

IOT Based automatic accident detection, over speeding and Rescue management system

Project -1

Final Report

Submitted in Fulfillment of the Requirements for the
Award of Degree of

Bachelor in Technology



Supervised by:

Dr. Komal Khurana

Submitted by:

Abhishek 2237723
Ayush Kumar 2237727
Vaibhav 2102087

Department of Electronics & Communication
Engineering
Chandigarh Engineering College CGC
Landran, Mohali

CERTIFICATE

This is to certify that the work presented in the thesis entitled “IOT Based automatic accident detection, over speeding and Rescue management system” is a bonafide record of the work done during at Chandigarh Engineering College CGC Landran, Mohali, PUNJAB.

The project work is an authentic record of my own work and is carried out under the supervision and guidance of Guide Dr. Komal Khurana, ECE Department. The matter presented in the report has not been submitted elsewhere, wholly or in part, for the award of any other degree or diploma.

Name of student

Abhishek 2237723
Ayush Kumar 2237727
Vaibhav 2102087

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Prof. (Dr.) Komal Khurana
Department of Electronics & Communication Engineering

Prof. (Dr.) Vinay Bhatia
HOD ECE
Electronics & Communication Engineering Department
Chandigarh Engineering College, Landran, Mohali, PUNJAB

ACKNOWLEDGEMENT

I take this opportunity to express my sincere gratitude to the Principal, Chandigarh Engineering College, CGC Landran, for providing this opportunity to carry out the present thesis work. The constant guidance and encouragement received from Dr. Komal Khurana Professor and Head, Department of Electronics and Communication Engineering, has been of great help in carrying our present work and helped us in completing this project with success.

I would like to express a deep sense of gratitude to my Project Advisor Dr. Komal Khurana ECE for the guidance and support in defining the design problem and towards the completion of my project work. Without her wise counsel and able guidance, it would have been impossible to complete the thesis in this manner.

I am also thankful to all the faculty and staff members of Department of Electronics and Communication Engineering, Chandigarh Engineering College, Landran for their intellectual support throughout the course of this work.

Abhishek 2237723
Ayush Kumar 2237727
Vaibhav 2102087

CODE

```
#include <Wire.h>
#include <Adafruit_MPU6050.h>
#include <Adafruit_Sensor.h>
#include <TinyGPS++.h>
#include <SoftwareSerial.h>

const int SWT_BTN = 7;
const int BUZZ = 6;

// Creating Object of TinyGPSPlus
static const uint32_t GPSBaud = 9600;

TinyGPSPlus gps;

Adafruit_MPU6050 mpu;

// The serial connection to the GPS module
SoftwareSerial GPS_GSM(2, 3);

// ***** Defining Hardware PINS *****
const int MOTOR_L_CLK = 10;
const int MOTOR_L_ACLK = 11;

// Define your mobile number
String mobileNumber = "+918810228013"; // Replace with your mobile number

bool fastRun = false, accidentDetected = false, eventSent = false;
bool smsSent = false;
bool overSpeed_smsSent = false;

int accidentCount = 0;

//30.68754550002415, 76.66464053068741
```

```

String longTi = "", laTi = "";
String def_longTi = "76.66464053068741", def_laTi = "30.68754550002415";

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);

  GPS_GSM.begin(GPSBaud);

  pinMode(SWT_BTN, INPUT_PULLUP);

  pinMode(MOTOR_L_CLK, OUTPUT);
  pinMode(MOTOR_L_ACLK, OUTPUT);

  pinMode(BUZZ, OUTPUT);

  if (!mpu.begin()) {
    Serial.println("Failed to find MPU6050 chip");
    while (1) {
      delay(10);
    }
  }
  Serial.println("MPU6050 Found!");

  startMotor(fastRun);

}

void loop() {
  // put your main code here, to run repeatedly:
  checkButtons();
  if (!smsSent) {
    startMotor(fastRun);
  }
}

```

```

else {
    stopMotor();
}
checkAccident();
checkAccelSensor();
getGPS();
delay(300);
}

void startMotor(bool fast) {
    if (fast) {
        analogWrite(MOTOR_L_CLK, 255);
        analogWrite(MOTOR_L_ACLK, 0);
    }
    else {
        analogWrite(MOTOR_L_CLK, 100);
        analogWrite(MOTOR_L_ACLK, 0);
    }
}

void stopMotor() {
    analogWrite(MOTOR_L_CLK, 0);
    analogWrite(MOTOR_L_ACLK, 0);
}

void checkButtons() {
    Serial.print("Switch Status = ");
    Serial.println(digitalRead(SWT_BTN));
    if (digitalRead(SWT_BTN) == LOW) {
        beep();
        fastRun = !fastRun;
        if (fastRun && !smsSent) {
            if (laTi == "" && longTi == "") {
                String msg = "Over Speed at this location : 

```

```

        sendSMS(mobileNumber, msg);
    }
    else {
        String msg = "Over Speed at this location : https://maps.google.com/?q=" +
String(laTi) + "," + String(longTi);
        sendSMS(mobileNumber, msg);
    }
}
Serial.print("Speed is high : ");
Serial.println(fastRun);
delay(300); // Button debounce delay
}
}

```

```

void checkAccelSensor() {
    sensors_event_t accel_event, g, t;
    mpu.getEvent(&accel_event, &g, &t);

    float x_accel = accel_event.acceleration.x;
    float y_accel = accel_event.acceleration.y;

    unsigned long currentTime = millis();
    bool inRange = (x_accel < -6 || x_accel > 6 || y_accel < -6 || y_accel > 6);

    Serial.print("x_accel : ");
    Serial.println(x_accel);
    Serial.print("y_accel : ");
    Serial.println(y_accel);

    // If the values are within range and event has not been sent yet, send 0 for field 3 to
    ThingSpeak
    if (!inRange && !eventSent) {
        accidentDetected = false;
        eventSent = true; // Set the eventSent flag
    } else if (inRange) {

```

```

// If the values are not in range, reset the eventSent flag
eventSent = false;

// Check if the values remain unchanged for 60 seconds before sending the event
again
static bool outOfRange = false;
static unsigned long outOfRangeStartTime = 0;

// If the values were previously in range and just went out of range, record the start
time
if (!outOfRange) {
    outOfRangeStartTime = currentTime;
    outOfRange = true;
}

// If the values have been out of range for 15 seconds accident detected
if (currentTime - outOfRangeStartTime >= 10000) {
    accidentDetected = true;
    Serial.println("Accident Detected");
    outOfRange = false; // Reset the outOfRange flag
}
}
}

void checkAccident() {
    if (accidentDetected) {
        if (accidentCount < 1) {
            beep();
        }
        accidentCount++;
        if (!smsSent) {
            String url = "";
            if (laTi == "" && longTi == "") {
                url = "Accident Occured at this location : https://maps.google.com/?q=" +
String(def_laTi) + "," + String(def_longTi);

```



```

    }
    else {
        url = "Accident Occured at this location : https://maps.google.com/?q=" +
String(laTi) + "," + String(longTi);
    }
    Serial.println(url);
    delay(200);
    sendSMS(mobileNumber, url);
    smsSent = true;
}
Serial.println("Accident Detected");
stopMotor();
}
}

```

```

void getGPS() {
    // This sketch displays information every time a new sentence is correctly encoded.
    while (GPS_GSM.available() > 0) {
        gps.encode(GPS_GSM.read());
        if (gps.location.isUpdated()) {
            Serial.print("Latitude= ");
            Serial.print(gps.location.lat(), 6);
            laTi = String(gps.location.lat(), 6);
            Serial.print(" Longitude= ");
            Serial.println(gps.location.lng(), 6);
            longTi = String(gps.location.lng(), 6);
        }
    }
}

```

// Function to send SMS

```

void sendSMS(String number, String msg) {

```

```

    // AT command to set SIM800L to text mode
    GPS_GSM.println("AT+CMGF=1");

```

```
delay(100);

// AT command to set the recipient's phone number
GPS_GSM.print("AT+CMGS=\"");
GPS_GSM.print(number);
GPS_GSM.println("\");
delay(100);

// Send the message
GPS_GSM.print(msg);
delay(100);

// End the SMS and send it
GPS_GSM.write(26);
beep();
delay(1000);
}

void beep() {
  digitalWrite(BUZZ, HIGH);
  delay(500);
  digitalWrite(BUZZ, LOW);
  delay(500);
}
```