```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClient.h>
#include <SoftwareSerial.h>
#include <Wire.h>
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 32 // OLED display height, in pixels
#define OLED_RESET     -1 // Reset pin # (or -1 if sharing Arduino reset pin)
#define SCREEN_ADDRESS 0x3C ///< See datasheet for Address; 0x3D for 128x64, 0x3C for 128x32

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
const char* ssid     = "iot";
const char* password = "12345678";
const char* serverName = "http://iotcloud22.in/2953_women_safety/post_value.php";
WiFiClient client;
HTTPClient http;
int updates;
int failedUpdates;
int pos;
int stringplace = 0;
```

```
String timeUp;
String nmea[15];
String labels[12] {"Time: ", "Status: ", "Latitude: ", "Hemisphere: ", "Longitude: ", "Hemisphere: ", "Speed: ", "Track Angle: ", "Date: "};

String ch;
String lat = "13.07373";
String lon = "80.26040";
int force;
void setup() {
 pinMode(A0, INPUT);
 Serial.begin(9600);
 if (!display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS)) {
  Serial.println(F("SSD1306 allocation failed"));
  for (;;);
 }
 display.clearDisplay();
 display.display();
 WiFi.begin(ssid, password);
 Serial.println("Connecting");
 while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
 }
```

```
 Serial.println("");
 Serial.print("Connected to WiFi network with IP Address: ");
 Serial.println(WiFi.localIP());

 delay(1000);
 testdrawstyles();
}
void loop() {
 force = analogRead(A0);
 Serial.println(force);

 if (force > 330) {
  //display.clearDisplay();
  display.setTextSize(2);           // Normal 1:1 pixel scale
  display.setTextColor(SSD1306_WHITE);       // Draw white text
  // display.clearDisplay();
  display.setCursor(80, 2);
  display.println("HELP");
  display.display();
  sendsms();
  delay(100);
 }
 display.clearDisplay();
 display.setTextSize(2);           // Normal 1:1 pixel scale
 display.setTextColor(SSD1306_WHITE);       // Draw white text
```

```
  //  display.clearDisplay();
  display.setCursor(10, 2);
  display.println(F("P:"));
  display.setCursor(30, 2);
  display.println(force);
  //  display.setCursor(30, 18);
  //  display.println(F("   SYSTEM  "));
  display.display();
  delay(500);
  sending_to_db();
}
void testdrawstyles(void) {
  display.clearDisplay();
  display.setTextSize(1);            // Normal 1:1 pixel scale
  display.setTextColor(SSD1306_WHITE);       // Draw white text
  display.clearDisplay();
  display.setCursor(10, 2);
  display.println(F("  WOMEN   SAFETY"));
  display.setCursor(30, 18);
  display.println(F("   SYSTEM  "));
  display.display();
  delay(4000);
}
void gps()
{
```

```
//  Serial.flush();
Serial.read();
if (Serial.find("$GPRMC,")) {
  String Msg = Serial.readStringUntil('\n');
  Serial.println(Msg);
  for (int i = 0; i < Msg.length(); i++) {
    if (Msg.substring(i, i + 1) == ",") {
      nmea[pos] = Msg.substring(stringplace, i);
      stringplace = i + 1;
      pos++;
    }
    if (i == Msg.length() - 1) {
      nmea[pos] = Msg.substring(stringplace, i);
    }
  }
  updates++;
  nmea[2] = ConvertLat();
  nmea[4] = ConvertLng();
  //for (int i = 0; i < 9; i++) {
  /*Serial.print(labels[0]);
    Serial.print(nmea[0]);
    Serial.print(labels[8]);
    Serial.println(nmea[8]);*/
  Serial.print("https://maps.google.com/maps?f=q&q=");
  Serial.print(nmea[2]);
```

```
  Serial.print(",");
  Serial.println(nmea[4]);

  int lat1 = nmea[2].toInt();
  if (lat1 > 0) {
    Serial.println("new data");
    lat = nmea[2];
    lon = nmea[4];
  }
  else {
    Serial.println("old data");

  }
  Serial.println("");
  //}

 }
 else {

  failedUpdates++;

 }
 stringplace = 0;
 pos = 0;
}
```

```
String ConvertLat() {
 String posneg = "";
 if (nmea[3] == "S") {
  posneg = "-";
 }
 String latfirst;
 float latsecond;
 for (int i = 0; i < nmea[2].length(); i++) {
  if (nmea[2].substring(i, i + 1) == ".") {
   latfirst = nmea[2].substring(0, i - 2);
   latsecond = nmea[2].substring(i - 2).toFloat();
  }
 }
 latsecond = latsecond / 60;
 String CalcLat = "";

 char charVal[9];
 dtostrf(latsecond, 4, 6, charVal);
 for (int i = 0; i < sizeof(charVal); i++)
 {
  CalcLat += charVal[i];
 }
 latfirst += CalcLat.substring(1);
 latfirst = posneg += latfirst;
```

```
 return latfirst;
}

String ConvertLng()
{
 String posneg = "";
 if (nmea[5] == "W") {
  posneg = "-";
 }
 String lngfirst;
 float lngsecond;
 for (int i = 0; i < nmea[4].length(); i++) {
  if (nmea[4].substring(i, i + 1) == ".") {
   lngfirst = nmea[4].substring(0, i - 2);
   //Serial.println(lngfirst);
   lngsecond = nmea[4].substring(i - 2).toFloat();
   //Serial.println(lngsecond);
  }
 }
 lngsecond = lngsecond / 60;
 String CalcLng = "";
 char charVal[9];
 dtostrf(lngsecond, 4, 6, charVal);
 for (int i = 0; i < sizeof(charVal); i++)
 {
```

```
  CalcLng += charVal[i];
 }
 lngfirst += CalcLng.substring(1);
 lngfirst = posneg += lngfirst;
 return lngfirst;
}

void sending_to_db()
{
 if (WiFi.status() == WL_CONNECTED)
 {

  http.begin(client, serverName);
  http.addHeader("Content-Type", "application/x-www-form-urlencoded");

  String httpRequestData = "&value1=" + String(force) + "&value2=" + String(lat) + "&value3=" + String(lon) +   "";
  //    Serial.print("httpRequestData: ");
  //    Serial.println(httpRequestData);
  int httpResponseCode = http.POST(httpRequestData);
  if (httpResponseCode > 0) {
   Serial.print("HTTP Response code: ");
   Serial.println(httpResponseCode);
  }
  else {
   Serial.print("Error code: ");
```

```
    Serial.println(httpResponseCode);
   }
   http.end();
  }
  else {
   Serial.println("WiFi Disconnected");
  }

  delay(500);



}
void sendsms()
{
  gps();
  +
  Serial.println("AT\r");
  delay(1000);
  Serial.println("AT+CMGF=1\r");
  delay(1000);
  Serial.println("AT+CMGS=\"+919344594021\"\r");
  delay(1000);
  Serial.println("***SOS EMERGENCY***");
  Serial.println("please help");
  delay(3000);
```

```
 Serial.println((char)26);

 delay(2000);

}
```