

**TEAM NO:** 19

**TEAM MEMBERS NAMES:**

1. R. KISHORE
2. KISHORE.R
3. MOULIESWARAN.A
4. SREE BALAJI.P

**COLLEGE:** SSN COLLEGE OF ENGINEERING

**HACKATHON:** LIONS CIRCUIT & PCB CUPID

---

## **SILENT GUARDIAN**

**ABSTRACT:**

In today's rapidly changing world, personal safety is a top concern, especially for women, children, elderly individuals, and solo workers. We the team 19 presents the "SilentGuardian" a compact IoT-based emergency alert system built using the ESP32-C3 (Glyph C3). This smart device instantly sends an SOS alert to a Telegram group and shares the user's real-time location via a Google Maps link when a panic button is pressed or tilt the device.

Designed to be low-cost, portable and this system is ideal for wearable or embedded safety devices. It can be extended with features like audio recording, touch detection and GPS modules for enhanced protection.

**OBJECTIVE:**

Silent Guardian is a smart IoT safety device designed to help women alert trusted contacts instantly in emergencies. With one-button press or motion-based activation, it sends live location and SOS messages over the internet, ensuring fast help when it matters most.

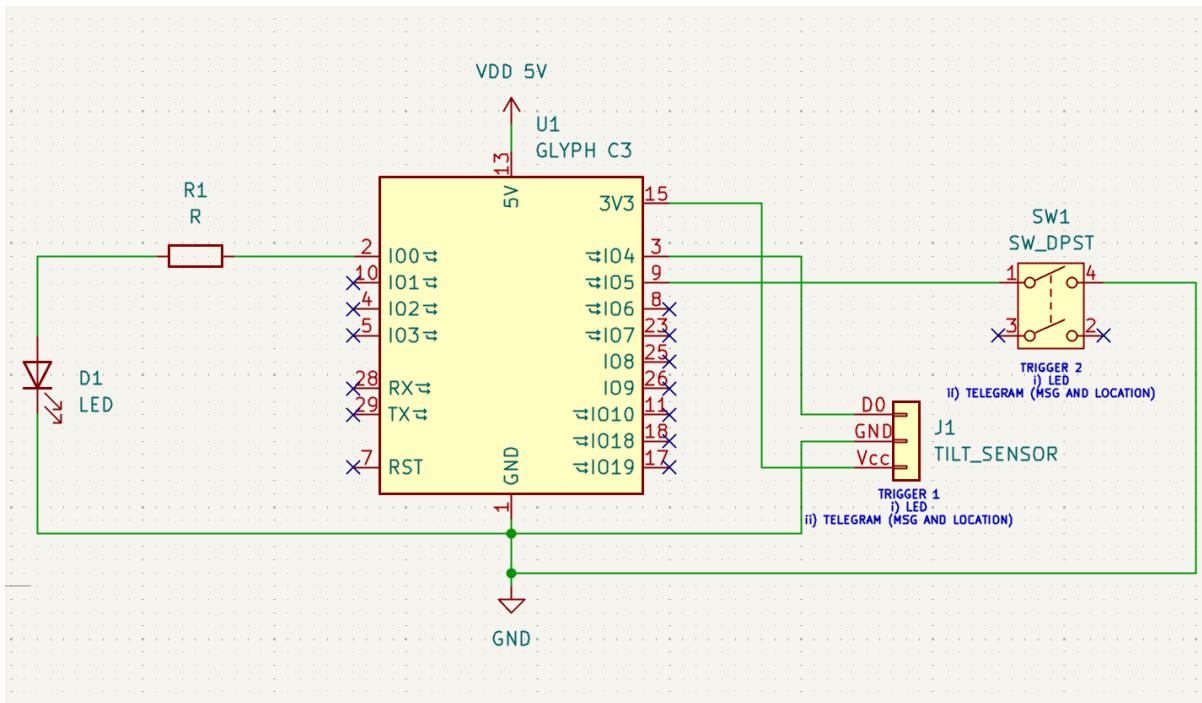
**COMPONENTS USED:**

1. Glyph C3(headers not soldered)
2. TS04-66-65-BK-160-SMT Tactile Switch (Push button small)
3. SW420 Vibration Sensor
4. Jumper wires (Male-Female)
5. Breadboard
6. Resistors
7. 5mm Red LED
8. USB Type -C Cable

## PIN CONFIGURATION:

SI. NO	COMPONENT	GPIO PIN	FUNCTION
1.	Push Button	GPIO 8	Emergency Button
2.	Vibration Sensor (SW-420)	GPIO 3	Vibration Detection
3.	Led	GPIO 2	Warning Sign
4.	Input (Sensor)	All Components	3V3 Pin
5.	Ground	All Components	GND Pin

## CIRCUIT DESIGN/ SCHEMATIC DIAGRAM:



## CODE:

```
#include <WiFi.h>
#include <HTTPClient.h>

// === Wi-Fi and Telegram Setup ===
const char* ssid = "Moulieswaran";
```

```
const char* password = "test123456";

String botToken = "8317752324:AAH3utyb5Nng9jxogF6S6D2kXpq0Am1HntA"; // Replace with your bot token

String chatID = "-1002851840477"; // Replace with your chat ID

// === Pins ===

#define TILT_PIN      3      // Tilt sensor
#define BUTTON_PIN    8      // Push button
#define LED_PIN       2      // LED

// === Logic Control ===

bool ledState = false;
bool lastButtonState = HIGH;
unsigned long lastDebounceTime = 0;
const unsigned long debounceDelay = 200;

void setup() {
    Serial.begin(115200);

    pinMode(TILT_PIN, INPUT);           // Tilt input
    pinMode(BUTTON_PIN, INPUT_PULLUP); // Button input
    pinMode(LED_PIN, OUTPUT);         // LED output
    digitalWrite(LED_PIN, LOW);

    // Connect to Wi-Fi
    Serial.print("Connecting to WiFi...");
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("\nConnected to WiFi!");
    Serial.println("System Ready");
}
```

```
void loop() {  
  
    bool currentButtonState = digitalRead(BUTTON_PIN);  
  
    bool tiltState = digitalRead(TILT_PIN);  
  
    unsigned long currentTime = millis();  
  
  
    // --- LED is OFF: Turn ON with Tilt or Button ---  
  
    if (!ledState) {  
  
        if (tiltState == LOW) {  
  
            Serial.println("SHE IS IN DANGER!!! RESCUE!!!!");  
  
            ledState = true;  
  
            digitalWrite(LED_PIN, HIGH);  
  
            sendTelegramMessage("⚠ Alert: SHE IS IN DANGER!!! RESCUE!!!!");  
  
            delay(1000); // Short delay to ensure requests don't collide  
  
            sendLocation(12.9716, 77.5946); // Replace with real coordinates if  
needed  
  
        }  
  
        else if (lastButtonState == HIGH && currentButtonState == LOW &&  
                (currentTime - lastDebounceTime > debounceDelay)) {  
  
            Serial.println("Button pressed → LED ON");  
  
            ledState = true;  
  
            digitalWrite(LED_PIN, HIGH);  
  
            lastDebounceTime = currentTime;  
  
            sendTelegramMessage("⚠ Alert: SHE IS IN DANGER!!! RESCUE!!!!");  
  
        }  
  
    }  
  
  
    // --- LED is ON: Turn OFF only with button ---  
  
    else {  
  
        if (lastButtonState == HIGH && currentButtonState == LOW &&  
            (currentTime - lastDebounceTime > debounceDelay)) {  
  
            Serial.println("Button pressed → LED OFF");  
  
            ledState = false;  
  
            digitalWrite(LED_PIN, LOW);  
  
            lastDebounceTime = currentTime;  
  
        }  
  
    }  
  
}
```

```
    sendTelegramMessage("✅ Info: SHE IS SAFE NOW.");

}

}

lastButtonState = currentButtonState;
}

// === Send Message to Telegram ===

void sendTelegramMessage(String message) {

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

HTTPClient http;

String url = "https://api.telegram.org/bot" + botToken +
    "/sendMessage?chat_id=" + chatID +
    "&text=" + message + "&parse_mode=Markdown";

http.begin(url);

int httpResponseCode = http.GET();

if (httpResponseCode > 0) {

Serial.println("Message sent!");

} else {

Serial.print("Telegram Error: ");
Serial.println(httpResponseCode);

}

http.end();

} else {

Serial.println("WiFi not connected. Cannot send message.");
}

}

// === Send Location to Telegram ===

void sendLocation(float latitude, float longitude) {

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

HTTPClient http;
```

```

String url = "https://api.telegram.org/bot" + botToken +
    "/sendLocation?chat_id=" + chatID +
    "&latitude=" + String(latitude, 6) +
    "&longitude=" + String(longitude, 6);

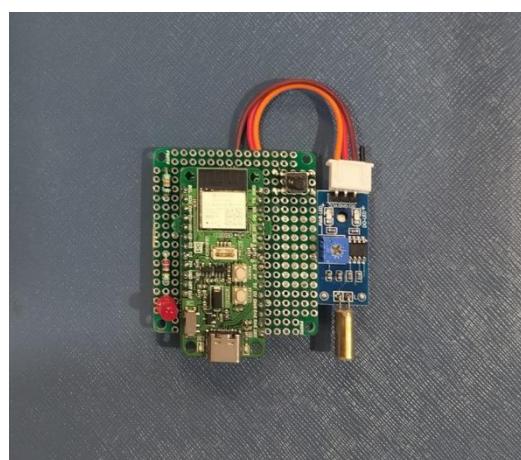
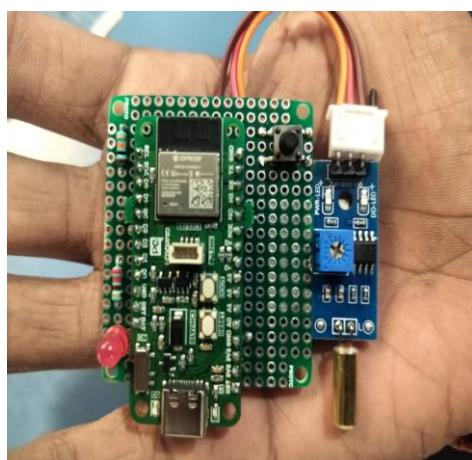
http.begin(url);
int httpResponseCode = http.GET();
if (httpResponseCode > 0) {
    Serial.println("Location sent!");
} else {
    Serial.print("Location Error: ");
    Serial.println(httpResponseCode);
}
http.end();
} else {
    Serial.println("WiFi not connected. Cannot send location.");
}
}

```

## **SOFTWARE/ TOOLS USED:**

1. Arduino IDE
2. ESP Board Manager for PCB Cupid C3
3. Telegram Bot API
4. KiCad 9.0

## **PROTOTYPE/ PICTURES:**



## **WORKING EXPLANATION AND OUTPUT:**

Silent Guardian is a compact wearable IoT safety device that helps a woman send an emergency alert (SOS) to her trusted contact via internet using Wi-Fi, Telegram, when she feels unsafe or is in danger.

### **1. How It Works:**

The device continuously monitors 2 emergency inputs:

- i. A Push Button – Pressed in panic or emergency.
- ii. A Vibration Sensor – Detects strong

### **2. Monitoring Triggers:**

- i. Motion (like tilting the device ).
- ii. Pressing the push button.

### **3. Emergency Trigger Activation:**

- If any one of these triggers is activated:
- The device instantly connects to Wi-Fi.
- It sends an SOS message like:
- "EMERGENCY! I need help
- Location will be sent.

### **4. Location Sharing**

- The device gets the live location using Wi-Fi geolocation or GPS (if integrated).
- The latitude and longitude are sent as a Google Maps link to the contact.

### **5. Alert Notification**

- Sends alert via Telegram bot.
- A LED gives feedback that the alert was sent.

## **What You See on the Contact's Side:**

The emergency contact will receive:

Example Telegram Alert:

EMERGENCY ALERT!

Silent Guardian triggered!

SHE IS IN DANGER.

Location: <https://maps.google.com/?q=12.9716,77.5946>

Time: 07:45 PM

## **Expected Output Summary:**

Action Taken	Output to Contact
Push button pressed	SOS message + location shared via app
Tilt detected	SOS message + location shared via app
LED feedback	Confirms alert sent

## **Real-Life Use Case Example:**

- You're walking home late. Someone suspicious follows you.
- You press the button or shake/Tilt your device.
- It silently sends your live location and a warning message to your trusted contact, and they know you're in danger.

## **APPLICATION AND RELEVANT PROJECT DETAILS:**

### **Applications:**

#### **1. Women's Safety**

- Quick distress alert to family/friends with location
- Discreet wearable form (e.g., keychain, pendant, band)

#### **2. Elderly Emergency Support**

- Instant alert if elderly fall or need assistance
- Optional vibration or fall sensor

#### **3. Child Tracking & Alert System**

- Notifies parents if child feels unsafe
- Could be integrated into a school bag

#### **4. Solo Workers or Remote Staff**

- Use in mines, factories, or remote areas
- Alert supervisors instantly

#### **5. Medical Emergency Trigger**

- Integrated with heart rate or medical sensors
- Sends alerts on abnormal readings

#### **6. Disaster Rescue Beacon**

- After a disaster (earthquake, etc.), survivors can press button to notify rescue teams

## **FUTURE WORK:**

### **1. GPS Module Integration (e.g., NEO-6M)**

Purpose: Accurately detect live geographic location independent of Wi-Fi.

Works outdoors even when Wi-Fi is unavailable.

Provides exact latitude and longitude for emergency responders.

Upgrade: Replace Wi-Fi-based approximate location with real-time GPS coordinates.

### **2. Touch Sensor for Silent Activation**

Purpose: Allow discreet and faster triggering of emergency alerts.

Use Case: If pressing a button is not feasible (e.g., under threat), a touch sensor hidden in clothing or accessories can silently trigger the alert.

Module: TTP223 capacitive touch sensor

Function: When touched, sends Telegram alert + location just like the push button.

### **3. Direct Calling Feature (to Father or Guardian)**

Purpose: Instantly call the primary contact without needing an app.

How: Use CallMeBot or a GSM module (like SIM800L) to make an automated voice call.

Enhancement:

Calls Parent/guardian with an emergency alert tone or message

Works in areas with poor internet by falling back to GSM.

### **4. Voice Message Recording**

Add-on: Microphone + SD card module

Function: Records a 10–30 second voice message when the alert is triggered.

Benefit: The recorded message can be later transmitted or retrieved for evidence.

### **5. Battery Backup and Power Efficiency**

Add a Li-Ion battery + charging circuit (TP4056) for portability.

Include deep sleep mode to save power and enable long-term usage.