

EMBEDDED SYSTEMS – PROJECT REPORT

Lost and Found GPS GSM tracker device

Done by:

Aravind Ananthakrishnan – EC21B1112

Vikrant Abhishek – EC21B1116

Introduction:

Lost and Found GPS GSM tracker device is a device that helps track lost or stolen items such as bags, keys, vehicles, and other valuable objects. It is designed to be compact and easy to carry or attach to objects. The device combines the power of GPS technology with GSM communication to enable remote tracking of objects.

Objective:

The primary objective of this project is to design and develop a Lost and Found GPS GSM tracker device that can be used to locate lost or stolen objects.

Hardware Components:

The following hardware components are required to build the Lost and Found GPS GSM tracker device:

- Arduino Uno board



- GSM module (SIM900)



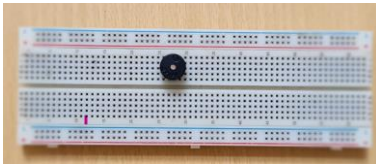
- GPS module (UBLOX NEO-6M)



- Li-Ion battery



- Battery charging module
- Buzzer and Breadboard



- Jumper wires
- Enclosure

Software Components:

The following software components are required to build the Lost and Found GPS GSM tracker device:

- Arduino IDE
- TinyGPS++ library
- SoftwareSerial library

Working:

1. The Lost and Found GPS GSM tracker device works as follows:
2. The GPS module receives location information from GPS satellites and sends it to the Arduino board.
3. The Arduino board reads the GPS location information and sends it to the GSM module using AT commands.
4. The GSM module sends the GPS location information to a predefined mobile number using SMS.
5. The mobile number owner can then use the information to locate the lost or stolen object.

6. The device is powered by a Li-Ion battery, which can be recharged using the battery charging module.

Code and Explanation:

```
#include <TinyGPS++.h>
#include <SoftwareSerial.h>
#include<SoftwareSerial.h>
SoftwareSerial GPS_SoftSerial(4, 3);
SoftwareSerial mySerial(10, 11);
TinyGPSPlus gps;
char msg;
double lat_val;
double long_val;
volatile float minutes, seconds;
volatile int degree, secs, mins;

void setup() {
  mySerial.begin(9600); // Setting the baud rate of GSM Module
  Serial.begin(9600);
  GPS_SoftSerial.begin(9600);
  delay(100);
}

void loop(){
  smartDelay(1000);
  unsigned long start;
  double alt_m_val;
  bool loc_valid, alt_valid;
  lat_val = gps.location.lat();
  loc_valid = gps.location.isValid();
  long_val = gps.location.lng();
```

```

alt_m_val = gps.altitude.meters();
alt_valid = gps.altitude.isValid();
if (!loc_valid)
{
    Serial.print("Latitude : ");
    Serial.println("0");
    Serial.print("Longitude : ");
    Serial.println("0");
    delay(4000);
}
else
{
    Serial.println("GPS READING: ");
    DegMinSec(lat_val);
    Serial.print("Latitude in Decimal Degrees : ");
    Serial.println(lat_val,6);
    DegMinSec(long_val);
    Serial.print("Longitude in Decimal Degrees : ");
    Serial.println(long_val,6);
    delay(4000);
}
SendMessage();
delay(5000);
}

```

```

static void smartDelay(unsigned long ms)
{
    unsigned long start = millis();
    do
    {

```

```

    while (GPS_SoftSerial.available())
    gps.encode(GPS_SoftSerial.read());
} while (millis() - start < ms);
}

```

```

void DegMinSec( double tot_val)
{
    degree = (int)tot_val;
    minutes = tot_val - degree;
    seconds = 60 * minutes;
    minutes = (int)seconds;
    mins = (int)minutes;
    seconds = seconds - minutes;
    seconds = 60 * seconds;
    secs =(int)seconds;
}

```

```

void SendMessage()
{
    mySerial.println("AT+CMGF=1"); //Sets the GSM Module in Text Mode
    delay(1000); // Delay of 1000 milli seconds or 1 second
    mySerial.println("AT+CMGS=\"+919676655675\"\\r"); // Replace x with mobile number
    delay(1000);
    mySerial.print(lat_val,4);
    mySerial.print(",");
    mySerial.println(long_val,4); // The SMS text you want to send
    delay(100);
    mySerial.println((char)26); // ASCII code of CTRL+Z
    delay(1000);
}

```

The code provided is an Arduino sketch for a Lost and Found GPS GSM Tracker device. It uses a GPS module to determine the device's current location and a GSM module to send the location data to a specified mobile number via SMS.

The code initializes the GPS and GSM modules and sets the required parameters such as baud rates. The GPS module continuously reads location data and stores it in variables. If a valid location is obtained, the latitude and longitude values are converted from decimal degrees to degree-minute-second format and printed to the serial monitor. If a valid location is not obtained, the latitude and longitude are set to 0 and printed to the serial monitor.

After obtaining the location data, the Send Message function is called, which sends an SMS to a specified phone number containing the latitude and longitude data. Overall, this code serves as a basic example of how to interface GPS and GSM modules with an Arduino to create a lost and found tracker device.

Android Application:

The android application was realized using Android Studio in kotlin language.

MainActivity.kt :

```
package com.example.avtracker
import android.content.BroadcastReceiver
import android.content.Context
import android.content.Intent
import android.content.IntentFilter
import android.os.Bundle
import android.provider.Telephony
import android.telephony.SmsManager
import android.webkit.WebView
import android.widget.Button
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity
import java.util.regex.Pattern

class MainActivity : AppCompatActivity() {
    private lateinit var latitudeTextView: TextView
    private lateinit var longitudeTextView: TextView
    private lateinit var webView: WebView
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        latitudeTextView = findViewById(R.id.textView2)
        longitudeTextView = findViewById(R.id.textView4)
        webView = findViewById(R.id.web_view)
        webView.settings.javaScriptEnabled=true
        webView.settings.javaScriptCanOpenWindowsAutomatically=true
        // Load a URL
        val smsReceiver = SmsReceiver()
        val intentFilter =
            IntentFilter(Telephony.Sms.Intents.SMS_RECEIVED_ACTION)
        registerReceiver(smsReceiver, intentFilter)
        val button = findViewById<Button>(R.id.button)
```

```

        button.setOnClickListener {
            val phoneNumber = "+919676655675"
            val message = "Location"
            val subscriptionId = SmsManager.getDefaultSmsSubscriptionId()
            val smsManager =
SmsManager.getSmsManagerForSubscriptionId(subscriptionId)

            // Send the message
            smsManager.sendTextMessage(phoneNumber, null, message, null,
null)
        }

        val button2 = findViewById<Button>(R.id.button2)
        button2.setOnClickListener {
            val phoneNumber = "+919676655675"
            val message = "Buzz"

            val smsManager = SmsManager.getDefault()

            smsManager.sendTextMessage(phoneNumber, null, message, null,
null)
        }
    }
    inner class SmsReceiver : BroadcastReceiver() {

        private var lat: Double = 0.0
        private var lon: Double = 0.0
        override fun onReceive(context: Context?, intent: Intent?) {
            val messages =
Telephony.Sms.Intents.getMessagesFromIntent(intent)
            val messageBody = messages[0].messageBody
            val sender = messages[0].originatingAddress
            if (sender == "+919676655675") {
                val pattern = "([-+]?[0-9]*\\.?[0-9]+),([-+]?[0-9]*\\.?[0-
9]+)"
                val matcher = Pattern.compile(pattern).matcher(messageBody)
                if (matcher.find()) {
                    lat = matcher.group(1).toDouble()
                    lon = matcher.group(2).toDouble()
                    latitudeTextView.text = lat.toString()
                    longitudeTextView.text = lon.toString()

webView.loadUrl("https://www.google.com/maps/search/?api=1&query=$lat%2C$lon"
)
                }
            }
        }
    }
}

```

AndriodManifest.xml :

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

```

```

<uses-feature
    android:name="android.hardware.telephony"
    android:required="false" />

<uses-permission android:name="android.permission.SEND_SMS" />
<uses-permission android:name="android.permission.RECEIVE_SMS" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"
/>

<application
    android:allowBackup="true"
    android:dataExtractionRules="@xml/data_extraction_rules"
    android:fullBackupContent="@xml/backup_rules"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/Theme.AVTracker"
    tools:targetApi="31">
    <activity
        android:name=".MainActivity"
        android:exported="true">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
    <receiver
        android:name="com.example.avtracker.MainActivity.SmsReceiver"
        android:exported="false"> <intent-filter> <action
        android:name="android.provider.Telephony.SMS_RECEIVED" /> </intent-filter>
    </receiver>
    </application>

</manifest>

```

Activity_main.xml :

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="412dp"
    android:layout_height="883dp"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Latitude: "
        app:layout_constraintBottom_toBottomOf="parent"

```



```

        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.216"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.562" />

<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Longitude: "
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.191"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.611" />

<Button
    android:id="@+id/button"
    android:layout_width="385dp"
    android:layout_height="55dp"
    android:text="LIVE LOCATION"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.555"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.745"
    tools:ignore="MissingConstraints,VisualLintButtonSize" />

<Button
    android:id="@+id/button2"
    android:layout_width="385dp"
    android:layout_height="55dp"
    android:text="BUZZ"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.592"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.865"
    tools:ignore="MissingConstraints,VisualLintButtonSize" />

<WebView
    android:id="@+id/web_view"
    android:layout_width="383dp"
    android:layout_height="404dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.62"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.129"
    tools:ignore="MissingConstraints" />

<TextView

```

```

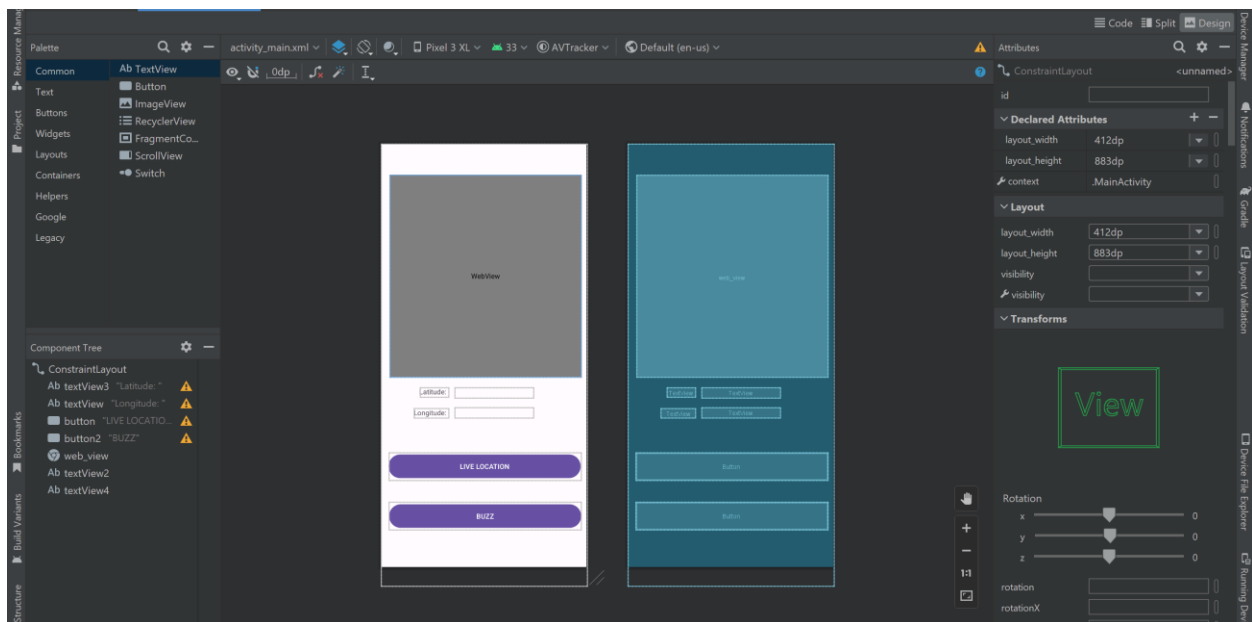
        android:id="@+id/textView2"
        android:layout_width="158dp"
        android:layout_height="20dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.578"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.563" />

<TextView
    android:id="@+id/textView4"
    android:layout_width="158dp"
    android:layout_height="20dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.578"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.611" />

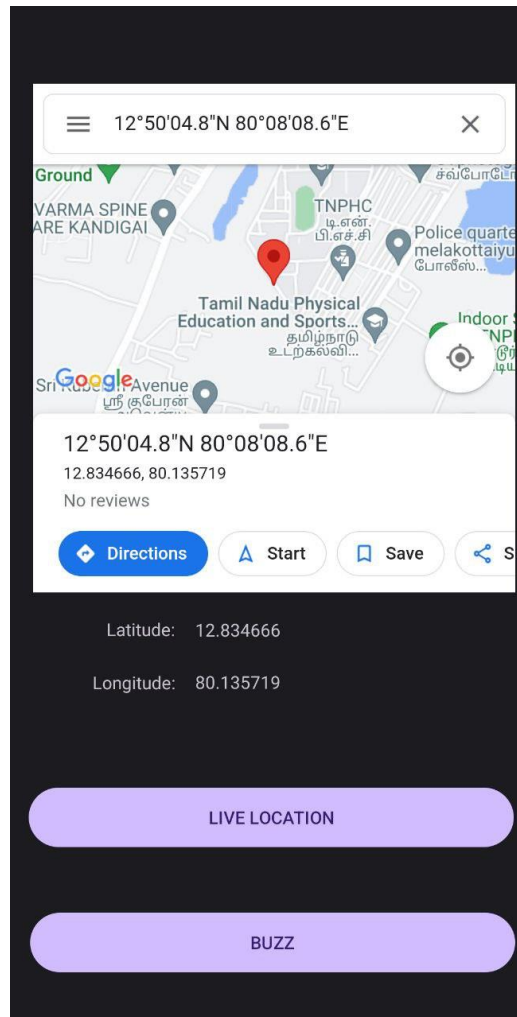
</androidx.constraintlayout.widget.ConstraintLayout>

```

Main Activity Area:



Application Screenshot:



Here when LIVE LOCATION button is pressed, a specific SMS is sent to GSM cellular number which triggers a code in Arduino microcontroller to send the current GPS data back to the client in <latitude>,<Longitude> format which is read by the app and it displays location on WebView.

Similarly, when Buzz is pressed, another SMS is sent which instructs microcontroller to switch Buzzer on.

Conclusion:

The Lost and Found GPS GSM tracker device is an innovative solution for tracking lost or stolen objects. It combines the power of GPS technology and GSM communication to enable remote tracking of objects. With the Lost and Found GPS GSM tracker device, individuals can now easily locate their lost or stolen items, providing them with greater peace of mind.