash), analog-to-digital converter, SPI serial ports, I/O lines, registers, timer, external and internal interrupts, and oscillator.

- ICSP for In-Circuit Serial Programming
- Power LED indicator
- Digital I/O pins (D0 to D13)
- TX and RX LEDs indicating data flow
- AREF for Analog Reference
- Reset button
- USB for board-computer connection
- 16MHz Crystal Oscillator
- Voltage Regulator for converting input voltage to 5V
- GND pins for grounding
- Vin for input voltage
- Analog Pins (A0 to A5) for reading analog sensors and functioning as GPIO pins.

It's important to note that while the information presented here is specific to the Arduino UNO, the ATmega328P microcontroller is also used in other Arduino boards like Nano and ProMini

6.2.4 NODEMCU

Figure6.2.4 Nodemcu

NodeMCU is an open-source LUA-based firmware designed for the ESP8266 wifi chip, commonly integrated into the NodeMCU Development board for ESP8266 functionality exploration.

While often referred to as a WiFi module, the ESP8266 is, in fact, a microcontroller developed by Espressif Systems. The ESP8266 is a versatile microcontroller with WiFi capabilities, widely utilized in IoT applications. Various ESP8266 modules are available, ranging from ESP8266-01 to ESP8266-12, differing mainly in breakout board configurations. For instance, ESP8266-01 features two GPIO pins, while ESP-12 boasts 16 GPIO pins.

Key components of the NodeMCU Development board include:

- **ESP8266 Microcontroller:** The NodeMCU board is based on the ESP8266 microcontroller, a 32-bit, low-power, high-performance device with integrated Wi-Fi functionality. It serves multiple purposes, including connecting IoT devices to the internet and handling wireless communication tasks.

- **USB-to-Serial Interface:** Typically equipped with a USB-to-serial interface, the NodeMCU board facilitates programming and USB-based communication with the ESP8266. This interface is essential for firmware uploads and debugging processes.

- **Power Supply:** The NodeMCU can be powered using a 5V micro-USB cable and incorporates a built-in voltage regulator, allowing compatibility with a broad range of input voltages, typically between 7-12V.

- **GPIO Pins:** Featuring multiple General-Purpose Input/Output (GPIO) pins, the NodeMCU allows for various digital input and output tasks. The ESP8266 offers a total of 17 GPIO pins, enabling interfacing with sensors, displays, and other devices.

- **Analog Pins**: NodeMCU includes one analog input pin (A0) designed for analog voltage measurements.

- **LEDs:** An onboard LED is often integrated into the NodeMCU board, serving as a useful tool for debugging or indicating status.

- **Reset Button:** The presence of a reset button allows users to reset the ESP8266 without the need to remove power.

This comprehensive set of features makes NodeMCU a versatile and user-friendly platform for developing applications involving WiFi communication and IoT functionality.

6.2.5 GLOBAL POSITIONING SYSTEM (GPS)

Figure 6.2.5 GPS

The Global Positioning System (GPS) is a navigation system that utilizes satellites to offer precise location and timing data globally. Initially developed and currently maintained by the United States Department of Defense, GPS has become an indispensable component of modern life, finding applications in navigation, mapping, surveying, and tracking.

This system comprises satellites orbiting Earth and ground-based control stations, ensuring widespread coverage and accurate positioning information. Satellites in the GPS constellation orbit at a medium altitude of approximately 20,000 kilometers, with around 30 operational satellites and spares in orbit at any given time, strategically positioned in multiple orbital planes for comprehensive global coverage. Users across the globe have access to several satellites simultaneously, ensuring continuous coverage.

The accuracy of GPS is influenced by factors such as the number and arrangement of satellites in view and the quality of the receiver. Standard civilian GPS receivers commonly provide location accuracy within a few meters. In contrast, specialized high-precision GPS systems, utilized in fields like surveying and scientific applications, can achieve remarkably precise accuracy, down to a few centimeters.

APPLICATIONS:

GPS technology is used in many ways. In cars, it helps with directions and tracking fleets. It's crucial for safe air and sea travel. People use it for outdoor activities and sports to track routes and performance. In farming, GPS optimizes planting and harvesting. The military depends on it for tasks like guiding missiles. It's also vital in search and rescue to find people accurately.

6.2.6 ACCELEROMETER SENSOR

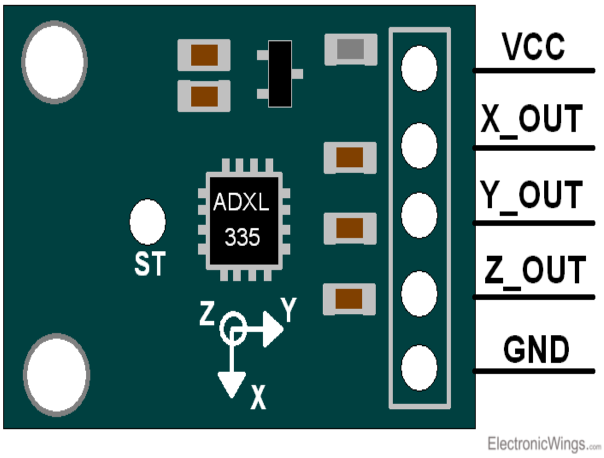


Figure 6.2.6 Accelerometer Sensor

The ADXL335 accelerometer operates on the fundamental principle of measuring the acceleration force acting on a mass suspended by a spring. This mechanism produces analog voltage outputs that signify the acceleration along three perpendicular axes.

Acceleration, being a vector quantity, characterizes how velocity changes concerning time. Velocity, in turn, is a combination of speed and direction. Acceleration can be elucidated through two facets: a change in speed and a change in direction, both of which can occur simultaneously. In the context of the ADXL335 accelerometer, it functions as a device measuring the acceleration of an object. The device tracks acceleration through analog inputs in three dimensions—X, Y, and Z. Known for its power efficiency and minimal noise, it seamlessly interfaces with various controllers like Arduino or microcontrollers when employed for acceleration measurements.

This accelerometer finds widespread application in diverse fields, including construction machinery such as drilling, pile driving, and demolition, as well as human activity monitoring in activities like running, walking, dancing, and skipping. Easily accessible in the market or online, the ADXL335 accelerometer stands out for its simplicity and effectiveness in measuring acceleration.

Pin Description

- VCC: Power supply pin i.e. connect 5V here.
- X_OUT: X axis analog output.
- Y_OUT: Y axis analog output.
- Z_OUT: Z axis analog output.
- GND: Ground pin i.e. connect ground here.

ADXL335 accelerometer provides analog voltage at the output X, Y, Z pins; which is proportional to the acceleration in respective directions i.e. X, Y, Z.

6.2.7 ULTRASONIC SENSOR:

Figure 6.2.7 Ultrasonic Sensor

The Ultrasonic Sensor plays a critical role in a voice-guided smart shoes system designed for visually impaired individuals, focusing on obstacle detection. Its primary function involves emitting high-frequency sound waves and measuring the time taken for these waves to bounce back from obstacles. This process aids in calculating the distance to objects in the user's path, allowing the system to identify and respond to potential hazards effectively.

Typically equipped with four pins, the Ultrasonic Sensor has distinct functionalities:

- **VCC (Voltage Supply):** Connects to the positive terminal of the power source (usually +5V) to provide the necessary operating power.
- **GND (Ground):** Connects to the ground (0V) to complete the electrical circuit and ensure proper sensor functioning.

- **Trigger (Trig) Pin:** Initiates the emission of ultrasonic pulses when triggered, sending out a pulse of sound waves.
- **Echo Pin:** Receives reflected sound waves and, by measuring the time delay between Trigger and Echo pins, allows the Arduino or microcontroller to calculate obstacle distance.

With its specific pin configuration, the Ultrasonic Sensor contributes to accurate obstacle detection in the smart shoes system. The real-time data generated is vital for delivering voice-guided instructions, enhancing the ability of visually impaired individuals to navigate their surroundings safely and independently.

APPLICATIONS

- Navigation and Way finding
- Education and Learning
- Employment and Workplace Integration
- Social Interaction
- Access to Public Services
- Recreation and Leisure Activities
- Daily Living Tasks
- Emergency Situations

6.2.8 DHT11 (TEMPERATURE SENSOR)

Figure 6.2.8 Temperature Sensor

The DHT11 is a popular digital sensor known for its simplicity and affordability in measuring temperature and humidity. It's widely used in various projects where keeping an eye on environmental conditions is crucial. This sensor combines a humidity and temperature sensor in one compact package, making it easy to work with. It uses special components to sense humidity and temperature, and its built-in converter helps provide digital readings. This user-friendly feature makes it a common choice for projects using microcontrollers like Arduino.

Key Features:

- Temperature Range: 0°C to 50°C (32°F to 122°F)
- Humidity Range: 20% to 80%
- Accuracy: ±2°C for temperature, ±5% for humidity
- Resolution: 1°C for temperature, 1% for humidity
- Operating Voltage: 3.3V to 5.5V

DHT11 Pin Configuration:

- **VCC (Power):** This pin serves to supply power to the sensor, typically linked to the 5V output of the microcontroller or an independent power source.
- **Data:** The data pin facilitates bidirectional communication between the sensor and the microcontroller. It transmits the digital signal conveying temperature and humidity readings.
- **Ground (GND):** This pin is coupled to the ground (0V) of the microcontroller or the power supply.

6.2.9 MQ2 SENSOR

Figure 6.2.9 MQ2 Sensor

The MQ-2 gas sensor is a frequently employed semiconductor-based sensor designed to detect various gases, primarily for monitoring air quality and ensuring safety. This sensor exhibits sensitivity to a range of gases, encompassing LPG, propane, hydrogen, methane, carbon monoxide, as well as smoke and various volatile organic compounds. Due to its versatility, the MQ-2 sensor offers a cost-effective and efficient solution for detecting the presence of these gases. It finds extensive use in applications such as gas leak detection, fire safety, and air quality monitoring.

Pin Configuration:

- **VCC (Pin 1):** This pin serves as the supply voltage input, typically linked to a power source (often +5V or +3.3V) to energize the sensor.
- **GND (Pin 2):** The GND pin connects to the ground (0V) reference point.
- **AOUT (Pin 3):** Responsible for analog output, this pin delivers a voltage signal proportional to the detected gas concentration. A microcontroller's analog-to-digital converter (ADC) can interpret this voltage.
- **DOUT (Pin 4):** The DOUT pin functions as the digital output. It produces a digital signal (usually HIGH or LOW) based on a predefined gas concentration threshold. When surpassed, this signal can activate an alarm. This pin is commonly used for basic gas detection without the need for precise concentration values.

It's crucial to acknowledge that the MQ-2 sensor necessitates a warm-up duration for accurate readings. During this period, the sensor must operate in clean air to ensure stable measurements. Furthermore, the sensor's sensitivity may fluctuate with variations in temperature and humidity, underscoring the significance of proper calibration and environmental considerations for precise gas detection.

6.2.10 FIRE SENSOR

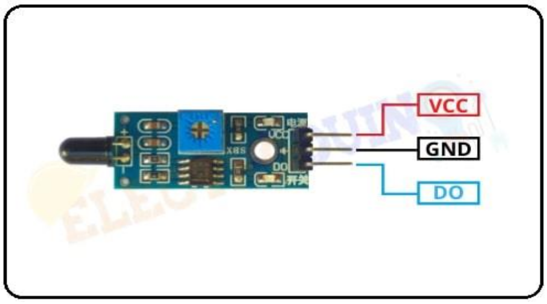


Figure 6.2.10 fire sensor

A fire sensor plays a vital role in identifying the presence of fire or excessive heat in its immediate surroundings, serving as a crucial element in fire alarm systems and various safety applications. Among the diverse types of fire sensors, one prevalent variant is the Flame Sensor or Flame Detector, designed to recognize the infrared (IR) radiation emitted by flames.

These sensors are typically engineered to react to specific environmental changes linked to fires, such as elevated heat, smoke, or the presence of flames. The Flame Sensor, in particular, discerns the infrared light emitted during combustion. Upon detecting a fire, the sensor generates a signal that can initiate an alarm or activate other safety protocols.

Pin Configuration:

- **VCC (Power):** This pin functions to provide power to the sensor and is commonly linked to a power source with the suitable voltage requirement.
- **GND (Ground):** The ground pin establishes a connection to the ground (0V) of either the microcontroller or the power supply.
- **Signal/OUT:** When the sensor identifies flames or elevated temperatures, this pin generates a signal. The nature of this signal, whether digital or analog, is contingent on the specific type of sensor in use.

6.2.11 DRIVE BOARD L293D

Figure 6.2.11 Drive Board L293d

The L293D stands out as a widely utilized motor driver IC (integrated circuit), playing a key role in steering the direction and speed of DC motors and stepper motors. Its uncomplicated operational mechanism positions it as a fundamental element in numerous electronics and robotics ventures. With four half-H-bridges, the L293D empowers independent control over two motors. These half-H-bridges facilitate bidirectional motor movement and speed regulation. Specifically crafted for seamless integration with microcontrollers and digital control systems, the L293D responds to control signals on its input pins. By appropriately manipulating these signals, users can activate or deactivate motor outputs, alter the motor direction, and manage its speed. Essentially, the L293D translates digital inputs into essential control signals, offering a versatile and straightforward solution for motor control across diverse projects.

6.2.12 DC MOTOR

Figure 6.2.12 DC motor

A Direct Current (DC) motor, an electrical device transforming electrical energy into mechanical motion, operates on the Lorentz force principle. This principle involves a current-carrying conductor within a magnetic field experiencing a force that induces movement. DC motors find widespread application in industrial machinery, robotics, household appliances, and automotive systems. Diverse types of DC motors exist, such as brushed DC motors featuring commutators and brushes, brushless DC motors utilizing electronic commutation, and coreless DC motors designed for enhanced performance. Renowned for their versatility, efficiency, and ability to offer precise control over speed and torque, DC motors are well-suited for a broad spectrum of applications.

CHAPTER-7

SYSTEM DESIGN & IMPLEMENTATION

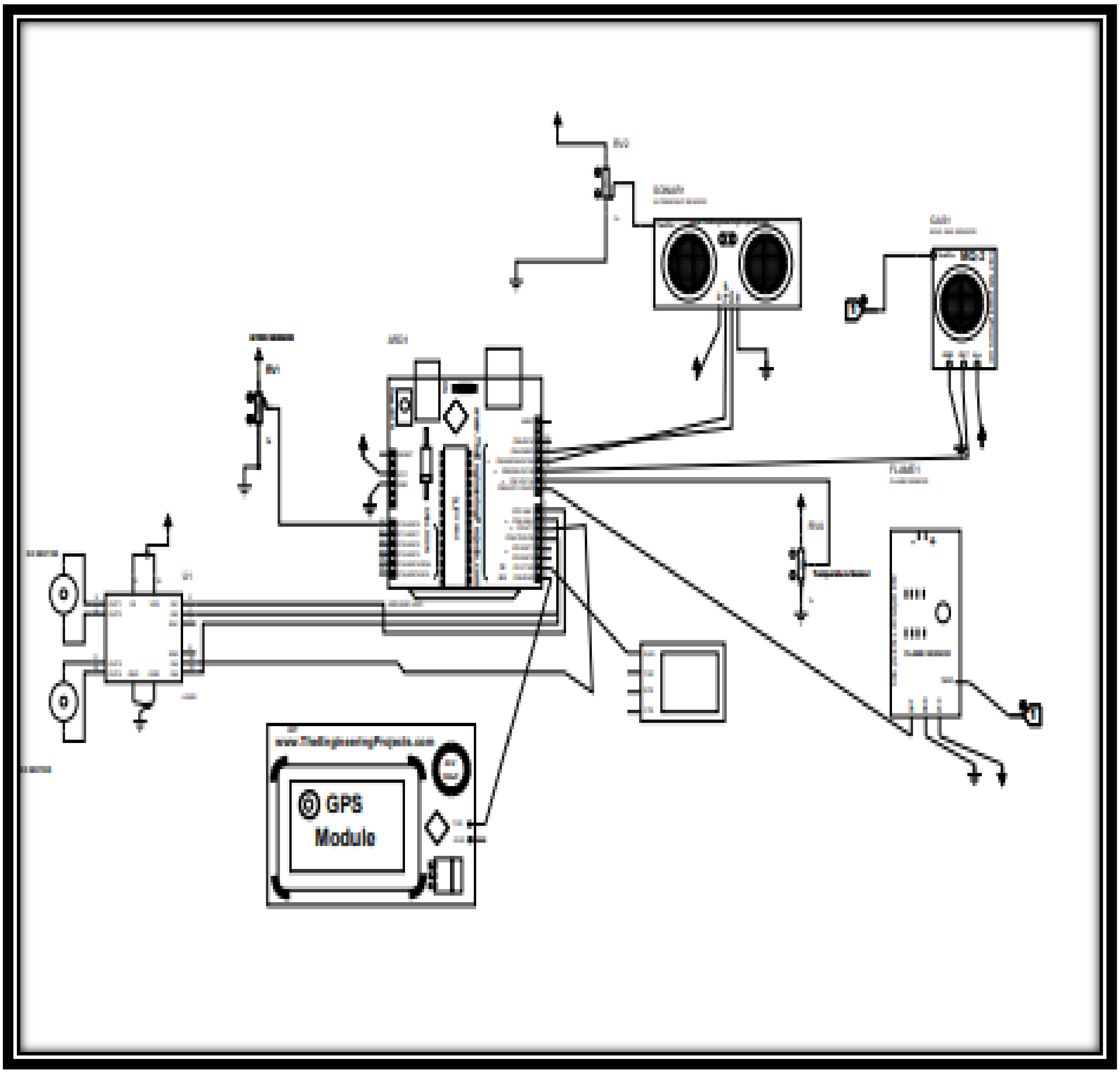


Figure 7.1 Circuit Diagram

7.2 FLOW DIAGRAM

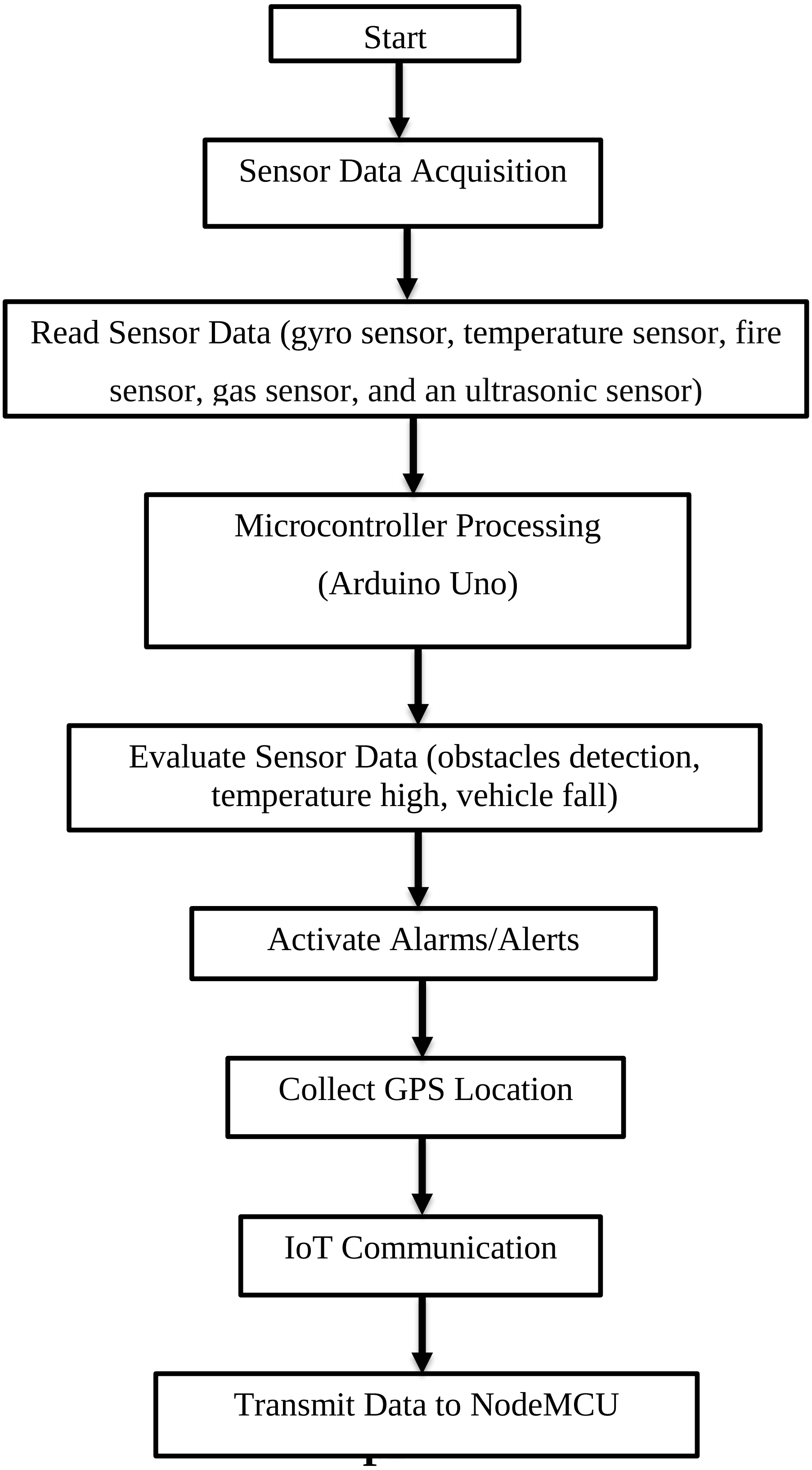


Figure 7.2 Flowchart

CHAPTER-8

TIMELINE FOR EXECUTION OF PROJECT

(GANTT CHART)

TASKS		Week							
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1	Doing the proposal	■	■						
2	Order the hardware online			■	■				
3	Design the Circuit Programming					■	■		
4	Testing the Elderly Care system							■	
5	Presentation								■

Figure 8.1 Timeline gantt chart

CHAPTER-9

RESULTS

SIMULATIONS:

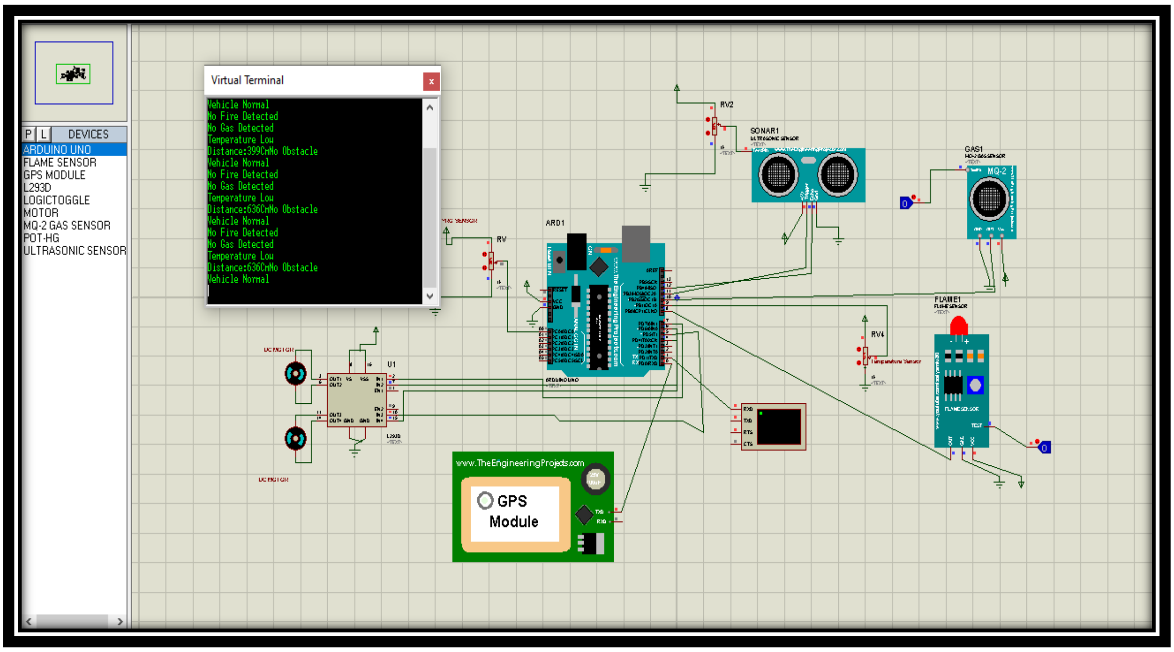


Figure 9.1 Vehicle is normal

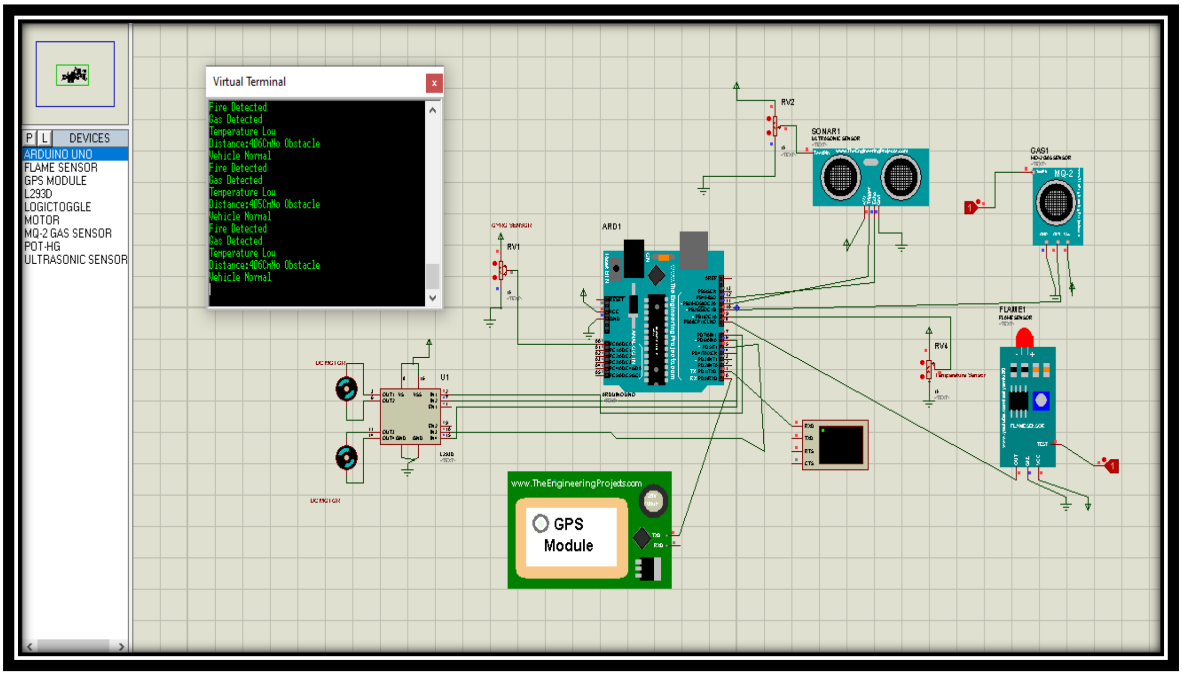


Figure 9.2 Fire &Gas Detected:

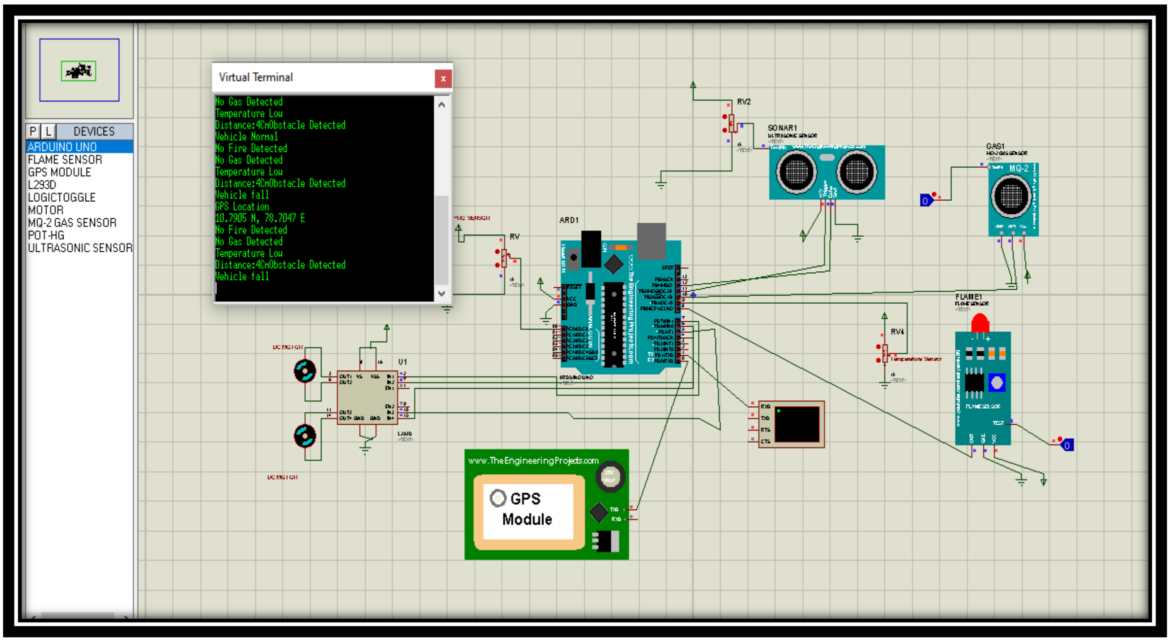


Figure 9.3 Vehicle fall and obstacle Detected,

HARDWARE IMPLEMENTED

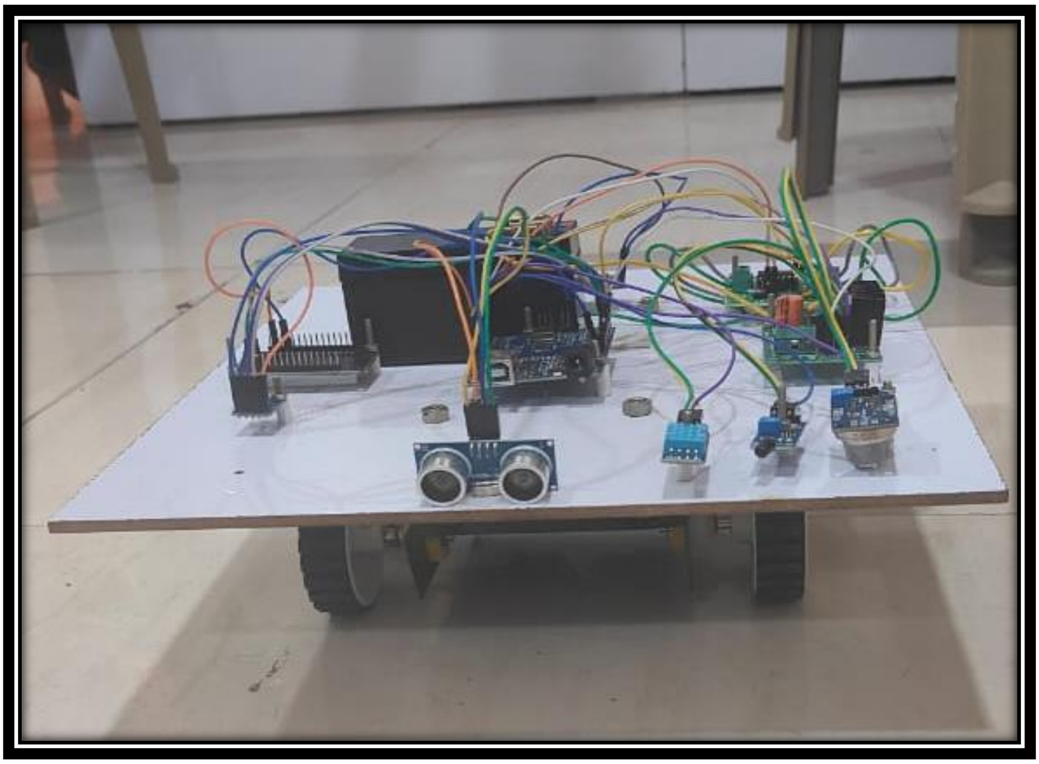


Figure 9.4 Hardware Model

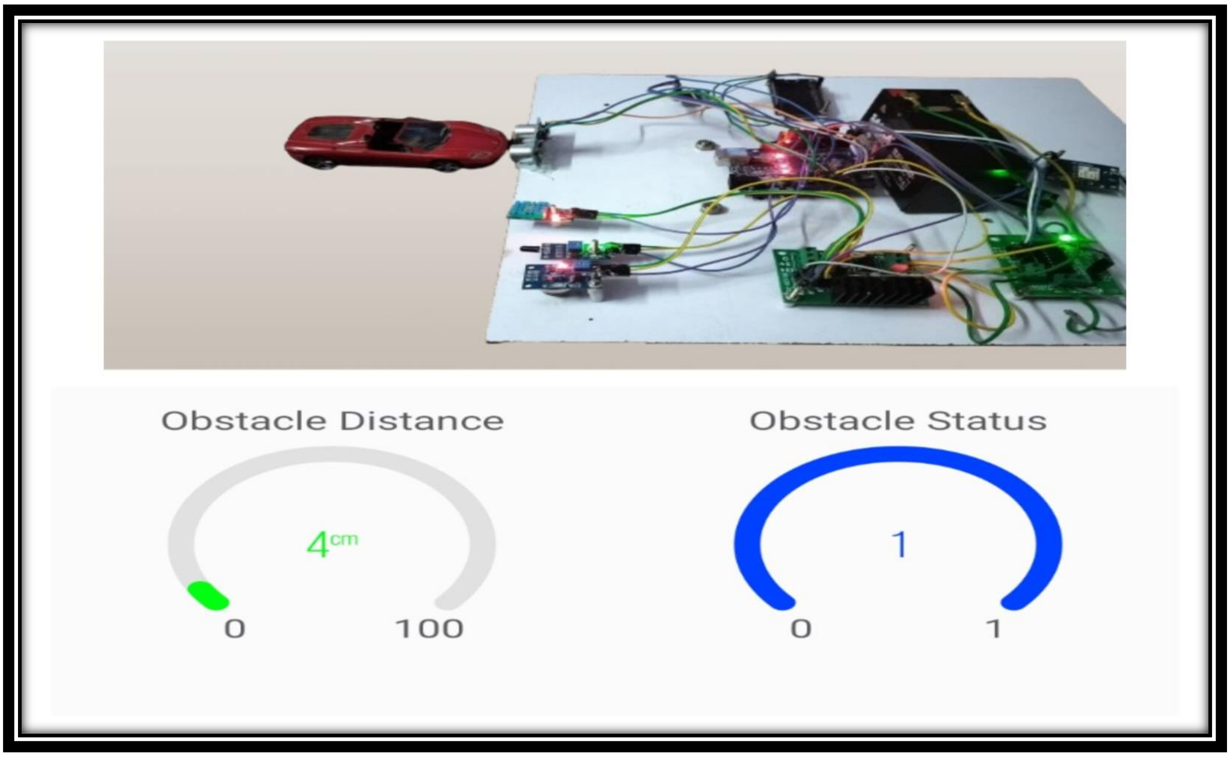


Figure 9.5 Obstacle detected

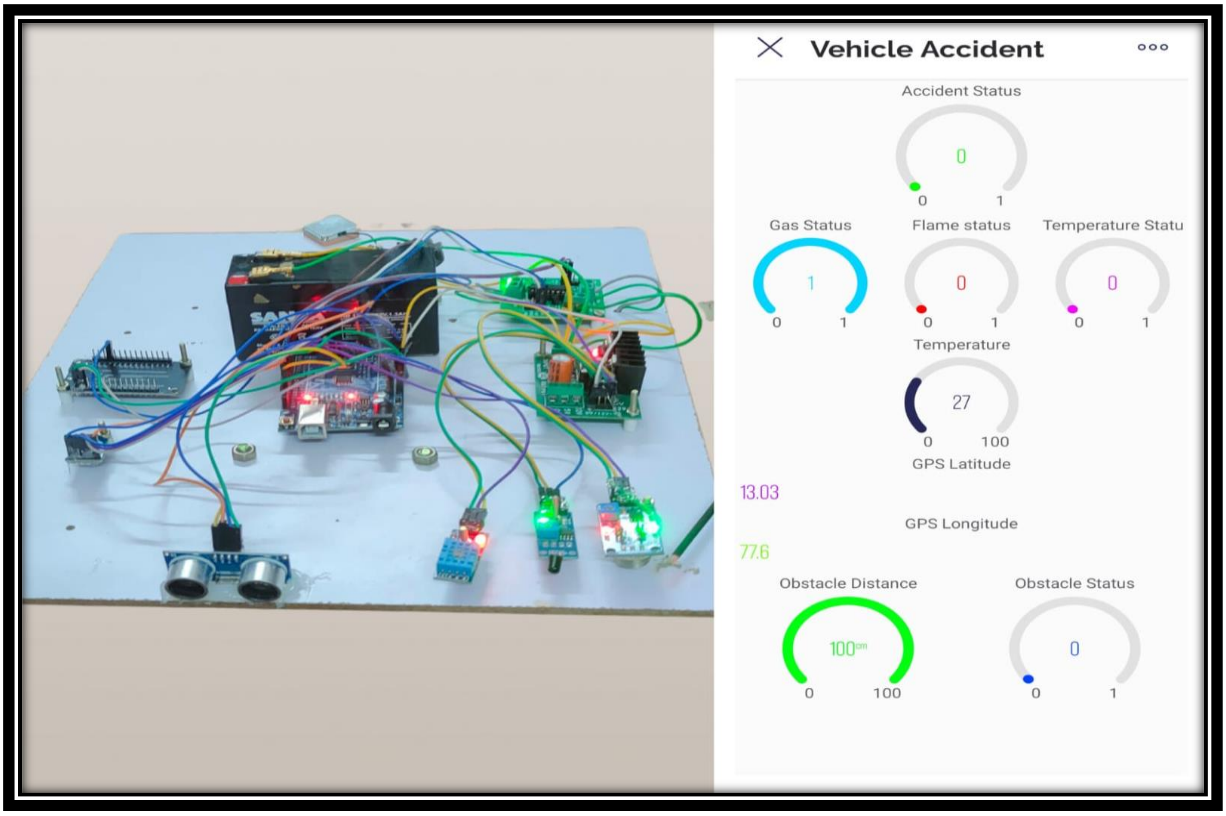


Figure 9.6 Gas detected

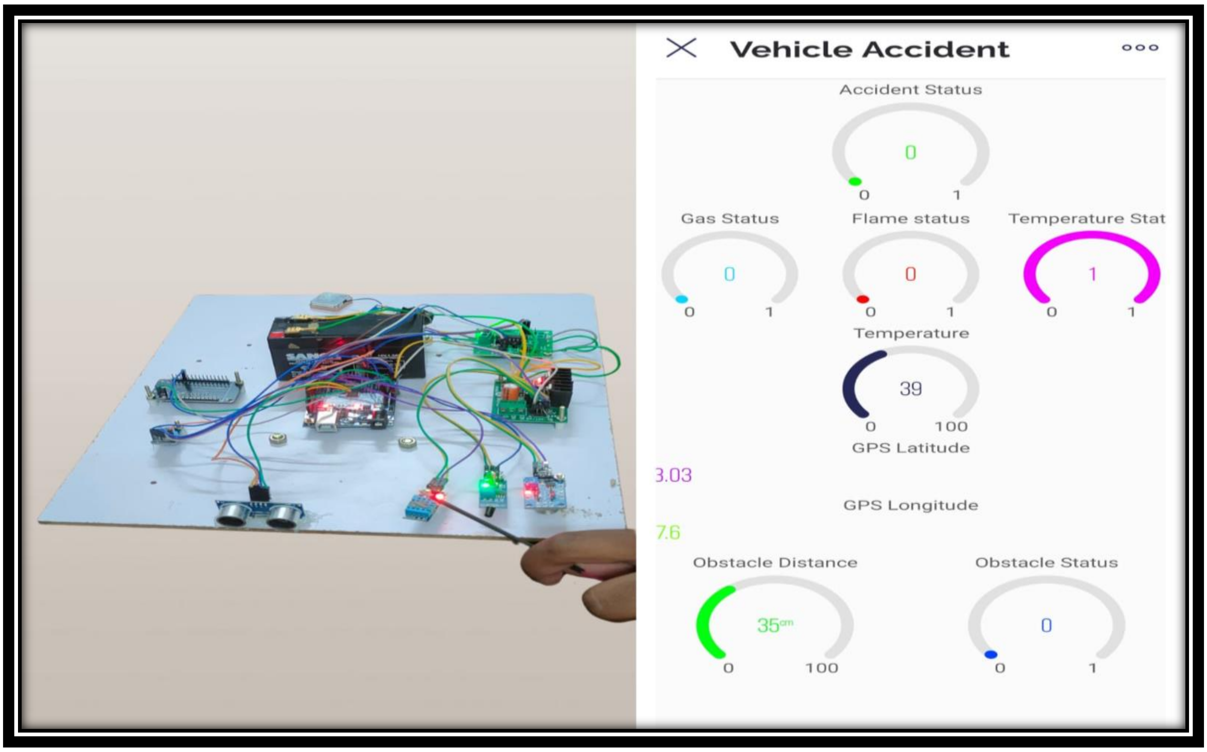


Figure 9.7 Temperature is greater than 35

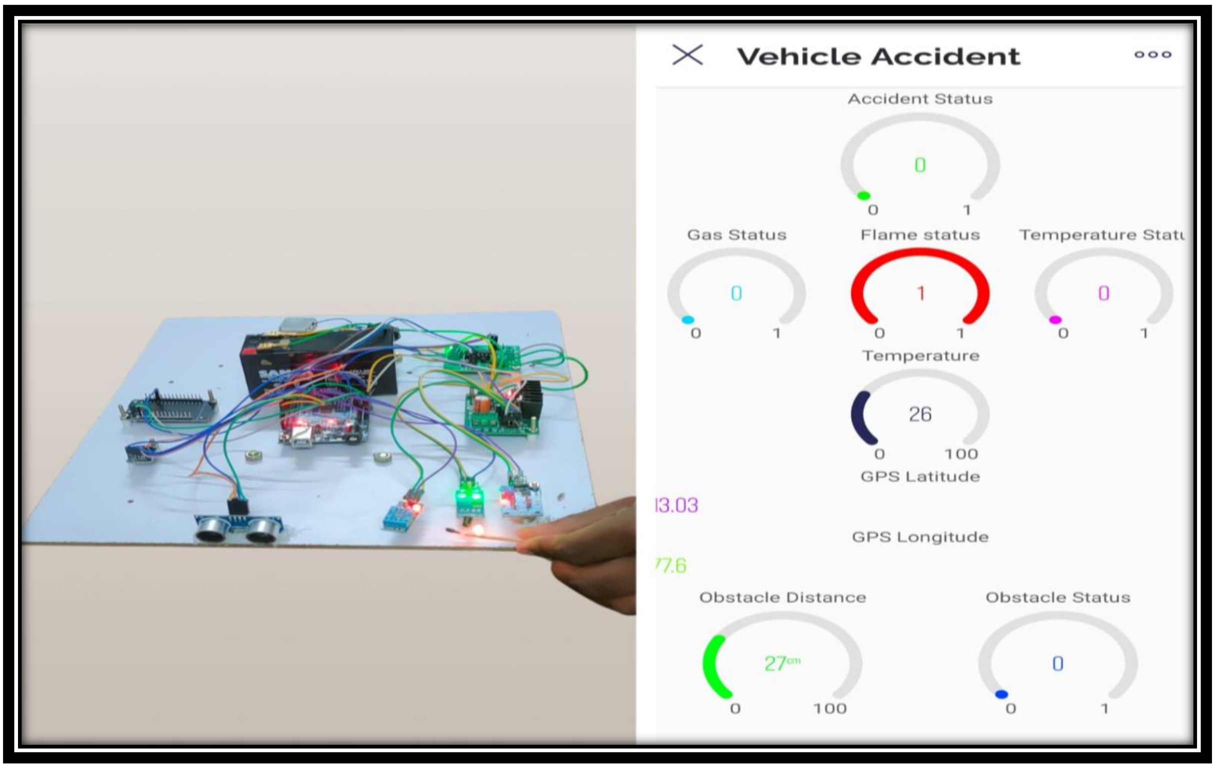


Figure 9.8 Flame detected

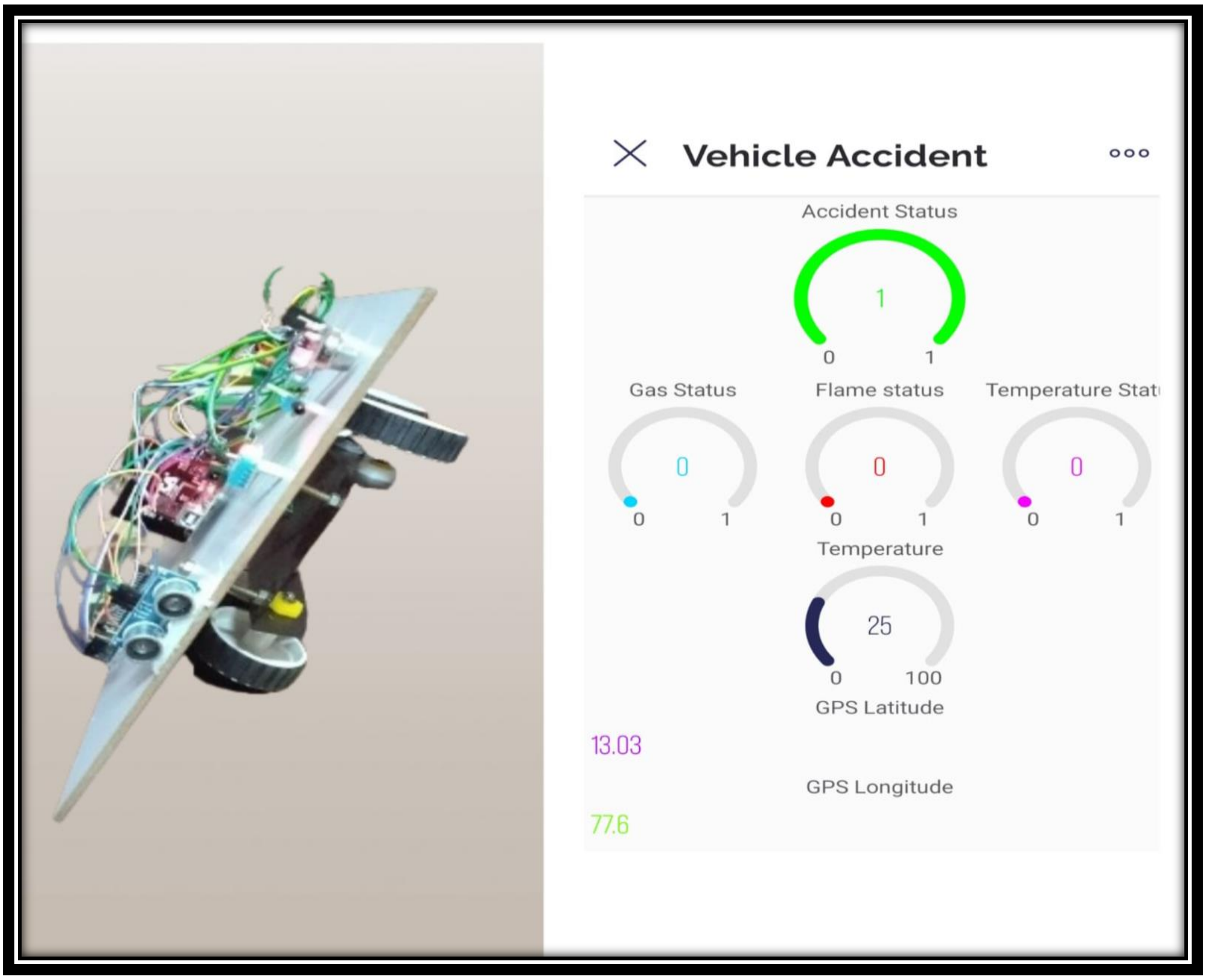


Figure 9.9 Accident status high and GPS location detected

CHAPTER-10
CONCLUSION

In summary, the integration of IoT and GPS technology in the Vehicle Fall Detection System using Arduino signifies a notable advancement in automotive safety and emergency response. This innovative system effectively addresses the crucial challenge of promptly detecting vehicular accidents or rollovers, enabling automated and swift responses during emergencies. Through the utilization of Arduino microcontrollers and diverse sensors, it establishes a robust platform for real-time data collection and analysis. In the event of an accident or obstacle detection, the system efficiently transmits vital information, such as the accident's location, to a central server or emergency services, potentially reducing response times and saving lives. Its adaptability makes it suitable for various vehicles, from personal cars to commercial fleets, promising improvements in road safety, vehicle tracking, and emergency response services. However, it is imperative to address certain challenges, including refining accident detection accuracy to minimize false alarms, addressing privacy concerns related to data collection, and ensuring the system's reliability across diverse driving conditions and environments. While the Vehicle Fall Detection System with IoT and GPS Integration using Arduino shows promise in revolutionizing vehicle safety and emergency response, ongoing research and development efforts are crucial to enhancing its accuracy and reliability, solidifying its role as a valuable asset in automotive safety and accident prevention.

REFERENCES

[1] R. Lavanya, S. Balamurugan, K. A Bhoobalan, C. Jeevanandham, V. Kishore Kumar "Accident Prevention and Identification System for Vehicles using Arduino" Internal Journal of Latest Engineering and Management Research (IJLEMR), vol. 09 Issue 5th May 2022, no. 454.

[2] Dilkhush, Nikhil Pal, Kedar Nath Parida, Nishant Singh, MK Jayanthi Kannan "Accident Detection and Notification System using GPS and GSM navigation technology" Internal Journal of Latest Engineering and Management Research (IJLEMR), vol. 07 Issue 6th June 2020, no. 84.

[3] P.G. Kate, S.T. Shinde, S.V. Jagtap, S.T. Kamble "Accident Detection and Alerting System" Internal Journal of Latest Engineering and Management Research (IJLEMR), vol. 06 Issue 3rd Mar 2019, no. 359.

[4] Tafadzwa Petros Chikaka ,Omowunmi Mary Longe "An Automatic Vehicle Accident Detection and Rescue System" 2021 IEEE 6th International Forum on Research and Technology for Society and Industry (RTSI) , Issue 6th September 2021.

[5] CH. Gowri, B. Raj Kumar "Accident Detection And Tracking System Using GSM, GPS And Arduino" Journal of Emerging Technologies and Innovative Research, Volume 7, Issue 4th April 2020.

[6] Arnav Chaudhari, Harsh Agrawal, Smiti Poddar, Kiran Talele, Manisha Bansode "Smart Accident Detection And Alert System" 2021 IEEE 6th International Forum on Research and Technology for Society and Industry (RTSI) , Issue 27th August 2021.

[7] N. D. Ram, V. V. Krishna, D. K. Kalika and J. Kokilavani, "Smart Traffic Light Controller," Internal Journal of Latest Engineering and Management Research (IJLEMR), vol. 04, no. 03, pp. 83-89, 2019.

[8] N. Kattukkaran, A. George and T. P. Haridas, "Intelligent Accident Detection and Alert System for Emergency Medical Assistance," International Conference on Computer Communication and Informatics, 2017.

[9] W. R. Mouleeshu, D. S. Niviya, E. G. Pearlstone, K. M. Sathish and A. M. Yashar, "Make Way: An Intelligent Real-Time Traffic Light Control System," International Journal of Engineering and Advanced Technology, vol. 9, no. 2, pp. 709- 712, 2019.

[10] Nicky Kattukkaran, Mithun Haridas T P, "Accident detection and alert system for emerg ency medical assistance"in International Conference on Computer Communication and Informatics (ICCCI) 2017.

[11] Rohit Ganiga, Rohit Maurya, Archana Nanade,"Accident detection system using Piezo Disk Sensor", International Journel of science, Engineering and Technology Research(IJSETR) volume6,Issue3,March 2017,ISSN 2278-7798

[12] Nimisha Chaturvedi, PallikaSrivastava . "Automatic Vehicle Accident Detection and Messaging System Using GSM and GPS Modem ",Volume: 05 Issue: 03 | Mar-2018.

[13] J. (2020, October 31). How to Interface GSM Module to Arduino-Send and Receive SMS. Electronic Circuits and Diagrams-Electronic Projects and Design.

[14] S. Sharma and S. Sebastian, "IoT based car accident detection and notification algorithm for general road accidents", International Journal of Electrical & Computer Engineering, vol. 9, no. 5, pp. 2088-8708, 2019.

[15] N. Kattukkaran, A. George and T. M. Haridas, "Intelligent accident detection and alert system for emergency medical assistance", 2017 International Conference on Computer Communication and Informatics (ICCCI), pp. 1-6, 2017, January.

[16] M. Murshed and M. S. Chowdhury, "An IoT based car accident prevention and detection system with smart brake control", Proc. Int. Conf. Appl. Techn. Inf. Sci.(iCATIS), pp. 23, 2019, January.

APPENDIX-A

PSUEDOCODE

```
#include <DHT.h>

/******************DHT11***********************************/
#include "DHT.h"
#define DHTPIN 9     // what pin we're connected to
#define DHTTYPE DHT11   // DHT 11
String DHT_status;
DHT dht(DHTPIN, DHTTYPE);
String Temp_status;
int present_condition = 0;
int previous_condition = 0;

const int trigPin = 11;
const int echoPin = 12;

long duration;
int distance;

int x,h;
int a,b,d,e,q;


void setup()
{
  pinMode(8,INPUT);
  pinMode(10,INPUT);
  pinMode(11,INPUT);
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin, INPUT); // Sets the echoPin as an Input
  pinMode(3,OUTPUT);//motor
  pinMode(2,OUTPUT);
  pinMode(5,OUTPUT);
  pinMode(4,OUTPUT);
  Serial.begin(9600);
  dht.begin();
}

void loop()
{

if((a==0)&&(b==0)&&(d==0)&&(e==0)&&(q==0))
{
  forward();
  //Serial.print("Hello");
 }
```

```
 else if (q==1)
 {
  stopp();
  gps(help);
 // Serial.print("Hi");
  }

   else if (e==1)
 {
  stopp();

// // Serial.print("Hi");
  }

/**else if ((a=1)||(b=1)||(d=1)||(e=1))
{
  stopp();
  // Serial.println("N");
}
else if((f=1))
{
  stopp();
  gps(help);
  // Serial.println("N");
}
*/
  //String stringThree = e + a + b + d;
// String result = String(e) + String(a) + String(b) + String(d);
 // Serial.println(  result);

 dht11();
 gyro();

 // ultrasonic Sensor
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = duration * 0.034 / 2;
  Serial.print('U');
  Serial.print(distance);

  if (distance>10)
   {
     Serial.print("V");
     e=0;
  // Serial.print(e);
   // forward();
    //delay(100);
```

```
  }
  else
  {
   Serial.print("v");
  // stopp();
   e=1;
  }
//previous_condition = present_condition;
//present_condition = digitalRead(A5); // Reading digital data from the A5 Pin of the
Arduino.
////Serial.println(present_condition);
//if (previous_condition != present_condition)
//{
////led_blink();
//e=1;
//Serial.print("v");
//delay(1000);
//
//}
//else
//{
//e=0;
//Serial.println("V");
//
}

// int Vib=digitalRead(11);
// if (Vib==HIGH)
//  {
//    Serial.print("V");
//    e=0;
//   // Serial.print(e);
//    delay(100);
//    }
//  else
//  {
//  Serial.print("v");
//  e=1;
//  delay(1000);
//    }
//
  int Gas =  digitalRead(10);
  if (Gas == LOW)
  {
   Serial.print("G");
   a=1;
  }
  else
  {
   Serial.print("g");
   a=0;
```

```
    }
  int flame = digitalRead(8);
  if (flame == LOW)
  {
    Serial.print("F");
    b=1;
  }
  else
  {
    Serial.print("f");
    b=0;
    }

 }

void dht11()
{
        h = dht.readHumidity();
          x = dht.readTemperature();
        float f = dht.readTemperature(true);


        if (isnan(h) || isnan(x) || isnan(f))
        {
          Serial.println("Failed to read from DHT sensor!");
          return;
        }

        /****************************************/

      if(x > 35)
        {
                Serial.print('T');
                Serial.print(x);
                delay(500);
                d=1;

        }
        else
       {
//
                     Serial.print('t');
                     Serial.print(x);
                    delay(500);
                    d=0;
        }

         delay(500);

}
void forward()
```

```
{
  digitalWrite(3,HIGH);
  digitalWrite(2,LOW);
  digitalWrite(5,HIGH);
  digitalWrite(4,LOW);
  }

void stopp()
{
  digitalWrite(3,LOW);
  digitalWrite(2,LOW);
  digitalWrite(5,LOW);
  digitalWrite(4,LOW);
  }

void gyro()

{
        int x = analogRead(A0);
        int y = analogRead(A1);
        int z = analogRead(A2);

        x = x / 100;
        y = y / 100;
        z = z / 100;
        /**************************************************/
 // Serial.print(x);
  //Serial.print(y);
 // Serial.print(z);
  //delay(1000);

    if(x==3&&y==3&&z==2)
      {
      q=0;
      //forward();
       Serial.print("y");
         delay(5);
      }

    else
      {
      q=1;
      //stopp();
       Serial.print("Y");
         delay(5);
      }
}
void gps(String msg)
{
```

```
                    while(Serial.available())
                    {

                      inChar = (char)Serial.read();
                      //Serial.println(inChar);


                      if(inChar != '\n')
                        {

                           if ( inChar == '$')
                           {
                             i=0;
                             inputString1[0]= inChar;

                           }
                           else if(inputString1[0] == '$')
                           {
                             i++;
                             inputString1[i]= inChar;
                           }


                        }

                      else if (inChar == '\n')
                        {

                            stringComplete = true;
                            //Serial.println(inputString1);

                            i=0;
                            if(  (inputString1[0] == '$') && (inputString1[1] == 'G') &&
(inputString1[2] == 'P') && (inputString1[3] == 'R')  && (inputString1[4] == 'M') &&
(inputString1[5] == 'C'))
                            {
                            //Serial.println(inputString1);
                            GPS_GPRMC(inputString1 ,msg);
                            Serial.flush();

                            }

                        }


                }


}
```

```
/*************************************gps_function**********************
*******************************************************************/
int GPS_GPRMC(String inputString,String get_message)
{


//**********************************************************************
**********************************************************************
******************

        Gps_infomation_variable = 0;

        GPS_information[0]='$';
        GPS_information[1]='G';
        GPS_information[2]='P';
        GPS_information[3]='S';

        if(  (inputString[0] == '$') && (inputString[1] == 'G') && (inputString[2] == 'P') &&
(inputString[3] == 'R')  && (inputString[4] == 'M') && (inputString[5] == 'C'))
        {

                k=0;
                i=0;
                split_variable=0;

                Gps_infomation_variable=4;

               //Serial.println("******");

               //********************************** split time,lat,lon,date using ','
character ******************************************************

                while(inputString[i] != '\0')
                {

                    if(inputString[i] == ',')

                     {

                          split_variable++;
                          z=0;
                          //i++;

                     }


                     switch (split_variable)
                     {

                          case 1:
```

```
    timee[z] = inputString[i];
    z++;
    break;

case 2:

    gps_fix = inputString[i];

    GPS_information[Gps_infomation_variable] = gps_fix ;

    Gps_infomation_variable++;

    break;

case 3:


    lattitude[z] = inputString[i];

    GPS_information[Gps_infomation_variable] = lattitude[z];

    Gps_infomation_variable++;
    z++;

    break;

case 4:

    lat_dir = inputString[i];

    GPS_information[Gps_infomation_variable] = lat_dir ;

    Gps_infomation_variable++;

    break;

case 5:

    longitude[z] = inputString[i];

    GPS_information[Gps_infomation_variable] = longitude[z];

    Gps_infomation_variable++;

    longitude_length = z;

    z++;

    break;

case 6:
```

```
                lon_dir = inputString[i];

                GPS_information[Gps_infomation_variable] = lon_dir ;

                Gps_infomation_variable++;

                break;

              case 9:

                date[z] = inputString[i];

                GPS_information[Gps_infomation_variable] = date[z] ;

                Gps_infomation_variable++;
                z++;

                break;

              case 11:
                 z=0;


          }

          i++;
       }


    //****************************************** indian_time
***************************************************************************
*****

       hours = (((timee[1]-48)*10)+(timee[2] - 48)) + 5;

       if (hours >=24 )
        {
          hours = hours-24;
        }

      minits = (((timee[3]-48)*10)+(timee[4] - 48))+30;

       if (minits >=60)
         {
            minits = minits-60;

            hours = hours+1;
         }

      seconds = ((timee[5]-48)*10)+(timee[6]-48);
```

```
        indian_time_gps[0]=(hours/10)+48;
        indian_time_gps[1]=(hours%10)+48;
        indian_time_gps[2]=':';

        indian_time_gps[3]=(minits/10)+48;
        indian_time_gps[4]=(minits%10)+48;
        indian_time_gps[5]=':';

        indian_time_gps[6]=(seconds/10)+48;
        indian_time_gps[7]=(seconds%10)+48;


      GPS_information[Gps_infomation_variable] = ',';
      Gps_infomation_variable++;

      for(i=0;indian_time_gps[i]!='\0';i++)
      {
         GPS_information[Gps_infomation_variable] = indian_time_gps[i];
         Gps_infomation_variable++;

      }

     // Serial.print("************");

    //****************************************here we calculated decimal
values for map**************************************************************

    //**************************************** check gps fix or not
***************************************************************************
********


      //Serial.println(final_lat_long);

      if (gps_fix == 'V')
      {

         Gps_infomation_variable1 = 0;

         while( GPS_information[Gps_infomation_variable1] != '\0')
         {
               if(Gps_infomation_variable1 < Gps_infomation_variable)
               {
                 //Serial.print( GPS_information[Gps_infomation_variable1]);
                 Gps_infomation_variable1++;
               }
               else
               {
                  //Serial.print( GPS_information[Gps_infomation_variable1]='\0');
                  Gps_infomation_variable1++;
               }
```

```
                    }

                    String nf_final_lat_long =  String("GPS_not_fix_")+String (get_message);

                      //lcd.setCursor(14,1);
                     // lcd.print("GN");
                      //Serial.print(" Gps:Not         ");


                    webdata_len = nf_final_lat_long .length();

                   String send_data  = String(webdata_len)+"_"+nf_final_lat_long;


                     // Serial.print('Z');
//                   Serial.print(send_data);
//
//                   delay(2000);

                   break_char ='s';

                  // String iot = send_data.substring(0,2);
                   //int len=iot.toInt();
                   //String iot1 = send_data.substring(3,(len+3));
                   //Serial.println("****************************");
                   //Serial.print(iot1);
                   //Serial.println("****************************");


                    //Serial.print(String(webdata_len)+"_"+nf_final_lat_long);

/*******************************************************************
******************************/


                }
                else if(gps_fix == 'A')
                {


                    //  lcd.setCursor(14,1);
                  //   lcd.print("GY");
                  //Serial.print("Y");

                      float decimal_lattitude = lat_long_calculation(lattitude);
                      float decimal_longitude = lat_long_calculation(longitude);
//

                    //  lcd.setCursor(3,0);
                    //  lcd.print("L:");
```

```
                    //   lcd.setCursor(5,0);
                     //  lcd.print(String(float(decimal_lattitude),10));
                     //  lcd.print(double(decimal_lattitude),7);
                     //  lcd.setCursor(10,0);
                     //  lcd.print("LG:");
                     //  lcd.setCursor(12,0);
                      // lcd.print(String(float(decimal_longitude),10));
                    // lcd.print(double(decimal_longitude),7);

//
//                       lcd.setCursor(10,1);
//                       lcd.print("G:Send");

                      String final_lat_long =
"Lat:"+String(float(decimal_lattitude),10)+String('_')+"LG:"+String(float(decimal_longitud
e),10)+"_"+String(get_message);
                      webdata_len = final_lat_long.length();

                      Serial.print('L');
                      Serial.print(double(decimal_lattitude),7);
                      delay(20);
                      Serial.print('P');
                      Serial.print(double(decimal_longitude),7);
                      delay(20);
//                    Serial.print(String(webdata_len)+"Help"+final_lat_long);
//                    delay(20);
//                    delay(4000);

                      break_char ='s';
//                    delay(1000);

                }

          }


//******************************************************CLEAR********
***************************************************************************
**************************
        for(i=0;inputString[i]!='\0';i++)
        {
                inputString[i]='\0';
        }
       // Serial.print("\n");

}
float lat_long_calculation(char *gps_location_lat_lon)
{

            int z,l,total_length;
            int dot_length;
```

```
        char decimal_gps[15];
        for(z=0;gps_location_lat_lon[z]!='\0';z++)
         {
            decimal_gps[z] = gps_location_lat_lon[z];
           // Serial.print(decimal_gps[z]);
             if(gps_location_lat_lon[z] == '.')
             {
                  dot_length=z;
             }
             total_length=z;
         }
           l=0;

           for(z=1;z<(dot_length-2);z++)
           {
              gps_degree[l]=decimal_gps[z];
              l++;
           }

         l=0;

         for(z=(dot_length-2);z<=total_length;z++)
           {
              gps_minutes[l]=decimal_gps[z];

              l++;
           }

           float pi = atof(gps_degree);
           float pi1 = atof( gps_minutes);
           float pi3 = pi+(pi1/60);

           for(z=0;gps_degree[z] !='\0';z++)
            {
               gps_degree[z]='\0';
            }
           return(pi3);

}
```

IOT

```
//girish2228@gmail.com - User
//changowda43@gmail.com - User
//projectiot2023


 #define BLYNK_TEMPLATE_ID "TMPL3VNjJ6s3Y"
#define BLYNK_TEMPLATE_NAME "Vehicle accident"
#define BLYNK_AUTH_TOKEN "UXQAoQIDyWlIQVO1pUi6vFCS_ILhC2Wr"
```

```
/* Comment this out to disable prints and save space */
#define BLYNK_PRINT Serial


#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "projectiot";
char pass[] = "123456789";

#define VPIN_1    V0  // Clash
#define VPIN_2    V1  // Acc_Status
#define VPIN_3    V2  // Gas
#define VPIN_4    V3  // Flame
#define VPIN_5    V4  // Temp_St
#define VPIN_6    V5  // Temp
#define VPIN_7    V6  // GPS_Lt
#define VPIN_8    V7  // GPS_Lg
#define VPIN_9    V8  // GPS_Lg


BLYNK_CONNECTED()
{

  Blynk.syncVirtual(V0);
  Blynk.syncVirtual(V1);
  Blynk.syncVirtual(V2);
  Blynk.syncVirtual(V3);
  Blynk.syncVirtual(V4);
  Blynk.syncVirtual(V5);
  Blynk.syncVirtual(V6);
  Blynk.syncVirtual(V7);
  Blynk.syncVirtual(V8);


}
void setup()
{
  // Debug console
  Serial.begin(9600);
  delay(100);
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
}

void loop()
{
  Blynk.run();
  if (Serial.available()>0)
```

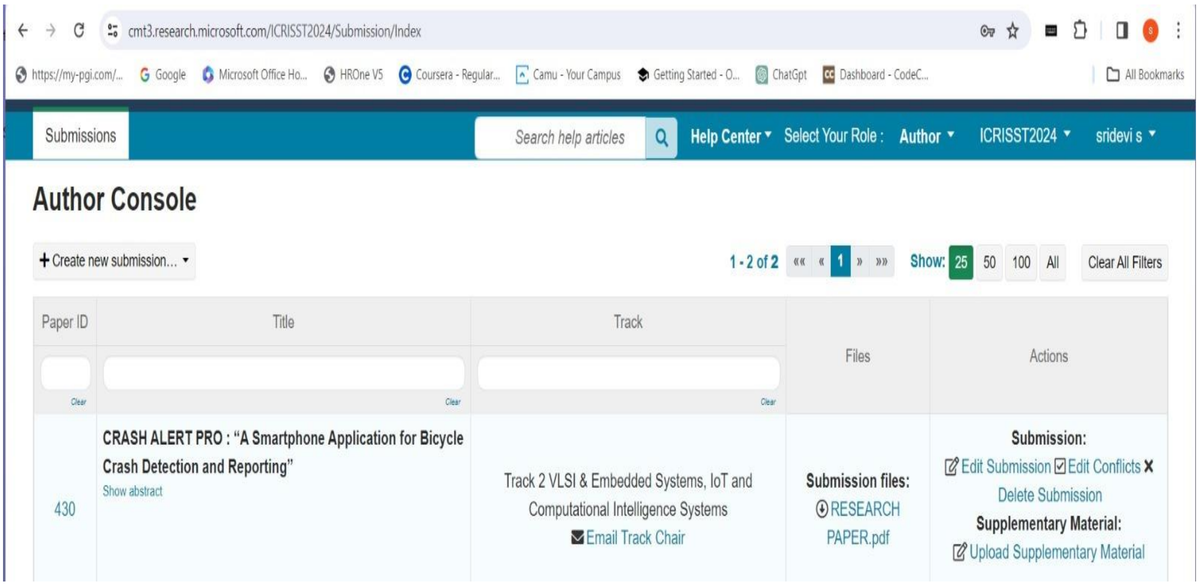
```
  {
  char a=Serial.read();
  //Serial.print(a);


  if(a=='v')
{
  Blynk.virtualWrite(VPIN_1,HIGH);
}
else if (a=='V')
{
  Blynk.virtualWrite(VPIN_1,LOW);
}
 if(a=='T')
{
    int T=Serial.parseInt();
    Blynk.virtualWrite(VPIN_5,HIGH);//Distance 1
    Blynk.virtualWrite(VPIN_6,T);//Distance 1
}
else if (a=='t')
{
  int t=Serial.parseInt();
  Blynk.virtualWrite(VPIN_5,LOW);//Distance 1
  Blynk.virtualWrite(VPIN_6,t);//Distance 1
}

if(a=='F')//Flame
    {

      Blynk.virtualWrite(VPIN_4,HIGH);
    }

if(a=='f')//Flame
    {

      Blynk.virtualWrite(VPIN_4,LOW);
    }


if(a=='G')//Gas
    {

      Blynk.virtualWrite(VPIN_3,HIGH);
    }


if(a=='g')//Gas
    {

      Blynk.virtualWrite(VPIN_3,LOW);
    }
```

```
if(a=='Y')//Position
   {

     Blynk.virtualWrite(VPIN_2,HIGH);
   }

if(a=='y')//Position
   {

     Blynk.virtualWrite(VPIN_2,LOW);


if(a=='L')
{
   float L=Serial.parseFloat();   ///Latitute
   Blynk.virtualWrite(VPIN_7,L);
}

if(a=='P')
{
   float p=Serial.parseFloat();   ///LANe
   Blynk.virtualWrite(VPIN_8,p);
}
if(a=='U')
{
   int u=Serial.parseInt();   ///Distance
   Blynk.virtualWrite(VPIN_9,u);
}


}
}
```

APPENDIX-B

IEEE-Research paper

Waiting for Acceptance



SUSTAINABLE DEVELOPMENT GOALS

The project work carried out here is mapped to SDG-3 Good Health and Well-Being.

This goal aims to ensure healthy lives and promote well-being for all ages. While bike crash detection specifically may not be explicitly mentioned in the SDGs, it aligns with the broader objective of promoting safety and preventing injuries, which falls under the umbrella of good health and well-being. SDG-9 Industry ,Innovation and Infrastructure also play a role in developing and implementing effective bike crash detection systems.

PLAGARISM CHECK OF A REPORT

CRASH ALERT PRO : "A Smartphone Application for Bicycle Crash Detection and Reporting"

ORIGINALITY REPORT

23%	15%	10%	18%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Presidency University Student Paper	7%
2	Arnav Chaudhari, Harsh Agrawal, Smiti Poddar, Kiran Talele, Manisha Bansode. "Smart Accident Detection And Alert System", 2021 IEEE India Council International Subsections Conference (INDISCON), 2021 Publication	1%
3	Submitted to Amrita Vishwa Vidyapeetham Student Paper	1%
4	forum.arduino.cc Internet Source	1%
5	pastebin.com Internet Source	1%
6	hdl.handle.net Internet Source	1%
7	Submitted to M S Ramaiah University of Applied Sciences Student Paper	1%

8	medium.com Internet Source	1 %
9	www.jetir.org Internet Source	1 %
10	Ton Duc Thang University Publication	<1 %
11	Submitted to Sreenidhi International School Student Paper	<1 %
12	Submitted to Federal University of Technology Student Paper	<1 %
13	cdnx.uobabylon.edu.iq Internet Source	<1 %
14	Submitted to University of Pretoria Student Paper	<1 %
15	dspace.iiuc.ac.bd:8080 Internet Source	<1 %
16	saviranadela.home.blog Internet Source	<1 %
17	Submitted to Liverpool John Moores University Student Paper	<1 %
18	Jamil Abedalrahim Jamil Alsayaydeh, Mohd Faizal bin Yusof, Mohamad Amirul Aliff bin Abdillah, Ahmed Jamal Abdullah Al-Gburi et al. "Enhancing Vehicle Safety: A Comprehensive	<1 %

Accident Detection and Alert System", International Journal of Advanced Computer Science and Applications, 2023
Publication

19	www.semanticscholar.org Internet Source	<1%
20	eprints.polsri.ac.id Internet Source	<1%
21	Submitted to Concordia University Student Paper	<1%
22	Submitted to University of Greenwich Student Paper	<1%
23	Submitted to University of Bedfordshire Student Paper	<1%
24	circuitdigest.com Internet Source	<1%
25	S.P. Rajamohana, Zbigniew M. Wawrzyniak, S. Krishna Prasath, R. Shevannth, I. Raja Kumar, M. Mohammed Rafi, T. Hariprasath D. "IoT-based heart disease prediction system", Institution of Engineering and Technology (IET), 2022 Publication	<1%
26	www.hackster.io Internet Source	<1%

www.sdgresources.relx.com

27	Internet Source	<1 %
28	Submitted to Universiti Malaysia Sabah Student Paper	<1 %
29	researchr.org Internet Source	<1 %
30	www.coursehero.com Internet Source	<1 %
31	Submitted to Pandit Deendayal Petroleum University Student Paper	<1 %
32	Submitted to UTC Reading Student Paper	<1 %
33	Submitted to University of Essex Student Paper	<1 %
34	Submitted to University of Wales, Lampeter Student Paper	<1 %
35	irdindia.in Internet Source	<1 %
36	scholarsbank.uoregon.edu Internet Source	<1 %
37	Submitted to National Institute of Technology Jamshedpur Student Paper	<1 %

38	Submitted to HTM (Haridus- ja Teadusministeerium) Student Paper	<1 %
39	Submitted to University Tun Hussein Onn Malaysia Student Paper	<1 %
40	ijarcce.com Internet Source	<1 %
41	vocal.media Internet Source	<1 %
42	www.ijariit.com Internet Source	<1 %
43	www.researchsquare.com Internet Source	<1 %
44	Submitted to South Bank University Student Paper	<1 %
45	iopscience.iop.org Internet Source	<1 %
46	J.M. González-González, Á. Paredes Parrilla, J.A. Aguado. "Chemical energy storage technologies", Elsevier BV, 2022 Publication	<1 %
47	Submitted to The University of Wolverhampton Student Paper	<1 %

48	Submitted to University of Salford Student Paper	<1 %
49	letmeask.co Internet Source	<1 %
50	www.irjet.net Internet Source	<1 %
51	Submitted to B.V. B College of Engineering and Technology, Hubli Student Paper	<1 %
52	Submitted to University College London Student Paper	<1 %
53	irjetblog.wordpress.com Internet Source	<1 %
54	"Table of Contents", 2021 IEEE India Council International Subsections Conference (INDISCON), 2021 Publication	<1 %
55	Abhishek Tiwari, Daksh Sharma, Harsh Garg, Albert Alexander Stonier. "IoT Based Smart Blind Stick with SoS and Live Tracking System", 2023 Second International Conference on Augmented Intelligence and Sustainable Systems (ICAISS), 2023 Publication	<1 %
56	Submitted to SSN COLLEGE OF ENGINEERING, Kalavakkam	<1 %

	Student Paper		
57	Submitted to Visvesvaraya Technological University, Belagavi Student Paper	<1 %
58	dspace.bracu.ac.bd:8080 Internet Source	<1 %
59	dokumen.tips Internet Source	<1 %
60	ebin.pub Internet Source	<1 %
61	www.careercup.com Internet Source	<1 %
62	Submitted to National University of Singapore Student Paper	<1 %
63	Submitted to Universiti Teknikal Malaysia Melaka Student Paper	<1 %
64	dokumen.pub Internet Source	<1 %
65	forum.arduinopolska.pl Internet Source	<1 %
66	how2electronics.com Internet Source	<1 %
67	ijrpr.com Internet Source	<1 %

68	iotbyhvm.ooo Internet Source	<1 %
69	pdffox.com Internet Source	<1 %
70	www.fireballpc.com Internet Source	<1 %
71	www.ijirset.com Internet Source	<1 %
72	www.ijraset.com Internet Source	<1 %
73	Audrey O'Shea. "A Geek Girl's Guide to Electronics and the Internet of Things", Wiley, 2020 Publication	<1 %

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		