**ACCIDENT DETECTION AND ALERT SYSTEM USING GPS AND GSM**

**SUBMITTED BY:**

**ADITYA KAUSHIK\_210950330001**

**ANSHUL CHAUHAN\_210950330003**

**ARE DEEKSHITHA\_210950330004**

**SUBMITTED TO:**

**Mr. AMALRAJ P C, CDAC-HYDERABAD**

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| **S NO.** | **CONTENTS** |
| **1** | **ABSTRACT & INTRODUCTION** |
| **2** | **MOTIVATION AND PROBLEM STATEMENT** |
| **3** | **REQUIREMENT SPECIFICATIONS & IMPLEMENTED FEATURES**   * **Hardware Requirements for this Project** * **Software Requirements for this Project** |
| **4** | **HIGH LEVEL DESIGN OF THE SYSTEM**   * **Architecture of the Proposed System** * **Block Diagram of the System** |
| **5** | **LOW LEVEL DESIGN OF THE SYSTEM**   * **Accident Detection System** * **Accident Alert System** * **Complete Code of this Project** |
| **6** | **RESULTS** |
| **7** | **CONCLUSION** |
| **8** | **LESSONS LEARNT** |
| **9** | **REFERENCES** |

**ABSTRACT**

In highly populated countries, people lose their lives because of accidents and poor emergency facilities every day. These lives could have been saved if medical facilities are provided at the right time. One approach to eliminate the delay between accident occurrence and first responder dispatch is to use automatic accident detection and notification systems in vehicles. This system aims to alert the near and dear ones of the person in the vehicle about the accident, so that they can provide immediate medical aid to the victim. In this System when a vehicle meets with an accident, sensors like 3-axis accelerometer, piezoelectric Sensor Transducer, fire sensor will detects the signal and sends it to a microcontroller (ESP32). Immediately microcontroller gathers the exact value of the geographical coordinates from GPS module which contains the value of longitude and latitude of the location of accident. After that the microcontroller sends the alert message through the GSM module to the near and dear ones or emergency medical facilities.

**INTRODUCTION**

The motor vehicle population is growing at a faster rate than the economic and population growth. Accidents and the death rate due to road accidents are also increasing. Most of the accident deaths that happen are due to the lack of immediate medical assistance on the roads like express highways, hilly regions. A facility for providing immediate medical assistance to the accident area can reduce the fatality to a greater extends. Thus comes the idea of an alert system that senses the accident and its seriousness to alert the medical centre, to the passenger’s near and dear ones for providing ambulance or medical aid to the accident area.

The proposed system will check whether an accident has occurred and identify the seriousness of the injury to the accident victim/driver. Once the decision of serious accident has taken, the system will alert the victim’s dear ones via short message service and they could inform the rescue team so that the rescue team can rush to the spot immediately without any delay as the correct location will be communicated by the GSM Module which is placed inside the vehicle.

**MOTIVATION AND PROBLEM STATEMENT**

Due to road accidents, 53 deaths occur in India every hour. About 450,000 accidents occur annually of which 150,000 people die every year which brings the death rate to 33.3% every year. This rate of accidents increases every day due to an increase in the motor vehicle population. Most of these deaths happen due to a lack of immediate medical assistance. A facility for providing immediate medical assistance can reduce the death rate by 20%. Thus comes the idea of an alert system that senses the accident and its seriousness and then alerts the medical centre nearby and also the near and dear of the person in the accident.

Every day around the world, a large percentage of people die from traffic accident injuries. An influential indicator of survival rates after detecting the accident is the time between the occurrence of the accident and the arrival of emergency responders to the scene. Reductions in this time, in turn, may affect the numbers of fatalities, and this is achieved through using automatic accident detection and notification systems which are either built-in in modern vehicles or available on the roads.

Earlier there was no particular system to detect accidents, usually when an accident occurs people who are passing by or people who saw the accident used to call for medical assistance. This particular alert system entirely depends on the probability of people who saw the accident happen, which causes a delay in the occurrence of the accident and call for help. This delay may be in minutes or hours is crucial for the person in the accident.

So, the proposed project helps reduce the delay which may save a lot of lives.

**REQUIREMENT SPECIFICATION & IMPLEMENTED FEATURES**

**Hardware Requirements for this project**

**1. ESP32 Development Board**

The ESP32 is a development board developed by Espressif systems. It can be programmed using Arduino IDE and ESP-IDF. It has a built in Bluetooth module and CAN protocol and SRAM. It has 36 GPIO Pins with a CPU clock of 160MHz. It has 12-bit ADC onboard and supports CAN, UART, I2C and I2S. It can be used in prototyping IoT products, Low power Battery operated application, small range networking projects, and with the projects which require many Input Output Pins and Wi-Fi and Bluetooth connectivity.

**ESP32 WROOM-32 MCU Module Specifications:**

* Microprocessor:TensilicaXtensa LX6
* Operating Voltage: 3.3V
* Analog Input Pins: 12-bit, 18 Channel
* DAC Pins: 8-bit, 2 Channel
* Digital I/O Pins: 39 (of which 34 is normal GPIO pin)
* SRAM: 520KB
* WiFi: 802.11 b/g/n

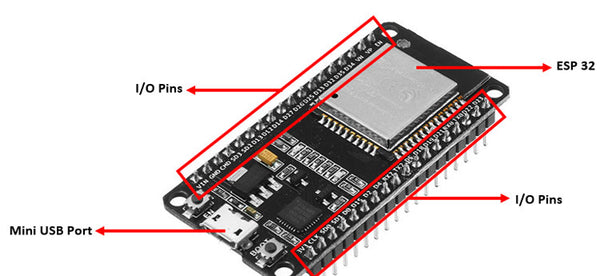


Figure: ESP32 Development Board

**2. SIM800L GPRS GSM Module**

The **SIM800L** is a micro SIM low cost **GSM/GPRS Development Module**. The Module supports TTL communication and hence can easily communicate with Microcontrollers without the need of additional data converters. The module also supports antenna with IPX connector.

**SIM800L**module can be used to make a call, receive a call, send and receive text messages, connecting to internet through GPRS, TCP/IP, etc. Moreover the module supports quad-band GSM/GPRS network, so it can operate globally.

The **SIM800L GPRS GSM Module** is compact in nature and hence can be directly used on final Designs. The on-board LED indicates the connection status of the Board, when there is no signal the LED flashes quickly and when a signal is established it flashes slowly.



Figure: SIM800L GSM Module

**3. NEO-6M GPS Module**

The **NEO-6M** is a stand-alone **GPS** (Global Positioning System) module featuring a high-performance 50 channel U-Blox 6 positioning engine. The **NEO6M GPS module** checks for location on Earth and provides the Latitude and Longitude of the position it is in. This is a low-cost module with a detachable antenna, which also comes with a logic level converter and a voltage regulator, which makes it compatible with both 5V and 3.3V powered boards like Arduino Uno, ESP32, etc. It can be used in the Navigation Systems of Smartphones and Tablets, Drones, in location-based services, etc.

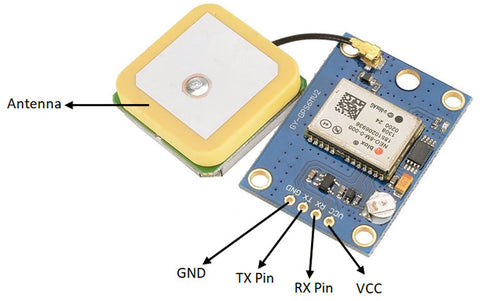


Figure: NEO6M GPS Module

**4. ADXL335 - Triple Axis Linear Accelerometer**

ADXL335 is a Breakout board based on 3 axis ADXL335 IC from Analog Devices. The Accelerometer Module requires no external devices and works on 5V power supply. It can be directly interfaced to ADC of a microcontroller without any external components.

The ADXL335 is a triple axis MEMS accelerometer with extremely low noise and power consumption - only 320uA.This module can be used to sense motion or tilt (in case of non moving) in 3 axis.

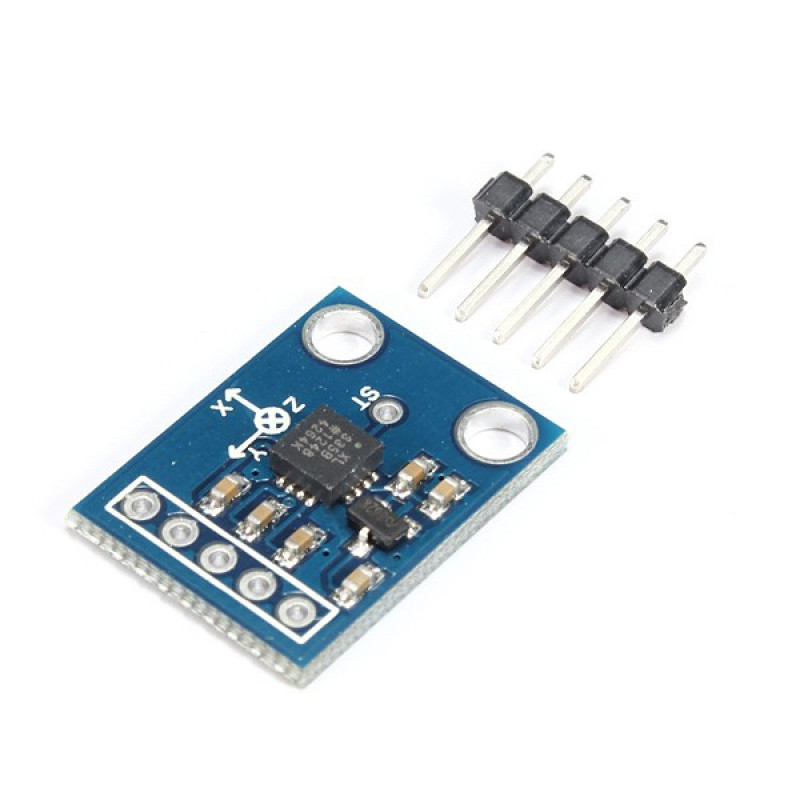


Figure: ADXL335 3-axis Accelerometer

**5. Piezoelectric sensor 35mm**

This piezoelectric sensor 35mm measures changes in pressure, acceleration, force, temperature or strain and converts them into electrical voltage.  This is generally used to detect vibration or a knock. This can also be used for a very small audio transducer such as a buzzer.

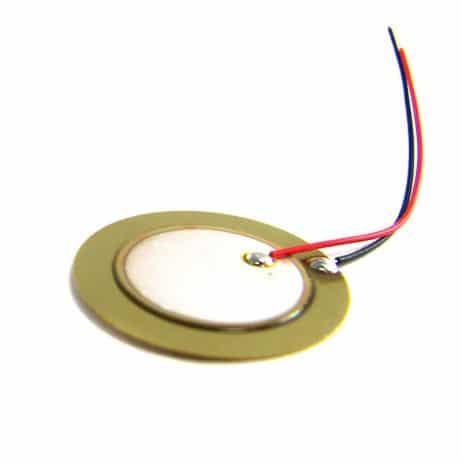


Figure: Piezoelectric Sensor

**6. IR Flame/Fire Sensor Module**

This **flame sensor** or **fire sensor module**works on the concept that when a flame or fire is burning it emits IR signals. This IR signal is then received by the IR receiver on the **fire sensor module** to detect the flame or fire.

The sensor has an operating voltage from 3V to 5.5V and has both digital and analog output. The sensitivity of the digital output can be controlled by the on-board potentiometer. Detection angle of sensor is 60 degree and range is theoretically 100cm but practically you can get upto 20-30cm.



Figure: IR Fire Sensor

**7. 18650 Li-ion 2000mAh Rechargeable Battery**

**18650 battery** is a Li-ion rechargeable battery with 2000 mAh Battery Capacity. It is very useful for applications which require continuous high current or high current in short bursts. An 18650 cell can be charged and discharged up to 1000 cycles without much loss in battery capacity. They are safe to use, environment friendly and has a long battery life. It comes with high energy density and provides excellent continuous power sources to your device.



Figure: 3.7V, 2A Li-ion Battery

**Software requirements for this project**

**PlatformIO IDE for VSCode**

PlatformIO is a cross-platform, cross-architecture, multiple framework, professional tool for embedded systems engineers and for software developers who write applications for embedded products. It can be used to develop software platforms for Arduino, ESP32, PIC32, and AVR. PlatformIO can be used with VS Code to provide a powerful set of tools to assist you in development.

**Visual Studio Code** is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux.PlatformIO can be used as extension in VSCode.

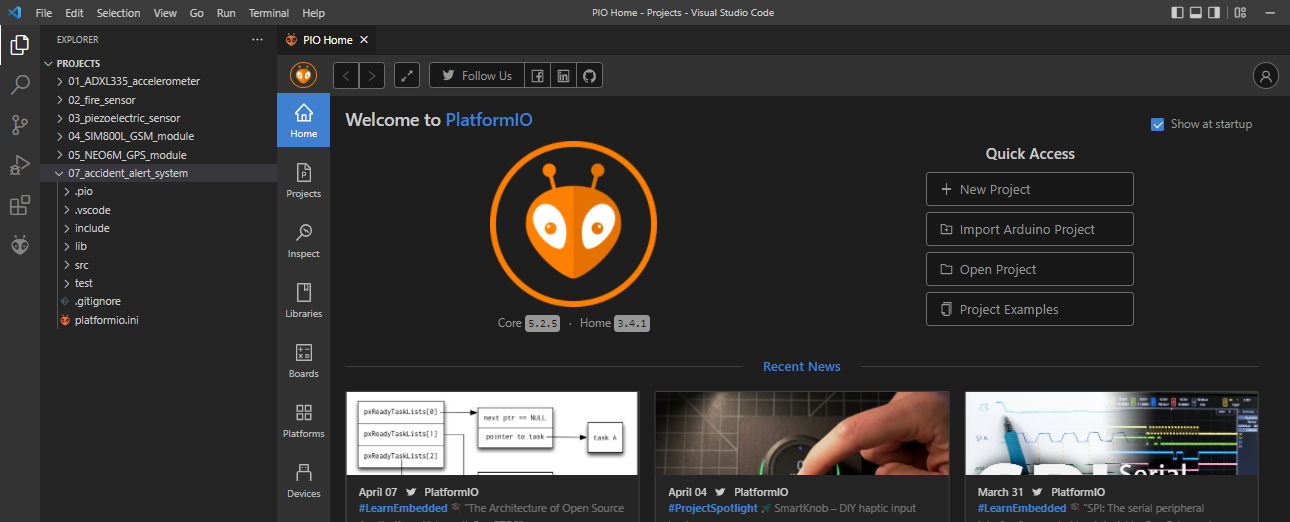


Figure: Home Page of PlatformIO IDE for VSCode

**Libraries used in this Project**

**1. TinyGPS**: TinyGPS is designed to provide most of the NMEA GPS functionality like position, date, time, altitude, speed and course – without the large size that seems to accompany similar bodies of code.  To use it, specify #include <TinyGPS.h>.

**2. EspSoftwareSerial**: EspSoftwareSerial is a library that enables serial communication with a digital pin other than the serial port. It is possible to have multiple software serial ports with speeds up to 115200bps. To use it, specify #include <SoftwareSerial.h>.

**HIGH LEVEL DESIGN OF THE SYSTEM**

**Architecture of the Proposed System**

The proposed system consists of two main parts: accident detection system and alert system. The accident detection system will constantly monitor the vehicle and detects whether the vehicle has fallen down or met with any impact or any fire accident has taken place. This module consist of three sensors namely, Piezoelectric sensor transducer, ADXL335 accelerometer, Fire sensor. Once the vehicle accident is detected, the information is send to the second part of the system. The second part consists of an ESP32 WROOM-32, GSM module, GPS module. When any of the three sensors gets activated based on the level of impact then the micro-controller fetches the location, date and time from the GPS receiver and sends an accident alert message along with the corresponding information to the contact number mentioned in the code which is dumped in the microcontroller i.e., to near and dear one’s via SMS. The high level architecture of the system is as shown below:



Figure: High level architecture of Accident detection and alerting system

**Block Diagram of the System**

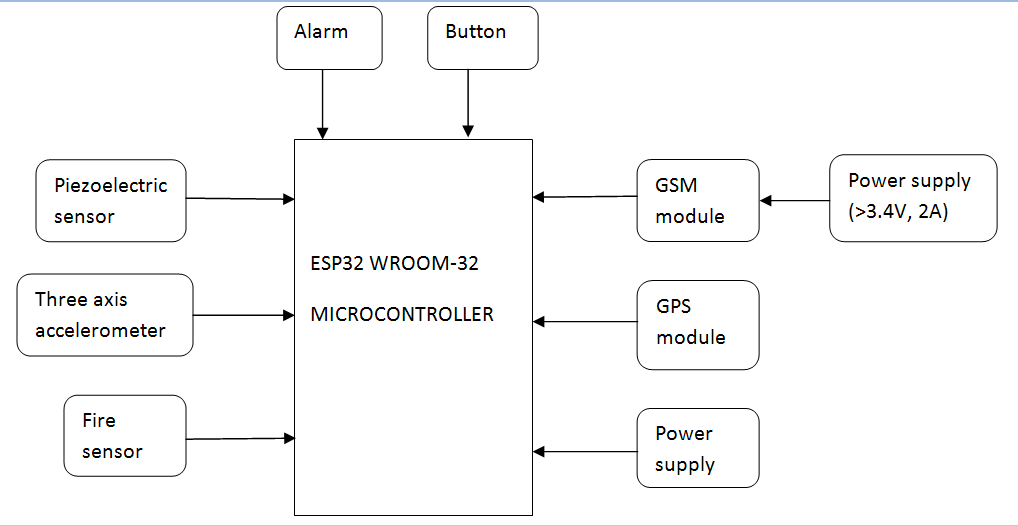


Figure: Block Diagram of the System

The operating voltage for all the components except SIM800L GSM module is 3.3V which we can provide through the board but for SIM800L GSM module operating voltage is from 3.4V to 4.4V so we have to use an external power supply for this module. We have used a 3.7V, 2A Lithium ion battery for powering SIM800L GSM module.

In this system we also have a button which we can use in case of accidents which are not serious and don’t require emergency medical aid. By pressing this button a SMS of “Accident is not serious” is sent to the contact number mentioned in the code.

**LOW LEVEL DESIGN OF THE SYSTEM**

The complete system is divided into two parts:

1. Accident detection module

2. Accident alert module

**1. Accident Detection Module**

This partof the system is used for detecting various types of accidents. It mainly consists of four components which are ADXL335 three axis accelerometer, Piezoelectric sensor, IR Fire sensor and Buzzer.

**ADXL335 accelerometer**is an analog output device which gives output in the range of 0 to 4096. It has to be connected at ADC pins of ESP32 board. The x-axis, y-axis and z-axis pins of ADXL335 are connected at GPIO13, GPIO12, GPIO14 pins of ESP32 respectively. For accident detection, the output of the ADXL335 can be calibrated with the predefined threshold values of the x-axis and y-axis. ADXL335 detects the angular motion of the vehicle, if the vehicle is fallen down or turned over due to a crash it will results in an angular motion and ADXL335 detects it as an accident.

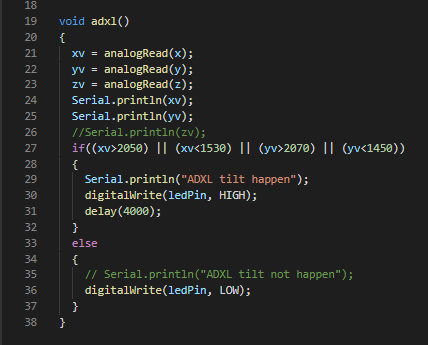


Figure: Test code for ADXL335 accelerometer

**IR Fire sensor** is a digital output device. Its D0 pin is connected to the GPIO15 pin of ESP32. Whenever it detects the fire, it sets this pin HIGH and fire accident is detected.

**Piezoelectric sensor** is also an analog output device. Its positive end is connected to the GPIO4 pin of ESP32 and negative is connected to the GND. Whenever it detects a pressure more than a threshold value (due to collision), it detects an accident.

If any of the above components detect any accident, microcontroller will activate the second part of this system.

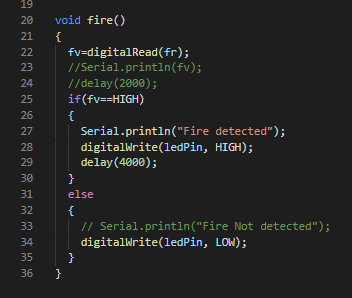


Figure: Test code for Fire sensor

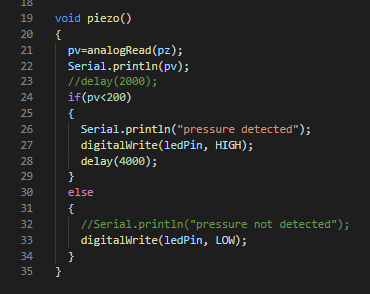


Figure: Test code for Piezoelectric sensor

A **Buzzer** is also provided to alarm the nearby people that an accident has occurred there, in case the vehicle and the victim are fallen somewhere from where they are not visible. Thus the chances of offering the help from the nearby people are also increased. Buzzer is connected to the GPIO2 pin of the ESP32.

**2. Accident Alert Module**

This part is used for alerting the near ones of the victim and notifies them about the place and time of accident. It mainly consists of a SIM800L GSM module, a NEO6M GPS module and a button.

**NEO6M GPS module** gets the location of the vehicle continuously at GPIO16 pin of ESP32. It communicates with a microcontroller over UART. We uses Espsoftwareserial library here to make this communication over UART smooth.

For our projects, we need to parse NMEA sentences into useful information.To simplify our work, we have a library called **TinyGPS library**. This library does a lot of heavy lifting required for receiving data from GPS modules, such as reading and extracting useful data in the background. We have used some functions from TinyGPS library to get the accurate location and time of the accident, they are “gps.f\_get\_position()” and “gps.crack\_datetime()”.

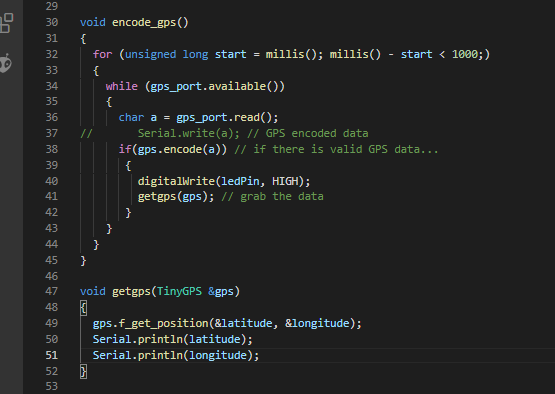


Figure: Test code for NEO6M GPS module

**SIM800L GSM module** sends an accident alert SMS to the mentioned phone number, the moment first part of this system detects an accident.Supply power for this module can be anywhere from 3.4V to 4.4 volts. Connecting it to 5V pin will likely destroy the module and it doesn’t even run on 3.3 V. So, an external power source like Li-ion battery or DC-DC buck converters rated 3.7V 2A is required.

Like other modems, the SIM800L communicates through **AT Commands**. ESP32 sends these commands serially to the SIM800L and the latter replies via the same serial port.

The first command is the initial handshake command, which is simply AT, if the SIM800L is good to go, it should reply with OK. Other AT commands we used in our project are:

“AT+ **CMGF**=(mode)”- This command is used to set the SMS mode. Either text or PDU mode can be selected by assigning 1 or 0 in the command.

“AT**+CMGS**=(serial number of message to be send)- This command is used to send a SMS message to a phone number.

Tx and Rx pins of this module are connected to GPIO3 and GPIO1 pins of ESP32 respectively.

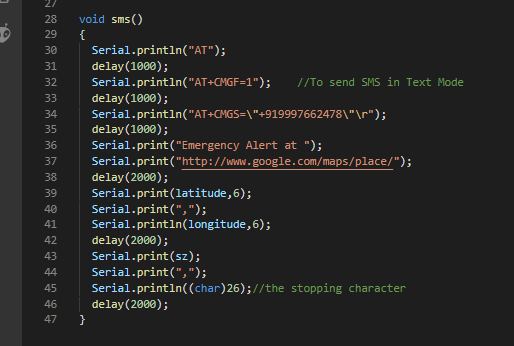


Figure: Test code for SIM800L GSM module

**Button** is used here to notify the near ones of the victim that the accident about which they just got the notification is not that serious. The decision of pressing that button is totally upon the victim, if he/she feels that they don’t need any help they can press that button. This can save the precious time of the medical rescue team. This also uses SIM800L GSM module to send the “Accident is not serious” message to the same phone number. Button is connected to the GPIO5 pin of ESP32.

**Complete Code of this Project**

#include<Arduino.h>

#include<TinyGPS.h>

#include<SoftwareSerial.h>

intx=13, y=12, z=14;

intxv=0, yv=0, zv=0;

intalarmPin = 2, ledPin = 27;

intbuttonPin = 5, bv = 0;

intfr = 15, fv = 0;

intpz = 4, pv = 0;

TinyGPSgps;

SoftwareSerialgps\_port(16,17); // Rx2 Tx2

chardata[32];

floatlatitude, longitude;

voidencode\_gps(void);

voidadxl(void);

voidfire(void);

voidpiezo(void);

voidsms(void);

voidsms\_s(void);

voidgetgps(TinyGPS&gps);

staticvoidprint\_date(TinyGPS&gps);

voidsetup()

{

 pinMode(alarmPin, OUTPUT);

 digitalWrite(alarmPin, LOW);

 pinMode(ledPin, OUTPUT);

 digitalWrite(ledPin, LOW);

 pinMode(fr, INPUT);

 digitalWrite(fr, LOW);

 pinMode(buttonPin, INPUT);

 pinMode(pz, INPUT);

 Serial.begin(9600);

 gps\_port.begin(9600);

 delay(1000);

}

voidloop()

{

  encode\_gps();

  adxl();

  fire();

  piezo();

  bv=digitalRead(buttonPin);

  if (bv == LOW) {

    sms\_s();

  }

  else {

  }

}

voidadxl()

{

  xv = analogRead(x);

  yv = analogRead(y);

  zv = analogRead(z);

  //Serial.println(xv);

  //Serial.println(yv);

  if((xv>2050) || (xv<1600) || (yv>2050) || (yv<1650))

  {

    Serial.println("ADXL tilt happen");

    digitalWrite(alarmPin, HIGH);

    sms();

  }

  else

  {

    digitalWrite(alarmPin, LOW);

  }

}

voidfire()

{

  fv=digitalRead(fr);

  if(fv==HIGH)

  {

    Serial.println("Fire detected");

    digitalWrite(alarmPin, HIGH);

    sms();

  }

  else

  {

    digitalWrite(alarmPin, LOW);

  }

}

voidpiezo()

{

  pv=analogRead(pz);

  //Serial.println(pv);

  if(pv<100)

  {

    Serial.println("Pressure detected");

    digitalWrite(alarmPin, HIGH);

    sms();

  }

  else

  {

    digitalWrite(alarmPin, LOW);

  }

}

voidsms()

{

  Serial.println("AT");

  delay(1000);

  Serial.println("AT+CMGF=1");    //To send SMS in Text Mode

  delay(1000);

  Serial.println("AT+CMGS=\"+919997662478\"\r");

  delay(1000);

  Serial.print("Accident Alert at ");

  Serial.print("http://www.google.com/maps/place/");

  delay(2000);

  Serial.print(latitude,6);

  Serial.print(",");

  Serial.println(longitude,6);

  delay(2000);

  Serial.print(data);

  Serial.println((char)26); //the stopping character

  delay(2000);

}

voidsms\_s()

{

  Serial.println("AT");

  delay(1000);

  Serial.println("AT+CMGF=1");    //To send SMS in Text Mode

  delay(1000);

  Serial.println("AT+CMGS=\"+919997662478\"\r");

  delay(1000);

  Serial.print("Nothing Serious.");

  delay(2000);

  Serial.print(data);

  Serial.println((char)26); //the stopping character

  delay(2000);

}

voidencode\_gps()

{

  for (unsignedlongstart = millis(); millis() - start<1000;)

  {

    while (gps\_port.available())

    {

      chara = gps\_port.read();

//       Serial.write(a); // GPS encoded data

      if(gps.encode(a)) // if there is valid GPS data...

       {

         digitalWrite(ledPin, HIGH);

         getgps(gps); // grab the data

       }

    }

  }

}

voidgetgps(TinyGPS&gps)

{

  gps.f\_get\_position(&latitude, &longitude);

  Serial.println(latitude);

  Serial.println(longitude);

  print\_date(gps);

}

staticvoidprint\_date(TinyGPS&gps)

{

  intyear;

  byte month, day, hour, minute, second, hundredths;

  unsignedlongage;

  gps.crack\_datetime(&year, &month, &day, &hour, &minute, &second, &hundredths, &age);

  if (age == TinyGPS::GPS\_INVALID\_AGE)

  {

  }

  else

  {

    sprintf(data, "\nDate: %02d/%02d/%02d\nTime: %02d:%02d:%02d\n", day, month, year, hour, minute, second);

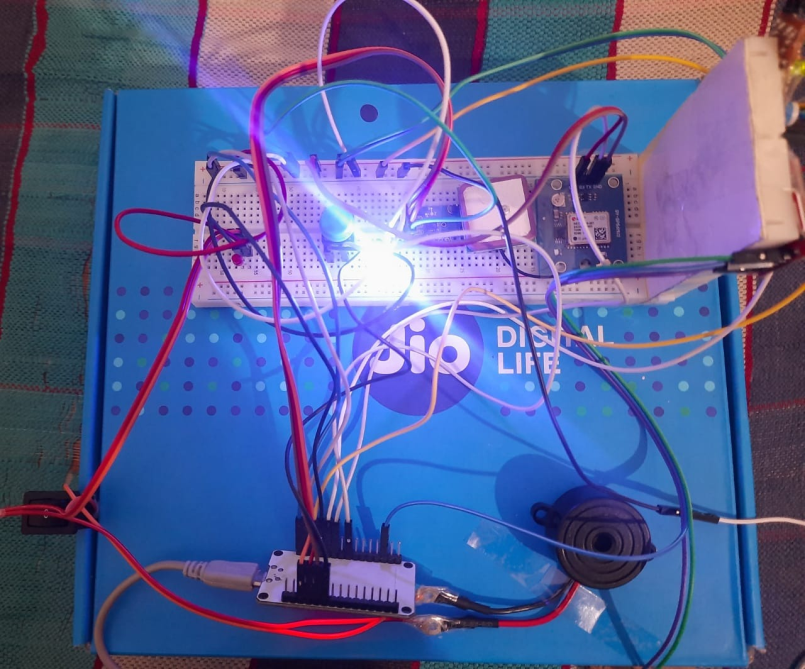
    Serial.print(data);

  }

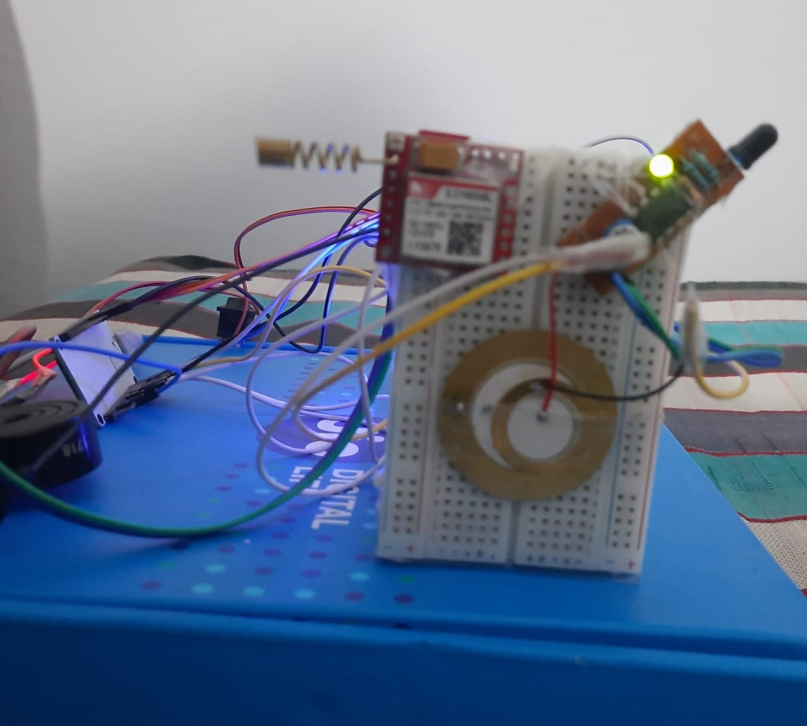
}

**RESULTS**

**Top view of the system:**

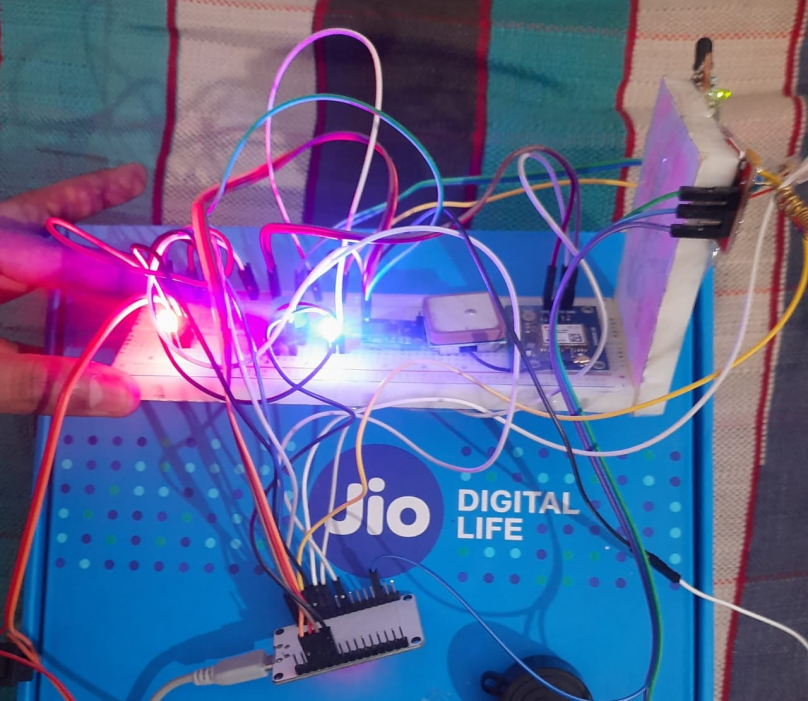


**Front view of the system:**

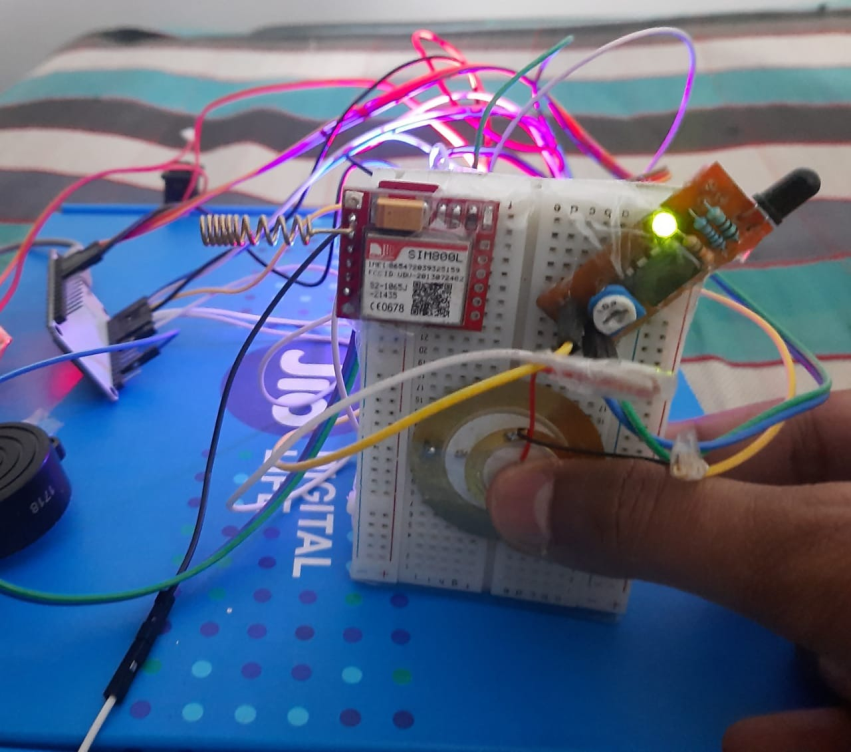


**Accident Detection (Red LED is indicating accident detection):**

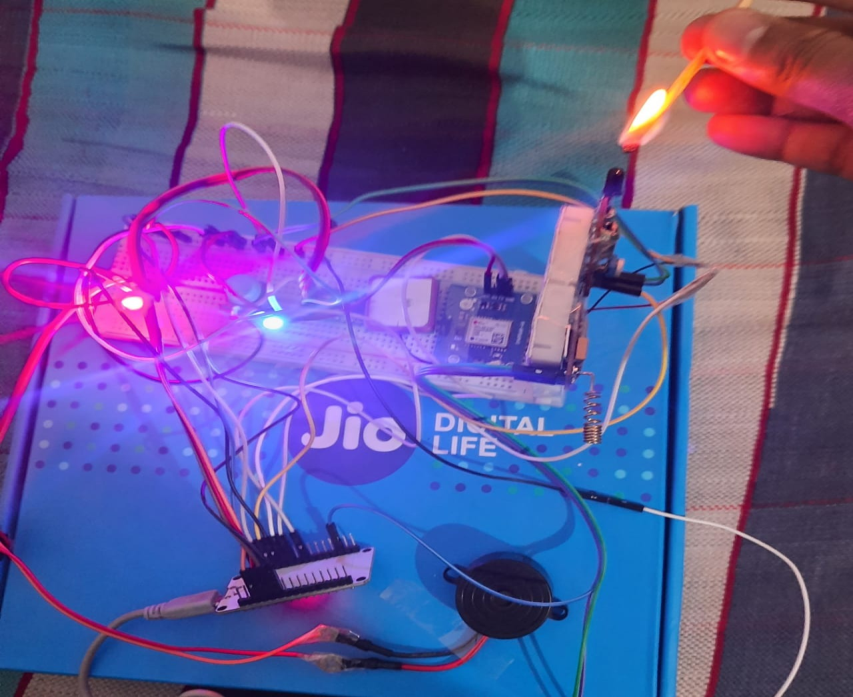
**1. Detecting vehicle turn over or vehicle fallen down types of accidents:**



**2. Detecting collision or crash types of accidents:**

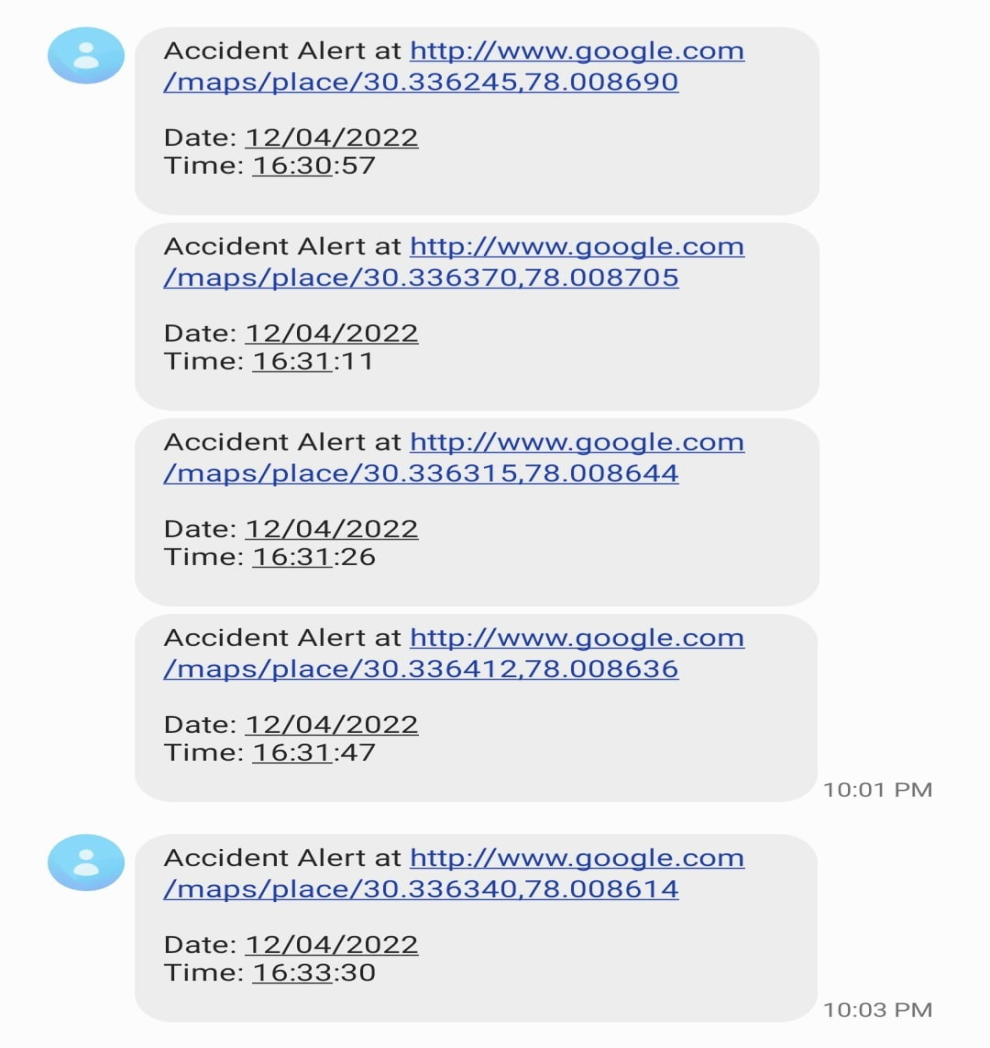


**3. Detecting fire accidents:**

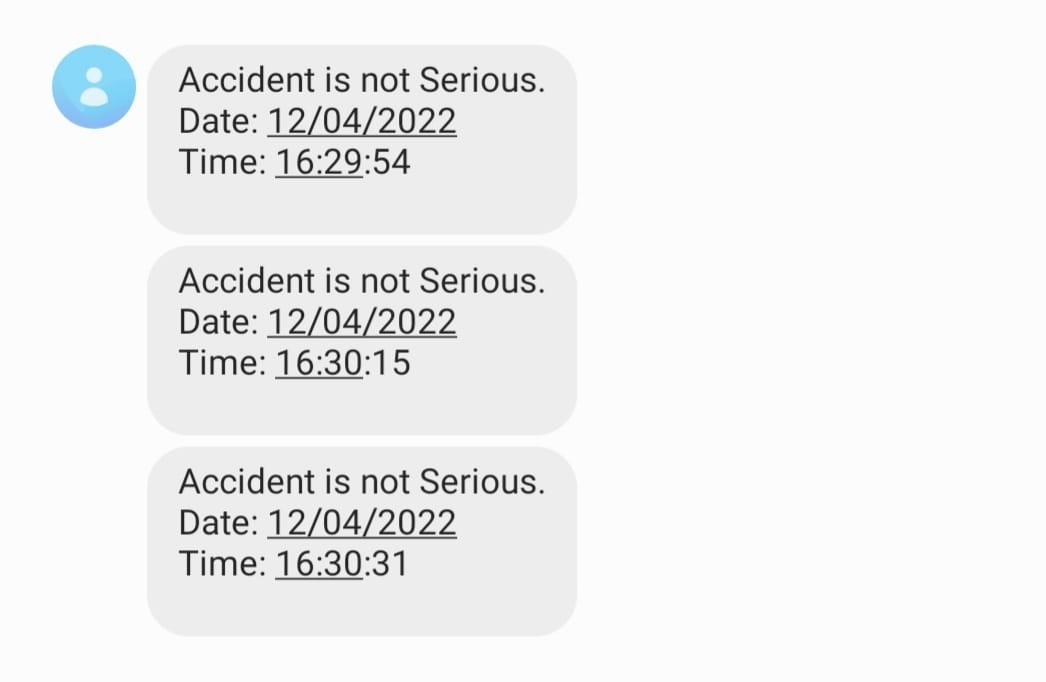


**SMS received on the mentioned phone number:**

**1. SMS received on accident detection:**

****

**2. SMS received on pressing the button:**



**CONCLUSION**

Every human life is precious and worth saving. Life should not end on road waiting for help in a crash. This projects shouts out for help where we are unable to shout for help. The system can detect the accident and then alert the victim’s near and dear ones to provide medical aid to accident victim. Piezoelectric Sensor Transducer sensor, fire sensor and MEMS are used to determine whether an accident had occurred and GPS is used to detect the location of accident. The communications between the system and the responder is done by GSM. Accident detection and alert systems are highly relevant in these days and this project aims at developing a low cost solution for the same for the benefit of the society.

**LESSONS LEARNT**

One important point we have to keep in our mind while creating this project, that the all GSM modules are power hungry devices and require sufficient power to work. If they didn’t get enough power they won’t get connected to the network. We cannot provide this much power from our board, so we have to use an external power source which could be Li-ion battery or a power adaptor with voltage step-down.

Also, most of the components we have used in our project don’t come with onboard voltage regulator (except GPS module), so we should not connect them directly to the 5v supply of our ESP32 board, it may damage them permanently.

GPS module which we have used sometimes may take several minutes (like, up to 7 or so) to get its initial fix. Make sure the antenna is out in the open with a clear view of the sky. Once it gets fixed with any of the satellite we can use it from anywhere.

**REFERENCES**

[1] Asian Journal of Applied Science and Technology (AJAST) Volume 5, Issue 2, Pages 81-89, April-June 2021, “Accident Alert and Vehicle Tracking System using GPS and GSM”.

[2] International Journal of Trend in Scientific Research and Development (IJTSRD) Volume 3 Issue 5, August 2019, “Accident Detection System using Arduino Uno”.

[3] <https://quartzcomponents.com/products>, “ADXL335 3-axis Accelerometer, IR Fire Sensor, Piezoelectric Sensor, NEO6M GPS Module, SIM800L GSM Module, ESP32 Development Board”.

[4] <https://ieeexplore.ieee.org/document/9544940>, “Vehicle Accident Detection & Alert System using IoT and Artificial Intelligence”.