KALASALINGAM
ACADEMY OF RESEARCH AND EDUCATION
(DEEMED TO BE UNIVERSITY)

Under sec. 3 of UGC Act 1956. Accredited by NAAC with "A" Grade

Department of Information Technology

Course Level Project Report

IOT-SENSORS AND DEVICES
(211ECE1400)

Team Number	**9**
Project Title	**The Automatic Vehicle Accident Detection and Messaging System**
Team Members (Reg. No & Name)	SARAVNA MUTHU PANDI- 9922008161
	SANTHOSH PANDI- 9922008160
	DHANUSH- 9922008181
	MUGENDRARAJA- 9922008128
	SATHYA SELAN - 9922008162
Faculty In Charge	DR.S.SUPRAKASH

KALASALINGAM
ACADEMY OF RESEARCH AND EDUCATION
(DEEMED TO BE UNIVERSITY)

Under sec. 3 of UGC Act 1956. Accredited by NAAC with "A" Grade

Department of Information Technology

BONAFIDE CERTIFICATE

Bonafide record of work done by _____

_____

_____

_____

of _____ Department during Odd semester in Academic Year 2023-24, as part of 211ECE1400 - IOT-Sensors and Devices Course level project.

Faculty In charge **Head of the Department**

Submitted for the evaluation of course level project of 211ECE1400 - IOT-Sensors and Devices course held on _____

Internal Examiner **External Examiner**

Table of Contents

ABSTRACT

Purpose:

The Rapid growth of technology and infrastructure has made our lives easier. The advent of technology has also increased the traffic hazards and the road accidents take place frequently which causes huge loss of life and property because of the poor emergency facilities. Our project will provide an optimum solution to this draw back. According to this project, when a vehicle meets with an accident, the Micro electro mechanical system (MEMS) sensor will detect the signal and this signal will be analyzed by Arduino. The Arduino sends the alert message through the GSM Module including the location to police control room or a rescue team. So, the police can immediately trace the location through the GPS Module, after receiving the information. Then after confirming the location necessary action will be taken. The aim of this work is to automatically detect an accident and alert the nearest hospital or medical services about the exact location of the accident.

Keywords — GSM, GPS, Arduino, sensor, MEM

.

Summary:

An automatic vehicle accident detection and messaging system is designed to detect accidents in vehicles and automatically send notifications to relevant authorities or emergency contacts. This system typically uses sensors, accelerometers, and GPS technology to monitor the vehicle's status. When it detects a sudden impact or a significant change in speed or direction, it triggers an alert. The system can then send notifications, including the vehicle's location and relevant information, to emergency services, family members, or other designated contacts. This technology aims to improve response times in emergency situations and enhance overall road safety.

INTRODUCTION

1.1 Background

Road accidents are a leading cause of injury and mortality worldwide. Swift response to accidents can significantly reduce the severity of injuries and save lives. This project addresses the need for an automated accident detection and notification system.

The purpose of this project is to create a system that can automatically detect vehicle accidents and send notifications to emergency services. This system has the potential to reduce response times, leading to better outcomes for accident victims.

1.3 Objectives

The main objectives of this project are:

Develop an accident detection system using GPS and sensors.
Implement a communication system with a GSM module for real-time messaging. Evaluate the system's effectiveness and reliability

1.4 Scope

This project focuses on the development of a proof-of-concept system. It may be further extended and integrated with existing emergency response services.

MOTIVATION:

Enhancing Safety: The primary motivation is to improve road safety by reducing response times to accidents. Rapid notification to emergency services can save lives by ensuring timely medical assistance.

Minimizing Fatalities: Traffic accidents can result in fatalities, and early detection and alerting can help minimize the severity of injuries and fatalities.

Reducing Injuries: For non-fatal accidents, early notification enables prompt medical attention, potentially reducing the severity of injuries and improving recovery outcomes.

Improving Emergency Response: These systems streamline communication with emergency services, ensuring that they have accurate information about the accident location and details, which can lead to faster response times.

Supporting Remote Areas: In remote or less populated areas, where accident detection may take longer due to fewer witnesses, these systems can be crucial in speeding up the response.

Providing Peace of Mind: For drivers and their families, knowing that such a system is in place can provide peace of mind and a sense of security.

Insurance Benefits: Some insurance companies offer discounts to drivers with these systems, which can be a motivating factor for both individuals and fleets to adopt them.

Data Collection: These systems can also collect data about driving behavior, which can be used for safety analysis and improving road infrastructure.

Legislation and Regulations: In some regions, there are regulations or incentives promoting the use of such systems, which motivates their adoption.

Overall, the automatic vehicle accident detection and messaging system aims to make our roads safer, reduce the impact of accidents, and save lives, which serves as a strong motivator for its development and adoption.

OBJECTIVE OF THE PROJECT:

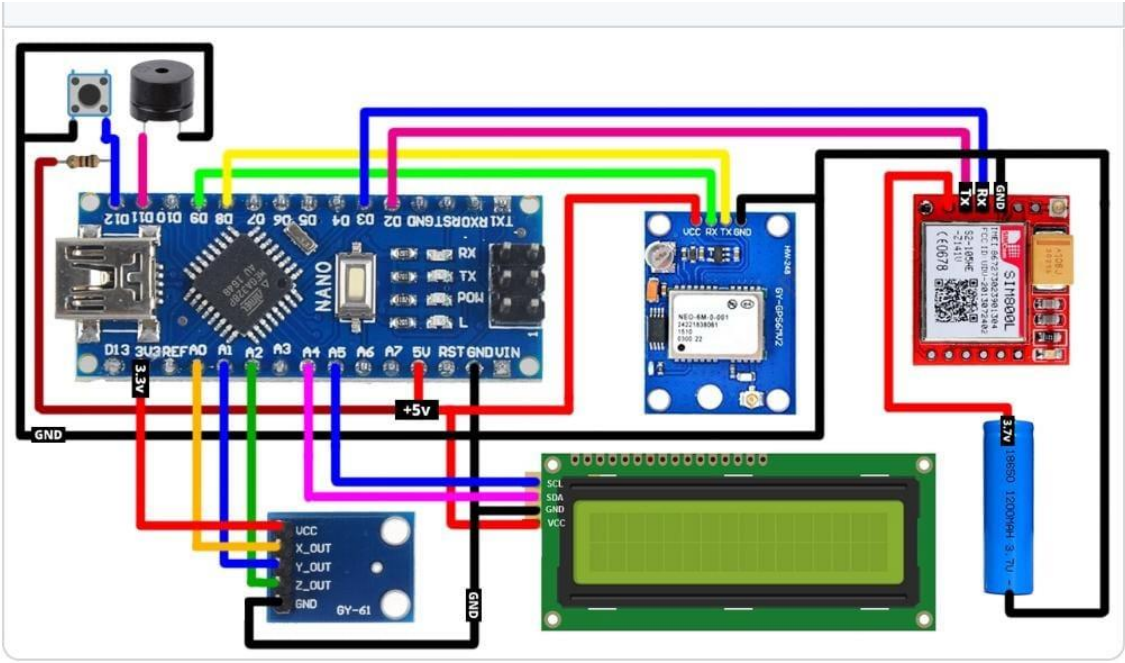
The main objectives of this project are:

. Develop an accident detection system using GPS and sensors.

. Implement a communication system with a GSM module for real-time messaging.

. Evaluate the system's effectiveness and reliability.

GENERAL BLOCK DIAGRAM



COMPONENTS:

- Arduino UNO
- GPS module
- GSM module
- 16*2 LCD Display
- ADXL335 Accelerometer

Methodology:

Accident Detection:

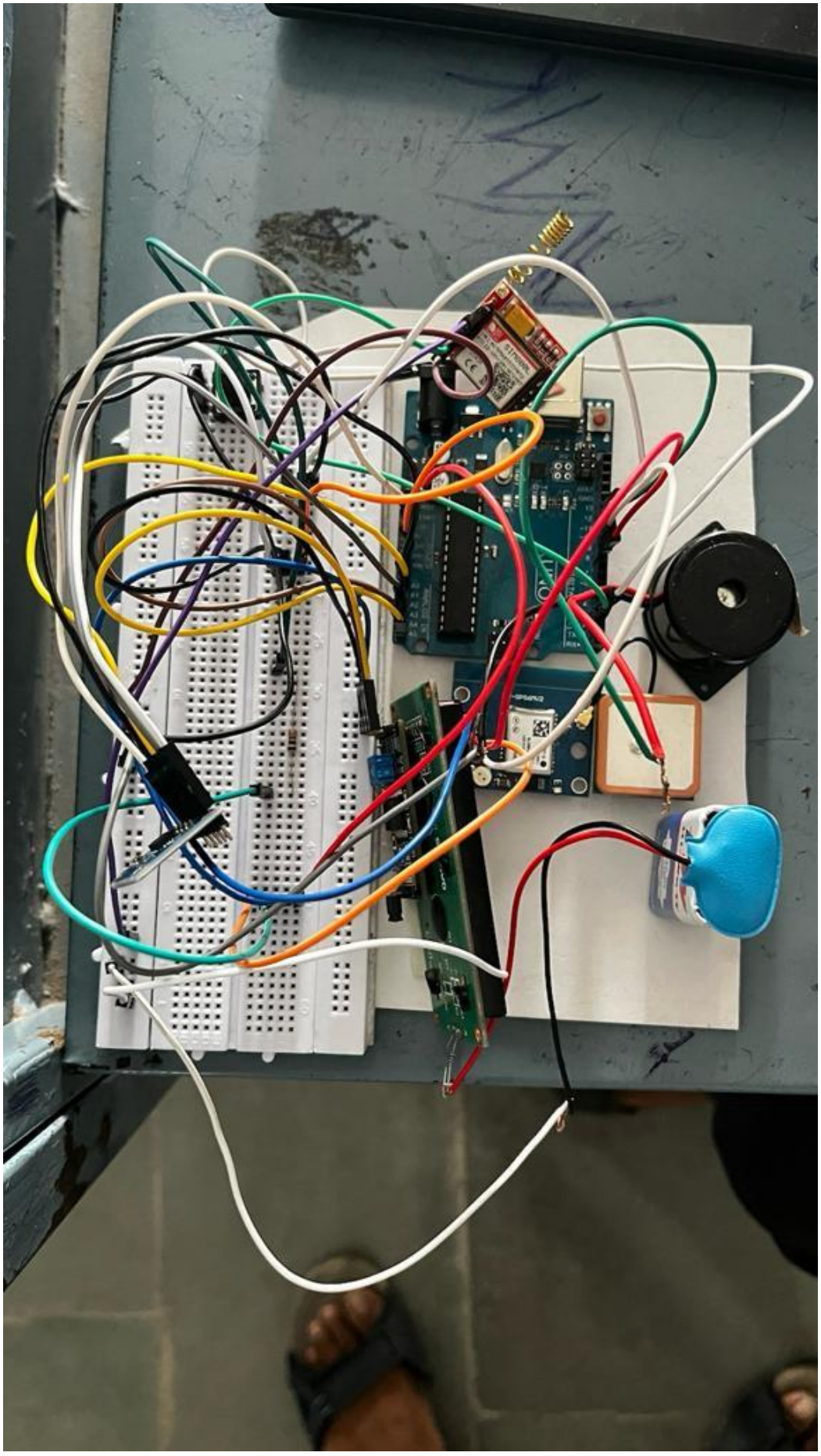
Accelerometer sensors continuously monitor vehicle motion. Sudden deceleration or abrupt changes in acceleration are indicative of an accident.
GPS coordinates are recorded at the moment of impact.

Notification:

If an accident is detected, the system sends an emergency message to a predefined phone number using the GSM module.

The message includes the accident location obtained from the GPSmodule.

IMPEMENTION:



Conclusion:

The Automatic Vehicle Accident Detection and Messaging System using GPS and GSM technology demonstrates the feasibility of a real-time accident detection and notification system. While the proof-of-concept has been successful, further refinements and integration with emergency services are necessary to achieve widespread impact.

REFERENCES

[1] M.Rajendra Prasad, P.Aswani ,"An automated traffic accident detection and alarm device", International Journal of Technological Exploration and Learning (IJTEL) Volume 1 Issue 1, August 2012.

[2] Ms. Sarika B. Kale, Gajanan P. Dhok, "Embedded system for intelligent ambulance and traffic control management International Journal of Computer and Electronics research",Volume 2, Issue 2, April 2013.

[3] Fengyuan Jia Hongyan Wang ,"A New Type of Automatic Alarming Device to Rescue Accident Injured in Time",September 2014.

[4] Sri Krishna Chaitanya Varma, Poornesh, Tarun Varma, Harsha ,"Automatic Vehicle Accident Detection and Messaging system using GPS and GSM Modems", International Journal of Scientific & Engineering Research, Volume 4, Issue 8, August-2013 ISSN 2229-5518.

[5] https://www.electronicwings.com/sensors-modules/adxl335- accelerometer-module.

[6] Tanaya Achalkar, Shrinath Panmand, Saurabh Naik, Dilip Patil, Rachna Sonkumwar "An Efficient Approach for Accident Detection System" International Journal of Engineering Trends

and Technology 67.4 (2019): 4-7.

[7] http://mtsystem.ru/sites/default/files/documents/sim800_hard ware_design_v1.09.pdf

[8] http://sensorembedded.com/product_extra_files/skg13bl.pdf

CODING:

```
#include<LiquidCrystal_I2C.h>
#include <AltSoftSerial.h>
#include <TinyGPS++.h>

#include <SoftwareSerial.h>
#include <math.h>

#include<Wire.h>
//must add i2c lcd address use i2c-scanner.ino file
LiquidCrystal_I2C lcd(0x27, 16, 2);
//-------------------------------------------------------------
//emergency phone number with country code
const String EMERGENCY_PHONE = "918056591862";
//-------------------------------------------------------------
//GSM Module RX pin to Arduino 3
//GSM Module TX pin to Arduino 2
#define rxPin 2
#define txPin 3
SoftwareSerial sim800(rxPin,txPin);
//-------------------------------------------------------------
//GPS Module RX pin to Arduino 9
//GPS Module TX pin to Arduino 8
AltSoftSerial neogps;
TinyGPSPlus gps;
//-------------------------------------------------------------
String sms_status,sender_number,received_date,msg;
String latitude, longitude;
//-------------------------------------------------------------
#define BUZZER 12
#define BUTTON 11
//-------------------------------------------------------------
```

```
#define xPin A1
#define yPin A2
#define zPin A3
//-------------------------------------------------------------

byte updateflag;

int xaxis = 0, yaxis = 0, zaxis = 0;
int deltx = 0, delty = 0, deltz = 0;
int vibration = 2, devibrate = 75;
int magnitude = 0;
int sensitivity = 20;
double angle;
boolean impact_detected = false;
//Used to run impact routine every 2mS.
unsigned long time1;
unsigned long impact_time;
unsigned long alert_delay = 30000; //30 seconds
//-------------------------------------------------------------




/*******************************
 * setup() function
 *******************************/
void setup()
{
  //-------------------------------------------------------------
  //Serial.println("Arduino serial initialize");
  Serial.begin(9600);
  //-------------------------------------------------------------
  //Serial.println("SIM800L serial initialize");
```

```
  sim800.begin(9600);
  //------------------------------------------------------------
  //Serial.println("NEO6M serial initialize");
  neogps.begin(9600);
  //------------------------------------------------------------
  pinMode(BUZZER, OUTPUT);
  pinMode(BUTTON, INPUT_PULLUP);
  //------------------------------------------------------------
  //initialize lcd screen
  lcd.begin();
  // turn on the backlight
  lcd.backlight();
  lcd.clear();
  //------------------------------------------------------------
  sms_status = "";
  sender_number="";
  received_date="";
  msg="";
  //------------------------------------------------------------
  sim800.println("AT"); //Check GSM Module
  delay(1000);
  //SendAT("AT", "OK", 2000); //Check GSM Module
  sim800.println("ATE1"); //Echo ON
  delay(1000);
  //SendAT("ATE1", "OK", 2000); //Echo ON
  sim800.println("AT+CPIN?"); //Check SIM ready
  delay(1000);
  //SendAT("AT+CPIN?", "READY", 2000); //Check SIM ready
  sim800.println("AT+CMGF=1"); //SMS text mode
  delay(1000);
  //SendAT("AT+CMGF=1", "OK", 2000); //SMS text mode
  sim800.println("AT+CNMI=1,1,0,0,0"); /// Decides how newly arrived SMS should be
handled
  delay(1000);
```

```
  //SendAT("AT+CNMI=1,1,0,0,0", "OK", 2000); //set sms received format
  //AT +CNMI = 2,1,0,0,0 - AT +CNMI = 2,2,0,0,0 (both are same)
  //-------------------------------------------------------------
  time1 = micros();
  //Serial.print("time1 = "); Serial.println(time1);
  //-------------------------------------------------------------
  //read calibrated values. otherwise false impact will trigger
  //when you reset your Arduino. (By pressing reset button)
  xaxis = analogRead(xPin);
  yaxis = analogRead(yPin);
  zaxis = analogRead(zPin);
  //-------------------------------------------------------------
}




/*******************************
 * loop() function
 *******************************/
void loop()
{
  //-------------------------------------------------------------
  //call impact routine every 2mS
  if (micros() - time1 > 1999) Impact();
  //-------------------------------------------------------------
  if(updateflag > 0)
  {
    updateflag=0;
    Serial.println("Impact detected!!");
    Serial.print("Magnitude:"); Serial.println(magnitude);
```

```
  getGps();
  digitalWrite(BUZZER, HIGH);
  impact_detected = true;
  impact_time = millis();

  lcd.clear();
  lcd.setCursor(0,0); //col=0 row=0
  lcd.print("Crash Detected");
  lcd.setCursor(0,1); //col=0 row=1
  lcd.print("Magnitude:"+String(magnitude));
}
//------------------------------------------------------------
if(impact_detected == true)
{
  if(millis() - impact_time >= alert_delay) {
    digitalWrite(BUZZER, LOW);
    makeCall();
    delay(1000);
    sendAlert();
    impact_detected = false;
    impact_time = 0;
  }
}

if(digitalRead(BUTTON) == LOW){
  delay(200);
  digitalWrite(BUZZER, LOW);
  impact_detected = false;
  impact_time = 0;
}
//------------------------------------------------------------
while(sim800.available()){
  parseData(sim800.readString());
```

```
  }
  //------------------------------------------------------------
  while(Serial.available())  {
    sim800.println(Serial.readString());
  }
  //------------------------------------------------------------


}
```

```
/*******************************
 * Impact() function
 *******************************/
void Impact()
{
  //------------------------------------------------------------
  time1 = micros(); // resets time value
  //------------------------------------------------------------
  int oldx = xaxis; //store previous axis readings for comparison
  int oldy = yaxis;
  int oldz = zaxis;

  xaxis = analogRead(xPin);
  yaxis = analogRead(yPin);
  zaxis = analogRead(zPin);

  //------------------------------------------------------------
  //loop counter prevents false triggering. Vibration resets if there is an impact. Don't detect new changes until that "time" has passed.
  vibration--;
```

```
  //Serial.print("Vibration = "); Serial.println(vibration);
  if(vibration < 0) vibration = 0;
  //Serial.println("Vibration Reset!");

  if(vibration > 0) return;
  //------------------------------------------------------------
  deltx = xaxis - oldx;
  delty = yaxis - oldy;
  deltz = zaxis - oldz;

  //Magnitude to calculate force of impact.
  magnitude = sqrt(sq(deltx) + sq(delty) + sq(deltz));

//NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
  if (magnitude >= sensitivity) //impact detected
  {
    updateflag=1;
    // reset anti-vibration counter
    vibration = devibrate;
  }

//NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
  else
  {
    //if (magnitude > 15)
      //Serial.println(magnitude);
    //reset magnitude of impact to 0
    magnitude=0;
  }

//NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
}
```

```
/******************************
 * parseData() function
 ******************************/
void parseData(String buff){
  Serial.println(buff);

  unsigned int len, index;
  //----------------------------------------------------------
  //Remove sent "AT Command" from the response string.
  index = buff.indexOf("\r");
  buff.remove(0, index+2);
  buff.trim();

//NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
  if(buff != "OK"){
    //----------------------------------------------------------
    index = buff.indexOf(":");
    String cmd = buff.substring(0, index);
    cmd.trim();

    buff.remove(0, index+2);
    //Serial.println(buff);
    //----------------------------------------------------------
    if(cmd == "+CMTI"){
      //get newly arrived memory location and store it in temp
      //temp = 4
      index = buff.indexOf(",");
      String temp = buff.substring(index+1, buff.length());
```

```
    temp = "AT+CMGR=" + temp + "\r";
    //AT+CMGR=4 i.e. get message stored at memory location 4
    sim800.println(temp);
   }
   //---------------------------------------------------------------
   else if(cmd == "+CMGR"){
    //extractSms(buff);
    //Serial.println(buff.indexOf(EMERGENCY_PHONE));
    if(buff.indexOf(918056591862) > 1){
     buff.toLowerCase();
     //Serial.println(buff.indexOf("get gps"));
     if(buff.indexOf("get gps") > 1){
      getGps();
      String sms_data;
      sms_data = "GPS Location Data\r";
      sms_data += "http://maps.google.com/maps?q=loc:";
      sms_data += latitude + "," + longitude;

      sendSms(sms_data);
     }
    }
   }
   //---------------------------------------------------------------
  }
  else{
  //The result of AT Command is "OK"
  }

//NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
}


/*****************************
```

```
 * getGps() Function
*****************************/
void getGps()
{
  // Can take up to 60 seconds
  boolean newData = false;
  for (unsigned long start = millis(); millis() - start < 2000;){
    while (neogps.available()){
      if (gps.encode(neogps.read())){
        newData = true;
        break;
      }
    }
  }

  if (newData) //If newData is true
  {
    latitude = String(gps.location.lat(), 6);
    longitude = String(gps.location.lng(), 6);
    newData = false;
  }
  else {
    Serial.println("No GPS data is available");
    latitude = "";
    longitude = "";
  }

  Serial.print("Latitude= "); Serial.println(latitude);
  Serial.print("Lngitude= "); Serial.println(longitude);
}
```

```
/*******************************
* sendAlert() function
*******************************/
void sendAlert()
{
  String sms_data;
  sms_data = "Accident Alert!!\r";
  sms_data += "http://maps.google.com/maps?q=loc:";
  sms_data += latitude + "," + longitude;

  sendSms(sms_data);
}




/*******************************
* makeCall() function
*******************************/
void makeCall()
{
  Serial.println("calling....");
  sim800.println("ATD918056591862");
  delay(20000); //20 sec delay
  sim800.println("ATH");
  delay(1000); //1 sec delay
}




/*******************************
```

```
 * sendSms() function
 *****************************/
 void sendSms(String text)
{
  //return;
  sim800.print("AT+CMGF=1\r");
  delay(1000);
  sim800.print("AT+CMGS=\"918056591862");
  delay(1000);
  sim800.print(text);
  delay(100);
  sim800.write(0x1A); //ascii code for ctrl-26 //sim800.println((char)26); //ascii code for ctrl-
26
  delay(1000);
  Serial.println("SMS Sent Successfully.");
}
```

```
/*****************************
 * SendAT() function
 *****************************/
boolean SendAT(String at_command, String expected_answer, unsigned int timeout){

    uint8_t x=0;
    boolean answer=0;
    String response;
    unsigned long previous;

    //Clean the input buffer
    while( sim800.available() > 0) sim800.read();
```

```
  sim800.println(at_command);

  x = 0;
  previous = millis();

  //this loop waits for the answer with time out
  do{
    //if there are data in the UART input buffer, reads it and checks for the asnwer
    if(sim800.available() != 0){
      response += sim800.read();
      x++;
      // check if the desired answer (OK) is in the response of the module
      if(response.indexOf(expected_answer) > 0){
        answer = 1;
        break;
      }
    }
  }while((answer == 0) && ((millis() - previous) < timeout));

 Serial.println(response);
 return answer;
}
```