19Z604 - EMBEDDED SYSTEMS

CAR ACCIDENT LOCATION TRACKING SYSTEM

PSG COLLEGE OF TECHNOLOGY

TEAM - 14

Team Members

20Z216 - Ganapathi Subramaniam K

20Z224 - Karan H

20Z230 -M Pavan Kumar Reddy

20Z238 - Praveen Kumar S

20Z246 – Sarath R A

ABSTRACT

Nowadays, road accidents are very high. On time medical aid can help in saving lives. Integrated engineering is the latest trend to solve problems. To be able to design a product using an integrated technology will be beneficial to any engineering problems and a huge contribution to the community. By eliminating the time between when an accident occurs and when the responder arrive at the scene decreases mortality rate and can save lives. One approach to eliminating the delay between accident occurrence and responder's dispatch is to use Car accident location tracking systems.

PURPOSE

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 a serious accident has been made, the system will alert the victim's loved 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 mobile phone of the accident victim.

COMPONENTS REQUIRED

- Arduino Uno
- Sim900A GSM Module
- DC 5V 2A Adaptor
- GPS module (neo-6m)
- Registered sim card
- Jumper wires
- Limit Switch

USES OF COMPONENTS

1.Limit switches: Limit switches are used to sense the physical position or limit of an object, in this application, they are used to detect if the vehicle has been struck by another vehicle or a wall. When the switch is pressed, it sends a signal to the Arduino , indicating that the drive has been charged.

2.GSM module ("sim900A "): The GSM module is used to send SMS messages with longitude and latitude values indicating the location of the vehicle to a mobile phone. The module communicates with the Arduino microcontroller via serial communication and uses a registered SIM card to connect to the mobile network.

3.GPS module("neo-6m") Ublox: The GPS module is used to display the longitude and latitude values of the vehicle's location. It communicates with the Arduino via serial communication and provides location information in a standard NMEA format.

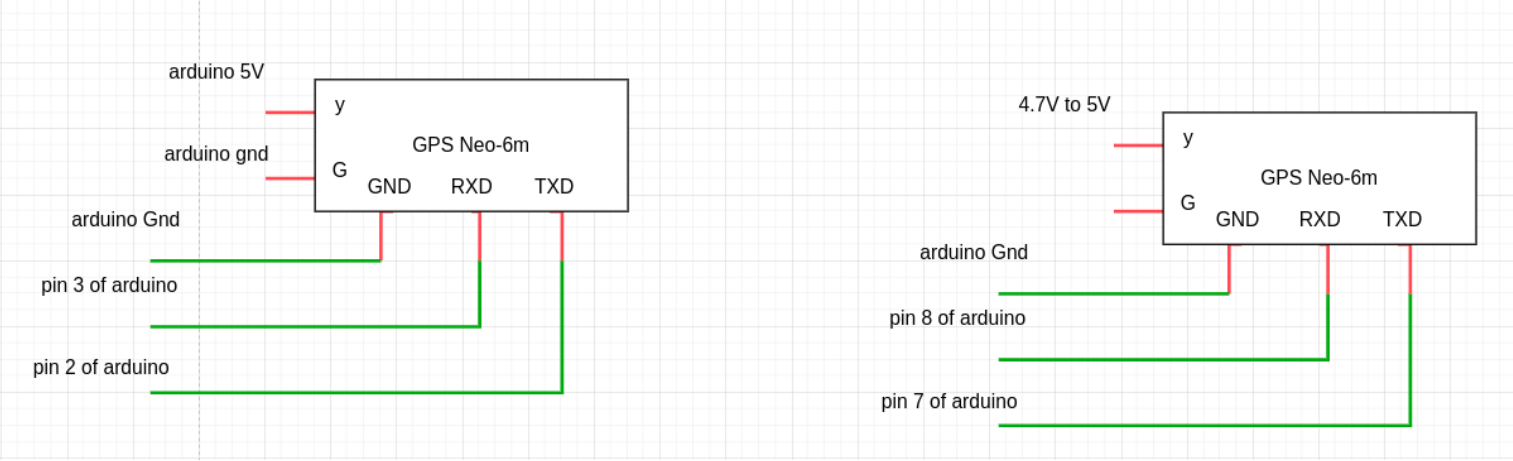
4.5v adaptor: The 5v adaptor is used to power the Arduino Uno, GPS module, and GSM module. It provides a stable power supply and ensures that the components operate correctly.

5.Jumper wires: Jumper wires are used to connect the different components together, allowing them to communicate with each other and share data.

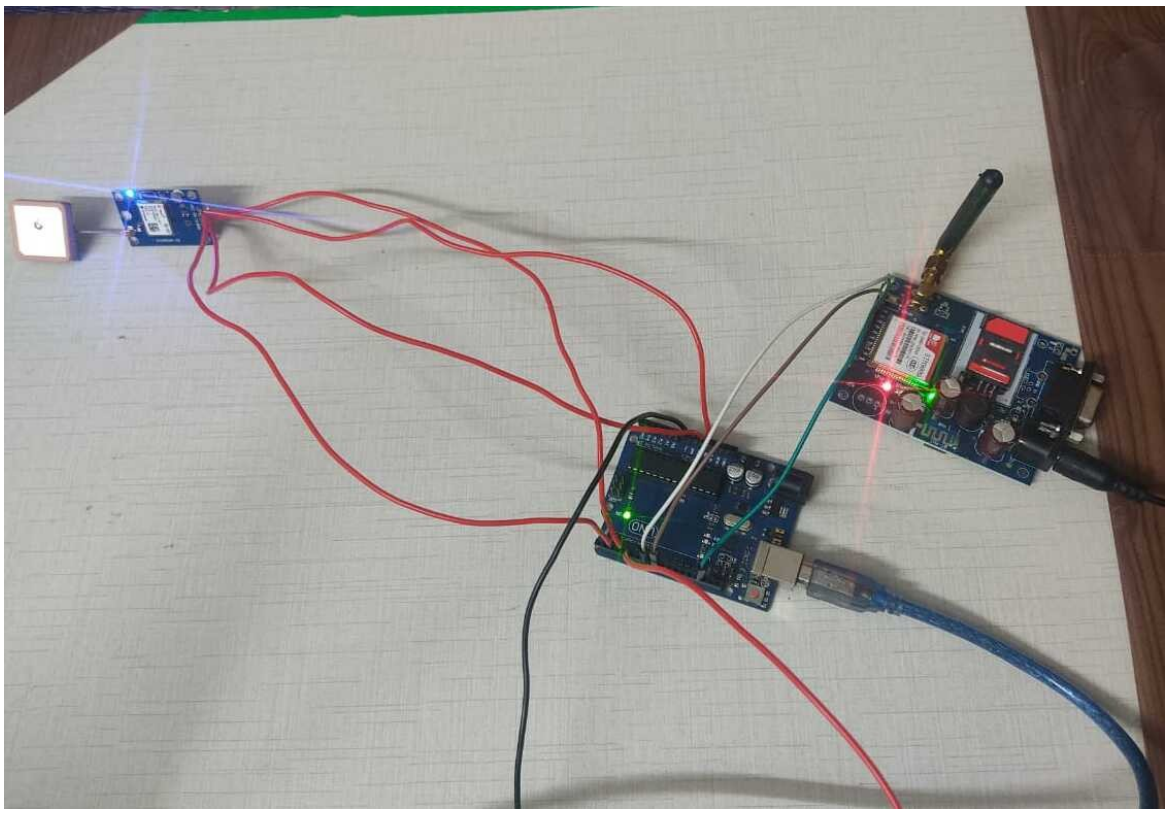
6.Registered SIM card : The registered SIM card is required to send SMS messages using the GSM module. It provides access to the mobile network and allows the module to communicate with other mobile devices.

7. Arduino Uno : Arduino microcontroller is the heart of this project which is responsible for controlling the different components and processing the data.

Circuit diagram model:



Working Circuit Images :



Working :

- When the car is moving, the GPS module continuously receives data from the satellites and calculates the longitude and latitude values of the car's location.The Arduino monitors the state of the limit switches. If a switch is pressed, it means that the car has collided with another car or a wall.When a collision is detected, the Arduino retrieves the longitude and latitude values from memory and sends them to the GSM module.
- The GSM module uses the registered SIM card to access the mobile network and sends an SMS message to a predefined phone number, containing the longitude and latitude values of the car's location.
- The recipient of the SMS message can use the longitude and latitude values to determine the location of the car and take appropriate action, such as calling emergency services or informing the car owner .The overall system continues to monitor the car's location and collision status, repeating the process if another collision is detected.

Code :

```
#include <SoftwareSerial.h>

#include <TinyGPS++.h>

static const int RXPin = 2, TXPin = 3;

static const uint32_t GPSBaud = 9600;


int  m = 9740;
```

```
int y = 71;
// The TinyGPS++ object
TinyGPSPlus gps;

// The serial connection to the GPS device
SoftwareSerial ss(RXPin, TXPin);                    // for gps

SoftwareSerial SIM900(7, 8);                        // for gsm module

int sensor = A1; // tilt sensor connected here
int led = 13; // tilt sensor output indication

String textForSMS;
 int limits = 4; // limit switch

 String datareal;
 String dataimaginary;
 String combined;
 int raw = 1000000;

 String datareal2;
 String dataimaginary2;
 String combined2;
```

```
 double longitude;
 double latitude;
void setup()
{
  SIM900.begin(19200);
  Serial.begin(9600);
  ss.begin(GPSBaud);
  delay(10000);  // give time to log on to network.
  Serial.println(" logging time completed!");
  randomSeed(analogRead(0));
  pinMode(limits, INPUT);
  digitalWrite(limits, HIGH);

  pinMode(sensor, INPUT);
  pinMode(led, OUTPUT);
  digitalWrite(led, LOW);

  //   Serial.println(F("DeviceExample.ino"));
  // Serial.println(F("A simple demonstration of TinyGPS++ with an
attached GPS module"));
  // Serial.print(F("Testing TinyGPS++ library v. "));
  Serial.print(F("Start Application "));

  Serial.println(TinyGPSPlus::libraryVersion());
```

```
  Serial.println();
}




void sendSMS(String message)
{
  SIM900.print("AT+CMGF=1\r");                    // AT command to
                                                    send SMS message
  delay(100);
 SIM900.println("AT + CMGS = \"+916383146519\"");  //
recipient's mobile number,
        In international format
  delay(100);
  SIM900.println(message);                        // message to send
  delay(100);
  SIM900.println((char)26);                       // End AT command with
                                                    a ^Z, ASCII code 26
  delay(100);
  SIM900.println();
  delay(5000);                                  // give module time to send
                                                   SMS


}


void loop()
```

```
{
  int reading;
    reading = analogRead(sensor);
// Serial.println(reading);
//  delay(1000);

 // for the tilt sensor
//**********************
//**********************
if(reading > 800)
{
   digitalWrite(led, HIGH);

 displayInfo();



latitude = gps.location.lat(), 6 ;
longitude = gps.location.lng(), 6 ;
// for latitude
long datareal = int(latitude);
int fahad = ( latitude - datareal) * 100000;

// for longitude
long datareal2 = int(longitude);
int fahad2 = (longitude - datareal2 ) * 100000;
```

```
  textForSMS = "Longitude:  ";
  textForSMS.concat(datareal2);
  textForSMS = textForSMS + ".";
  textForSMS.concat(fahad2);
  textForSMS = textForSMS + " Latitude: ";
   textForSMS.concat(datareal);
    textForSMS = textForSMS + ".";
  textForSMS.concat(fahad);
 textForSMS = textForSMS + " Message to Car";
  sendSMS(textForSMS);
  Serial.println(textForSMS);
  Serial.println("message sent.");
  delay(5000);


}


    // This sketch displays information every time a new sentence is
correctly encoded.
  while (ss.available() > 0)
    if (gps.encode(ss.read()))
      displayInfo();


  if (millis() > 5000 && gps.charsProcessed() < 10)
  {
```

```
    Serial.println(F("No GPS detected: check wiring."));
    while(true);
  }
 // for the button
//************************
//************************
  if(digitalRead(limits) == LOW)
  {
    displayInfo();
latitude = gps.location.lat(), 6 ;
longitude = gps.location.lng(), 6 ;
// for latitude
long datareal = int(latitude);
int fahad = ( latitude - datareal) * 100000;

// for longitude
long datareal2 = int(longitude);
int fahad2 = (longitude - datareal2 ) * 100000;

  textForSMS = "Longitude:  ";
  textForSMS.concat(datareal2);
  textForSMS = textForSMS + ".";
  textForSMS.concat(fahad2);
  textForSMS = textForSMS + " Latitude: ";
   textForSMS.concat(datareal);
```

```
    textForSMS = textForSMS + ".";
  textForSMS.concat(fahad);
 textForSMS = textForSMS + "  limit switch activated";
  sendSMS(textForSMS);
  Serial.println(textForSMS);
  Serial.println("message sent.");
  delay(5000);
  }
  else
  digitalWrite(limits, HIGH);
   digitalWrite(led, LOW);
}



void displayInfo()
{
  Serial.print(F("Location: "));
  if (gps.location.isValid())
  {
    Serial.print(gps.location.lat(), 6);
    Serial.print(F(","));
    Serial.print(gps.location.lng(), 6);
    Serial.print(" ");
    Serial.print(F("Speed:"));
    Serial.print(gps.speed.kmph());
```

```
}
else
{
  Serial.print(F("INVALID"));
}

Serial.print(F("  Date/Time: "));
if (gps.date.isValid())
{
  Serial.print(gps.date.month());
  Serial.print(F("/"));
  Serial.print(gps.date.day());
  Serial.print(F("/"));
  Serial.print(gps.date.year());
}
else
{
  Serial.print(F("INVALID"));
}

Serial.print(F(" "));
if (gps.time.isValid())
{
  if (gps.time.hour() < 10) Serial.print(F("0"));
  Serial.print(gps.time.hour());
```

```
    Serial.print(F(":"));
    if (gps.time.minute() < 10) Serial.print(F("0"));
    Serial.print(gps.time.minute());
    Serial.print(F(":"));
    if (gps.time.second() < 10) Serial.print(F("0"));
    Serial.print(gps.time.second());
    Serial.print(F("."));
    if (gps.time.centisecond() < 10) Serial.print(F("0"));
    Serial.print(gps.time.centisecond());
  }
  else
  {
    Serial.print(F("INVALID"));
  }
   Serial.println();
}
```

Output :