UNIVERSITY OF MORATUWA, SRI LANKA
Faculty of Engineering
Department of Electronic and Telecommunication Engineering
Semester 4 (Intake 2020)

EN2160 - Electronic Design Realization

Design Report

A. T. P. Amarasekara
200023C

Table of Contents

Abstract

The smart tracker presented in this design report is an innovative device equipped with advanced GPS and GSM technologies to provide accurate real-time location tracking and monitoring capabilities for assets, vehicles, and individuals. In addition to tracking functionality, the smart tracker is integrated with an emergency SOS alert feature, further enhancing its utility and safety aspects.

The primary objective of this smart tracker is to deliver precise location data and insights into the movement of tracked entities, enabling efficient asset management and ensuring personal security and safety. It accurately tracks and displays the real-time location of the device, empowering users to monitor and safeguard their valuable assets or loved ones remotely.

The core functionality of this smart tracker revolves around two essential metrics: tracking the location of the device using GPS connectivity and continuously updating it onto the User Interface platform.

Furthermore, the integration of an emergency SOS alert feature adds an extra layer of safety to the smart tracker. In times of distress or emergencies, users can activate the SOS alert, triggering an immediate notification to designated contacts. This functionality ensures quick response and assistance in critical situations, enhancing personal security and peace of mind for users.

In conclusion, the smart tracker's multifunctionality revolves around precise real-time location tracking, and the integration of an emergency SOS alert feature. Its adaptability and seamless communication capabilities make it an invaluable tool for individuals, and organizations seeking reliable and comprehensive tracking solutions with enhanced safety features.

1. General Information

1.1 Overview

The Smart Tracker is an innovative and advanced device designed to provide real-time tracking and monitoring capabilities for assets, vehicles, or individuals. The primary objective of this tracker is to offer accurate location data and seamless communication through the Global Positioning System (GPS) and the Global System for Mobile Communications (GSM) network. The key specifications of the Smart Tracker are as follows:

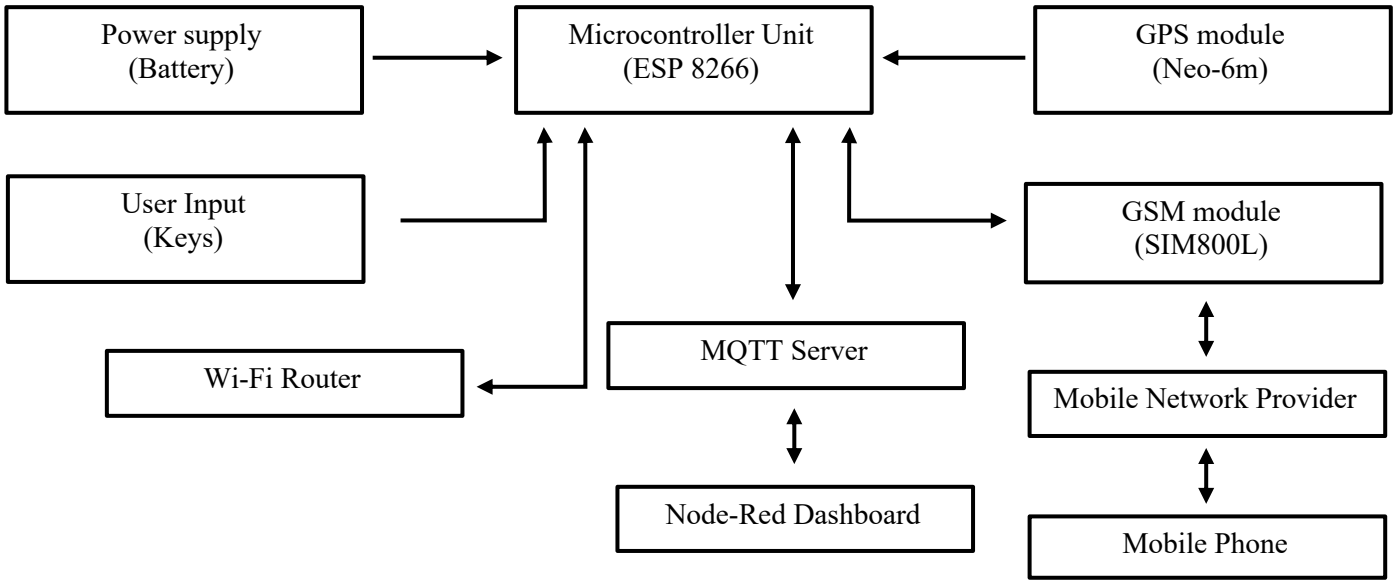
- **GPS Technology:** The Smart tracker incorporates a high-sensitivity GPS receiver module with U-blox NEO-6M chip, capable of acquiring signals from multiple satellite systems. This ensures optimal positioning accuracy even in challenging environments.
- **Communication:** To transmit the real-time location data, the Smart tracker communicates with MQTT server and utilizes the Node-red dashboard. Also, this device is equipped with a GSM/GPRS module which establishes a reliable and secure communication link to a mobile device, allowing users to access the location information effortlessly.
- **Microcontroller:** The GPS tracker employs an efficient and low-power microcontroller to handle data processing, manage communication protocols, and control the overall device operation. The microcontroller plays a crucial role in maximizing battery life and reducing power consumption.
- **Power Source:** The device is powered by a high-capacity rechargeable lithium-ion battery, which ensures extended operation without frequent recharging. The device also incorporates power-saving features to optimize battery usage.
- **Interfaces:** For user convenience, the GPS tracker is equipped with a Micro-USB interface, enabling easy charging and configuration. LED indicators are integrated into the design to provide intuitive status feedback, such as power status, GPS signal strength, and communication status.

1.2 Functionality

- Real-time location tracking through dashboard
- Real-time speed tracking through dashboard
- Real-time altitude tracking through dashboard
- Emergency SOS button allowing the user to notify pre-defined phone number by SMS and Call
- Allows user to change the SOS number through the dashboard
- Wi-Fi connectivity using Soft-AP
- Battery Rechargeability

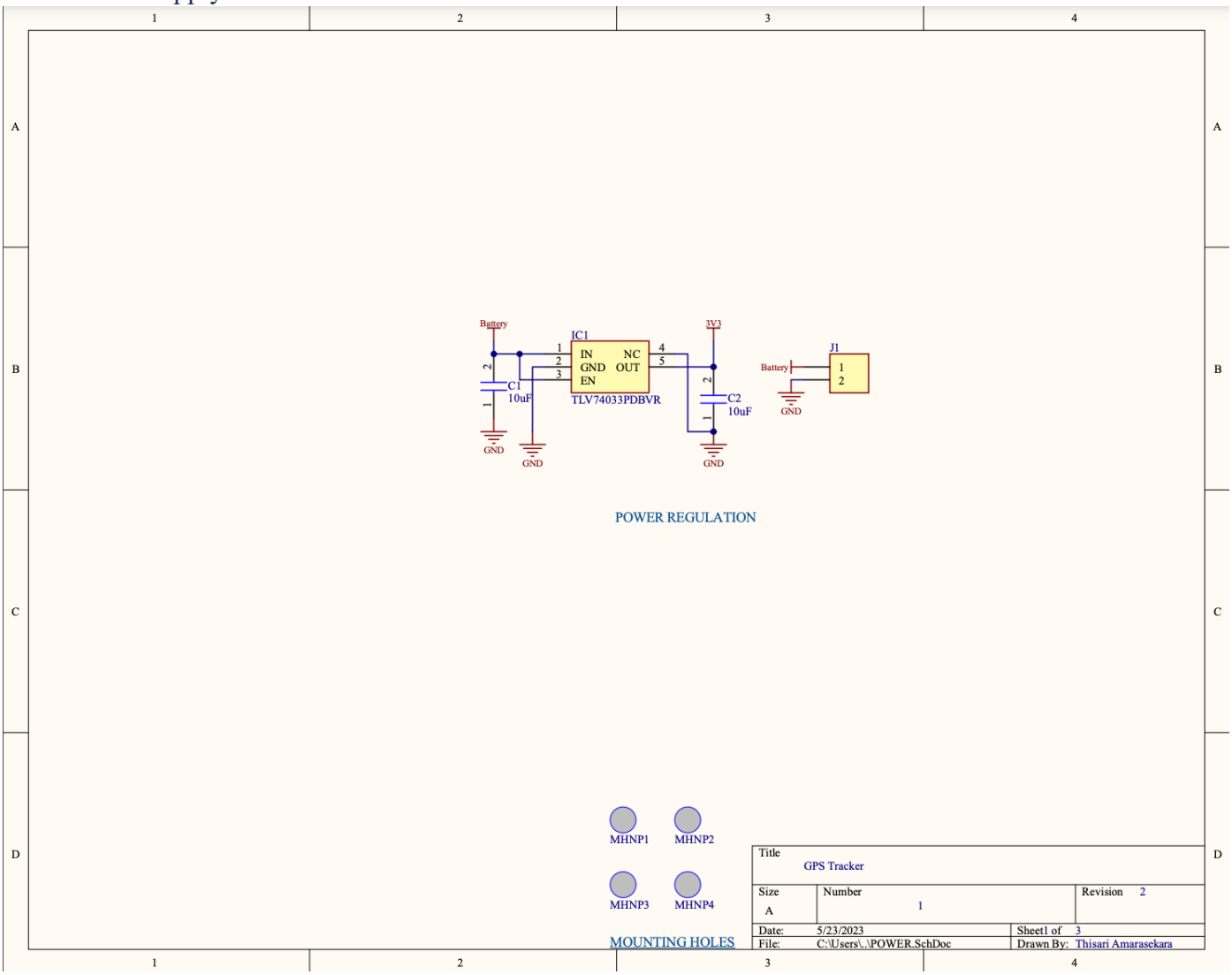
2. Functional Circuit Design

2.1 Functional Block Diagram

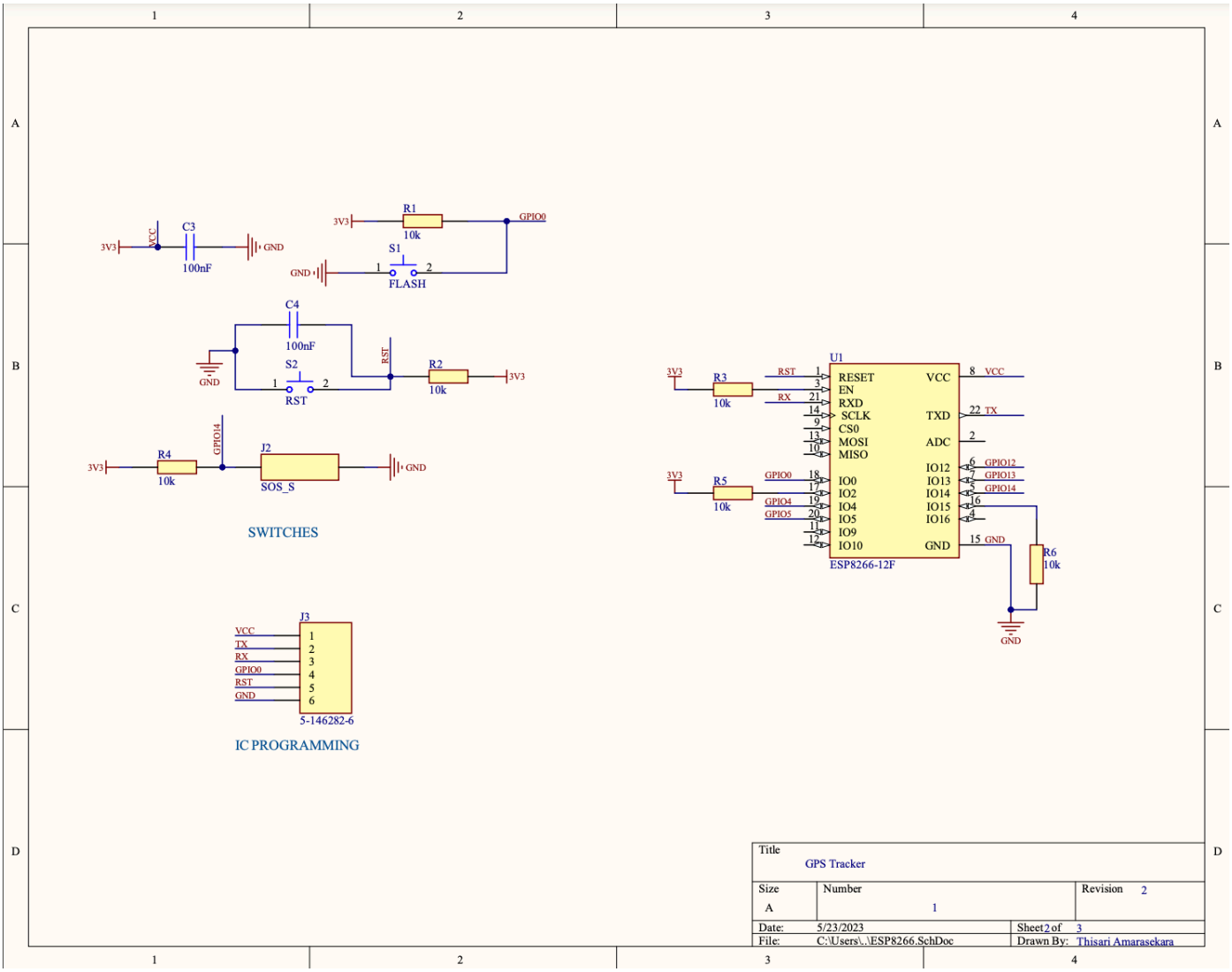


2.2 Schematic design

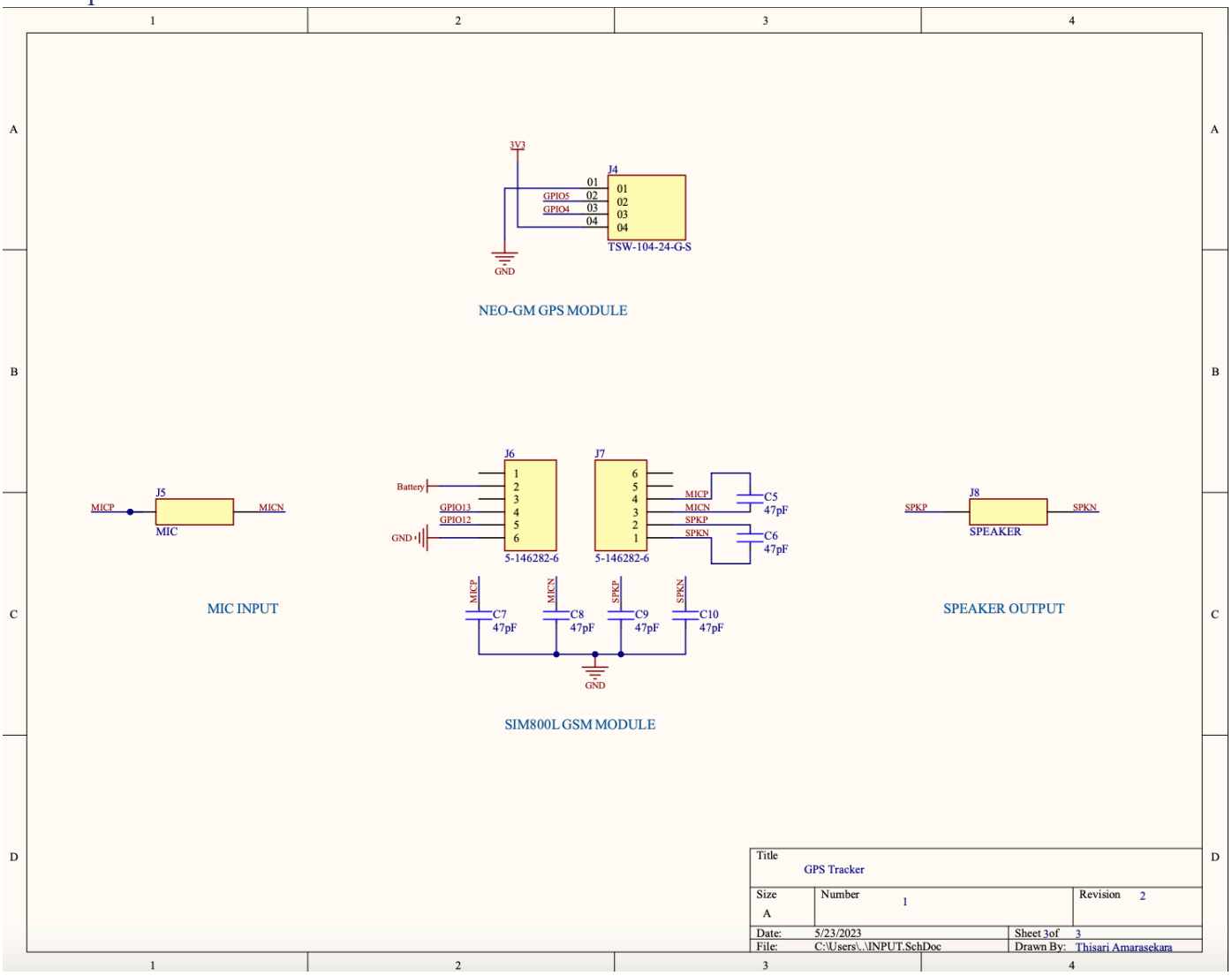
2.2.1 Power Supply



2.2.2 Microcontroller



2.2.3 Input Modules



3. Component Selection

The smart tracker is designed to provide accurate real-time location tracking using a combination of the ESP8266 SMD chip as the microcontroller, u-blox NEO-6M GPS module, and SIM800L GSM module. Each component was carefully selected based on its features, performance, and compatibility to ensure optimal functionality and seamless integration within the smart tracker system.

3.1 ESP8266 SMD Chip

The ESP8266 SMD chip is a popular and highly capable microcontroller known for its Wi-Fi connectivity and low power consumption. It serves as the brains of the GPS Tracker, providing essential processing capabilities and efficient control over the device's operation. The ESP8266 SMD chip has been selected for the following reasons:

- **Wi-Fi Connectivity:** The built-in Wi-Fi capability of the ESP8266 enables easy and secure communication with external servers, cloud platforms, or mobile apps. This allows users to remotely access real-time location data and control the GPS Tracker effectively.

- **Low Power Consumption:** The ESP8266 is designed with power efficiency in mind, making it ideal for battery-powered applications like the GPS Tracker. It ensures prolonged battery life, a crucial factor for uninterrupted tracking over extended periods.

- **Abundant GPIO Pins:** The ESP8266 offers a sufficient number of General-Purpose Input/Output (GPIO) pins, facilitating seamless interfacing with other components like GPS and GSM modules.

- **Compact Size:** As an SMD (Surface Mount Device) chip, the ESP8266 comes in a compact form factor, making it ideal for space-constrained designs, such as the smart tracker.

3.2 U-blox NEO-6M GPS Module

The u-blox NEO-6M GPS module is a highly accurate and reliable GPS receiver, making it an excellent choice for precise location tracking in the GPS Tracker. The key reasons for selecting the u-blox NEO-6M GPS module are as follows:

- **Multi-Constellation Support:** The NEO-6M supports multiple satellite constellations, including GPS, GLONASS, and Galileo. This enhances the accuracy and availability of positioning data, particularly in urban environments and challenging terrains.

- **High Sensitivity:** The NEO-6M GPS module features high-sensitivity tracking and acquisition, ensuring a faster and more stable lock on satellite signals, even under weak signal conditions.

- **Compact Form Factor:** The small and compact design of the NEO-6M module allows for easy integration into the GPS Tracker, without compromising the overall size and portability of the device.

- **Power Efficiency:** The NEO-6M is optimized for low power consumption, complementing the ESP8266's power-saving features to extend the GPS Tracker's battery life significantly.

3.3 SIM800L GSM Module

The SIM800L GSM module is a robust and versatile communication module that enables the GPS Tracker to establish a GSM/GPRS connection for transmitting location data to remote servers or mobile apps. The decision to use the SIM800L GSM module is based on the following considerations:

- **Wide Network Compatibility:** The SIM800L module supports quad-band GSM frequencies, ensuring compatibility with various network operators worldwide, allowing the GPS Tracker to function in different geographical regions.
- **GPRS Data Transfer:** With GPRS support, the module can efficiently send location data to a central server, making it suitable for real-time tracking applications.
- **Small Form Factor:** The compact size of the SIM800L module makes it an ideal choice for space-constrained applications like the GPS Tracker.
- **Low Power Consumption:** Similar to the other selected components, the SIM800L is designed to be energy-efficient, ensuring it doesn't unduly drain the GPS Tracker's battery.

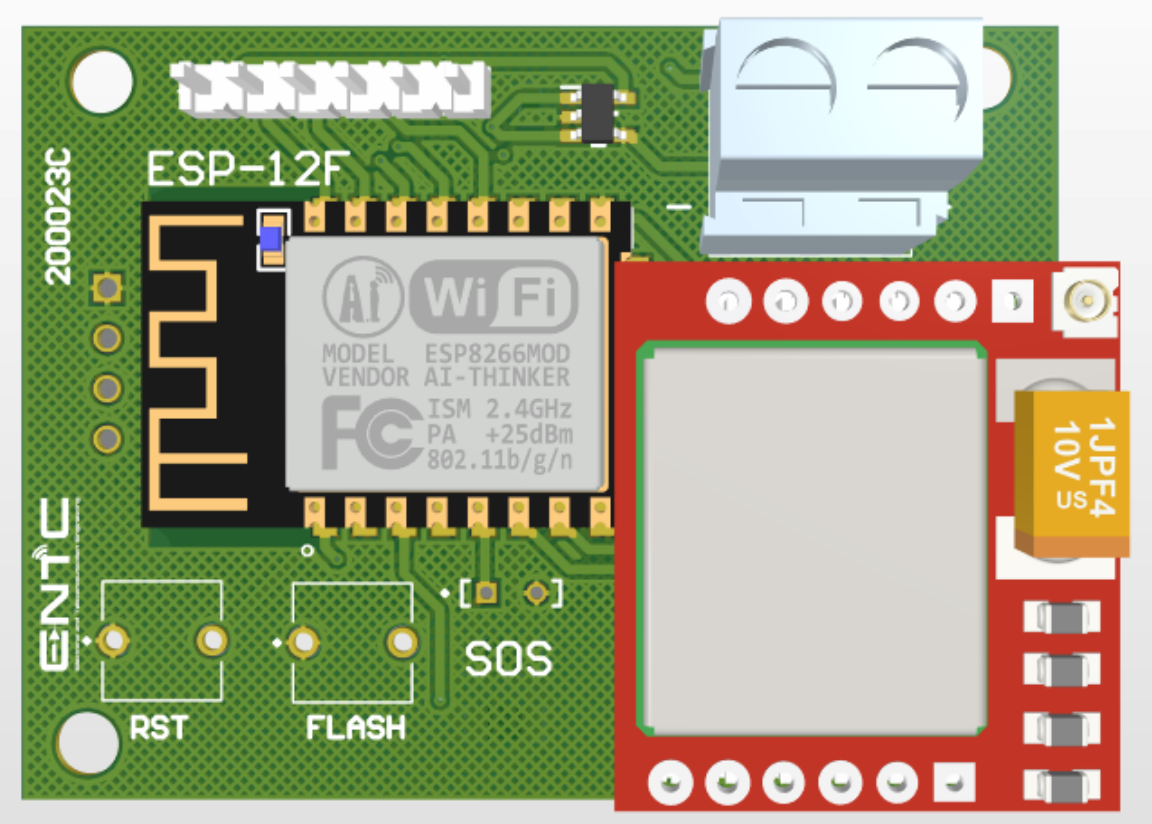
3.4 TLV74033PDBVR LDO Voltage Regulator

The 3.3V LDO voltage regulator was selected for its ability to provide a reliable and steady 3.3V output voltage, which is essential for the proper operation of the ESP8266 microcontroller, GPS module. The key features and advantages of these voltage regulators include:

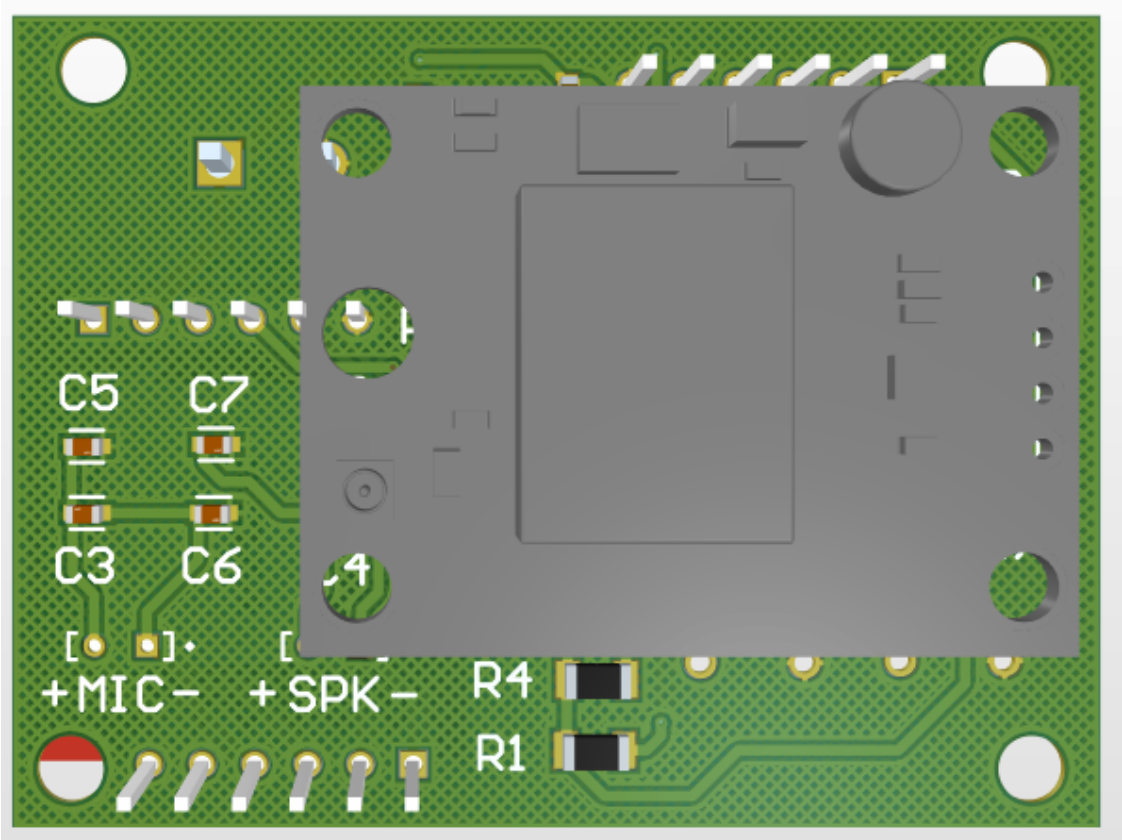
- **Voltage Regulation:** The LDO voltage regulators maintain a constant output voltage of 3.3V, even when the input voltage fluctuates. This feature is crucial for preventing damage to sensitive electronic components that require a stable power supply.
- **Low Dropout Voltage:** LDO regulators have a low dropout voltage, meaning they can maintain the regulated output voltage even when the input voltage is only slightly higher than the desired output voltage. This ensures efficient power conversion and minimizes power dissipation. The TLV74033PDBVR regulator has a 460mV dropout voltage.
- **Current Rating (300 mA):** The selected LDO voltage regulator has a current rating of 300 mA, which is sufficient to meet the power requirements of the ESP8266 microcontroller, GPS module, and other peripherals within the smart tracker.
- **Compact and Lightweight:** The chosen voltage regulators is available in compact and lightweight packages, ensuring they fit seamlessly within the space-constrained design of the smart tracker.
- **Efficiency and Low Power Consumption:** These LDO regulators are designed for high efficiency and low quiescent current, contributing to the overall energy efficiency and extended battery life of the smart tracker.

4. PCB Layout

4.1 3D View

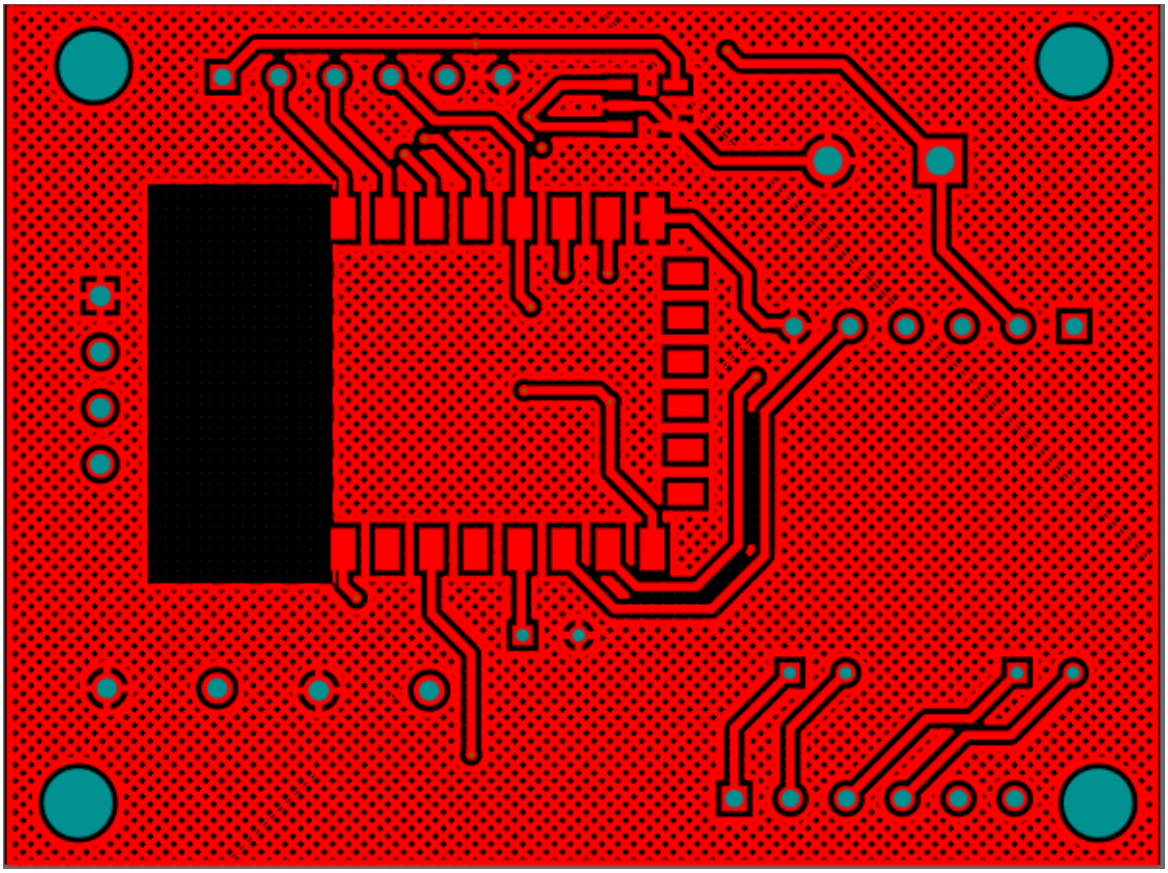


Top View

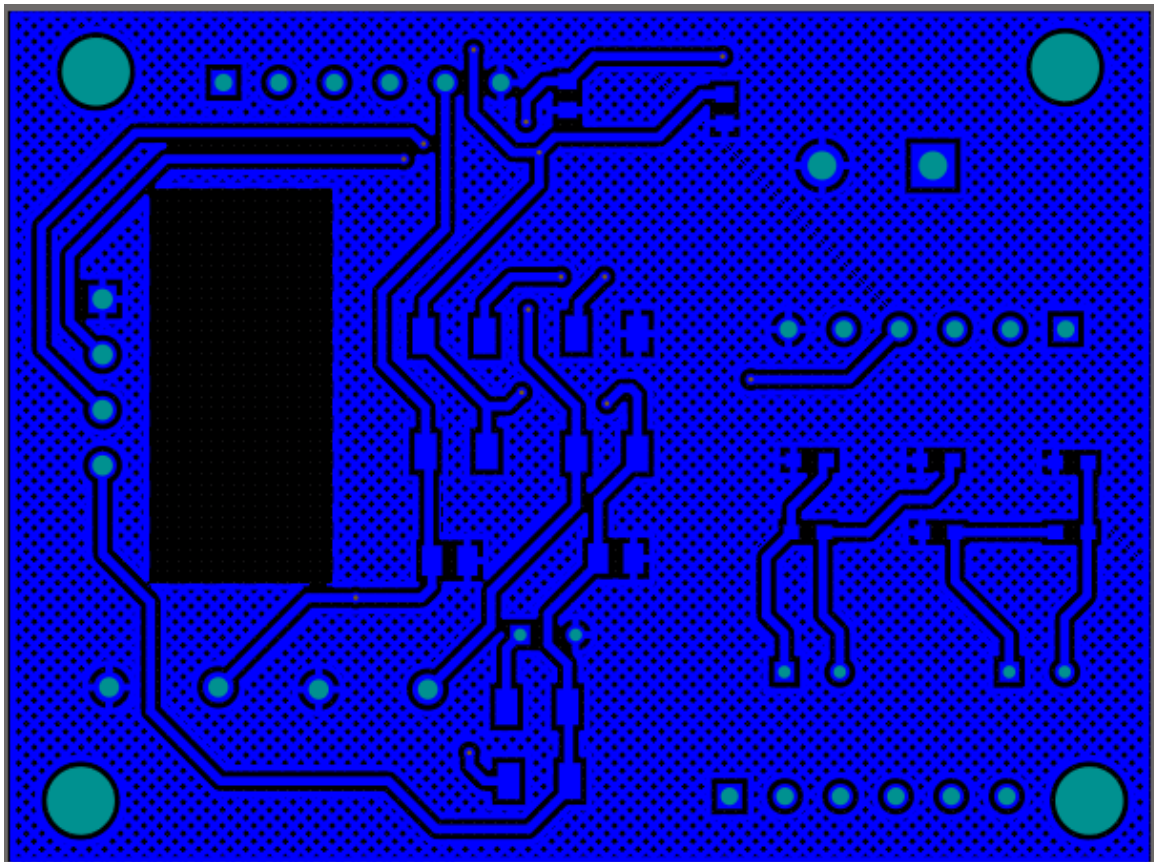


Bottom View

4.2 Top Layer



4.3 Bottom Layer



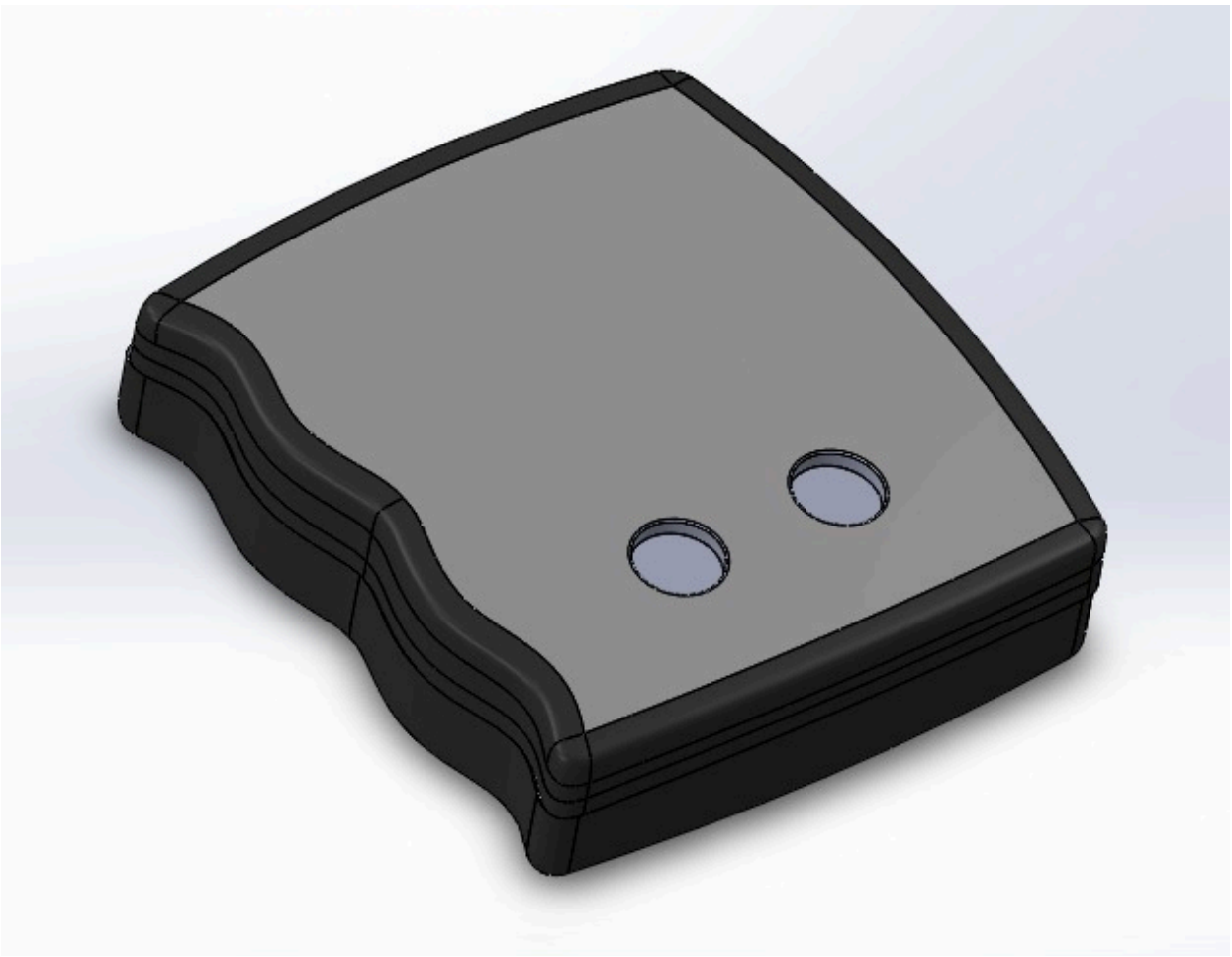
5. Bill Of Material

Item Name	Quantity	Unit Price (Rs.)	Supplier
ESP 8266 chip	1	1101.18	https://www.elecrow.com/
Sim800L GPS Module	1	1200.00	https://tronic.lk/
Neo6m GSM Module	1	1425.00	https://www.duino.lk/
Li-ion Battery (1800mAh)	1	600.00	https://tronic.lk/
Mini 8 Ohm Speaker	1	535.92	https://www.mouser.com/
Electret condenser microphone	1	127.6	https://www.mouser.com/
3.3 V LDO Voltage Regulators 300-mA	1	79.75	https://www.mouser.com/
10kΩ 250mW Thick Film Resistors	6	5.00	https://www.lcsc.com/
50V 100nF Multilayer Ceramic Capacitors	2	7.31	https://www.lcsc.com/
6.3V 10uF Multilayer Ceramic Capacitors	2	13.94	https://www.lcsc.com/
50V 47pF Multilayer Ceramic Capacitors	4	0.83	https://www.lcsc.com/
2-Way Mini JST 2.0 PH Connector Wire	1	20.00	https://tronic.lk/
2-Pin 2-Way Terminal Block	1	20.00	https://tronic.lk/
Battery holder	1	70.00	https://tronic.lk/
3.7V Li-Ion battery (1800mAh)	1	600.00	https://tronic.lk/
SPST Rectangular Rocker Switch	1	20.00	https://tronic.lk/
Stainless Steel Metal Push Button	1	170.00	https://tronic.lk/
M3 Screws	4	20.00	https://tronic.lk/
M3 Copper Female Spacer	4	80.00	https://tronic.lk/
PCB	1	1900.00	https://jlcpcb.com/
Enclosure Box	1	470.00	https://tronic.lk/

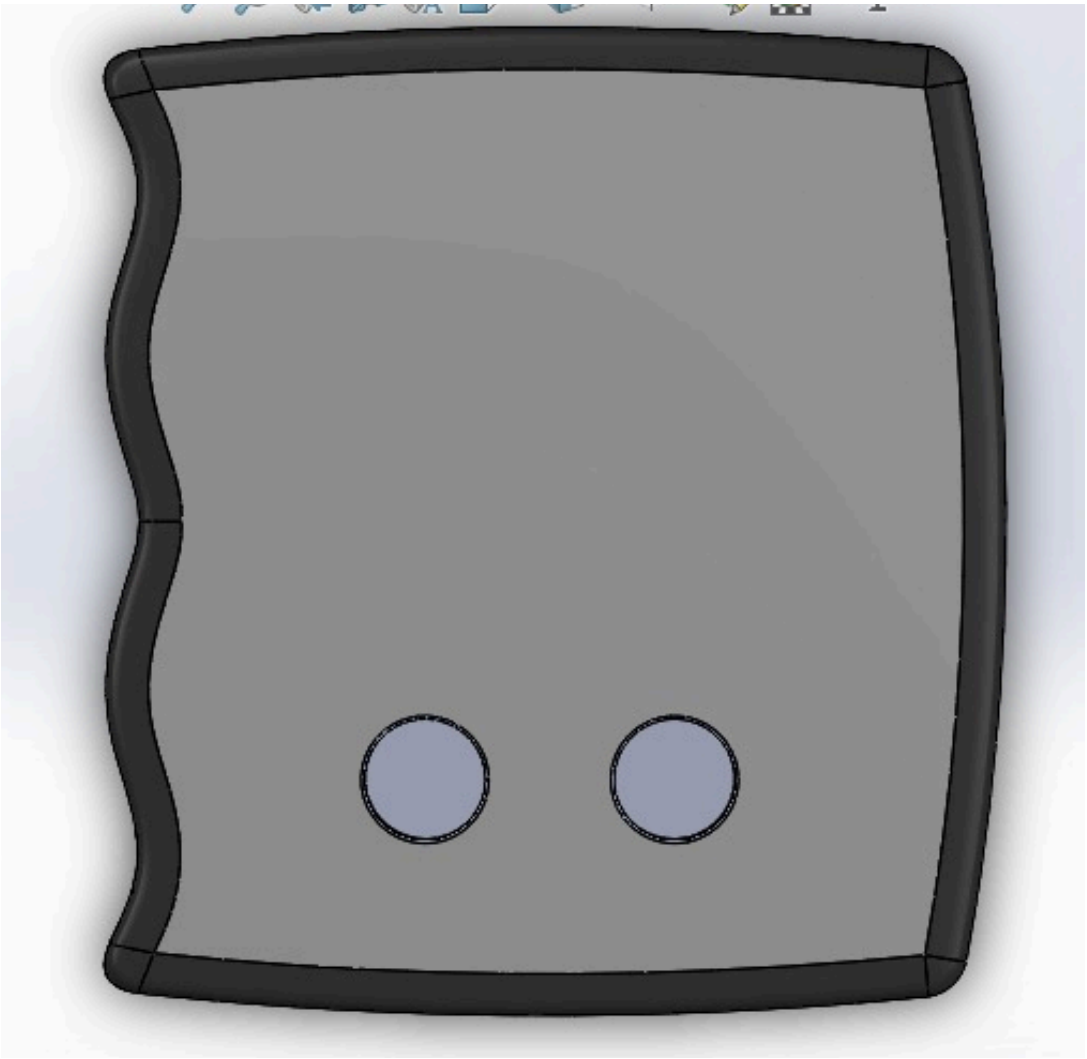
6. Enclosure Design

The Smart Tracker enclosure is designed to protect the internal components from environmental factors and provide a sleek and sturdy housing.
The enclosure is modelled from CAD software Solidworks 2020. In order to manufacture the enclosure, injection molding is used and the designs for the molds are also modelled from Solidworks. Material of the enclosure is Acrylonitrile Butadiene Styrene (ABS). With the lack of facilities in-house, enclosure manufacture is outsourced, and the injection molded parts will be provided.

6.1 Solidworks Design

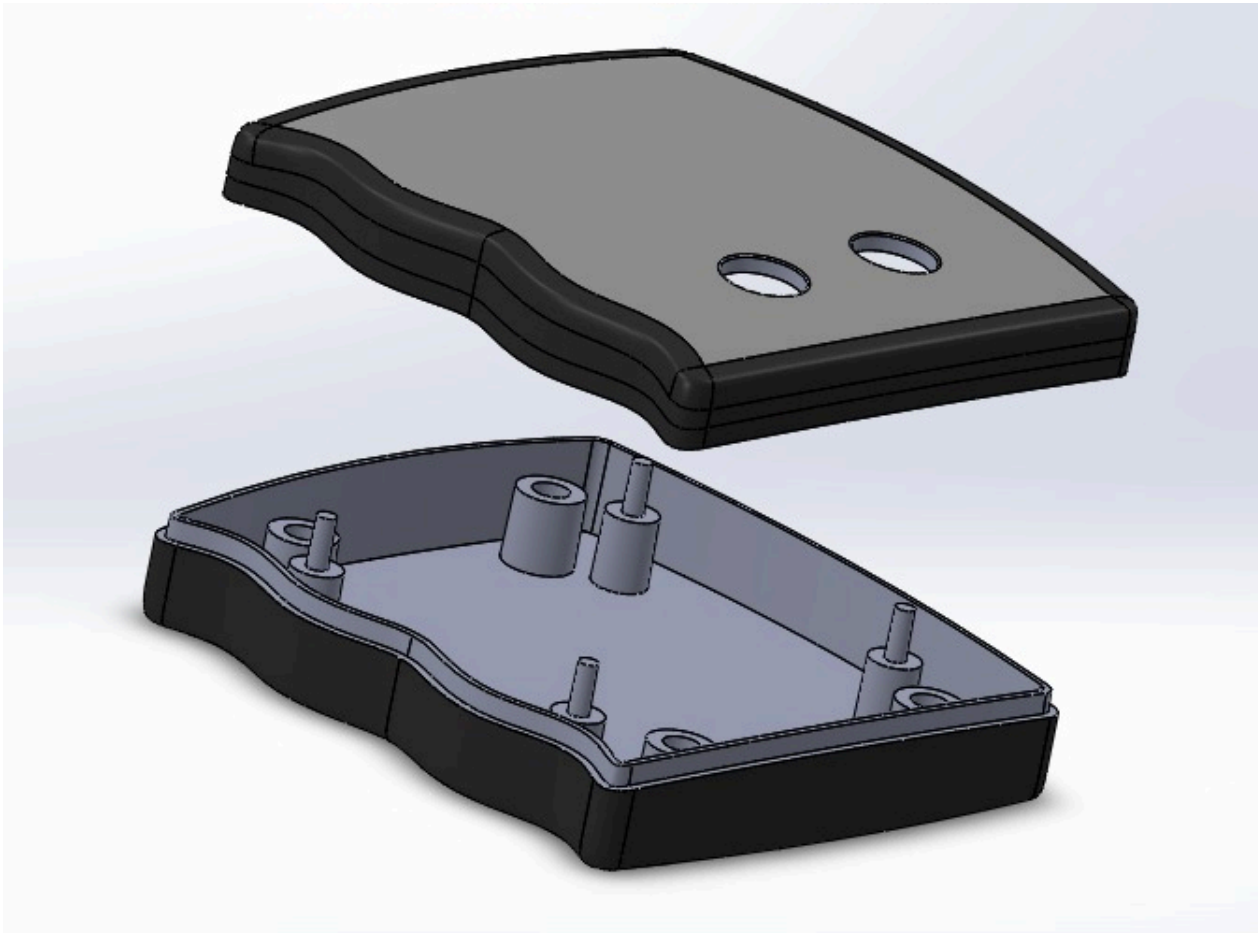


Isometric View

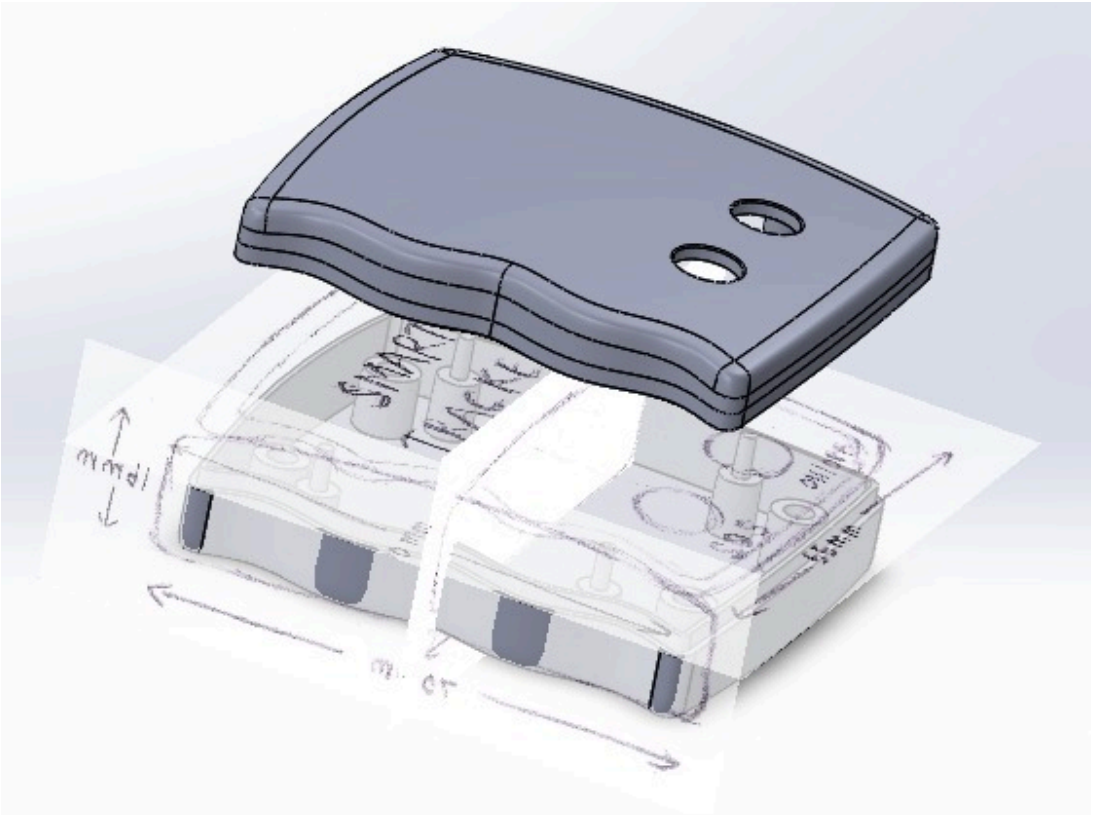


Top View

Side View

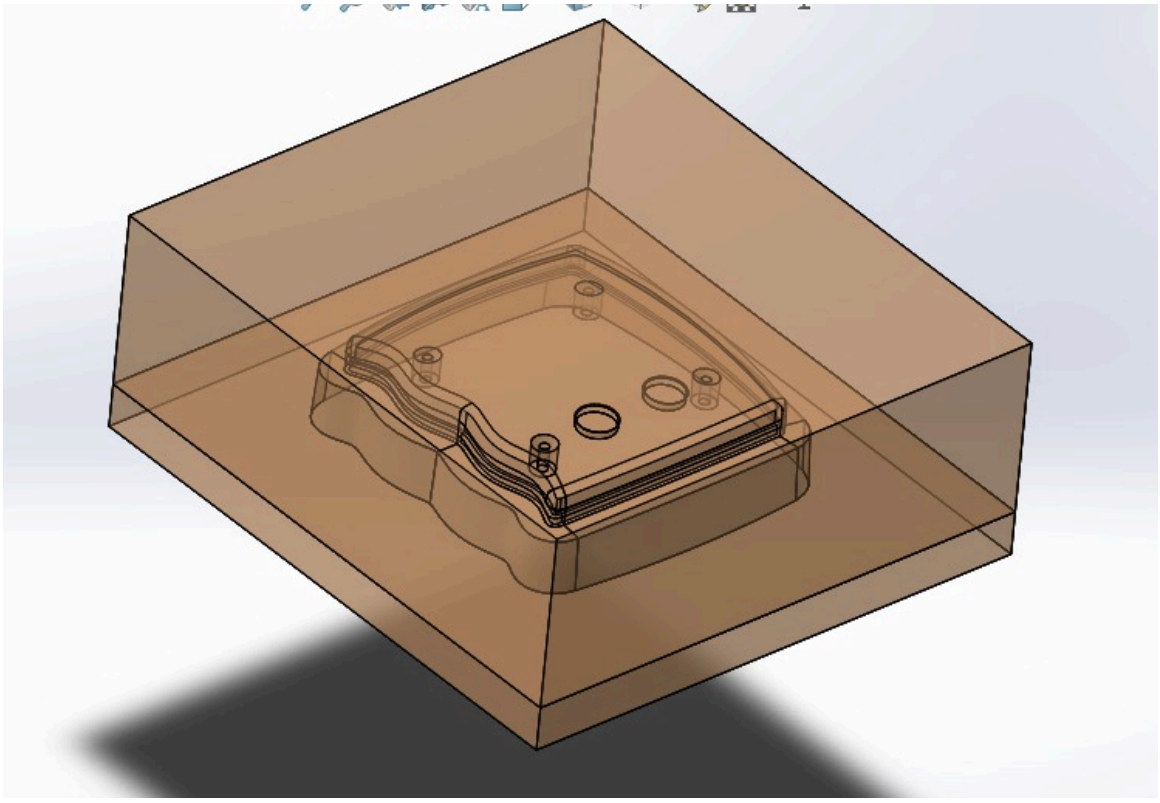


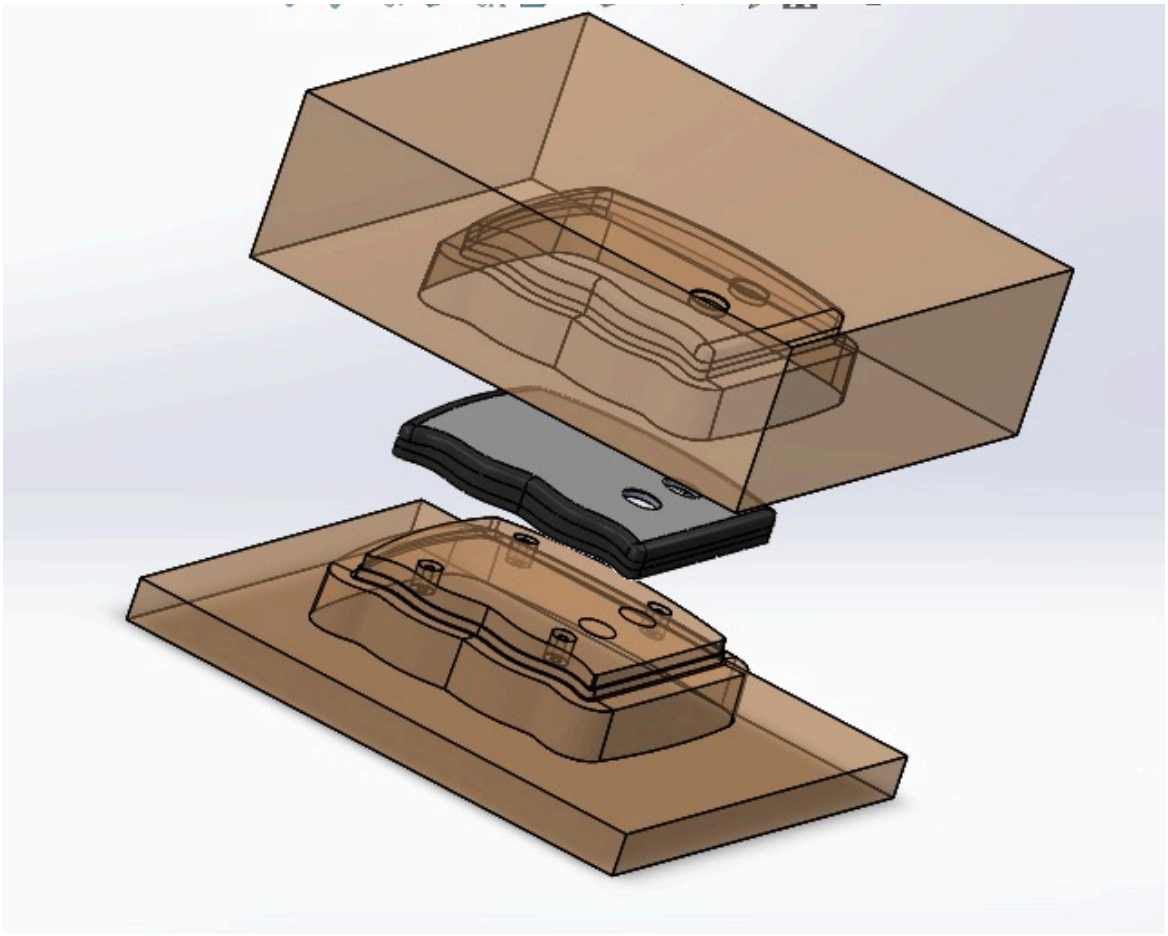
Top and Bottom Parts



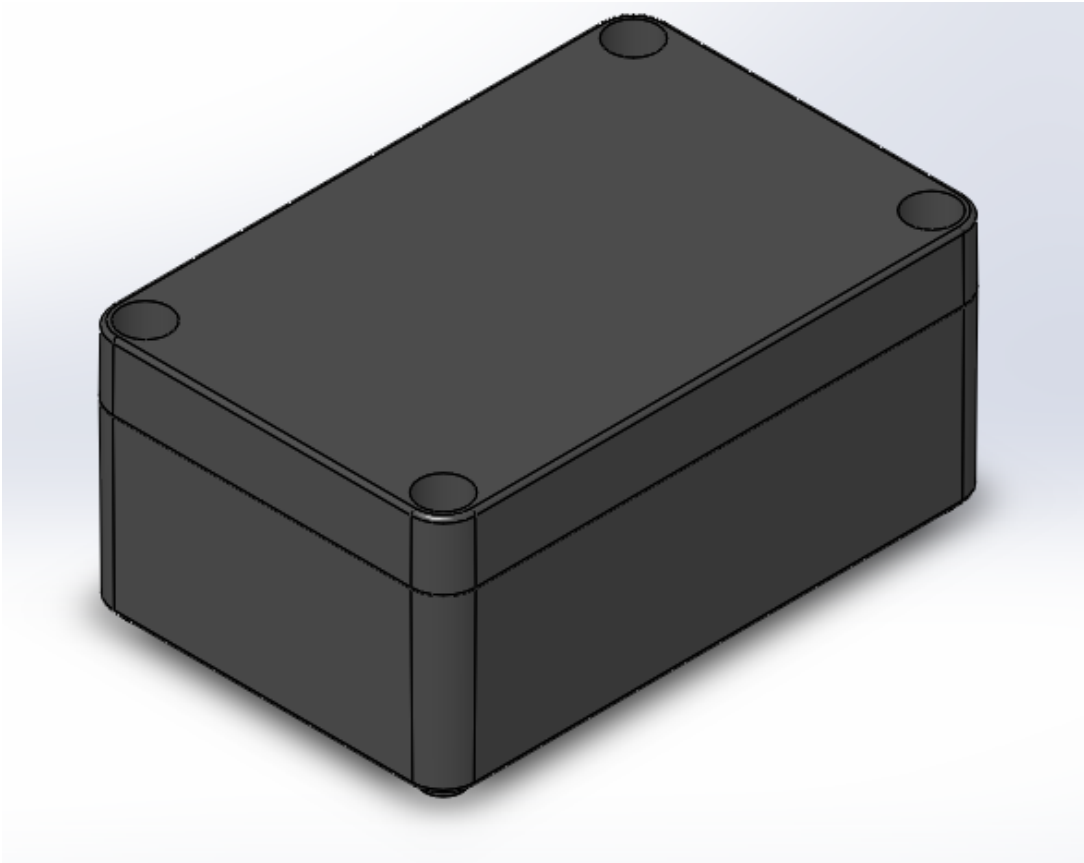
Design using Sketches

6.2 Mold Design





As Injection Molding is not accessible at the moment, a pre-manufactured enclosure was used for the evaluation. The necessary holes for the switches were cut using laser cutting.



7. Software Programming

7.1 Main Code

```
#include <SoftwareSerial.h>
#include <TinyGPS++.h>
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <DNSServer.h>
#include <ESP8266WebServer.h>
#include <WiFiManager.h>

WiFiClient espClient;
PubSubClient mqttClient(espClient);

//Initially defined SOS phone number with country code
String PHONE = "+94703760969";

#define rxGSM 12
#define txGSM 13
SoftwareSerial sim800(rxGSM,txGSM);

#define rxGPS 5
#define txGPS 4
SoftwareSerial neogps(rxGPS,txGPS);
TinyGPSPlus gps;

String smsStatus,senderNumber,receivedDate,msg;

#define SOS 14
int SOS_Time = 5;
int c = 0;
float latitude;
float longitude;
float velocity;
float height;
char latArr[10];
char longArr[10];
char veloArr[10];
char altiArr[10];
char phoneArr[12];

void setup() {

  Serial.begin(115200);
  Serial.println("Arduino serial initialize");

  sim800.begin(9600);
  Serial.println("SIM800L serial initialize");

  neogps.begin(9600);
  Serial.println("NodeMCU serial initialize");
```

```
  smsStatus = "";
  senderNumber="";
  receivedDate="";
  msg="";

  sim800.print("AT+CMGF=1\r"); //SMS text mode
  delay(1000);

  sim800.print("AT+CMGD=1,4");

  WiFiManager wifiManager;
  //wifiManager.resetSettings();
  wifiManager.autoConnect("GPSTrackerAP");
  Serial.println("Connected.......");
  sim800.print("AT+CMGF=1\r");
  delay(1000);
  sim800.print("AT+CMGS=\""+PHONE+"\"\r");
  delay(1000);
  sim800.print("Connected to WiFi");
  delay(100);
  sim800.write(0x1A); //ascii code for ctrl-26
  delay(1000);
  setupMqtt();
}

void loop() {
  if (digitalRead(SOS) == LOW){
      Serial.print("Calling In.."); // Waiting for 5 sec
      for (c = 0; c < SOS_Time; c++)
      {
        Serial.println((SOS_Time - c));
        delay(1000);
        if (digitalRead(SOS) == HIGH)
          break;
      }

      if (c == 5)
      {
        Serial.println("Sending location");
        sendLocation();
        delay(10000);
        sim800.println("ATD"+PHONE+";");
        delay(10000);
      }
  }

  if (!mqttClient.connected()) {
        connectToBroker();
  }

  mqttClient.loop();
  getLocation();
  dtostrf(latitude, 0, 6, latArr);
  dtostrf(longitude, 0, 6, longArr);
```

```
  dtostrf(velocity, 0, 6, veloArr);
  dtostrf(height, 0, 6, altiArr);
  PHONE.toCharArray(phoneArr,13);
  mqttClient.publish("200023C_LONGITUDE", longArr);
  mqttClient.publish("200023C_LATITUDE", latArr);
  mqttClient.publish("200023C_VELOCITY", veloArr);
  mqttClient.publish("200023C_ALTITUDE", altiArr);
  mqttClient.publish("200023C_SOSNUM", phoneArr);
  delay(1000);
}

void sendLocation()
{
  boolean newData = false;

  for (unsigned long start = millis(); millis() - start < 2000;)
  {
    while (neogps.available())
    {
      if (gps.encode(neogps.read()))
        {newData = true;}
    }
  }

  //If newData is true
  if(newData)
  {
    yield();
    Serial.print("Latitude= ");
    Serial.print(gps.location.lat(), 6);
    Serial.print(" Longitude= ");
    Serial.println(gps.location.lng(), 6);
    newData = false;

    String latitudeS = String(gps.location.lat(), 6);
    String longitudeS = String(gps.location.lng(), 6);

    String text = "Latitude= " + latitudeS;
    text += "\n\r";
    text += "Longitude= " + longitudeS;
    text += "\n\r";
    text += "Speed= " + String(gps.speed.kmph()) + " km/h";
    text += "\n\r";
    text += "Altitude= " + String(gps.altitude.meters()) + " meters";
    text += "\n\r";
    text += "http://maps.google.com/maps?q=" + latitudeS + "+" + longitudeS;

    delay(300);
    yield();
    sim800.print("AT+CMGF=1\r");
    delay(1000);
    sim800.print("AT+CMGS=\""+PHONE+"\"\r");
    delay(1000);
    sim800.print(text);
```

```
    delay(100);
    sim800.write(0x1A); //ascii code for ctrl-26
    delay(1000);
    Serial.println("GPS Location SMS Sent Successfully.");
  }
  else {Serial.println("Invalid GPS data");}
}

void getLocation() {
  boolean newData = false;

  for (unsigned long start = millis(); millis() - start < 2000;)
  {
    while (neogps.available())
    {
      if (gps.encode(neogps.read()))
        {newData = true;}
    }
  }

  //If newData is true
  if(newData)
  {
    latitude = gps.location.lat();
    longitude = gps.location.lng();
    velocity = gps.speed.kmph();
    height = gps.altitude.meters();
    newData = false;
  }
  else {Serial.println("Invalid GPS data");}
}

void setupMqtt() {
    mqttClient.setServer("test.mosquitto.org", 1883);
    mqttClient.setCallback(receiveCallback);
}

void receiveCallback(char *topic, byte *payload, unsigned int length) {
    Serial.print("Message arrived [");
    Serial.print(topic);
    Serial.print("] ");

    char payloadCharAr[length];
    for (int i = 0; i < length; i++) {
        Serial.print((char)payload[i]);
        payloadCharAr[i] = (char)payload[i];
    }
    Serial.println();
    Serial.println(PHONE);
    Serial.println(payloadCharAr);
    if(strcmp(topic,"200023C_SOS_SET") == 0) {
      for (int i = 0; i < 12; i++) {
        PHONE[i] = payloadCharAr[i];
      }
```

```
      Serial.println(PHONE);
    }
}

void connectToBroker() {
    while (!mqttClient.connected()) {
        Serial.print("Attempting MQTT connection... ");
        if (mqttClient.connect("ESP8266-200023C")) {
              sim800.print("AT+CMGF=1\r");
              delay(1000);
              sim800.print("AT+CMGS=\""+PHONE+"\"\r");
              delay(1000);
              sim800.print("Connected to Server");
              delay(100);
              sim800.write(0x1A); //ascii code for ctrl-26
              delay(1000);
            Serial.println("connected");
            //Subscribing To Topics
            mqttClient.subscribe("200023C_SOS_SET");
        }
        else {
            Serial.print("failed");
            Serial.println(mqttClient.state());
            delay(5000);
        }
    }
}
```

7.2 Node-Red JSON file

Code is attached as Appendix.

8. Production Procedure

PCB fabrication, soldering, testing, and enclosure manufacturing process are discussed in the following sections.

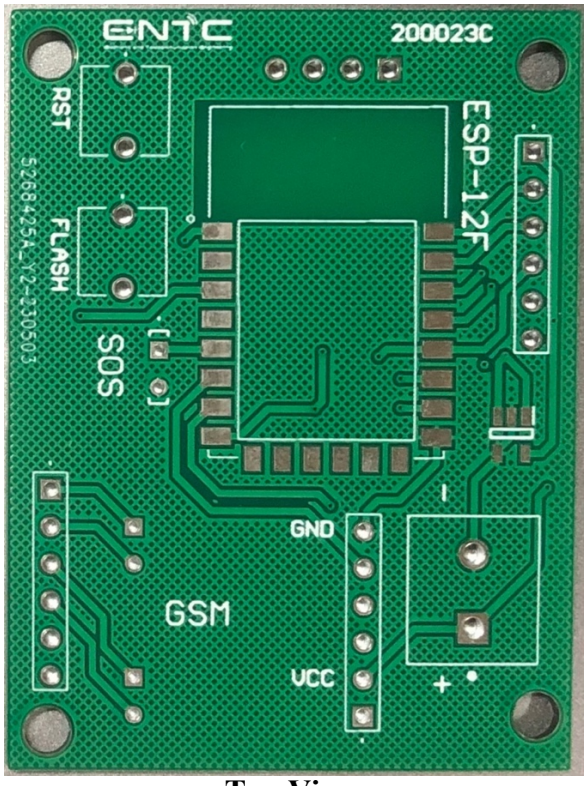
8.1 PCB Manufacturing

PCBs are designed using Altium Software. A single PCB is used for this Smart tracker. The PCB is double sided and the components are placed on both top and bottom layers. All the components except for the switch connections, battery connector, GPS and GSM modules are SMD (Surface Mounted Device). Since the facilities for manufacturing PCBs locally are lacking, PCB manufacturing is outsourced to a Chinese company.

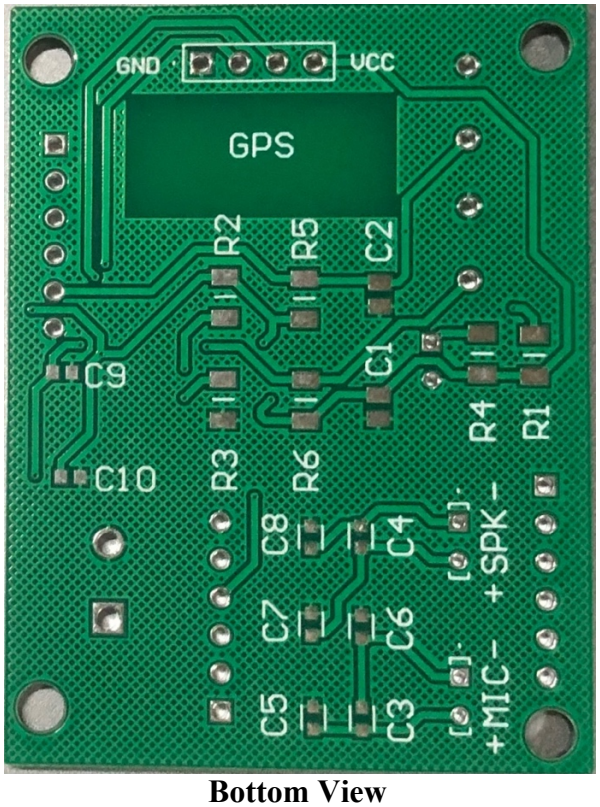
PCB Manufacturer: Jia Li Chuang (Hong Kong) Co., Limited (JLC PCB)

PCB Specifications:

- Layers : 2
- Thickness : 1.6 mm
- Surface Finish : HASL (with lead)
- Copper weight : 1 oz
- Material type : FR4-Standard Tg 130-140C



Top View



Bottom View

8.2 Soldering

8.2.1 Safety Precautions

Soldering process should be carried out by qualified technicians with a proper training.

- Workstation should be clean and dry.
- Wear protective gear to protect to inhalation of fumes.

- It is highly recommended not to touch or handle ICs with your bare hands, as there is a possibility for electrostatic discharge through the IC, which will damage it permanently. Use a conducting wristband and a grounding mat which will discharge the charge build-up to the ground.
- Soldering should be done with proper control of temperature and it is recommended to refer the datasheets of the components for temperature sensitivity to prevent any damage to components.
- Used solder sponges and contaminated rags should be placed in a sealable bag for disposal as hazardous waste
- Other than these, all the safety methods that are normally practiced during soldering must be used.

8.2.2 PCB Soldering

Both through-hole and SMD components are utilized in the smart tracker design, and it is advised to employ the hand soldering technique. Hand soldering is recommended due to the relatively fewer components, which helps in minimizing unnecessary expenses. Additionally, this method can reduce errors during the soldering process. Detailed instructions for component preparation, soldering equipment, and soldering components are provided below.

To prepare and perform the soldering process safely and effectively, follow the following guidelines:

1. Component Preparation:
   - Brush the component tips to eliminate any oxide layer or impurities that might hinder proper solder adhesion.
   - Wash the components with 99% isopropyl alcohol to ensure they are thoroughly clean and free from contaminants.

2. Application of Rosin Flux:
   - Apply Rosin flux to the surfaces that need to be soldered, as it aids in removing oxidized metal and ensures better soldering connections.

3. Soldering Iron Wattage:
   - Use a 30W soldering iron to strike an optimal balance between efficient soldering and preventing excessive heat that could damage sensitive components.

4. Recommended Soldering Iron Tip:
   - Opt for a Round B series tip, as it is specifically recommended for the soldering iron. This tip shape offers better control and precision during the soldering process.

5. Soldering Wire Gauge:
   - Use soldering wire with a gauge of 20 (0.813mm) to ensure an appropriate amount of solder for reliable connections.

6. Soldering Sequence:
   - Begin soldering with low-temperature sensitive components, gradually progressing to high-temperature sensitive ones. This approach minimizes the risk of heat damage to delicate components.
7. Caution with ESP 8266 Chip:
   - Be cautious with the ESP 8266 chip and avoid subjecting it to continuous high temperatures during soldering. Excessive heat can potentially damage the integrated circuit.

By following these guidelines, you can ensure that the soldering process is carried out meticulously, resulting in well-executed solder joints and minimizing the risk of damage to sensitive components. Proper soldering techniques and precautions contribute to the overall reliability and functionality of the smart tracker.

9. Assembly Process

The assembly of the GPS Tracker involves the following step-by-step instructions:

Step 1: **Prepare the PCB:**

- Gather all the components listed in the Bill of Materials (BOM).
- Refer to the electronic schematics for accurate placement and soldering of each component onto the PCB.
- Pay special attention to polarity and orientation to avoid any potential issues during assembly.

Step 2: **Assemble the Enclosure:**

- Utilize the provided 3D model and technical drawings to fabricate the GPS tracker's enclosure.
- Carefully place the assembled PCB inside the enclosure, ensuring that all interfaces align precisely with the designated openings.
- Secure the PCB in place using appropriate mounting hardware to prevent movement within the enclosure.

Assembled Product

Step 3: **Power Up and Initial Configuration:**

- Charge the GPS tracker's lithium-ion battery using the standard Micro-USB interface until it reaches full capacity.
- Power on the device and follow the initial configuration steps, such as setting communication parameters (Wi-Fi Connectivity) and device-specific settings (SOS Number).

Step 4: **Functional Testing:**

- Perform a series of functional tests to validate the GPS tracker's performance and reliability.
- Conduct a GPS acquisition test by taking the device outdoors to acquire satellite signals and verify the accuracy of the location data.
- Test the communication module by establishing a GSM/GPRS connection with a mobile phone, and ensure that location data is transmitted and received correctly.
- Evaluate the device's power-saving features to validate the efficiency of the battery utilization.

10. Testing for Functionality

To ensure the GPS Tracker's functionality, conduct comprehensive testing as follows:

a) **GPS Acquisition Test:**
- Take the GPS Tracker to an outdoor location with a clear view of the sky.
- Observe the time required for the GPS module to acquire satellite signals and determine the location.
- Record and analyze the accuracy of the location data displayed on the device.

b) **Communication Test:**
- Establish a GSM/GPRS connection between the GPS tracker and the remote server or the designated mobile phone.
- Monitor the successful transmission of real-time location data to the server or mobile device.
- Verify the accuracy and consistency of location data received on the server or mobile device.

c) **Power Saving Test:**
- Monitor and record the GPS tracker's power consumption during active tracking and standby modes.
- Assess the effectiveness of power-saving features in optimizing battery life without compromising performance.

d) **Enclosure Durability Test:**
- Subject the GPS tracker to environmental factors such as dust, humidity, and temperature variations.
- Examine the enclosure's ability to protect internal components from potential damage.

e) **Overall Performance Test:**
- Evaluate the GPS tracker's overall performance and reliability during an extended operational period.
- Address and resolve any issues encountered during testing to ensure a high-quality product.

11. Datasheets and Manuals

- ESP8266 Microcontroller
- U-blox Neo-6m GPS Module
- SIM800L GPS Module
- LDO Voltage Regulator

Appendix

Node-Red JSON File


```
[
    {
        "id": "9825727f6d477a0f",
        "type": "tab",
        "label": "GPS Tracker",
        "disabled": false,
        "info": "",
        "env": []
    },
    {
        "id": "496682f289842ef4",
        "type": "mqtt in",
        "z": "9825727f6d477a0f",
        "name": "",
        "topic": "200023C_LATITUDE",
        "qos": "2",
        "datatype": "auto-detect",
        "broker": "3211b65328dc6a82",
        "nl": false,
        "rap": true,
        "rh": 0,
        "inputs": 0,
        "x": 240,
        "y": 120,
        "wires": [
            [
                "b2f27e202cc23725"
            ]
        ]
    },
    {
        "id": "10052c42ec48db13",
        "type": "mqtt in",
        "z": "9825727f6d477a0f",
        "name": "",
        "topic": "200023C_LONGITUDE",
        "qos": "2",
        "datatype": "auto-detect",
        "broker": "3211b65328dc6a82",
        "nl": false,
        "rap": true,
        "rh": 0,
        "inputs": 0,
        "x": 240,
        "y": 180,
        "wires": [
            [
                "c396527084b94368"
            ]
        ]
```

```
},
{
    "id": "d80f04adbb1f18c2",
    "type": "debug",
    "z": "9825727f6d477a0f",
    "name": "debug 3",
    "active": true,
    "tosidebar": true,
    "console": false,
    "tostatus": false,
    "complete": "payload",
    "targetType": "msg",
    "statusVal": "",
    "statusType": "auto",
    "x": 600,
    "y": 120,
    "wires": []
},
{
    "id": "3679d6d8314b51e8",
    "type": "debug",
    "z": "9825727f6d477a0f",
    "name": "debug 4",
    "active": true,
    "tosidebar": true,
    "console": false,
    "tostatus": false,
    "complete": "payload",
    "targetType": "msg",
    "statusVal": "",
    "statusType": "auto",
    "x": 600,
    "y": 180,
    "wires": []
},
{
    "id": "851736b48cfc71d0",
    "type": "ui_text_input",
    "z": "9825727f6d477a0f",
    "name": "",
    "label": "Enter SOS Number :",
    "tooltip": "Enter Phone Number with the Country Code",
    "group": "1fdce47bb9e7e82c",
    "order": 2,
    "width": 12,
    "height": 1,
    "passthru": true,
    "mode": "text",
    "delay": "0",
    "topic": "topic",
    "sendOnBlur": true,
    "className": "",
    "topicType": "msg",
    "x": 200,
```

```
        "y": 580,
        "wires": [
            [
                "f9c611d02849ddb7",
                "0aaa3b70fc50d19f"
            ]
        ]
    },
    {
        "id": "b2f27e202cc23725",
        "type": "function",
        "z": "9825727f6d477a0f",
        "name": "function 4",
        "func": "flow.set(\"latitude\",msg.payload);\nreturn msg;",
        "outputs": 1,
        "noerr": 0,
        "initialize": "",
        "finalize": "",
        "libs": [],
        "x": 440,
        "y": 120,
        "wires": [
            [
                "d80f04adbb1f18c2"
            ]
        ]
    },
    {
        "id": "c396527084b94368",
        "type": "function",
        "z": "9825727f6d477a0f",
        "name": "function 5",
        "func": "flow.set(\"longitude\", msg.payload);\nreturn msg;",
        "outputs": 1,
        "noerr": 0,
        "initialize": "",
        "finalize": "",
        "libs": [],
        "x": 440,
        "y": 180,
        "wires": [
            [
                "3679d6d8314b51e8"
            ]
        ]
    },
    {
        "id": "57f30c1a224e3cb0",
        "type": "mqtt in",
        "z": "9825727f6d477a0f",
        "name": "",
        "topic": "200023C_VELOCITY",
        "qos": "2",
        "datatype": "auto-detect",
```

```
        "broker": "3211b65328dc6a82",
        "nl": false,
        "rap": true,
        "rh": 0,
        "inputs": 0,
        "x": 240,
        "y": 260,
        "wires": [
            [
                "a69b82f2c82ae157"
            ]
        ]
    },
    {
        "id": "953d958c682f2a83",
        "type": "debug",
        "z": "9825727f6d477a0f",
        "name": "debug 9",
        "active": true,
        "tosidebar": true,
        "console": false,
        "tostatus": false,
        "complete": "payload",
        "targetType": "msg",
        "statusVal": "",
        "statusType": "auto",
        "x": 600,
        "y": 220,
        "wires": []
    },
    {
        "id": "a69b82f2c82ae157",
        "type": "function",
        "z": "9825727f6d477a0f",
        "name": "function 9",
        "func": "flow.set(\"velocity\",msg.payload);\nmsg.payload =
msg.payload.toFixed(2);\nreturn msg;",
        "outputs": 1,
        "noerr": 0,
        "initialize": "",
        "finalize": "",
        "libs": [],
        "x": 440,
        "y": 260,
        "wires": [
            [
                "abdbc468e6a1c541",
                "e108f6c0c331d724"
            ]
        ]
    },
    {
        "id": "abdbc468e6a1c541",
        "type": "ui_gauge",
```

```
    "z": "9825727f6d477a0f",
    "name": "",
    "group": "1fdce47bb9e7e82c",
    "order": 3,
    "width": 6,
    "height": 5,
    "gtype": "gage",
    "title": "Speed",
    "label": "km/h ",
    "format": "{{value}}",
    "min": 0,
    "max": "100",
    "colors": [
        "#00b500",
        "#e6e600",
        "#ca3838"
    ],
    "seg1": "",
    "seg2": "",
    "diff": false,
    "classNamе": "",
    "x": 770,
    "y": 240,
    "wires": []
},
{
    "id": "e108f6c0c331d724",
    "type": "ui_chart",
    "z": "9825727f6d477a0f",
    "name": "",
    "group": "1fdce47bb9e7e82c",
    "order": 5,
    "width": 12,
    "height": 4,
    "label": "Speed",
    "chartType": "line",
    "legend": "false",
    "xformat": "HH:mm:ss",
    "interpolate": "cubic",
    "nodata": "",
    "dot": false,
    "ymin": "0",
    "ymax": "",
    "removeOlder": 1,
    "removeOlderPoints": "",
    "removeOlderUnit": "604800",
    "cutout": 0,
    "useOneColor": false,
    "useUTC": false,
    "colors": [
        "#1f77b4",
        "#aec7e8",
        "#ff7f0e",
        "#2ca02c",
```

```
            "#98df8a",
            "#d62728",
            "#ff9896",
            "#9467bd",
            "#c5b0d5"
        ],
        "outputs": 1,
        "useDifferentColor": false,
        "classname": "",
        "x": 790,
        "y": 300,
        "wires": [
            []
        ]
    },
    {
        "id": "398b28a3549b7c1b",
        "type": "ui_toast",
        "z": "9825727f6d477a0f",
        "position": "top right",
        "displayTime": "3",
        "highlight": "",
        "sendall": true,
        "outputs": 0,
        "ok": "OK",
        "cancel": "",
        "raw": false,
        "classname": "",
        "topic": "",
        "name": "",
        "x": 510,
        "y": 800,
        "wires": []
    },
    {
        "id": "0aaa3b70fc50d19f",
        "type": "debug",
        "z": "9825727f6d477a0f",
        "name": "debug 10",
        "active": true,
        "tosidebar": true,
        "console": false,
        "tostatus": false,
        "complete": "false",
        "statusVal": "",
        "statusType": "auto",
        "x": 460,
        "y": 680,
        "wires": []
    },
    {
        "id": "f9c611d02849ddb7",
        "type": "function",
        "z": "9825727f6d477a0f",
```

```
        "name": "function 10",
        "func": "if ((msg.payload.length != 12) || (msg.payload.slice(0,3) != \"+94\")) {\n
return { payload: \"Please Enter Valid Phone Number\" };\n}\nreturn {payload: msg.payload};",
        "outputs": 1,
        "noerr": 0,
        "initialize": "",
        "finalize": "",
        "libs": [],
        "x": 210,
        "y": 660,
        "wires": [
            [
                "109bfddbe98848d4"
            ]
        ]
    },
    {
        "id": "109bfddbe98848d4",
        "type": "switch",
        "z": "9825727f6d477a0f",
        "name": "",
        "property": "payload",
        "propertyType": "msg",
        "rules": [
            {
                "t": "eq",
                "v": "Please Enter Valid Phone Number",
                "vt": "str"
            },
            {
                "t": "nnull"
            }
        ],
        "checkall": "false",
        "repair": false,
        "outputs": 2,
        "x": 310,
        "y": 740,
        "wires": [
            [
                "398b28a3549b7c1b",
                "e1d48f8883617559"
            ],
            [
                "6af3fba7c6fe9249",
                "564eb63b9e5e1fed"
            ]
        ]
    },
    {
        "id": "e1d48f8883617559",
        "type": "debug",
        "z": "9825727f6d477a0f",
        "name": "debug 11",
```

```
        "active": true,
        "tosidebar": true,
        "console": false,
        "tostatus": false,
        "complete": "false",
        "statusVal": "",
        "statusType": "auto",
        "x": 540,
        "y": 740,
        "wires": []
    },
    {
        "id": "6af3fba7c6fe9249",
        "type": "debug",
        "z": "9825727f6d477a0f",
        "name": "debug 12",
        "active": true,
        "tosidebar": true,
        "console": false,
        "tostatus": false,
        "complete": "payload",
        "targetType": "msg",
        "statusVal": "",
        "statusType": "auto",
        "x": 500,
        "y": 860,
        "wires": []
    },
    {
        "id": "a9bf49ae1388b3f2",
        "type": "function",
        "z": "9825727f6d477a0f",
        "name": "function 11",
        "func": "msg.payload = {\n    \"name\" : \"Tracker map\",\n    \"lat\" :
flow.get(\"latitude\"),\n    \"lon\" : flow.get(\"longitude\"),\n    \"iconColor\" :
\"red\"\n}\nreturn msg;",
        "outputs": 1,
        "noerr": 0,
        "initialize": "",
        "finalize": "",
        "libs": [],
        "x": 610,
        "y": 540,
        "wires": [
            [
                "c5a8b4a4e8584acf",
                "ccea6746e655c64b"
            ]
        ]
    },
    {
        "id": "fa6f042017acea9c",
        "type": "debug",
        "z": "9825727f6d477a0f",
```

```
        "name": "debug 13",
        "active": true,
        "tosidebar": true,
        "console": false,
        "tostatus": false,
        "complete": "payload",
        "targetType": "msg",
        "statusVal": "",
        "statusType": "auto",
        "x": 780,
        "y": 680,
        "wires": []
    },
    {
        "id": "4d0b2c3756787c78",
        "type": "inject",
        "z": "9825727f6d477a0f",
        "name": "",
        "props": [
            {
                "p": "payload"
            }
        ],
        "repeat": "3",
        "crontab": "",
        "once": false,
        "onceDelay": 0.1,
        "topic": "",
        "payload": "{\"name\":\"Tracker map\"}",
        "payloadType": "json",
        "x": 370,
        "y": 480,
        "wires": [
            [
                "a9bf49ae1388b3f2"
            ]
        ]
    },
    {
        "id": "c5a8b4a4e8584acf",
        "type": "worldmap",
        "z": "9825727f6d477a0f",
        "name": "",
        "lat": "",
        "lon": "",
        "zoom": "",
        "layer": "OSMC",
        "cluster": "",
        "maxage": "",
        "usermenu": "show",
        "layers": "show",
        "panit": "false",
        "panlock": "false",
        "zoomlock": "false",
```

```
        "hiderightclick": "false",
        "coords": "none",
        "showgrid": "false",
        "showruler": "false",
        "allowFileDrop": "false",
        "path": "/worldmap",
        "overlist": "DR,CO,RA,DN,HM",
        "maplist": "OSMG,OSMC,EsriC,EsriS,EsriT,EsriDG,NatGeo",
        "mapname": "",
        "mapurl": "",
        "mapopt": "",
        "mapwms": false,
        "x": 800,
        "y": 620,
        "wires": []
    },
    {
        "id": "ce976b0ea46c316c",
        "type": "debug",
        "z": "9825727f6d477a0f",
        "name": "debug 14",
        "active": true,
        "tosidebar": true,
        "console": false,
        "tostatus": false,
        "complete": "payload",
        "targetType": "msg",
        "statusVal": "",
        "statusType": "auto",
        "x": 500,
        "y": 600,
        "wires": []
    },
    {
        "id": "ccea6746e655c64b",
        "type": "ui_worldmap",
        "z": "9825727f6d477a0f",
        "group": "bd76a47a08de8acc",
        "order": 1,
        "width": 12,
        "height": 14,
        "name": "",
        "lat": "7.8731",
        "lon": "80.7718",
        "zoom": "8",
        "layer": "OSMC",
        "cluster": "",
        "maxage": "600",
        "usermenu": "hide",
        "layers": "show",
        "panit": "false",
        "panlock": "false",
        "zoomlock": "false",
        "hiderightclick": "true",
```

```
        "coords": "deg",
        "showgrid": "false",
        "showruler": "false",
        "allowFileDrop": "false",
        "path": "/worldmap",
        "overlist": "DR,CO,RA,DN,HM",
        "maplist": "OSMG,OSMC,EsriC,EsriS,EsriT,EsriDG,NatGeo",
        "mapname": "",
        "mapurl": "",
        "mapopt": "",
        "mapwms": false,
        "x": 840,
        "y": 540,
        "wires": []
    },
    {
        "id": "05876c3451c55328",
        "type": "mqtt in",
        "z": "9825727f6d477a0f",
        "name": "",
        "topic": "200023C_SOSNUM",
        "qos": "2",
        "datatype": "auto-detect",
        "broker": "3211b65328dc6a82",
        "nl": false,
        "rap": true,
        "rh": 0,
        "inputs": 0,
        "x": 250,
        "y": 1020,
        "wires": [
            [
                "0894f454116064f4"
            ]
        ]
    },
    {
        "id": "1d8b674d0d052940",
        "type": "debug",
        "z": "9825727f6d477a0f",
        "name": "debug 15",
        "active": true,
        "tosidebar": true,
        "console": false,
        "tostatus": false,
        "complete": "true",
        "targetType": "full",
        "statusVal": "",
        "statusType": "auto",
        "x": 460,
        "y": 1020,
        "wires": []
    },
    {
```

```
    "id": "0894f454116064f4",
    "type": "ui_text",
    "z": "9825727f6d477a0f",
    "group": "1fdce47bb9e7e82c",
    "order": 1,
    "width": 0,
    "height": 0,
    "name": "",
    "label": "Current SOS Number : ",
    "format": "{{msg.payload}}",
    "layout": "row-spread",
    "className": "",
    "style": false,
    "font": "",
    "fontSize": 16,
    "color": "#000000",
    "x": 480,
    "y": 1100,
    "wires": []
},
{
    "id": "564eb63b9e5e1fed",
    "type": "mqtt out",
    "z": "9825727f6d477a0f",
    "name": "",
    "topic": "200023C_SOS_SET",
    "qos": "",
    "retain": "",
    "respTopic": "",
    "contentType": "",
    "userProps": "",
    "correl": "",
    "expiry": "",
    "broker": "3211b65328dc6a82",
    "x": 540,
    "y": 920,
    "wires": []
},
{
    "id": "9e4e963773ef8419",
    "type": "mqtt in",
    "z": "9825727f6d477a0f",
    "name": "",
    "topic": "200023C_ALTITUDE",
    "qos": "2",
    "datatype": "auto-detect",
    "broker": "3211b65328dc6a82",
    "nl": false,
    "rap": true,
    "rh": 0,
    "inputs": 0,
    "x": 300,
    "y": 400,
    "wires": [
```

```
            [
                "5450bd5d39fe03c6"
            ]
        ]
    },
    {
        "id": "dcaf8943879a3b50",
        "type": "debug",
        "z": "9825727f6d477a0f",
        "name": "debug 16",
        "active": true,
        "tosidebar": true,
        "console": false,
        "tostatus": false,
        "complete": "payload",
        "targetType": "msg",
        "statusVal": "",
        "statusType": "auto",
        "x": 480,
        "y": 340,
        "wires": []
    },
    {
        "id": "5450bd5d39fe03c6",
        "type": "function",
        "z": "9825727f6d477a0f",
        "name": "function 12",
        "func": "flow.set(\"altitude\",msg.payload);\nmsg.payload =
msg.payload.toFixed(2);\nreturn msg;",
        "outputs": 1,
        "noerr": 0,
        "initialize": "",
        "finalize": "",
        "libs": [],
        "x": 500,
        "y": 400,
        "wires": [
            [
                "66b32db09d9e361c",
                "196a0568fda0ee56"
            ]
        ]
    },
    {
        "id": "66b32db09d9e361c",
        "type": "ui_gauge",
        "z": "9825727f6d477a0f",
        "name": "",
        "group": "1fdce47bb9e7e82c",
        "order": 3,
        "width": 6,
        "height": 5,
        "gtype": "gage",
        "title": "Altitude",
```

```
        "label": "m",
        "format": "{{value}}",
        "min": 0,
        "max": "1000",
        "colors": [
            "#00b500",
            "#e6e600",
            "#ca3838"
        ],
        "seg1": "",
        "seg2": "",
        "diff": false,
        "className": "",
        "x": 840,
        "y": 380,
        "wires": []
    },
    {
        "id": "196a0568fda0ee56",
        "type": "ui_chart",
        "z": "9825727f6d477a0f",
        "name": "",
        "group": "1fdce47bb9e7e82c",
        "order": 5,
        "width": 12,
        "height": 4,
        "label": "Altitude",
        "chartType": "line",
        "legend": "false",
        "xformat": "HH:mm:ss",
        "interpolate": "cubic",
        "nodata": "",
        "dot": false,
        "ymin": "0",
        "ymax": "",
        "removeOlder": 1,
        "removeOlderPoints": "",
        "removeOlderUnit": "604800",
        "cutout": 0,
        "useOneColor": false,
        "useUTC": false,
        "colors": [
            "#1f77b4",
            "#aec7e8",
            "#ff7f0e",
            "#2ca02c",
            "#98df8a",
            "#d62728",
            "#ff9896",
            "#9467bd",
            "#c5b0d5"
        ],
        "outputs": 1,
        "useDifferentColor": false,
```

```
        "className": "",
        "x": 860,
        "y": 440,
        "wires": [
            []
        ]
    },
    {
        "id": "b3955c0f3bee21ce",
        "type": "mqtt in",
        "z": "9825727f6d477a0f",
        "name": "",
        "topic": "200023C_SOSALERT",
        "qos": "2",
        "datatype": "auto-detect",
        "broker": "3211b65328dc6a82",
        "nl": false,
        "rap": true,
        "rh": 0,
        "inputs": 0,
        "x": 260,
        "y": 1220,
        "wires": [
            [
                "2cfc93e151ec8b10",
                "a5e75a5d3c7d89c2"
            ]
        ]
    },
    {
        "id": "2cfc93e151ec8b10",
        "type": "ui_toast",
        "z": "9825727f6d477a0f",
        "position": "top left",
        "displayTime": "3",
        "highlight": "",
        "sendall": true,
        "outputs": 0,
        "ok": "OK",
        "cancel": "",
        "raw": false,
        "className": "",
        "topic": "ALERT",
        "name": "ALERT",
        "x": 500,
        "y": 1240,
        "wires": []
    },
    {
        "id": "a5e75a5d3c7d89c2",
        "type": "debug",
        "z": "9825727f6d477a0f",
        "name": "debug 17",
        "active": true,
```

```
        "tosidebar": true,
        "console": false,
        "tostatus": false,
        "complete": "true",
        "targetType": "full",
        "statusVal": "",
        "statusType": "auto",
        "x": 460,
        "y": 1340,
        "wires": []
    },
    {
        "id": "3211b65328dc6a82",
        "type": "mqtt-broker",
        "name": "",
        "broker": "test.mosquitto.org",
        "port": "1883",
        "clientid": "",
        "autoConnect": true,
        "usetls": false,
        "protocolVersion": "4",
        "keepalive": "60",
        "cleansession": true,
        "birthTopic": "",
        "birthQos": "0",
        "birthPayload": "",
        "birthMsg": {},
        "closeTopic": "",
        "closeQos": "0",
        "closePayload": "",
        "closeMsg": {},
        "willTopic": "",
        "willQos": "0",
        "willPayload": "",
        "willMsg": {},
        "userProps": "",
        "sessionExpiry": ""
    },
    {
        "id": "1fdce47bb9e7e82c",
        "type": "ui_group",
        "name": "SOS Number",
        "tab": "99355982a366f98f",
        "order": 2,
        "disp": true,
        "width": 12,
        "collapse": false,
        "className": ""
    },
    {
        "id": "bd76a47a08de8acc",
        "type": "ui_group",
        "name": "Location",
        "tab": "99355982a366f98f",
```

```
        "order": 1,
        "disp": true,
        "width": 12,
        "collapse": false,
        "className": ""
    },
    {
        "id": "99355982a366f98f",
        "type": "ui_tab",
        "name": "Home",
        "icon": "dashboard",
        "order": 1,
        "disabled": false,
        "hidden": false
    }
]
```