B. Tech Project Report

Auto SOS
Smart Vehicle Safety System with instant Emergency Notifications

Submitted by

Anshuman Shah	20201CSE0662
Deepak V	20201CSE0673
Jayanth B S	20201CSE0680
Aishwarya Oji	20201CSE0701
Greeshma S Devadiga	20201CSE0712

Department of Computer Science and Engineering

Under the Guidance of

Prof. DAKHOLE DIPALI KHUSHALRAO

Assistant Professor
Department of Computer Science and Engineering
School of Engineering

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

Department of Computer Science & Engineering
Bengaluru

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

Certificate

This is to certify that the project entitled "**Auto SOS Smart Vehicle Safety System with instant Emergency Notifications"** has been successfully completed by Anshuman Shah, Deepak V, Jayanth B S, Aishwarya Oji, Greeshma S Devadiga of seventh semester B. Tech at **Presidency University, Bengaluru** as the Internet Of Things project. The Project Report presented here is the bonafide work of the student.

Guide:
Prof. DAKHOLE DIPALI KHUSHALRAO

Head of the Department

Group Members:

Anshuman Shah	20201CSE0662
Deepak V	20201CSE0673
Jayanth B S	20201CSE0680
Aishwarya Oji	20201CSE0701
Greeshma S Devadiga	20201CSE0712

Acknowledgement

While performing our project, we had to take the help and guidelines of some respected persons who deserve our greatest gratitude. The completion of this project gave us immense pleasure. We are highly indebted to Ms. Dakhole Dipali Khushalrao, for their guidance, constant supervision and for their support in completing our project.

Our profound thanks go to the participants whose enthusiastic involvement and valuable insights enriched this project's development and implementation.

We would like to express our gratitude to our parents for their kind co-operation and encouragement.

Abstract

This report explores IoT-enabled smart Accident detection and Emergency notification system, a tech-driven approach enhancing safety on roads. It outlines the following objectives such as accident detection, location tracking, emergency notification signal. Hardware includes Arduino UNO R3, Accelerometer and GSM module etc., with software relying on Arduino IDE and C programming. Hence, this report encapsulates IoT's potential to ensure road safety and response efficiently.

Introduction and Overview

“Auto SOS: Smart Vehicle Safety System with instant Emergency Notifications” is a modern technology to propose an intelligent accident detection, location tracking and notification system to prevent accidents and reduce traffic hazards.

By using an Accelerometer sensor and a GSM module, the system can detect accidents, and sends notification message to the nearest police control room, trusted contacts and hospitals.

With the received information and data, the emergency services can find the shortest route of the accident spot and take initiatives to speed up the rescue process.

Existing Technologies

1.Collision Detection Systems: These use radar, lidar, cameras, or a combination of these sensors to detect potential collisions. When an imminent crash is detected, these systems can trigger emergency notifications.

2. GPS and Location Services: Integration with GPS technology enables precise location tracking, allowing for accurate reporting of incidents to emergency services and designated contacts.

3. Cloud-Based Platforms: Utilizing cloud services allows for real-time data storage, analysis, and sharing, enabling quicker emergency notifications and response.

Objectives

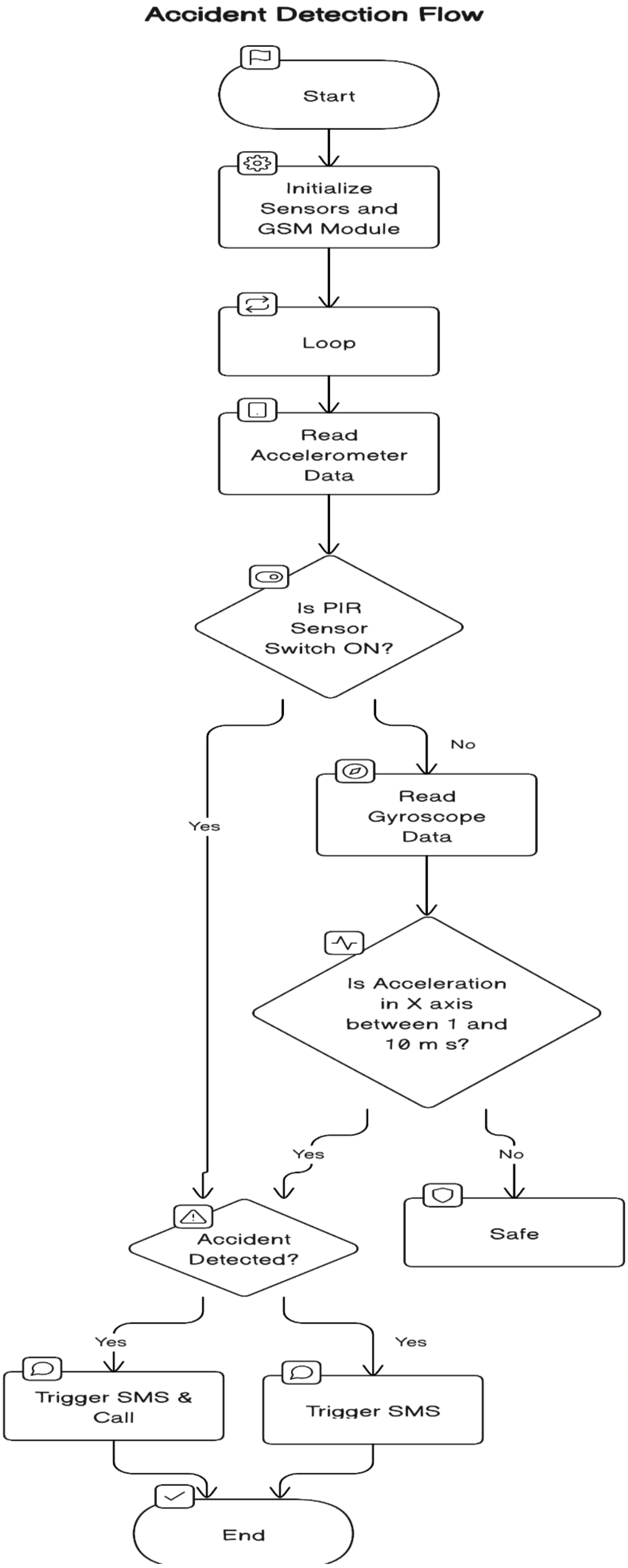
1. **Real-time Threat Detection:** Develop algorithms and sensors capable of identifying potential hazards on the road, such as accidents or impending collisions.

2. **Immediate Emergency Alerts:** Enable the system to send instant notifications to emergency services, designated contacts, and relevant authorities in case of an accident or emergency situation.

3. **Enhanced Communication Interfaces:** Create intuitive user interfaces that facilitate communication between the vehicle's system, emergency services, and concerned individuals to provide real-time updates and information exchange.

4. **Integration with Emergency Services:** Establish seamless integration with local emergency services to relay accurate location data and critical information, expediting their response time in emergencies.

Working of the model



Hardware Components:

1. Arduino Board:

- Model: Arduino UNO

- Purpose: This microcontroller board serves as the brain of the project. It processes the code, interfaces with sensors, and controls communication with the GSM module.

2. Accelerometer Sensor:

- Model: MPU6050

- Purpose: The MPU6050 is an Inertial Measurement Unit (IMU) that combines a 3-axis accelerometer and a 3-axis gyroscope. It measures acceleration in the X, Y, and Z axes and gyroscopic rotational motion, providing data about the vehicle's orientation and changes in motion.

3. GSM Module:

- Model: SIM900A

- Purpose: The SIM900A GSM module facilitates communication over the GSM (Global System for Mobile Communications) network. It enables the Arduino to send SMS messages and make phone calls. In this project, it's utilized for alerting emergency services in the event of an accident.

Software Components:

1. Operating System: Arduino IDE:

- Purpose: The Arduino Integrated Development Environment (IDE) is a software platform for writing, compiling, and uploading code to the Arduino board. It provides a user-friendly interface for programming Arduino microcontrollers.

2. Arduino C Programming Language:

- Purpose: The programming language used to write code for the Arduino. The C-based Arduino language simplifies embedded programming, making it accessible to a wide range of users.

3. Adafruit Libraries:

- Purpose: Adafruit provides libraries that simplify interfacing with various sensors and modules. In this project, Adafruit libraries such as Adafruit_MPU6050 and Adafruit_FONA are utilized for easy integration with the MPU6050 accelerometer and the SIM900A GSM module, respectively.

Interaction and Workflow:

1. Arduino Code Execution:

- The Arduino board runs a program written in the Arduino C programming language.

- The program reads data from the MPU6050 accelerometer, processes it, and checks for conditions indicative of an accident.

2. MPU6050 Data Processing:

- The MPU6050 sensor continuously measures acceleration and gyroscope readings.

- The Arduino processes this data to determine if a significant change in motion, indicative of an accident, has occurred.

3. Accident Detection:

- If the Arduino detects an accident based on the programmed conditions (e.g., PIR sensor or switch state, and/or accelerometer readings), it triggers the emergency response sequence.

4. GSM Communication:

- The Arduino uses the Adafruit_FONA library to interact with the SIM900A GSM module.

- When an accident is detected, the Arduino sends an SMS message to a predefined emergency contact number using send_multi_sms() and initiates a call using make_multi_call().

5. Serial Monitor Output:

- The Arduino continuously prints relevant information to the Serial Monitor for debugging and monitoring purposes. This can be viewed using the Arduino IDE.

6. Emergency Response:

- The GSM module sends an SMS alert to the predefined emergency contact number.

- The Arduino initiates a call to the same emergency contact number, notifying them of the accident.

7. Repeat and Continuous Monitoring:

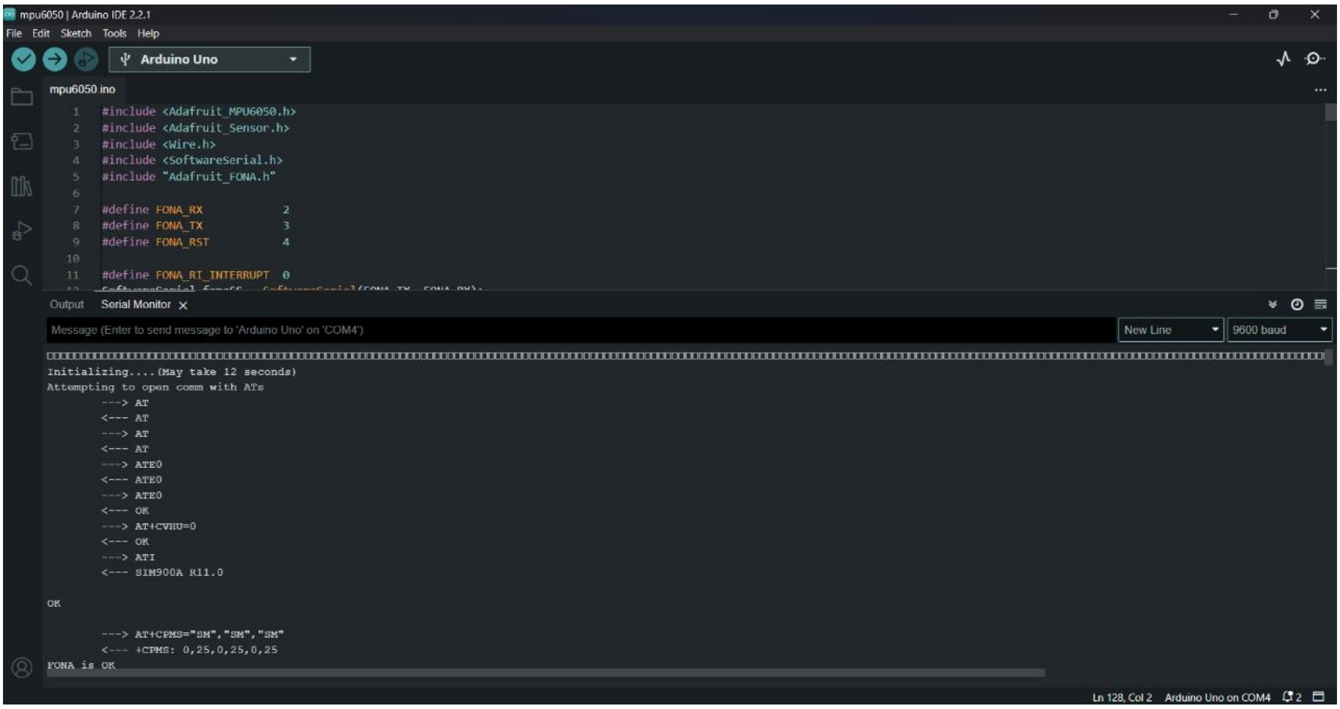
- The entire process is repeated in a loop, allowing the system to continuously monitor for accidents and respond accordingly.

This integrated hardware and software system is designed to enhance vehicle safety by detecting accidents and autonomously alerting emergency services using GSM communication. The combination of an accelerometer, GSM module, and Arduino microcontroller creates a versatile platform for building safety and monitoring applications.

Budget

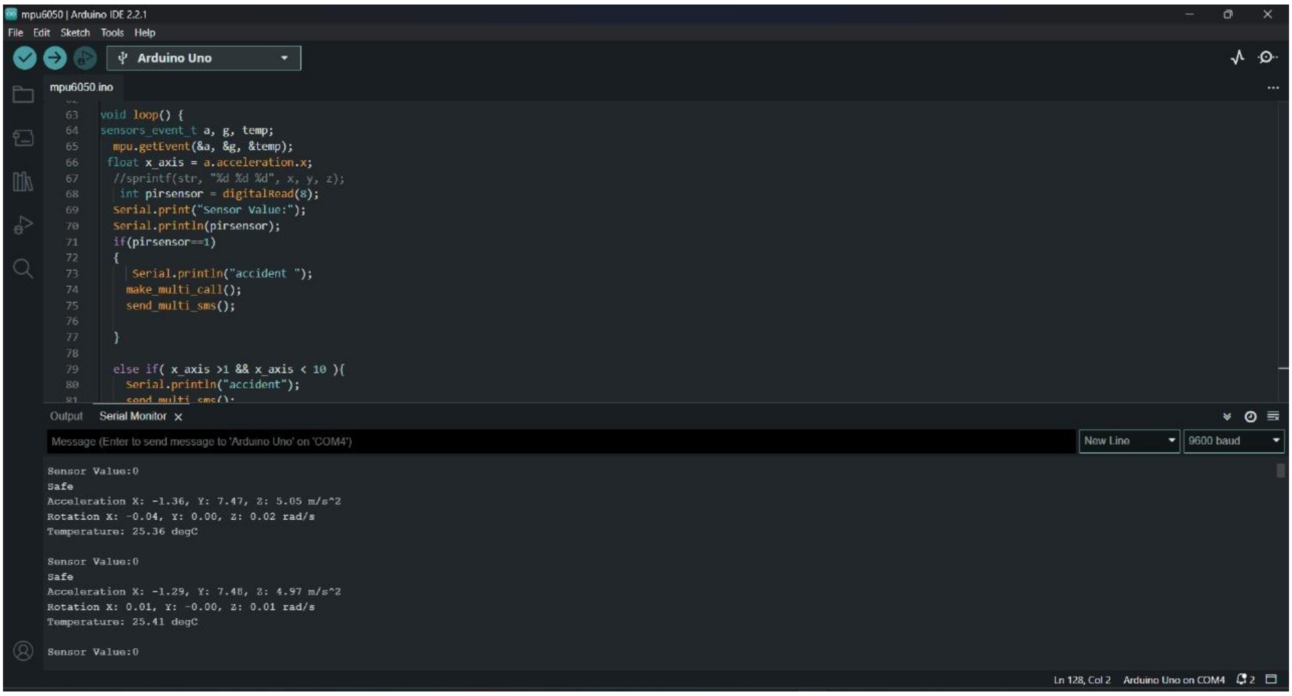
- Arduino: 750Rs
- Accelerometer sensor (ADXL335/MPU6050): 350Rs
- GSM Module (SIM900A): 400Rs
- Breadboard: 85Rs
- Jumper wires: 145Rs

Program code:


mpu6050 | Arduino IDE 2.2.1
File Edit Sketch Tools Help
Arduino Uno
mpu6050.ino
Output
Serial Monitor
Message (Enter to send message to 'Arduino Uno' on 'COM4')
New Line
9600 baud
Ln 128, Col 2 Arduino Uno on COM4 2

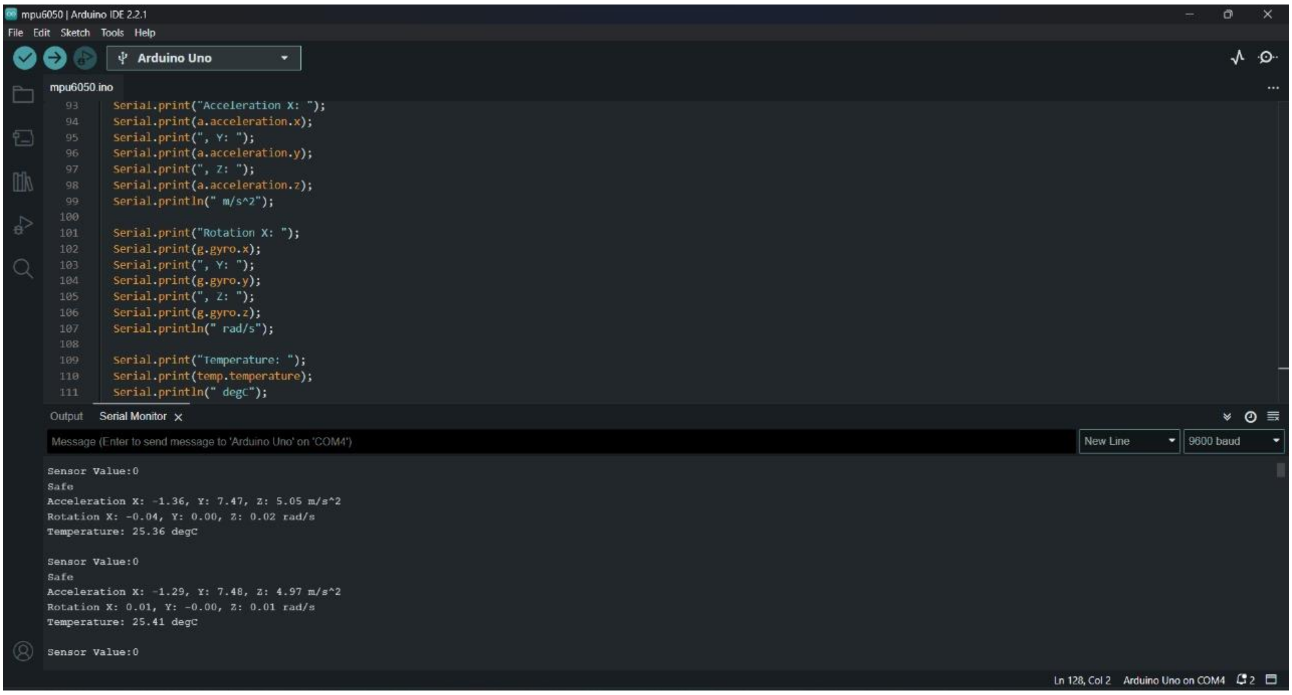
```
#include <Adafruit_MPU6050.h>
#include <Adafruit_Sensor.h>
#include <Wire.h>
#include <SoftwareSerial.h>
#include "Adafruit_FONA.h"

#define FONA_RX            2
#define FONA_TX            3
#define FONA_RST           4

#define FONA_RI_INTERRUPT  0
SoftwareSerial fonaSS = SoftwareSerial(FONA_TX, FONA_RX);
```

```
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
Initializing....(May take 12 seconds)
Attempting to open comm with ATs
        ---> AT
        <--- AT
        ---> AT
        <--- AT
        ---> ATE0
        <--- ATE0
        ---> ATE0
        <--- OK
        ---> AT+CVHU=0
        <--- OK
        ---> ATI
        <--- SIM900A R11.0

OK

        ---> AT+CPMS="SM","SM","SM"
        <--- +CPMS: 0,25,0,25,0,25
FONA is OK
```


mpu6050 | Arduino IDE 2.2.1
File Edit Sketch Tools Help
Arduino Uno
mpu6050.ino
Output
Serial Monitor
Message (Enter to send message to 'Arduino Uno' on 'COM4')
New Line
9600 baud
Ln 128, Col 2 Arduino Uno on COM4 2

```
void loop() {
sensors_event_t a, g, temp;
  mpu.getEvent(&a, &g, &temp);
 float x_axis = a.acceleration.x;
  //sprintf(str, "%d %d %d", x, y, z);
   int pirsensor = digitalRead(8);
  Serial.print("Sensor Value:");
  Serial.println(pirsensor);
  if(pirsensor==1)
  {
     Serial.println("accident ");
    make_multi_call();
    send_multi_sms();

  }

  else if( x_axis >1 && x_axis < 10 ){
    Serial.println("accident");
    send_multi_sms();
```

```
Sensor Value:0
Safe
Acceleration X: -1.36, Y: 7.47, Z: 5.05 m/s^2
Rotation X: -0.04, Y: 0.00, Z: 0.02 rad/s
Temperature: 25.36 degC

Sensor Value:0
Safe
Acceleration X: -1.29, Y: 7.48, Z: 4.97 m/s^2
Rotation X: 0.01, Y: -0.00, Z: 0.01 rad/s
Temperature: 25.41 degC

Sensor Value:0
```



```
  Serial.print("Acceleration X: ");
  Serial.print(a.acceleration.x);
  Serial.print(", Y: ");
  Serial.print(a.acceleration.y);
  Serial.print(", Z: ");
  Serial.print(a.acceleration.z);
  Serial.println(" m/s^2");

  Serial.print("Rotation X: ");
  Serial.print(g.gyro.x);
  Serial.print(", Y: ");
  Serial.print(g.gyro.y);
  Serial.print(", Z: ");
  Serial.print(g.gyro.z);
  Serial.println(" rad/s");

  Serial.print("Temperature: ");
  Serial.print(temp.temperature);
  Serial.println(" degC");
```

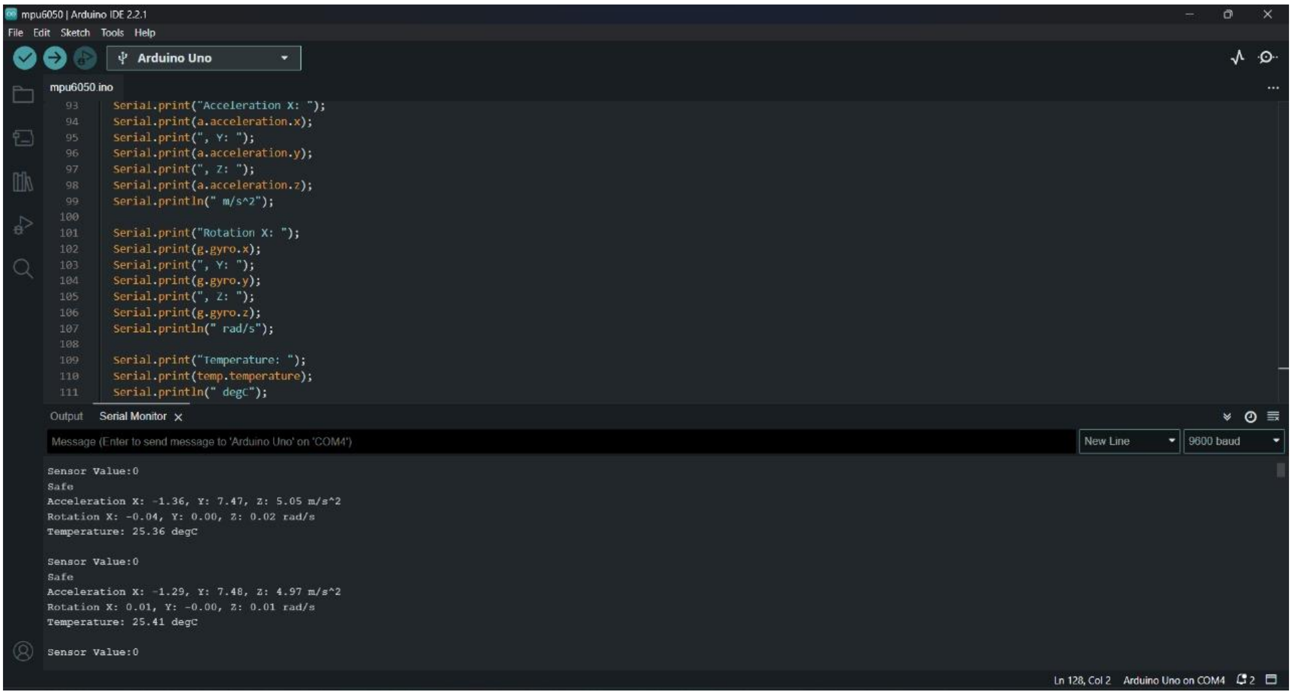
Message (Enter to send message to 'Arduino Uno' on 'COM4')

New Line

9600 baud

```
Sensor Value:0
Safe
Acceleration X: -1.36, Y: 7.47, Z: 5.05 m/s^2
Rotation X: -0.04, Y: 0.00, Z: 0.02 rad/s
Temperature: 25.36 degC

Sensor Value:0
Safe
Acceleration X: -1.29, Y: 7.48, Z: 4.97 m/s^2
Rotation X: 0.01, Y: -0.00, Z: 0.01 rad/s
Temperature: 25.41 degC

Sensor Value:0
```

Ln 128, Col 2 Arduino Uno on COM4



```
  Serial.print("Acceleration X: ");
  Serial.print(a.acceleration.x);
  Serial.print(", Y: ");
  Serial.print(a.acceleration.y);
  Serial.print(", Z: ");
  Serial.print(a.acceleration.z);
  Serial.println(" m/s^2");

  Serial.print("Rotation X: ");
  Serial.print(g.gyro.x);
  Serial.print(", Y: ");
  Serial.print(g.gyro.y);
  Serial.print(", Z: ");
  Serial.print(g.gyro.z);
  Serial.println(" rad/s");

  Serial.print("Temperature: ");
  Serial.print(temp.temperature);
  Serial.println(" degC");
```

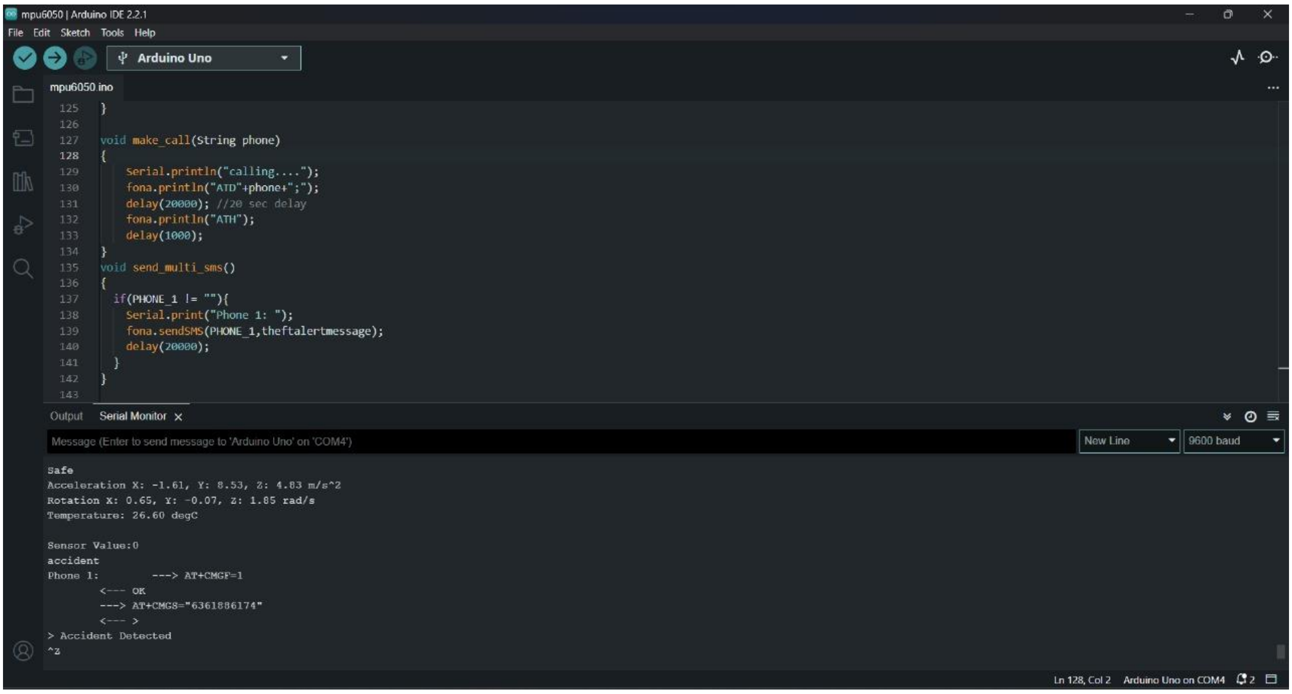
Message (Enter to send message to 'Arduino Uno' on 'COM4')

New Line

9600 baud

```
Sensor Value:0
Safe
Acceleration X: -1.36, Y: 7.47, Z: 5.05 m/s^2
Rotation X: -0.04, Y: 0.00, Z: 0.02 rad/s
Temperature: 25.36 degC

Sensor Value:0
Safe
Acceleration X: -1.29, Y: 7.48, Z: 4.97 m/s^2
Rotation X: 0.01, Y: -0.00, Z: 0.01 rad/s
Temperature: 25.41 degC

Sensor Value:0
```

Ln 128, Col 2 Arduino Uno on COM4

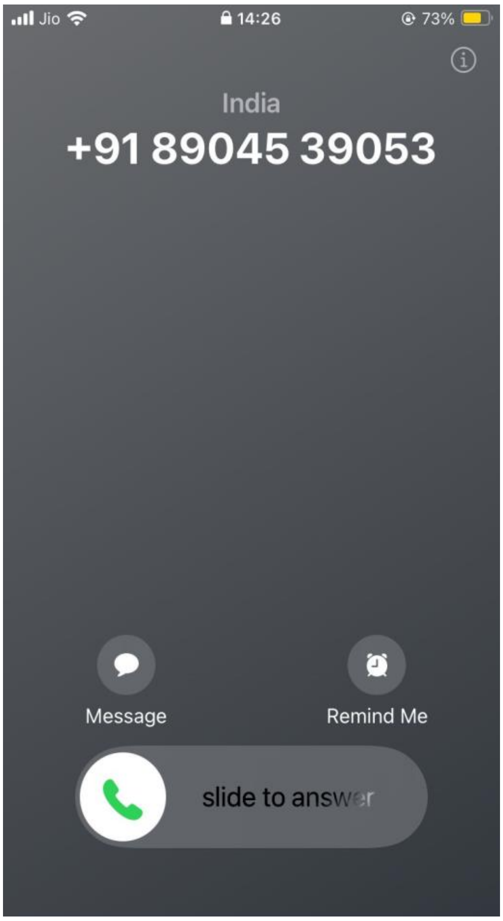


mpu6050 | Arduino IDE 2.2.1
File Edit Sketch Tools Help
Arduino Uno
mpu6050.ino
Output
Serial Monitor
Message (Enter to send message to 'Arduino Uno' on 'COM4')
New Line
9600 baud
Ln 128, Col 2
Arduino Uno on COM4
2

```
}

void make_call(String phone)
{
    Serial.println("calling....");
    fona.println("ATD"+phone+";");
    delay(20000); //20 sec delay
    fona.println("ATH");
    delay(1000);
}
void send_multi_sms()
{
  if(PHONE_1 != ""){
    Serial.print("Phone 1: ");
    fona.sendSMS(PHONE_1,theftalertmessage);
    delay(20000);
  }
}
```

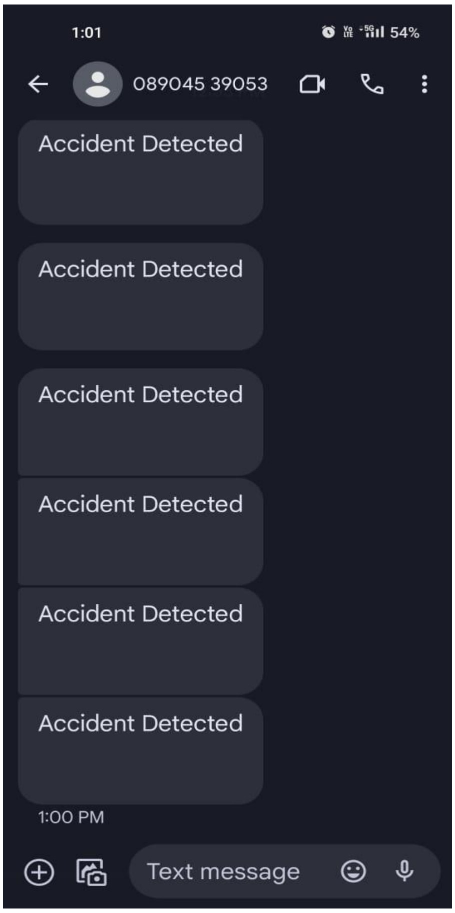
```
Safe
Acceleration X: -1.61, Y: 8.53, Z: 4.83 m/s^2
Rotation X: 0.65, Y: -0.07, Z: 1.85 rad/s
Temperature: 26.60 degC

Sensor Value:0
accident
Phone 1:        ---> AT+CMGF=1
        <--- OK
        ---> AT+CMGS="6361886174"
        <--- >
> Accident Detected
^Z
```

Output:


Jio 14:26 73%

India

+91 89045 39053


1:01 54%

089045 39053

Accident Detected

Accident Detected

Accident Detected

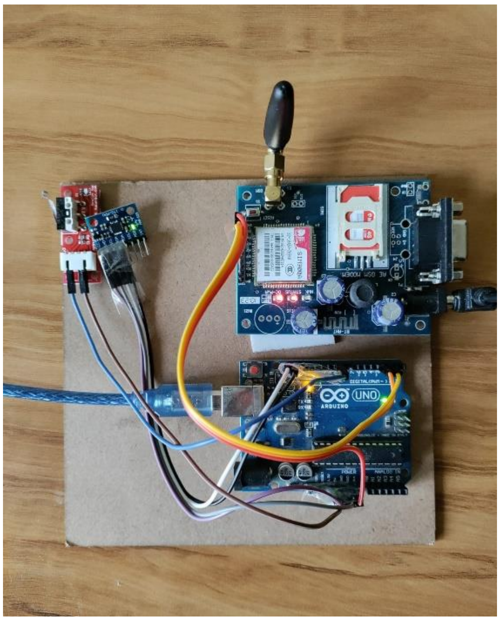
Accident Detected

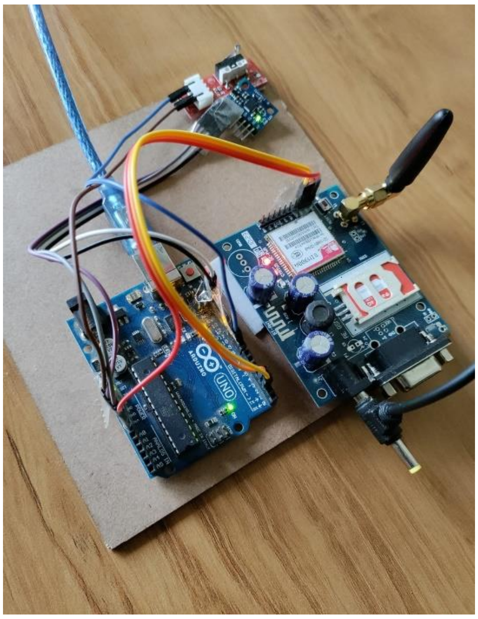
Accident Detected

Accident Detected

1:00 PM

Model:





Conclusion

In conclusion, the development of a Smart Vehicle Safety System equipped with instant Emergency Notifications represents a significant leap forward in enhancing road safety and response efficiency during critical situations. This system, designed with a multifaceted approach, aims to detect and respond to emergencies promptly while prioritizing user safety and communication with emergency services.

The integration of real-time threat detection, immediate emergency alerts, automated response mechanisms, and seamless communication interfaces ensures a proactive and swift reaction to potential hazards or accidents.

Ultimately, the deployment of such a system has the potential to save lives, reduce the severity of accidents, and significantly enhance the overall safety landscape on our roads. Its success relies on a holistic approach that combines technological innovation, regulatory compliance, user education, and seamless collaboration with emergency services, marking a significant stride towards a safer and more responsive driving experience.