

**Project Title:Tragedy Assisting Band System (TABS)**

**Group Members:**

|  |  |
| --- | --- |
| **Names:** | **CMS-ID:** |
| **Muhammad Shees** | **455842** |
| **Muhammad Asjad Kashif** | **454205** |
| **Muhammad Umar Malik** | **474125** |
| **Syed Muhammad Khizer Haider** | **474476** |
| **Aiza Tariq** | **474430** |
| **Muskan Khalil** | **472605** |

**Institute: National University of Science and Technology (NUST)**

**Date: 27-05-2024.**

**Abstract**

Project “**TABS”** (Tragedy Assisting Band System) is an innovative safety device designed specifically for age group ranging between 7-15 globally ,TABS aims at providing immediate assistance in any event of panic situation mostly harassment . This wearable band is equipped with a location-tracking feature that sends real-time GPS coordinates to provide emergency contact when activated to the parents or local authorities. The system is intended to offer a discrete and reliable means of summoning help, thereby enhancing the personal safety of young children in potentially dangerous situations.

The band features a panic switch for immediate activation, a long-lasting battery, and durable materials for daily use and the device is user-friendly.

The results demonstrate that the **“TABS”** (Tragedy Assisting Band System) significantly improves the sense of security among users and provides a practical solution to the pressing issue of harassment. The project enlightens the importance of technological interventions in safeguarding vulnerable populations and sets the stage for further developments in personal safety devices.

**Introduction**

**Background:**

According to the latest report of BBC news 30 percent of children age ranging from 7 to 13 are major victim of harassment which majorly affects the safety and health of these young children across Pakistan. keeping this age group in mind, TABS is designed as they are particularly vulnerable and culture often discourage them from openly discussing threats. TABS will provide instant assistance by sending real-time GPS to pre-set emergency contacts, thus facilitating rapid response in an emergency.

**Objectives:**

TABS should be compact, mobile light weight and camouflage band that can effectively sent SOS signal to authorities of parents (based on program written).

**Project Design and Development**

**Conceptual Design:**

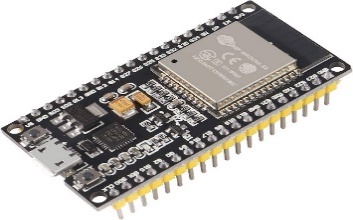
To make sure all objectives are filled. **“TABS”** (Tragedy Assisting Band System) is integrated with several hardware components to ensure functionality, reliability, and ease of use.

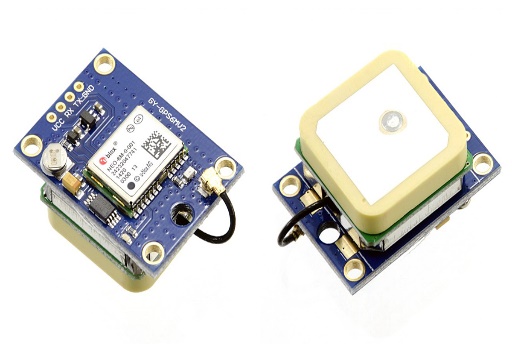
**Hardware Components**

The hardware used for implementation of Tragedy Assisting Band System includes:

1. **Esp-32** ( WIFI & BLUETOOTH)
2. **Neo-6M** (GPS module)
3. **SIM-800L** (GSM MODULE)
4. **5V lithium battery**
5. **Charging module**
6. **PCB board**
7. **SIM**
8. **PUSH BUTTON**.

**ESP-32:**

ESP32 is a chip that provides Wi-Fi and Bluetooth connectivity for embedded devices. It has a fast processing speed it is a signal sender and reciever or we can say a communicator it can be defined as the brain of the Tragedy Assisting Band System and it is compact & lightweight as compared to Arduino or UNO. Apart from that it has dual core increasing the processing speed to 240 MHz.

**Neo-6M:**

Neo-6M communicates with satellites and provide with the GPS location it is compact and lightweight. It communicates with ESP 32 using RX and TX nodes.

**SIM-800L:**

SIM800L is a miniature cellular module which allows for GPRS/GPMS transmission, sending and receiving SMS along with making and receiving voice calls. It is low cost compact and lightweight uses 3G/4G Sim and will send GPS location.

**5V lithium Battery:**

This battery provides a backup of 3 to 4 days and is rechargeable



**Push Button:**

To be used to turn on/off the circuit and sending SOS signal.

**Main Working of the "Tragedy Assisting Band System":**

ESP 32 acts as the brain of TABS communicating between GPS and GSM module using RX and TX nodes. TX nodes transmit data and RX retrieves data. Thus they work simultaneously making sure all the hardware communicates with each other and making sure that tabs works efficiently and effectively.

Let us guide you how all the data is interconnected with each other.

1. Connect ESP 32 with VCC and GND
2. Connect D2 of ESP32 to RX of SIM800l
3. Connect D4 of ESP32 to TX of SIM800l
4. Connect RX2 of ESP32 to TX of NEO6M
5. Connect TX2 of ESP32 to RX of NEO6M
6. Provide GPS and GSM VCC and GND
7. Connect battery
8. All components should have local GND and VCC from the battery

**CODE**

#include <TinyGPS++.h>

#include <HardwareSerial.h>

// Function prototypes

void sendCommand(String command, const int timeout, boolean debug);

void readSMS(int index);

void sendSMS(String number, String text);

void sendGPSLocation(String number);

// Define the hardware serial port for SIM800L

#define SIM800LSerial Serial2

// Define the baud rate for SIM800L

#define SIM800L\_BAUD 9600

// Define the serial port connected to the GPS module

#define GPS\_SERIAL\_PORT Serial1

// Define the baud rate of the GPS module

#define GPS\_BAUD 9600

// Define the GPS data parser object

TinyGPSPlus gps;

// Define the phone number to send the response to

#define RESPONSE\_NUMBER "+923350305078"

// Define the response message

#define RESPONSE\_MESSAGE "Hello"

// Define the interval for sending location (5 minutes in milliseconds)

#define LOCATION\_SEND\_INTERVAL 300000

unsigned long lastLocationSendTime = 0;

void setup() {

Serial.begin(9600); // Initialize Serial Monitor

SIM800LSerial.begin(SIM800L\_BAUD, SERIAL\_8N1, 4, 2); // Initialize SIM800L hardware serial

GPS\_SERIAL\_PORT.begin(GPS\_BAUD, SERIAL\_8N1, 16, 17); // Initialize GPS serial port

delay(30000); // Allow time for the SIM800L module to initialize Serial.println("Initializing SIM800L..."); // Send initialization commands to SIM800L

sendCommand("AT", 1000, true); // Check if the module is ready

sendCommand("AT+CMGF=1", 1000, true); // Set SMS mode to text mode

sendCommand("AT+CNMI=1,2,0,0,0", 1000, true); // Configure the module to notify about new SMS

Serial.println("Setup complete.");}

void loop() {

// Check if there is any data from the SIM800L module

if (SIM800LSerial.available()) {

String message = SIM800LSerial.readString();

Serial.println("Received from SIM800L: " + message);

// Check if the message contains "+CMTI" (new SMS indicator)

if (message.indexOf("+CMT:") != -1) {

// Extract the index of the new message

int index = message.substring(message.lastIndexOf(",") + 1).toInt();

readSMS(index); // Read and display the new message

// Print out the content of the received message for debugging

Serial.println("Received message content: " + message);

// Check if the received message is "Hello" from the specified number

if (message.indexOf(RESPONSE\_NUMBER) != -1 && message.indexOf(RESPONSE\_MESSAGE) != -1) {

Serial.println("Conditions met for response.");

A

// Attempt to send the GPS location as a response to the specified number

if (gps.location.isValid()) {

sendGPSLocation(RESPONSE\_NUMBER);

Serial.println("GPS location sent!");

} else {

Serial.println("No valid GPS location available.");

}

} else {

Serial.println("Message does not meet conditions for response.");

}

}

}

// Keep reading data from the GPS module

while (GPS\_SERIAL\_PORT.available() > 0) {

gps.encode(GPS\_SERIAL\_PORT.read()); // Feed GPS data to the parser

}

// Check if valid GPS data is available and send it immediately if it is the first time

if (gps.location.isValid()) {

if (lastLocationSendTime == 0 || millis() - lastLocationSendTime >= LOCATION\_SEND\_INTERVAL) {

sendGPSLocation(RESPONSE\_NUMBER);

lastLocationSendTime = millis(); // Update the last send time

Serial.println("GPS location sent!");

}

} else {

Serial.println("No valid GPS data.");

}

// Check if there is any data from the Serial Monitor

if (Serial.available()) {

String command = Serial.readStringUntil('\n');

command.trim(); // Remove any leading or trailing whitespace

// Check if the command is to send an SMS

if (command.startsWith("sendSMS")) {

// Extract the phone number and message

int firstComma = command.indexOf(',');

int firstQuote = command.indexOf('\"');

int secondQuote = command.indexOf('\"', firstQuote + 1);

int thirdQuote = command.indexOf('\"', secondQuote + 1);

int fourthQuote = command.indexOf('\"', thirdQuote + 1);

String number = command.substring(firstQuote + 1, secondQuote);

String text = command.substring(thirdQuote + 1, fourthQuote);

sendSMS(number, text);

} else {

// Send the entered command to SIM800L

sendCommand(command, 1000, true);}}}

// Function definitions

void sendCommand(String command, const int timeout, boolean debug) {

SIM800LSerial.println(command);

delay(timeout);

if (debug) {

while (SIM800LSerial.available()) {

Serial.write(SIM800LSerial.read());}

Serial.println();}}

void readSMS(int index) {

// Send command to read the SMS at the given index

sendCommand("AT+CMGR=" + String(index), 1000, true);}

void sendSMS(String number, String text) {

// Send the command to set the recipient's phone number

sendCommand("AT+CMGS=\"" + number + "\"", 1000, true);

delay(1000);

// Send the SMS text and Ctrl+Z to indicate the end of the message

SIM800LSerial.print(text);

delay(1000);

SIM800LSerial.write(26); // Ctrl+Z ASCII code

delay(1000);

while (SIM800LSerial.available()) {

Serial.write(SIM800LSerial.read());}

Serial.println();}

// Function to send the GPS location via SMS

void sendGPSLocation(String number) {

if (gps.location.isValid()) {

// Construct Google Maps URL

String googleMapsURL = "http://maps.google.com/maps?q=" + String(gps.location.lat(), 6) + "," + String(gps.location.lng(), 6);

sendSMS(number, googleMapsURL);} else {

Serial.println("No valid GPS location available.");}}

// Code written by Syed Muhammad Khizer Haider BSCS 23 (NBC)

**Problems faced and leaning out comes:**

There were several issues faced during this innovative idea. Firstly, we used A9g board which is an integrated board which has both GPS and GSM which I would recommend to use but in our case it was faulty. Then we got our hands on SIM 900D GSM module which was quite old and large and failed to send the SMS. Lastly, we used SIM800l which is very small and integrated board suitable for this project.

Learning out comes of out team were that there is a high chance that electrical components can be faulty and not be functional. Apart from that the code of TABS is self-written as AI tools were not much help full. To make sure the code works fine first try connect each module with the module separately and then try to connect the code to get the desired outcome.

**Conclusion/Results:**

In conclusion TABS is a small mobile device that aims to save lives of our upcoming generation and assuring their safety with its small compact and reliable hardware. This project helped us to push our boundaries and to help solve the issues in our society. TABS is in the early stages of its build but surely full fills its tasks efficiently and effectively