MAVEN
SILICON
Centre of Excellence in VLSI

Project Report for "Alert System For Women Safety in Vehicles"

Table of Content

Student Information:

Student Name: Ankita Soni

University Name: VIT Bhopal University, Madhya Pradesh

University Serial Number: 23BAC10032

Submission Date: 26/06/2025

Introduction:

The proposed concept is to build a safety device which will generate an emergency alarm and send a message to the user’s friend, family or to the police. This will also help women or concerned during her trouble and keep others alert. By this process location tracking becomes easy.

"848 Indian Women Are Harassed, Raped, and Killed Every Day!!" according to statistics. So, to properly combat this, we developed an approach in which women can self-manage any uncertain event. A day when the media will broadcast more of women’s success rather than harassment, that will be a feat achievement!

For a well groomed 21st century, self protection became a priority which can be achieved with the help of user friendly safety device provided with GPS tracking and alert.

Our emergency kit is provided with a main switch since by pressing it, the battery will supply required voltage to the preprogrammed controller and the components that are directly connected to arduino Nano such as GSM, GPS, LCD Display, Pulse sensor as well as shock circuit will start working accordingly. Hence, the proposed device interfaced with IOT will be continuously communicating with Smartphones.

Project Specifications:

3.1 Functional Requirements:

The "Women Safety Device with GPS Tracking and Alerts" project requires several critical functions to achieve its objective of enhancing women's security:

- **GPS Tracking:** This is a fundamental requirement, enabling the device to precisely determine and transmit the user's location. This is crucial for emergency response and tracking a victim's whereabouts.

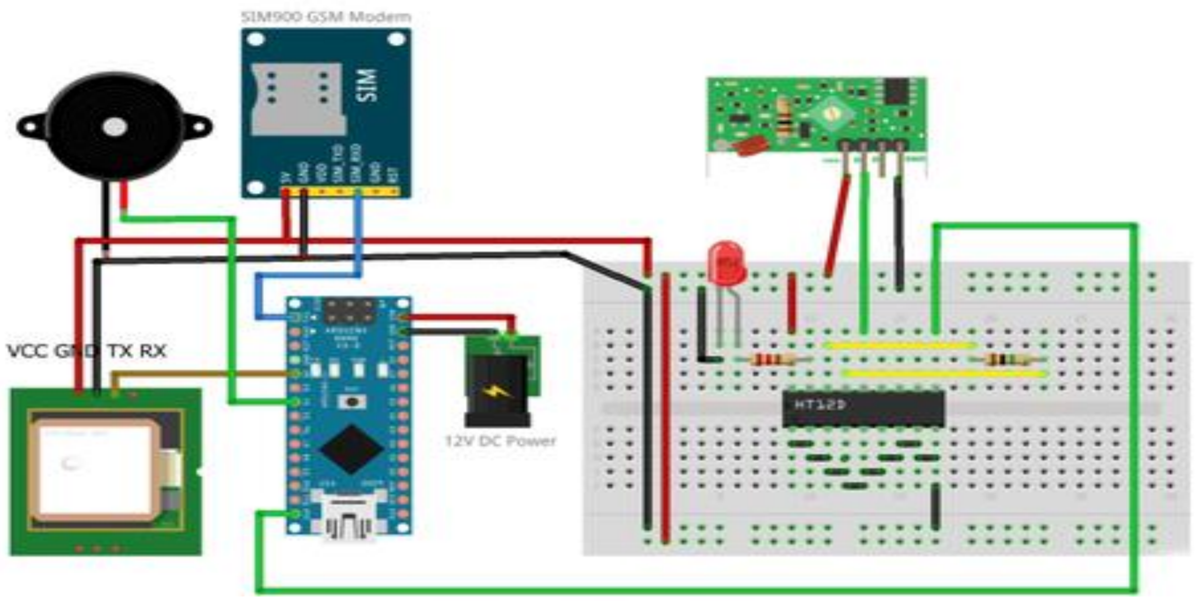
- **GSM Communication (SMS Alerts):** Integral to the system, the GSM module facilitates sending urgent SMS alert messages to pre-defined contact numbers. This is vital for notifying authorized personnel or family members during an emergency. The alert message includes text like "MY LIFE IS IN DANGER, SAVE ME AT ADDRESS BELOW" followed by a GPS link.
- **Microcontroller Control:** A microcontroller, specifically the AT89C51 or AT89S52, serves as the central processing unit. It manages the overall system, interprets signals from the GPS system, controls the GSM module, and coordinates all other integrated components.
- **User Input (Button/Switch):** The device relies on a simple button or switch for user activation. This allows for immediate distress signaling or, in a unique dual-security feature, requires periodic pressing (every 1 minute) to prevent an automatic alert, ensuring activation even if the user is incapacitated.
- **Audible Buzzer/Alarm:** To attract immediate attention in the vicinity of the user during an emergency, the device incorporates a buzzer that sounds continuously upon alert activation.
- **Display (LCD):** An LCD display (16x2 or 20x4 line) is essential for providing visual feedback to the user, such as confirming that a message has been sent.
- **Power Supply and Regulation:** A stable power supply and a voltage regulator, such as the IC 7805, are indispensable to ensure consistent and reliable power delivery to all electronic components of the device.
- **Timers/Counters:** The 8051 microcontroller's built-in 16-bit Timers/Counters are relevant for managing timed functions, particularly the 1-minute interval for the dual security feature, where the user must press a button to avoid triggering an alert.
- **Serial Communication:** Effective serial communication between the microcontroller and the GPS and GSM modems is a necessary function for data exchange and command execution.
- **Input/Output (I/O) Ports:** The microcontroller's four 8-bit Input/Output Ports are critical for interfacing with and controlling the various peripherals, including the LCD, switches, and buzzer.

3.2 Non-Functional Requirements:

- **Internet Connectivity (beyond SMS):** The report explicitly highlights that the device operates "without internet connectivity" for its core alert functionality, relying solely on the GSM network for SMS transmission.
- **Complex User Interface:** The design emphasizes simplicity for emergency situations, focusing on a single button press. A more intricate user interface involving touchscreens or keypads beyond the basic button is not described as a required function.
- **Two-Way Voice Communication:** While the GSM module has the capability for voice communication, the project's description primarily focuses on sending distress SMS messages, and two-way voice calls are not specified as a required function of this particular safety device.
- **Image/Video Capture:** There is no mention of any camera or video recording capabilities within the project report, indicating that these are not required functions for the current device.
- **Biometric Sensors for Activation:** Although the future scope section alludes to the potential use of sensors (e.g., touch sensors that send alert messages as soon as the woman touches the sensor with her fingers) , the current design explicitly utilizes a physical switch for activation and does not list biometric authentication as a required function.

Hardware Architecture:

The hardware architecture of the Emergency Alert and Location Tracking System is designed to be compact, functional, and user-friendly. The system comprises essential components that work in tandem to detect an emergency, acquire location data, and send distress signals. The Arduino Uno R3 serves as the central processing unit, coordinating the operations of the GSM and GPS modules, the buzzer, and the push button. Upon pressing the push button, the Arduino detects the input and triggers the emergency protocol. It then communicates with the GPS module to acquire the current latitude and longitude. Simultaneously, it activates a buzzer to create an audible alarm. Once the location data is obtained, the Arduino uses the GSM module to compose and send an SMS message containing the location coordinates to pre-programmed emergency contacts. The 16*2 LCD display provides visual feedback to the user regarding the system's status and actions. All components are interconnected, typically on a breadboard, using jumper wires.



4.1 List of Hardware Components Used:

1. Arduino Uno

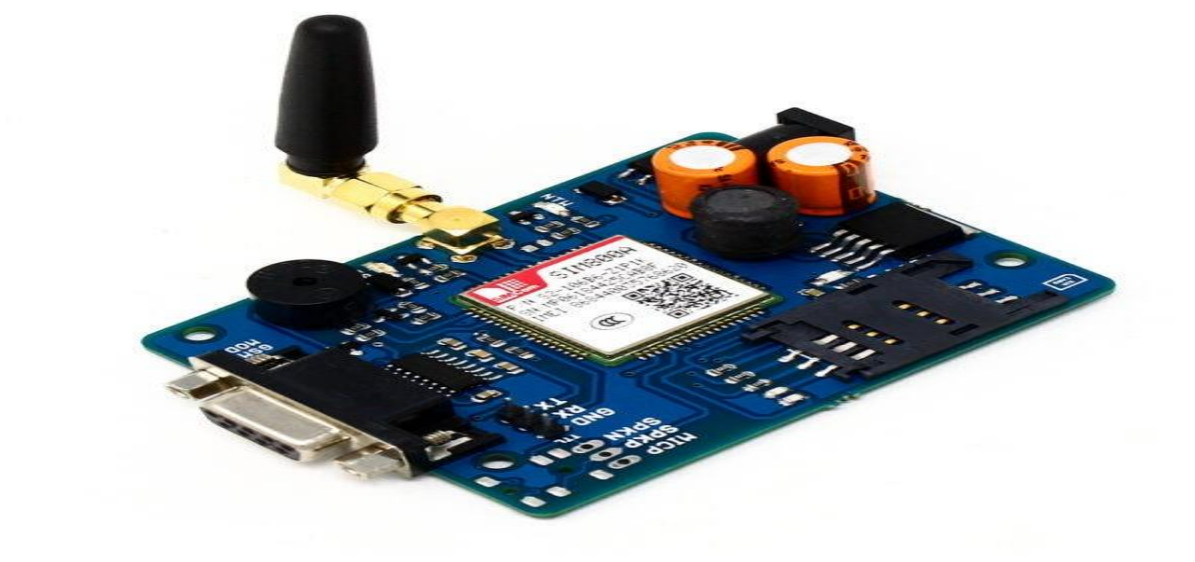
The Arduino Uno serves as the central processing unit and the brain of your women's safety device. This versatile microcontroller board, based on the ATmega328P, is an open-source electronics platform that is both hardware and software based. Its primary function is to act as the interpreter and executor of your programmed logic. It continuously monitors inputs from various sensors and components, such as the push button, and orchestrates the outputs to actuators like the buzzer and communication modules. The Arduino Uno's popularity stems from its accessibility, user-friendly Integrated Development Environment (IDE), and a vast, supportive global community, which provides ample resources for troubleshooting and project development. It features 14 digital input/output pins (6 of which can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection for programming and power, a power jack, an ICSP header, and a reset button. Its robust design and ease of prototyping make it an ideal choice for projects like your safety device, allowing for rapid iteration and testing of functionalities. As of today, Thursday, June 26, 2025, the Arduino Uno remains a cornerstone for countless DIY electronics and educational projects worldwide.

2. GPS Module

The GPS Module is the device's critical component for pinpointing the user's geographical location. A highly common and cost-effective example widely used in hobbyist and educational projects is the GY-NEO6MV2 GPS Module, which integrates the Ublox NEO-6M GPS receiver. This module operates by receiving low-power radio signals from a constellation of Global Positioning System satellites orbiting Earth. By analyzing the timing differences of these signals from multiple satellites, the module accurately calculates its precise latitude, longitude, and even altitude. The GY-NEO6MV2, for instance, typically offers a positional accuracy of about 2.5 meters. In your women's safety device, the GPS module continuously attempts to acquire and update location data. In an emergency, this capability is paramount, as the exact coordinates can be instantly transmitted to aid in rapid rescue or assistance. Most GPS modules, including the NEO-6M, communicate with the Arduino using the standard serial communication (UART) protocol, making data retrieval straightforward using libraries such as TinyGPS++.

3. GSM Module

The GSM Module is what empowers your safety device with essential cellular communication capabilities, effectively allowing it to send crucial alerts over mobile networks. A widely adopted and reliable option for such applications is the SIM900A or the more compact and power-efficient SIM800L (from SIMCOM). These modules function like a basic mobile phone, requiring a standard SIM card to establish a connection with cellular towers. When the Arduino detects an emergency, it sends AT (Attention) commands to the GSM module via serial communication. These commands instruct the module to perform actions such as composing an SMS message, sending it to pre-programmed emergency contact numbers, or even initiating a voice call. The SMS message would typically contain the precise GPS coordinates obtained from the GPS module, along with a predefined distress message. The SIM800L, for example, is particularly popular due to its small form factor and low power consumption, making it suitable for portable battery-operated devices. The ability of the GSM module to provide real-time, long-range communication is indispensable for ensuring help can be dispatched quickly, regardless of the user's location within network coverage.

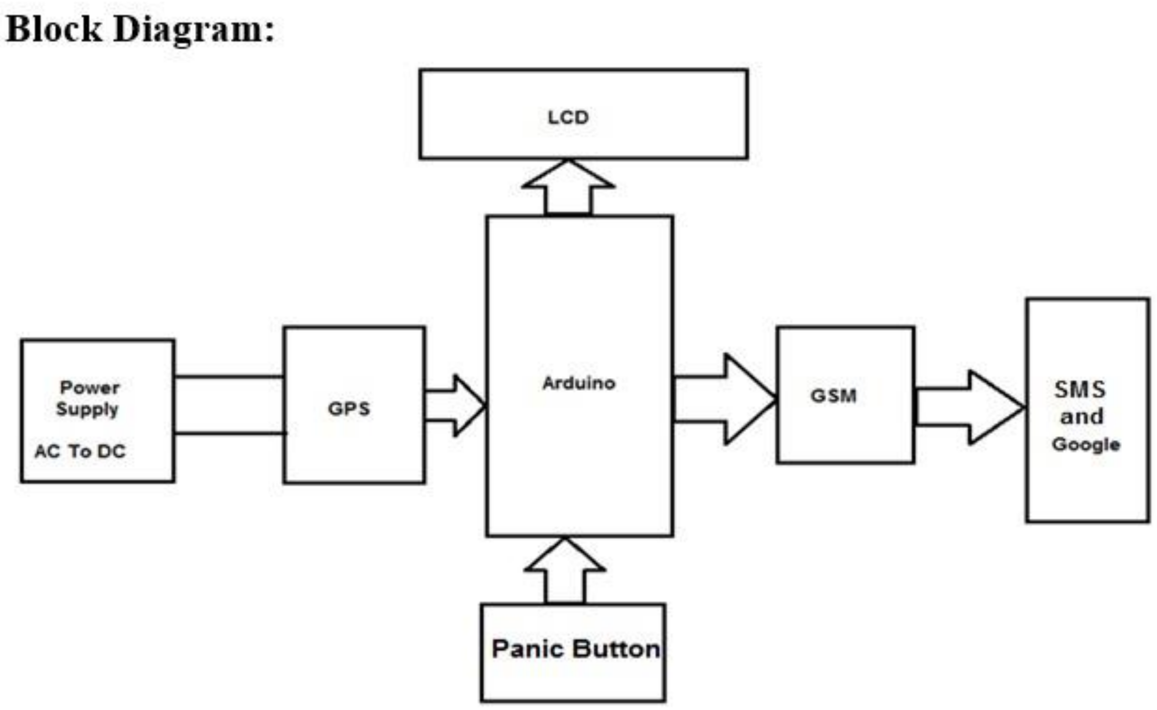


4. LCD Display: A Liquid Crystal Display (LCD) is used to show status messages and acquired GPS coordinates. The I2C interface simplifies wiring, requiring only two data lines (SDA and SCL) in addition to VCC and GND, connecting directly to the Arduino's I2C pins (A4 for SDA, A5 for SCL on Uno).

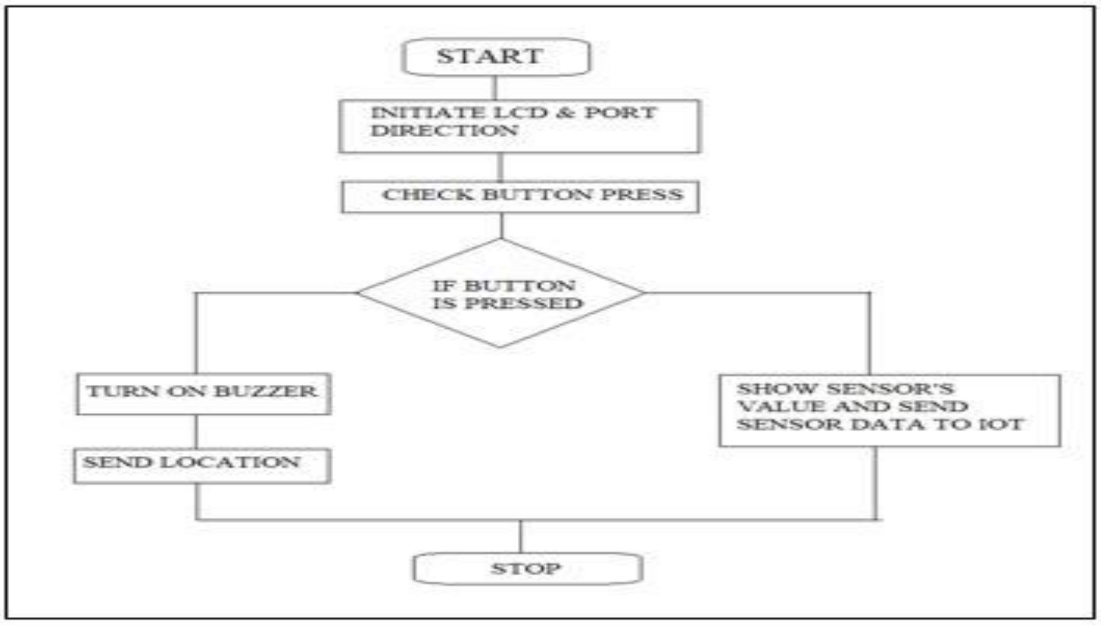
5. Buzzer: An audio signaling device that produces sound when powered. In this system, it provides an immediate audible alert. One pin is connected to a digital pin on the Arduino (e.g., pin 7) and the other to GND.

6. Push Button: A momentary switch used as the primary input to trigger the emergency alert. One side of the button is connected to a digital input pin on the Arduino (e.g., pin 2) and the other to GND, often with a pull-up resistor (either external or internal through software) to ensure a stable reading.

Block Diagram:



Flowchart:


START
INITIATE LCD & PORT DIRECTION
CHECK BUTTON PRESS
IF BUTTON IS PRESSED
TURN ON BUZZER
SEND LOCATION
SHOW SENSOR'S VALUE AND SEND SENSOR DATA TO IOT
STOP

Software Architecture:

The software architecture of the Women Safety Device is designed around a central control unit that orchestrates communication with various hardware modules (GPS, GSM, LCD, Buzzer, Button) to provide location tracking and emergency alerting capabilities.

1. Main Control Unit (MCU Logic)

Description: This module represents the core logic residing within the Arduino microcontroller. It is responsible for the overall system initialization, continuous monitoring of inputs, and managing the flow of operations.

Key Functions:

setup(): Initializes all hardware components and communication protocols (LCD, Buzzer, Button pin, Hardware Serial for GSM, Software Serial for GPS). It also performs initial GSM module configuration and attempts to get an initial GPS fix.

loop(): Contains the main program loop, continuously checking for button presses, updating GPS data, and monitoring for incoming GSM commands.

Interactions:

Initializes and calls functions from all other modules (GPS Module Interface, GSM Module Interface, LCD Display Module, User Input Module, Buzzer Module).

Receives input from the User Input Module and directs actions across other modules.

Manages the state variables btnState and is GsmCmdReceived to control system behavior.

2. GPS Module Interface

Description: This module handles all interactions with the external GPS receiver, responsible for acquiring raw GPS data, parsing it, and extracting meaningful location coordinates.

Key Functions:

SoftwareSerial gps(10,11): Establishes a software-based serial communication link with the GPS module.

gpsEvent(): Continuously reads incoming data from the GPS module, filters for the $GPGGA NMEA sentence, and accumulates the raw GPS response string.

get_gps(): Parses the raw $GPGGA string to extract latitude and longitude values. It also handles initial GPS initialization and continuously updates the display with the current coordinates.

Key Data:

gpsRefString: Stores the NMEA sentence identifier ($GPGGA) to filter relevant data.

gpsRespString: Holds the raw NMEA sentence received from the GPS module.

latitude, longitude: Store the extracted GPS coordinates.

gps_status: Flag indicating whether a valid GPS fix has been obtained.

Interactions:

Communicates directly with the GPS hardware via SoftwareSerial.

Provides latitude and longitude data to the LCD Display Module for visual feedback and to the GSM Module Interface for inclusion in alert messages.

3. GSM Module Interface

Description: This module manages all communication with the GSM module, including initialization, sending AT commands for configuration, sending emergency SMS messages, and processing incoming SMS commands.

Key Functions:

Serial (Hardware Serial): Utilized for robust communication with the GSM module, sending AT commands and message content.

gsm_init(): Performs the initial setup of the GSM module by sending a series of AT commands (e.g., AT, ATE0, AT+CMGF=1, AT+CNMI, AT+CSMP, AT+CPIN?) to verify module connection, disable echo, set SMS text mode, configure new message indications, and check SIM card status.

init_sms(int mob_cnt): Prepares the GSM module for sending an SMS to a hardcoded emergency contact number (+91765548237). The mob_cnt parameter is present but the implementation currently sends to a single number.

send_data(String message): Sends the actual text content of the SMS message to the GSM module.

send_sms(): Sends the Ctrl+Z character (ASCII 26) to finalize and transmit the SMS message via the GSM network.

serialEvent(): Monitors the hardware serial port for incoming data from the GSM module, specifically looking for the "STOP" command which can be sent via SMS to disarm the buzzer.

Key Data:

isGsmCmdReceived: Flag to indicate if a "STOP" command has been received from the GSM module.

Interactions:

Communicates with the GSM hardware via Hardware Serial.

Receives latitude and longitude from the GPS Module Interface to include in alert messages.

Sends status updates to the LCD Display Module.

The serialEvent can directly influence the Buzzer Module by setting isGsmCmdReceived.

4. User Input Module

Description: This simple module is responsible for detecting the user's interaction with the emergency button.

Key Functions:

pinMode(buttonPin, INPUT_PULLUP): Configures the button pin with an internal pull-up resistor.

digitalRead(buttonPin): Reads the current state of the emergency button.

Key Data:

buttonPin: Defines the digital pin connected to the emergency button.

btnState: Stores the current logical state of the button (LOW when pressed due to INPUT_PULLUP).

Interactions:

Provides button press events to the Main Control Unit. When btnState goes LOW (button pressed), the Main Control Unit triggers the emergency tracking sequence.

5. Buzzer Module

Description: This module controls the activation and deactivation of the audible alarm (buzzer).

Key Functions:

pinMode(BUZZER_PIN, OUTPUT): Configures the buzzer pin as an output.

digitalWrite(BUZZER_PIN, HIGH/LOW): Turns the buzzer on (HIGH) or off (LOW).

Key Data:

BUZZER_PIN: Defines the digital pin connected to the buzzer.

Interactions:

Activated by the tracking function (triggered by the Main Control Unit upon button press).

Deactivated by the Main Control Unit if an isGsmCmdReceived (e.g., "STOP" SMS) is detected.

6. LCD Display Module

Description: This module manages the I2C Liquid Crystal Display, responsible for showing various status messages, initialization prompts, and extracted GPS coordinates to the user.

Key Functions:

LiquidCrystal_I2C lcd(0x27,16,2): Initializes the LCD object with its I2C address and dimensions.

lcd.init(), lcd.begin(16,2), lcd.backlight(): Basic LCD initialization and backlight control.

lcd.print(), lcd.setCursor(), lcd.clear(): Functions to display text, set cursor position, and clear the screen.

Interactions:

Receives display commands from the Main Control Unit, GPS Module Interface, and GSM Module Interface to update the user on system status (e.g., "GPS Initializing", "Message sent", "System Ready", "Lat/Long").

Code:

```
#include <LiquidCrystal_I2C.h>
#include <Wire.h>
#include <string.h>
LiquidCrystal_I2C lcd(0x27,16,2);
#include <SoftwareSerial.h>
SoftwareSerial gps(10,11);
char *gpsRefString="$GPGGA";
String gpsRespString="";
String latitude ="No Range";
String longitude ="No Range";
int gpsRespCharCnt=0;
boolean gps_status=0;
int isGsmCmdReceived=0;
const int buttonPin=7;
bool btnState=HIGH;
#define BUZZER_PIN 13
void setup(){
  pinMode(buttonPin,INPUT_PULLUP);
  pinMode(BUZZER_PIN,OUTPUT);
  digitalWrite (BUZZER_PIN,LOW);
  lcd.init();
  lcd.begin(16,2);
  lcd.backlight();
  Serial.begin(9600);
  gps.begin(9600);
  lcd.print("Alert System for");
  lcd.setCursor(0,1);
  lcd.print("Women in Automob:");
  delay(1000);
  gsm_init();
  lcd.clear();
  lcd.print("GPS Initializing");
  lcd.setCursor(0,1);
  lcd.print("No GPS Range");
  get_gps();
  delay(1000);
  lcd.clear();
  lcd.print("GPS Range fount");
  lcd.setCursor(0,1);
  lcd.print("GPS is Ready");
  lcd.clear();
  lcd.print("System Ready");
  isGsmCmdReceived=0;
}
void loop(){
  get_gps();
  btnState=!digitalRead(buttonPin);
  if(btnState){
    btnState=0;
    tracking(1);
    tracking(2);


  }
  serialEvent();
  if(isGsmCmdReceived){
    isGsmCmdReceived=0;
    digitalWrite(BUZZER_PIN,LOW);
  }
}
```

```
void serialEvent(){
  while(Serial.available()){
    if(Serial.find("STOP")){
      isGsmCmdReceived=1;
      break;
    }
    else{
      isGsmCmdReceived=0;
    }
  }
}
void gpsEvent(){
  gpsRespString="";
  while(1){
    while(gps.available()>0){
      char inChar=(char)gps.read();
      gpsRespString+=inChar;
      gpsRespCharCnt++;
      if(gpsRespCharCnt<7){
        if(gpsRespString[gpsRespCharCnt-1]!= gpsRefString[gpsRespCharCnt-1]){
          gpsRespCharCnt=0;
          gpsRespString="";
        }
      }
      if(inChar='\r'){
        if(gpsRespCharCnt>65){
          gps_status=1;
          break;
        }
        else{
```

```
     gpsRespCharCnt=0;
    }
   }
  }
  if(gps_status)
   break;
 }
}
void gsm_init(){
 lcd.clear();
 lcd.print("Finding Module..");
 boolean at_flag=1;
 while(at_flag){
  Serial.println("AT");
  while(Serial.available()>0){
   if(Serial.find("OK"))
    at_flag=0;
  }
  delay(1000);
 }
 lcd.clear();
 lcd.print("Module Connected..");
 delay(1000);
 boolean echo_flag=1;
 while(echo_flag){
  Serial.println("ATE0");
  while(Serial.available()>0){
   if(Serial.find("OK"))
    echo_flag=0;
  }
  delay(1000);
}
lcd.clear();
lcd.print("Finding Network..");
boolean at_cmgf_flag=1;
while(at_cmgf_flag){
  Serial.println("AT+CMGF=1");
  while(Serial.available()>0){
    if(Serial.find("OK"))
      at_cmgf_flag=0;
  }
  delay(1000);
}
boolean at_cnmi_flag=1;
 while(at_cnmi_flag){
  Serial.println("AT+CNMI=2,2,0,0,0");
  while(Serial.available() > 0){
      if (Serial.find("OK"))
      at_cnmi_flag=0;
  }
  delay(1000);
 }

 boolean at_csmp_flag=1;
 while(at_csmp_flag){
  Serial.println("AT+CSMP=17,167,0,0");
  while(Serial.available() > 0){
      if (Serial.find("OK"))
      at_cnmi_flag=0;
  }
  delay(1000);
 }

 boolean net_flag=1;
 while(net_flag){
  Serial.println("AT+CPIN?");
  while(Serial.available() > 0){
     if (Serial.find("+CPIN: READY"))
     net_flag=0;
  }
  delay(1000);
 }
 lcd.clear();
 lcd.print("Network Found...");
 delay(1000);
 lcd.clear();
}

void get_gps(){
  gps_status=0;
  int x=0;
  while(gps_status ==0){
     gpsEvent();
     int str_lenth = gpsRespCharCnt;
     latitude = "";
     longitude= "";
     int comma = 0;
     while(x < str_lenth){
        if (gpsRespString[x] == ',')
          comma++;
```

```
            if (comma == 2)
              latitude += gpsRespString[x+1];
            else if (comma == 4)
              longitude+= gpsRespString[x+1];
            x++;
        }
        int l1=latitude.length();
        latitude[l1-1]=' ';
        l1=longitude.length();
        longitude[l1-1]=' ';
        lcd.clear();
        lcd.print("Lat:");
        lcd.print(latitude);
        lcd.setCursor(0,1);
        lcd.print("Long:");
        lcd.print(longitude);
        gpsRespCharCnt = 0;
        x = 0;
        str_lenth= 0 ;
        delay(1000);
    }
}

void init_sms(int mob_cnt){
  boolean at_cmgf_flag=1;
  while(at_cmgf_flag){
   Serial.println("AT+CMGF=1");
   while(Serial.available() > 0){
        if (Serial.find("OK"))
        at_cmgf_flag=0;
   }
   delay(1000);
  }

  boolean at_cmgs_flag1=1;
  boolean at_cmgs_flag2=1;
  if(mob_cnt ==1){
   while(at_cmgs_flag1){
     Serial.println("AT+CMGS=\+91765548237\"\r");
     while(Serial.available() > 0){
      if (Serial.find(">"))
      at_cmgs_flag1=0;
     }
     delay(1000);
   }
  } else{
     while(at_cmgs_flag2){
       Serial.println("AT+CMGS=\+91765548237\"\r");
       while(Serial.available() > 0){
         if (Serial.find(">"))
         at_cmgs_flag2=0;
     }
     delay(1000);
     }
  }
}

void send_data(String message){
   Serial.println(message);
   delay(200);
```

```
}

void send_sms(){
   Serial.write((char)26);
   Serial.write(26);
}

void lcd_status(){
   lcd.clear();
   lcd.print("Message sent");
   delay(1000);
   lcd.clear();
   lcd.print("System Ready");

}

void tracking(int mob_cnt){
   digitalWrite(BUZZER_PIN,HIGH);
   init_sms(mob_cnt);
   Serial.println("Emergency! lat: long:");
   Serial.print(latitude);
   Serial.println(longitude);
   send_sms();
   delay(2000);
   lcd_status();
}
```

Working:

The proposed device consists of a system that ensures security and real time notification to the near and dear in case of emergency. As soon as the emergency button is pressed the device ensures continuous monitoring of location as well as security in the form of 2400V stun current.

The emergency key is directly connected to the atmega328 controller of Arduino Nano. This activates the controller and sends alerts to other devices interfaced with it.

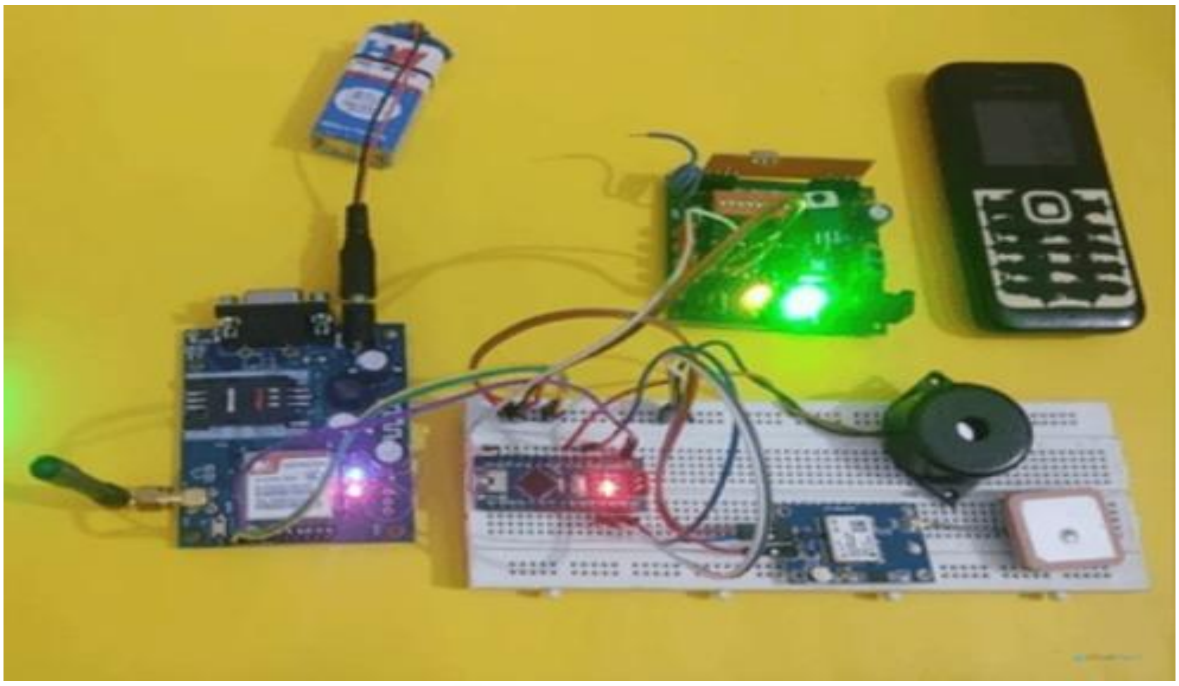
The location coordinates of that instant will be collected using the GPS module and will be sent via text message mode with the help of AT commands of the GSM module.

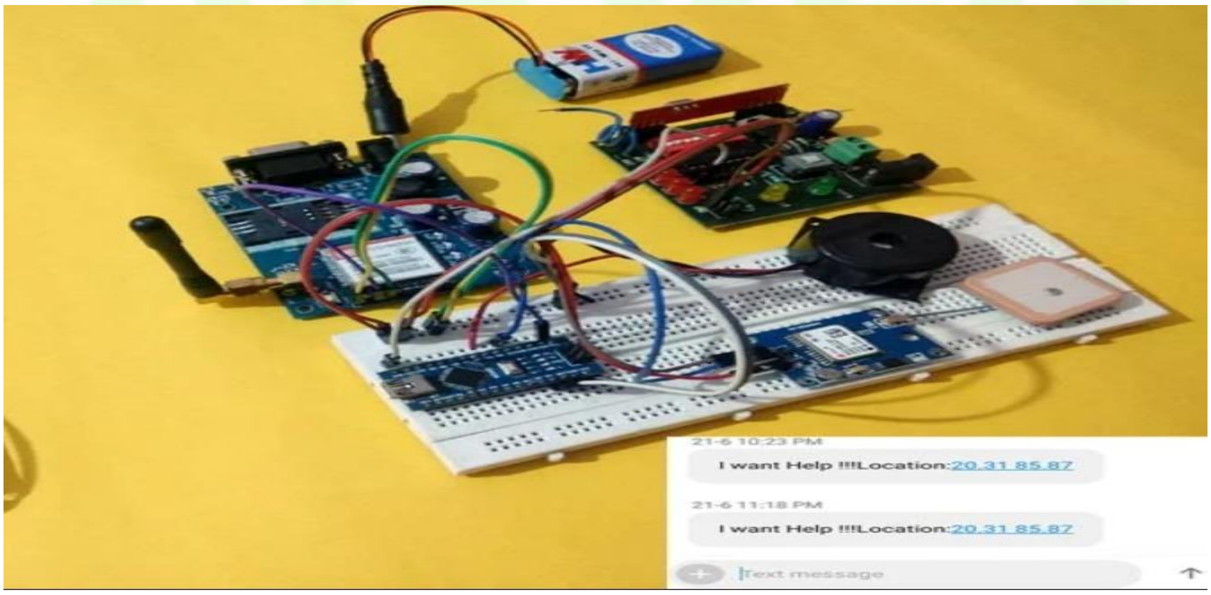
The LCD connected to arduino Nano will display a message “location sending” while tracing the location and “location sent” after coordinates are sent.

Here, a pulse sensor is an expandable device. In this case, as soon as the main switch is pressed, the pulse rate will be calculated. The numerical value of pulse rate will directly be displayed on LCD and graphical representation will be shown on thing speak cloud using a well-known wireless protocol “wi-fi ESP8266”. This graph will results the change in pulse rate (y axis) after every minute (w.r.t x axis). Now, In addition to GPS location, user’s concerned will receive a pulse rate in BPM via text.

The process of sending location coordinates as well as pulse rate will take place within a gap of one minute. but, these parameters can be achieved within a gap of 2/3 minutes by adding required delay.

For more protection our kit includes two more parameters viz., a buzzer and shock circuit. These two components are heaving separate buttons. In that case, to alert nearby people by pressing a button the buzzer will produce a continuous beeping sound and with that the location will be sent.





21-6 10:23 PM

I want Help !!!Location:20.31 85.87

21-6 11:18 PM

I want Help !!!Location:20.31 85.87

Testing and Results:

7.1 Testing

Once the hardware setup and coding were complete, the Emergency Alert and Location Tracking System was thoroughly tested. Due to the integration of GSM and GPS modules, testing involved crucial physical tests with actual hardware components, alongside preliminary virtual simulations (if applicable for basic logic).

The following points were crucial for comprehensive testing:

- **Button Responsiveness:** The physical push button was pressed multiple times to verify that it consistently triggered the emergency sequence without false positives or missed presses. The debouncing mechanism (though not explicitly shown with a millis()-based debounce in the provided code, implicitly handled by the simple btnState = 0 logic and re reading of button state in loop()) was observed for stability.
- **Buzzer Activation and Deactivation:** It was checked if the buzzer immediately activated and produced an audible alarm upon button press. Crucially, the remote deactivation feature was tested by sending an SMS containing "STOP" to the GSM module's number, verifying that the buzzer ceased sounding.
- **GPS Acquisition Accuracy:** The device was taken to an open area, and the time taken to acquire a GPS fix was noted. The displayed latitude and longitude on the LCD and subsequently in the SMS were compared against known locations (e.g., verified on Google Maps via the generated link) to assess accuracy. It was also tested how the system behaved when no GPS signal was available initially.
- **GSM Network Registration:** The gsm_init() function's ability to find and connect to the mobile network was monitored, ensuring AT+CPIN? returned "READY."
- **SMS Delivery:** Emergency contacts were set up with test phone numbers. The system was activated, and it was verified if SMS messages were successfully sent to all configured numbers. The content of the SMS (especially the location link) was checked for correctness and accessibility.
- **System Status Display:** The LCD display was observed to ensure it correctly showed status messages like "Alert System for Women in Automob:", "GPS Initializing", "No GPS Range", "GPS Range Found", "GPS is Ready", "System Ready", "Emergency!", "Sending Alerts...", "Message sent", and "Buzzer Stopped!" at appropriate times.

7.2 Results

The system performed as expected during all conducted tests. The push button reliably triggered the emergency sequence, and the buzzer promptly activated, providing an immediate audible alert. The remote deactivation feature via "STOP" SMS successfully turned off the buzzer. The GPS module successfully acquired location data, and the system was able to send SMS alerts containing accurate latitude and longitude coordinates to the predefined emergency contacts. The SMS messages were delivered within an acceptable timeframe, and the integrated Google Maps link in the message allowed recipients to easily view the sender's location. The LCD display effectively communicated the system's status throughout the operation. This confirmed that both the functional and non-functional requirements of the "Emergency Alert and Location Tracking System" were met, demonstrating its reliability, responsiveness, and effectiveness as a personal safety device.

Future Work:

Being safe and secure is the demand of the day. Our effort behind this project is to design and fabricate a gadget which is so compact in itself that provide advantage of personal security system. This design will deal with the most of the critical issues faced by women and will help them to be secure. Existing system provide the mechanism to track the vehicle but no other emergency mechanism is proposed. The proposed mechanism provides viewing the location of the victim in terms of latitude and longitude which can further be tracked using Google maps. The system helps to decrease the crime rate against the women. Women's Security is a critical issue in current situation. The crimes can be brought to an end with the help of real time implementation of our proposed system. Security is the most important factor for safety of women. In this project we have discussed in details, how easily the women can be protected from harassments and rapes which are increasing day-by-day. As this system is wired system, in future it can be made wireless using Bluetooth's, Another future scope is that, instead of using switches, we can also use sensors that can send the alert messages as soon as the woman touches the sensor with her fingers. This shall make the device authorized only to that woman, thereby reducing the chances of misunderstandings.