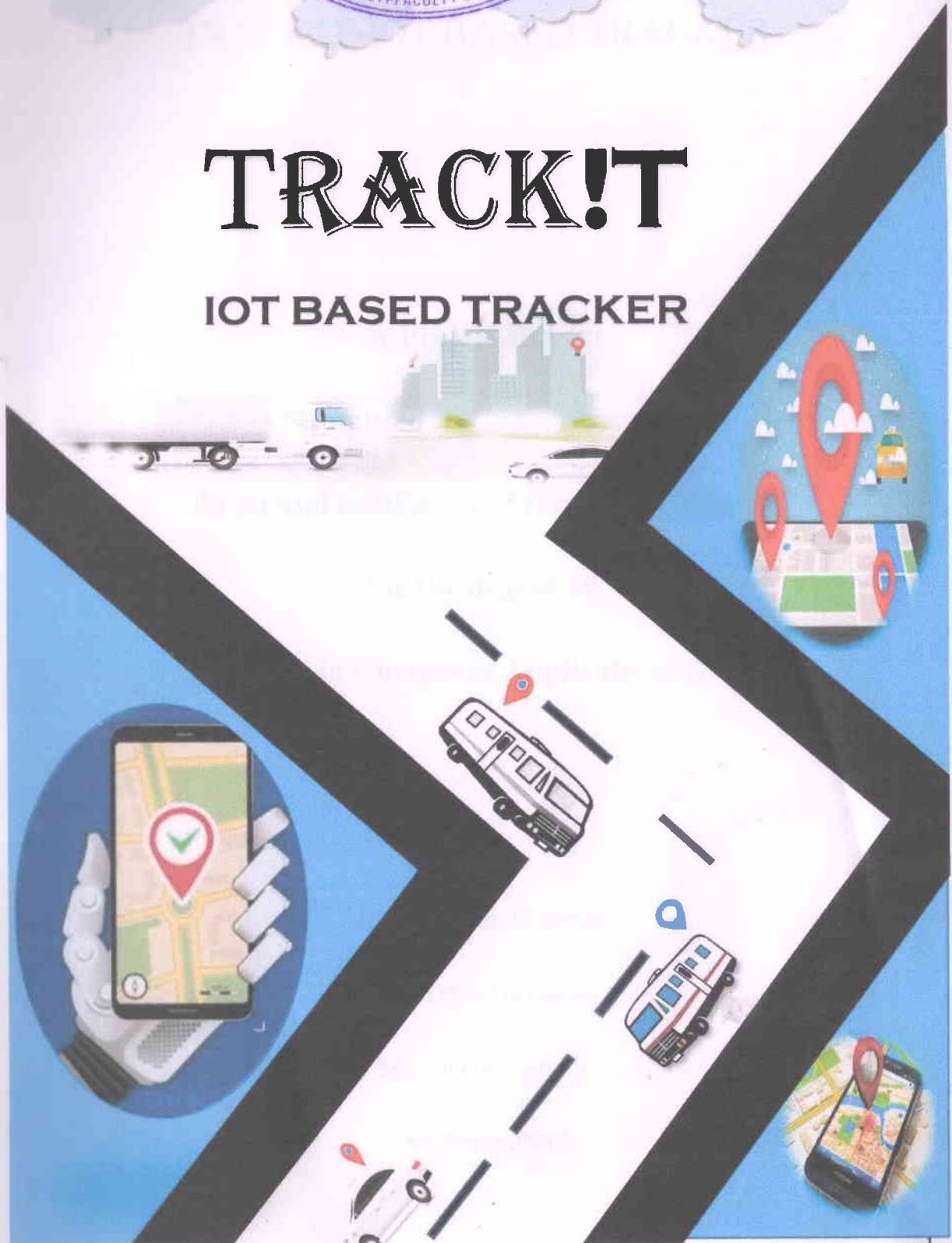


TRACK!T

IOT BASED TRACKER



TRACK!T-IOT BASED TRACKER

A Project Report

Submitted to Goa University

in partial fulfillment of the requirements

for the degree of

Bachelor in Computer Applications (BCA)

By

Mr. Manthan Gaunkar

Mr. Tejas Girodkar

Ms. Geeta Lamani

Mr. Aniket Naik

Ms. Shramaksha Naik

Mr. Diwyash Nitin Patade

Mr. Omkar Bibolkar

Guided By

Ms. Nisha Sawant



SPES's Shri Gopal Gaonkar Memorial Goa Multi-Faculty College

Affiliated to Goa University

CERTIFICATE

This is to certify that the project on

Track!t-IoT Based Tracker

has been successfully completed and submitted by

Mr. Manthan Gaunkar

Mr. Tejas Girodkar

Ms. Geeta Lamani

Mr. Aniket Naik

Ms. Shramaksha Naik


Mr. Diwyash Nitin Patade

Mr. Omkar Bibolkar



Ms. Nisha Sawant

(Internal Guide)



Dr. Shaikh M. Parvez Al-Usmani

(Principal)



Ms. Nisha Sawant

(Project Co-ordinator)

External

(Examiner)

Track!t-IoT Based Tracker

Roll No	Name	Goa University Reg. No	Signature
8308	Manthan Miranath Gaunkar	201801241	
8313	Tejas Anand Girodkar	201801238	
8315	Geeta Channappa Lamani	201801238	
8320	Aniket Mangaldas Naik	201801246	
8325	Shramaksha Shrikant Naik	201801257	
8327	Diwyash Nitin Patade	201801241	
8341	Omkar Damodar Bibolkar	201801248	



SPES Shri Gopal Gaonkar Memorial Goa Multi-Faculty College

Affiliated to Goa University

CERTIFICATE BY THE GUIDE

This is to certify that the project report is the record of the whole work done by the candidates themselves under my guidance during the period of study and that to the best of my knowledge; it has not previously formed the basis for the award of any diploma or degree by this on any other university.

Name of college: SPES's Goa Multi-Faculty College

Course: Bachelor in Computer Applications (BCA)

Academic Year: 2020-2021

Table of Contents

Ms. Nisha Sawant

Project Guide

A square box containing a handwritten signature in black ink. The signature reads "Nisha Sawant" with a horizontal line underneath the name.

Table of Contents

ACKNOWLEDGEMENT	11
PROJECT PROFILE	12
INTRODUCTION	13
EXISTING SYSTEM	15
PROPOSED SYSTEM	16
SYSTEM FEATURES	17
OBJECTIVE	18
WORKING PRINCIPAL	19
HARDWARE	20
Sim808 Module (GeoLocation)	20
Arduino Uno	21
Jumper Cable	22
GPS Antenna	23
GSM Antenna	24
Power Adapter	25
USB A to B cable	26
RS 232 cable	27
SOFTWARE	28
Arduino IDE	28
GSM Technology	29
GPS Technology	30
ACTIVITY DIAGRAMS	31
USE CASE DIAGRAM	33
CIRCUIT DIAGRAM	34
MODEL	35
ARDUINO IDE SETUP	38
TESTING CODE	40
HARDWARE CONNECTION TEST	52
Serial Monitor	52
Serial Connections	52
BIBLIOGRAPHY	56

ACKNOWLEDGEMENT

It is our immense pleasure to have worked on the project titled “Track!T – IOT Based Tracker”. This project consumed huge amount of work, research and dedication, still implementation would not have been possible if we did not have support of many individuals. Therefore, we would like to extend our sincere gratitude to all of them.

We would like to show our gratitude to our internal guide Ms. Nisha Sawant for her constant support, motivation, inspiration for this project throughout numerous consultations and for guiding us to make this project a success. As well as Tehnical support & guidance Nisha Sawat from Edutronics.

We also extend our sincere gratitude toward our Principal, Dr. Shaikh Mohammad Parvez Al-Usmani, for his wholehearted support in this project.

Nevertheless, we express our sincere gratitude to our teaching and non-teaching staff members and friends for their kind co-operation and encouragement throughout this project.

The Project Team

PROJECT PROFILE

Project name	Track!t-IoTBased Tracker
Objective	To find & locate lost Assets.
Software Required	Arduino IDE, GPS technology, GSM technology
Hardware requirements	Arduino board, Breadboard, Servo motor, Ultrasonic sensor, jumper wire, Led.
Internal Guide	Ms.Nisha Sawant

INTRODUCTION

Every day we see technology moving ahead with a steady pace, introducing new features & gimmicks to get our attention into buying them. Though there are people who migrate to the newer products, there are still a significant amount of people which tend to stick to the old ones they have. Due to reasons ranging from the product carrying emotional value or just not being able to afford the latest technology.

Let us take the example of vehicle, since it is one of the primary object in our project. Vehicles come equipped with GPS. It helps you to navigate your way and also keep track of the car, but not all released cars come equipped with GPS. New cars released often have variants .The lower variants, sometimes exclude GPS tracking along with many other things to make it more affordable for consumer. We also have to take the old vehicles into consideration since they didn't have GPS tracking system at that time.

In a country where we prefer mileage over any luxury, it is easy for people to take the budget approach and go for a vehicle that gets the job done, excluding any other features. There are still lots of people driving older cars, not having any security features besides the lock and alarm in some cars. But these don't matter when your car is stolen and the thief is on the run. Since the car didn't had any tracking system integrated into it.

This is where we would like to introduce TrackIt, A device utilizing global positioning system (GPS) to keep track of the latitudes & longitudes 24x7 , giving you real time position of the vehicle, so even if it is stolen and away from you, you'll always know it's location. This will help you to catch the thief.

Track!t-IoT Based Tracker

Since it's a separate device you can hide it in any place where you feel convenient, away from prying eyes. Thieves can often destroy the GPS system on the dashboard to prevent the car from being tracked but Track!t is an independent device, so they will be unaware of a tracker giving away the cars' location.

We plan on making it as compact as we can. So you can find easier to hide and not restrict it to only bigger assets like vehicles or big baggage. We want to make sure you never lose your stuff again.

EXISTING SYSTEM

Older generation cars as well as new cars (Lower variants) does not come equipped with GPS tracking features in case of theft. However, having pre-installed GPS might not be a completely foolproof solution. In a Worst case Scenario cars that have GPS system can easily be sabotaged by the carjacker when he is on the run to avoid being followed.

The same can be applied to Luggage or Large Valuable Assets that needs to be monitored.

Having a compact portable tracker hidden in unsuspecting location might as well be the Anti-Theft Insurance.

PROPOSED SYSTEM

The device will serve as a viable backup in case of future theft and will allow the user to catch the thief immediately in case the asset is stolen. It will also allow the user to monitor his car/asset by knowing its location. Its small size will allow the user to hide it discreetly from prying eyes in any location he desires.

SYSTEM FEATURES

- Location Tracking.
- Easy installation.
- Versatile.
- Affordable.
- Reliable
- Value for Money

OBJECTIVE

Our main objective is to secure the assets of the user. The user can easily place it in his car or other assets, so in case it is stolen, the user can track it down by sending it a message and getting its geographical location coordinates (latitude and longitude).

WORKING PRINCIPAL

In this project, Arduino will use for controlling the whole process with a GPS Receiver and GSM module. GPS Receiver is used for detecting coordinates of the embedded system and GSM module is used for sending the coordinates to user by SMS . When we turn on embedded system on first GPS module will collect the latitude and longitude of the embedded system and it will send the data to the Arduino and then Arduino will send the data to the GSM module and GSM module will send the SMS to the user.

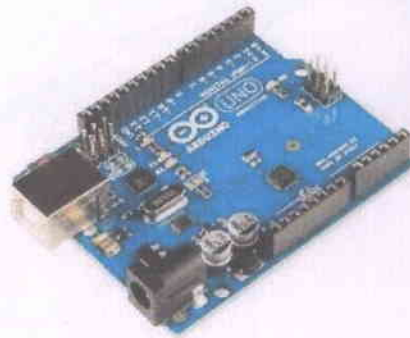


HARDWARE



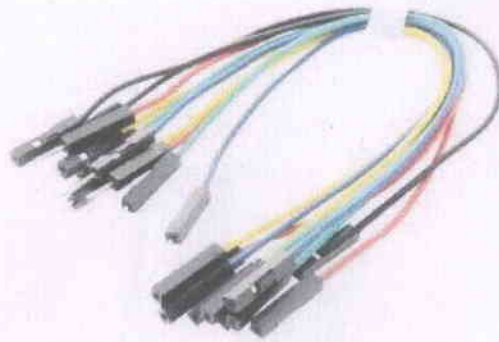
Sim808 Module (GeoLocation)

SIM808 Module is a complete Quad-Band GSM/GPRS module which combines GPS technology for satellite navigation. The compact design which integrated GPRS and GPS in a SMT package will significantly save both time and costs for customers to develop GPS enabled applications. Featuring an industry standard interface and GPS function, it allows variable assets to be tracked seamlessly at any location and anytime with signal coverage.



Arduino Uno

It is an open-source platform, means the boards and software are readily available and anyone can modify and optimize the boards for better functionality .The software used for Arduino devices is called IDE (Integrated Development Environment) which is free to use and required some basic skills to learn it. It can be programmed using C and C++ language. Some people get confused between Microcontroller and Arduino. While former is just an on system 40 pin chip that comes with a built-in microprocessor and later is a board that comes with the microcontroller in the base of the board, boot loader and allows easy access to input-output pins and makes uploading or burning of the program very easy.



Jumper Cable

Jumper cables, also known as booster cables or jump leads, are a pair of insulated wires of sufficient capacity with alligator clips at each end to interconnect the disabled equipment/vehicle with an auxiliary source, such as another vehicle or equipment with the same system voltage or to another battery.



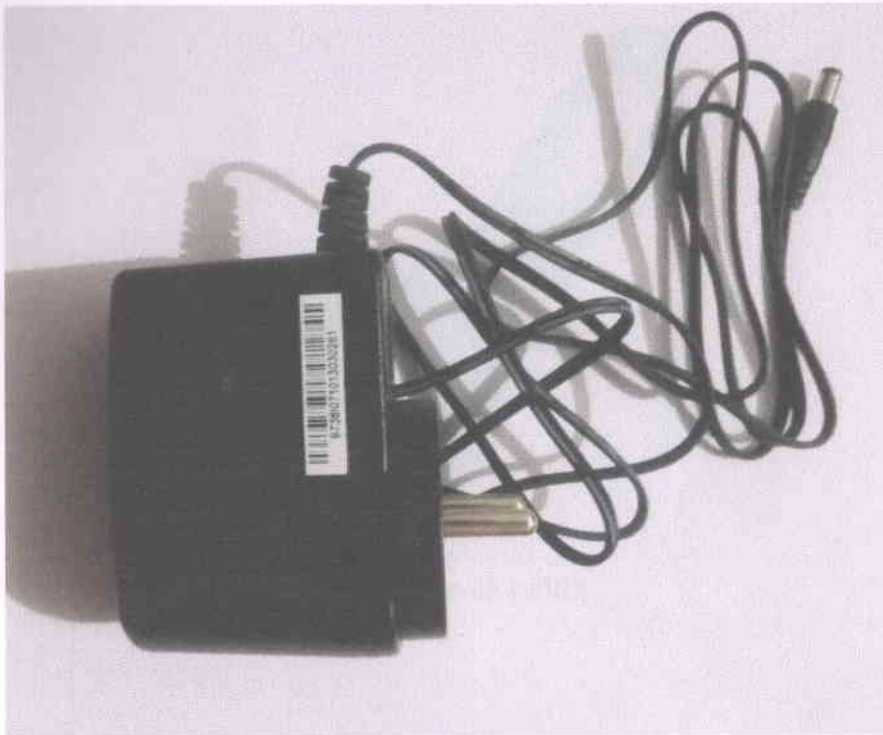
GPS Antenna

A GNSS or GPS antenna is a device designed to receive and amplify the radio signals transmitted on specific frequencies by GNSS satellites and convert them to an electronic signal for use by a GNSS or GPS receiver. The output of the GNSS or GPS antenna is fed into a GNSS or GPS receiver that can compute the position.



GSM Antenna

GSM Antenna is one kind of Antenna to transmit GSM signal at specified frequency 850, 900, 1800, 1900, 2100MHz. 2100MHz is separated into UMTS (Universal Mobile Telecommunications System) frequency band for 3G, and other frequencies are separated into 2G.GSM Coverage Maps and Roaming Information.



Power Adapter

A power supply for electronic devices. Also called an "AC adapter" or "charger," power adapters plug into a wall outlet and convert AC to a single DC voltage. Computers use multiple DC voltages, and the power adapter is the external part of the power supply for a laptop.

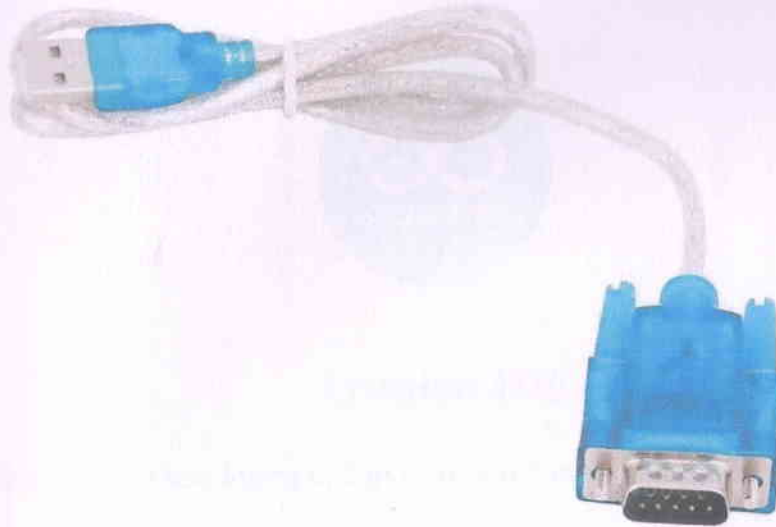


USB A to B cable

This is a Cable for board compatible with arduino (USB A to B). You can use it to connect "Arduino Uno", "Arduino Mega 2560" compatible boards or any board with the USB female A port of your computer.

Features:

- USB Type: Type-A to Type-B.
- Length: 25-30 cm.
- Fully compatible with the PC.



RS 232 cable

It is a form of serial data transmission. It is a form of communication which is also called as serial connection. It is one of the most used form of data transmission. It has the standard 9 pin D-sub connector. The RS232 connection transmits signals using a positive voltage for a binary 0 and a negative voltage for a binary 1.

SOFTWARE



Arduino IDE

Arduino IDE (Integrated Development Environment) introduced by Arduino.cc is opensource development software, uses for writing and compiling programs then uploading them to any Arduino shields. Arduino IDE software runs in a different operating system like Windows, Linux, and MAC that supports both C and C++ programming languages. Arduino IDE is available online and offline.



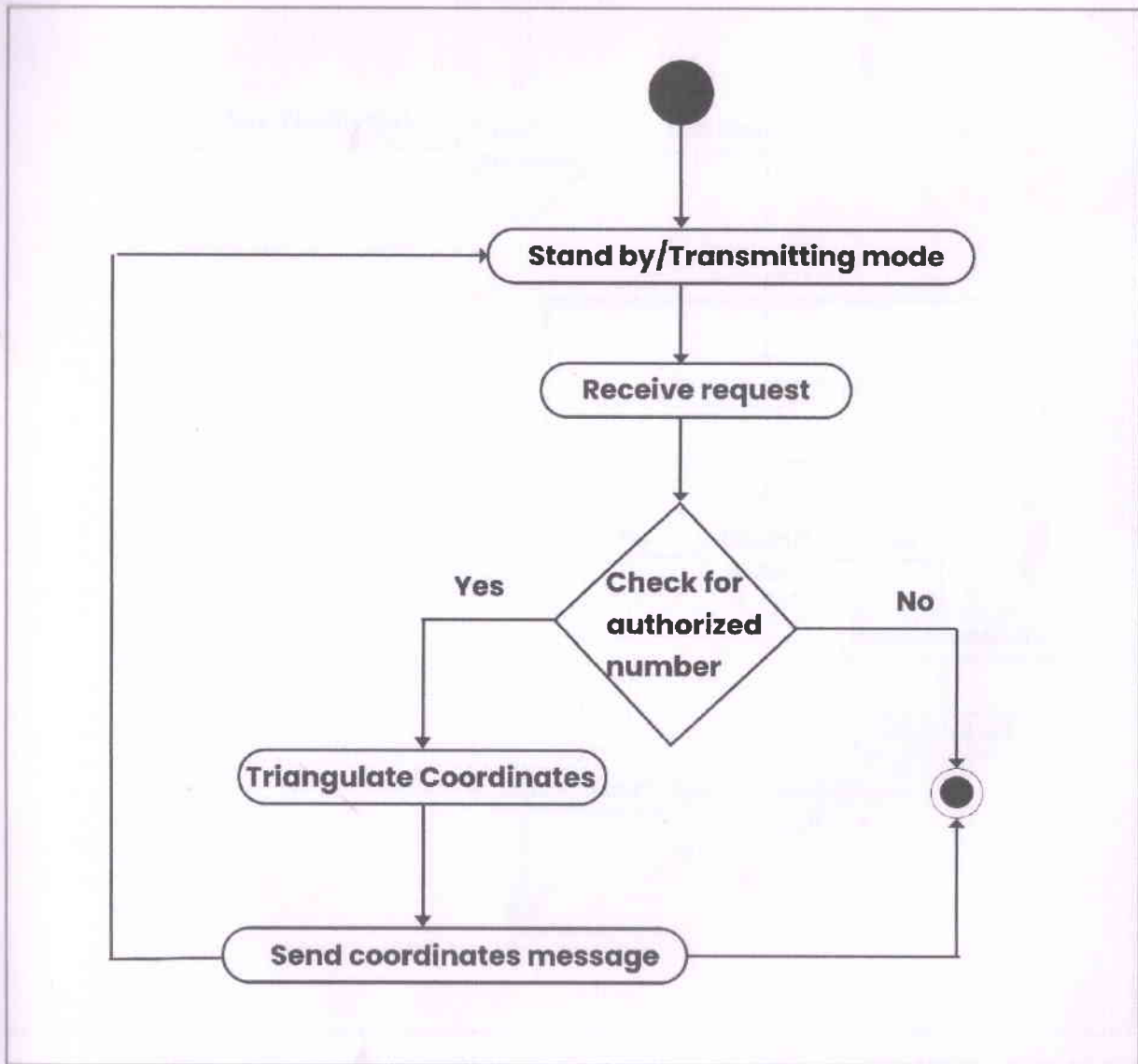
GSM Technology

GSM, a short form of Global System for Mobile Communication, is a cellular network that operates in various frequency ranges, i.e. 850MHz, 900MHz, 1800MHz and 1900MHz for transmitting data and broadly used by mobile phone customers in Europe and Asia including other parts of the world. The technology is accepted globally for digital cellular communication.

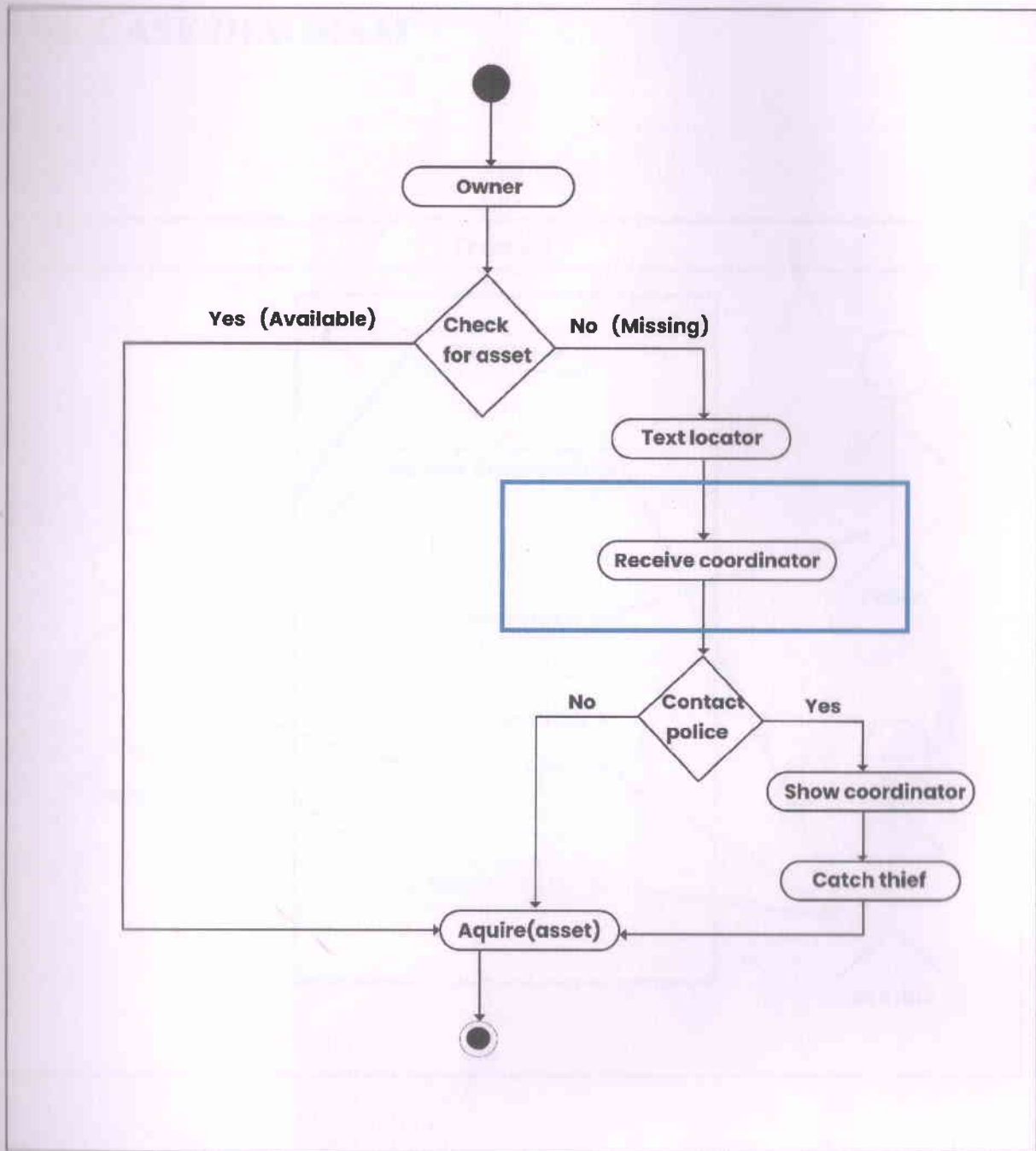
GPS Technology

GPS Technology GPS (Global Positioning System) is a navigation system based on the network of 24 satellites initially and positioned into trajectory or orbit. The satellites send signals or data to the GPS receiver on the earth to find the position, time, speed, and direction on any point on the planet. GPS works without any subscription fee at any situation regardless of weather condition everywhere in the world at any time. The system is worldwide accepted by aviation, navy, civilians, land survey and many more.

ACTIVITY DIAGRAMS

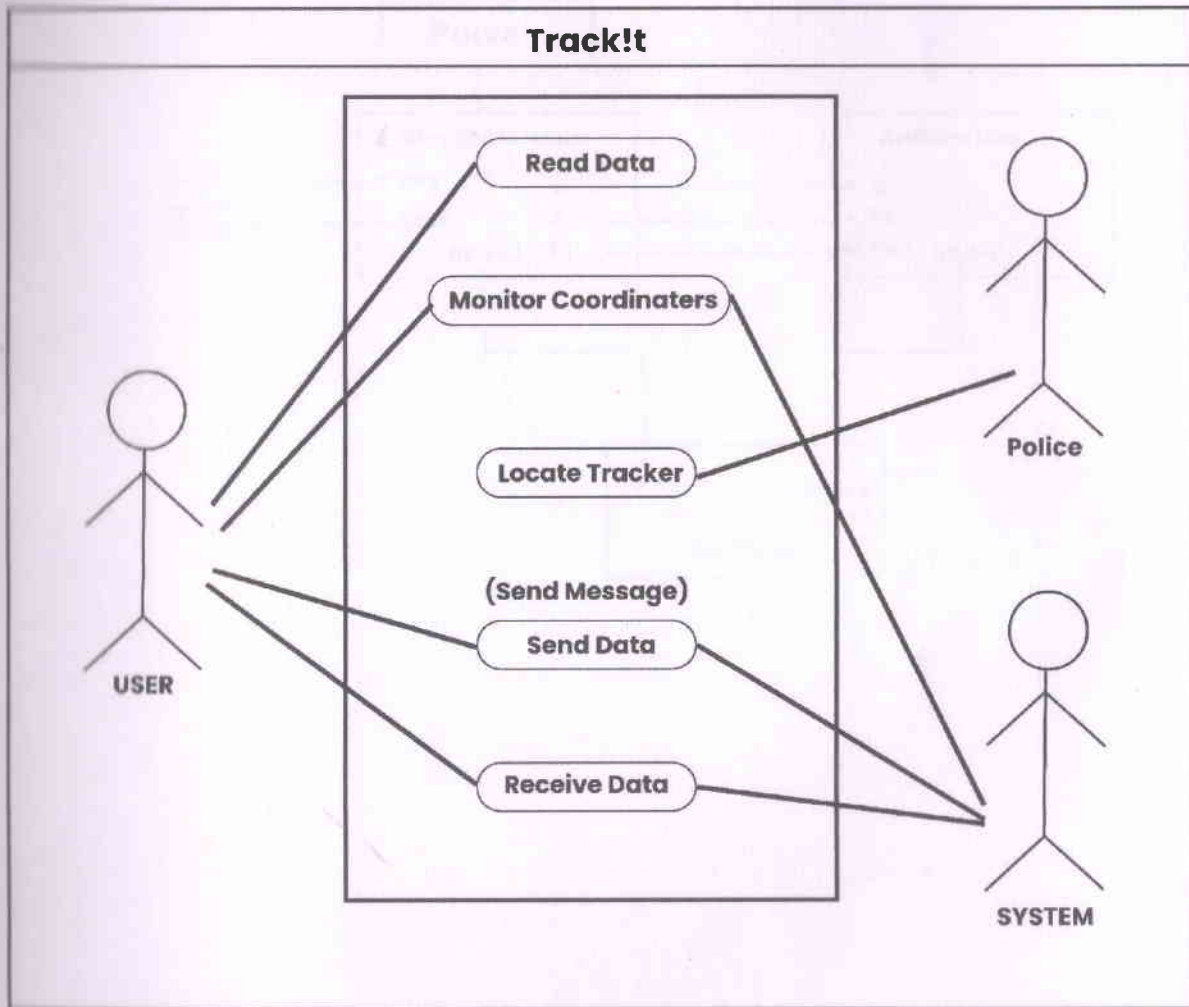


Device Point of View

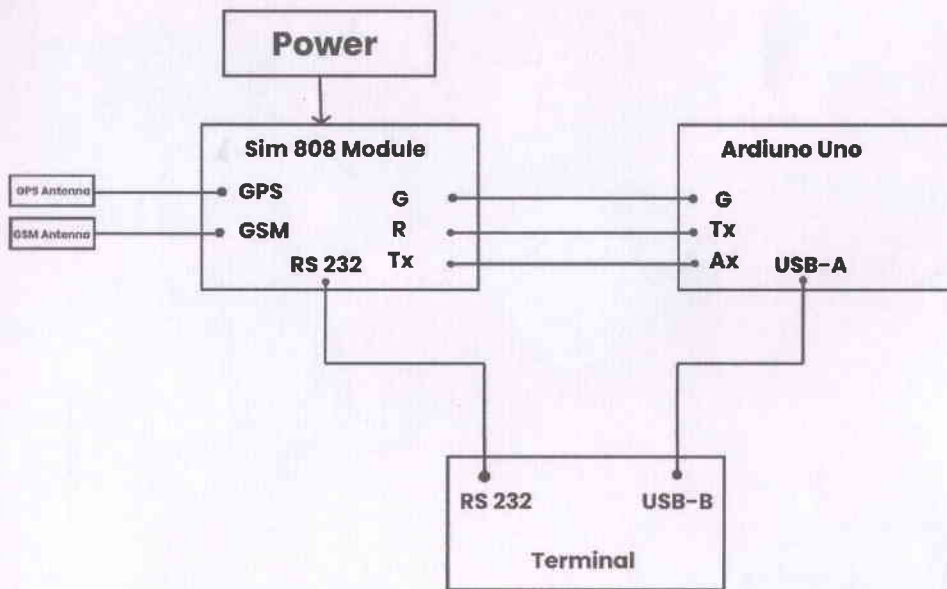


User Point of View

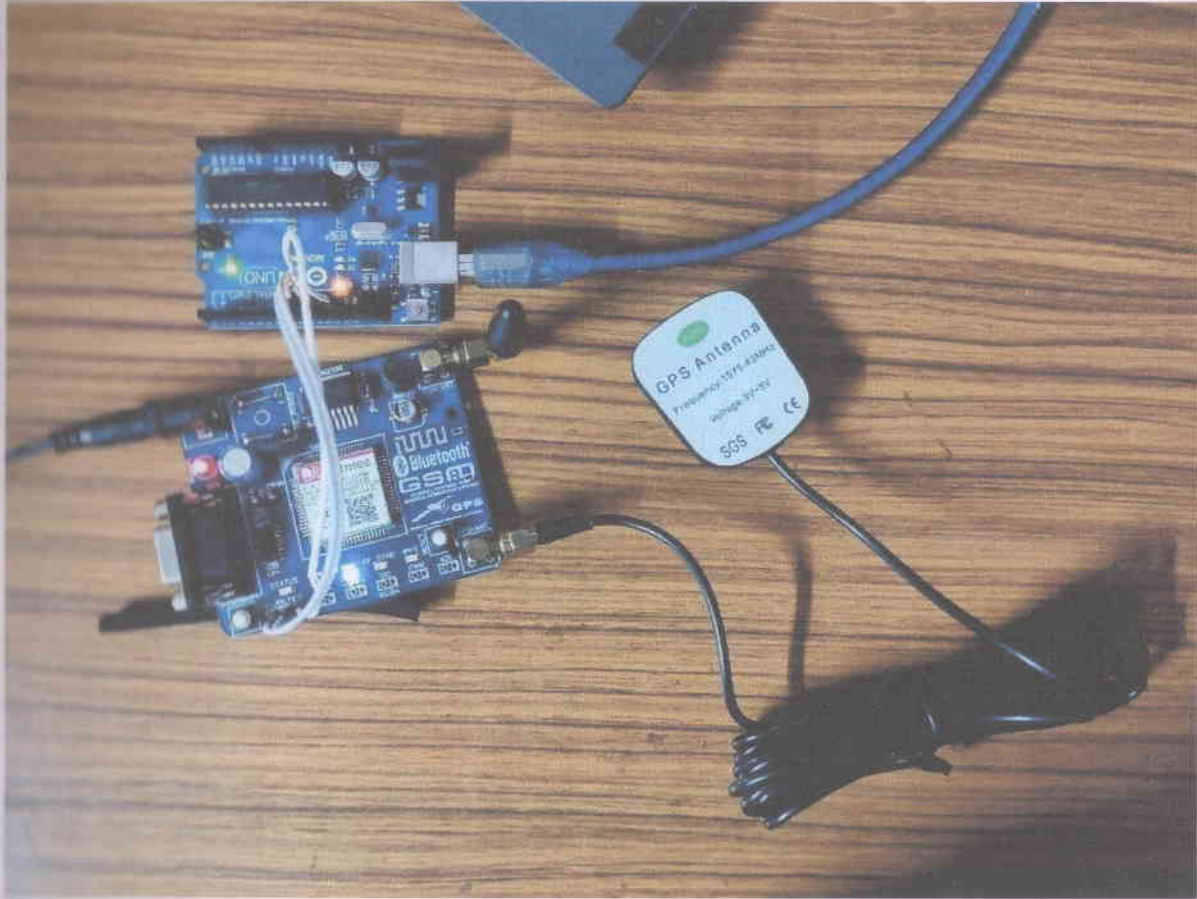
USE CASE DIAGRAM

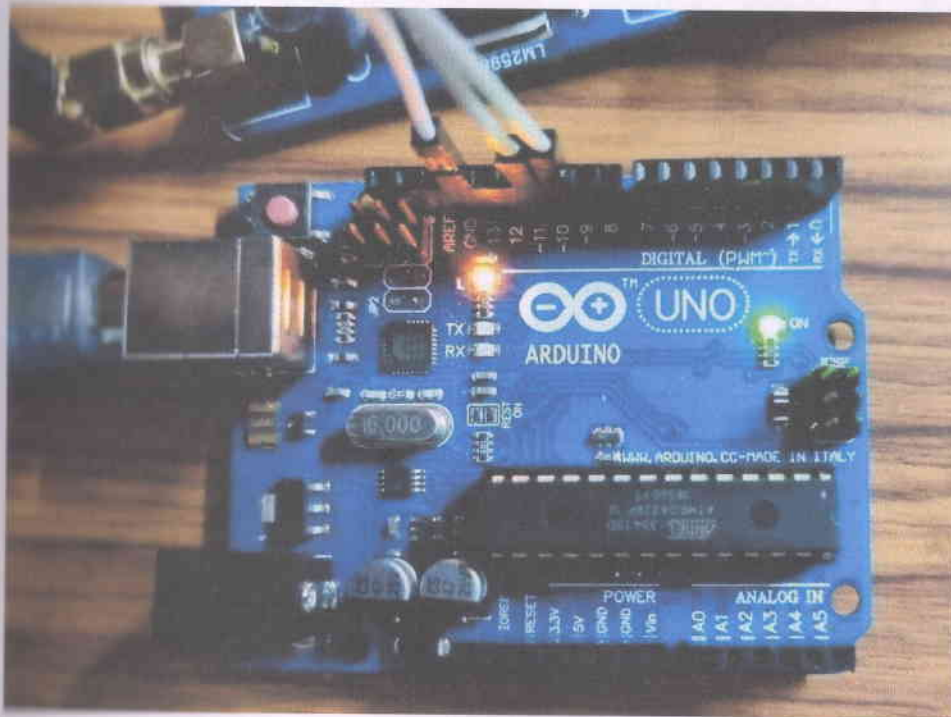
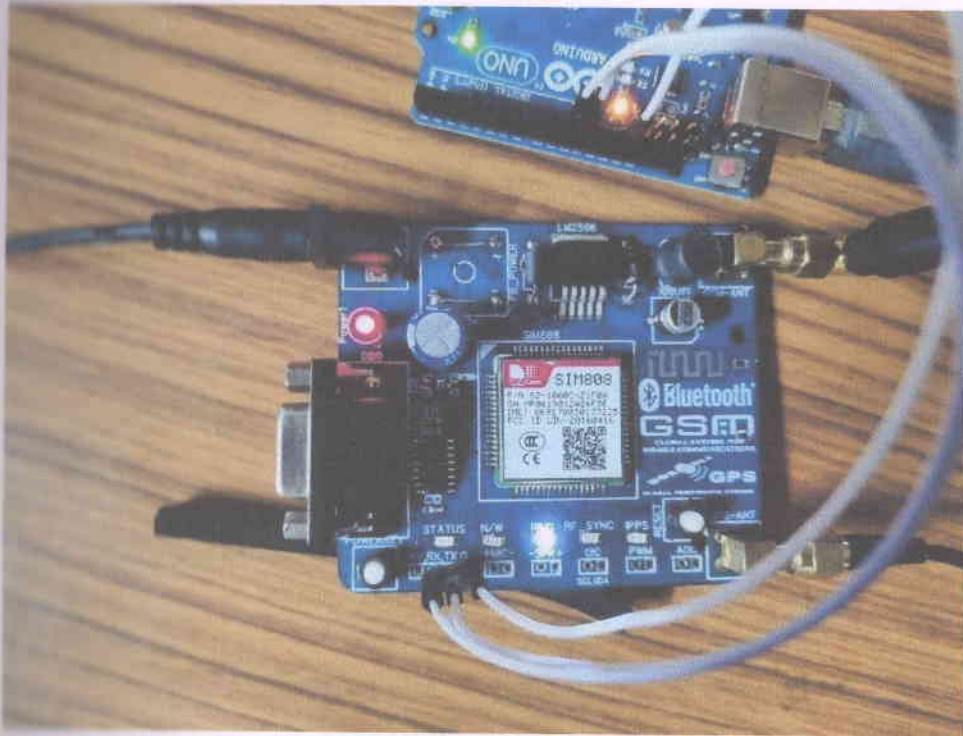


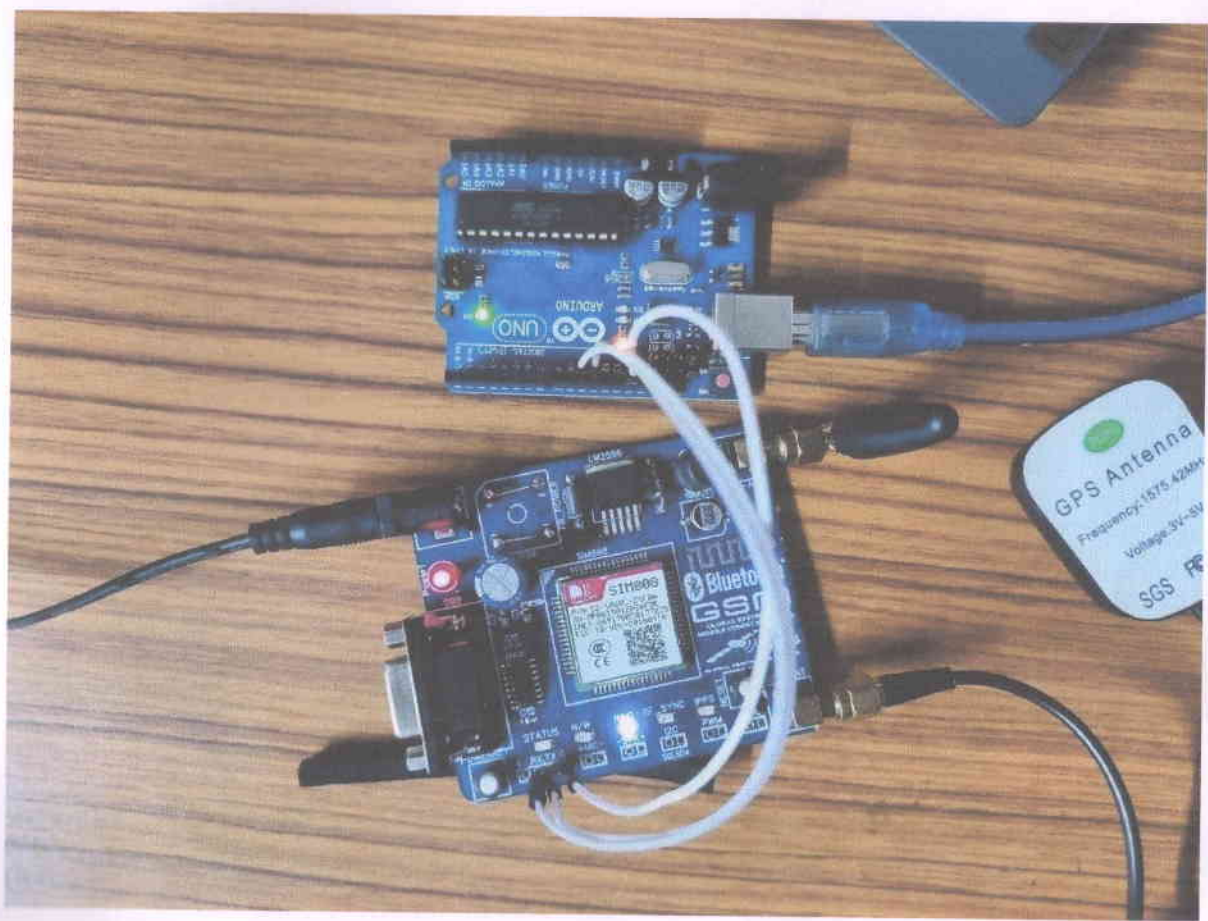
CIRCUIT DIAGRAM



MODEL







ARDUINO IDE SETUP

```

sketch_mqtt (Arduino)
File Edit Sketch Tools Help
sketch_mqtt
void readSMS();

void setup()
{
  mySerial.begin(5600);
  Serial.begin(9600);

  //***** Initialize SIM700 module *****
  while(!sim700.init())
  {
    Serial.println("SIM700 init error!\n");
    delay(1000);
  }
  delay(3000);

  Serial.println("SIM init success");

  Serial.println("Init success, please send SMS message to me!");
}

void loop()
{
  //***** Detecting unread SMS *****
  messageIndex = sim700.isSMSunread();
}

```

Copy error message

```

sketch_mqtt (Arduino)
File Edit Sketch Tools Help
sketch_mqtt
void loop()
{
  //***** Detecting unread SMS *****
  messageIndex = sim700.isSMSunread();

  //***** At least, there is one UNREAD SMS *****
  if (messageIndex > 0)
  {
    readSMS();
    getGPS();
    sendSMS();

    //***** Turn OFF the GPS power *****
    sim700.detachGPS();

    Serial.println("Please send SMS message to me!");
  }
}

void readSMS()
{
  Serial.println("messageToRead: ");
}

```

Copy error message

TrackIt-IoT Based Tracker

```
void readSMS()
{
  Serial.print("messageIndex: ");
  Serial.println(messageIndex);

  sim908.readSMS(messageIndex, message, MESSAGE_LENGTH, phone, datetime);

  //*****In order not to full SIM Memory, is better to delete it*****
  sim908.deleteSMS(messageIndex);
  Serial.println("From number: ");
  Serial.println(phone);
  Serial.println("Datetime: ");
  Serial.println(datetime);
  Serial.println("Received Message: ");
  Serial.println(message);
}

void getGPS()
{
  while (!sim908.attachGPS())
  {
    Serial.println("Open the GPS-power failure");
    delay(1000);
  }
}
```

```
void sendSMS()
{
  Serial.println("start to send message ...");

  Serial.println(MESSAGE);
  Serial.println(phone);

  sim908.sendSMS(phone, MESSAGE);
}
```

TESTING CODE

```
void setup()
{
  mySerial.begin(9600);
  Serial.begin(9600);

  //***** Initialize sim808 module *****

  while(!sim808.init())
  {
    Serial.print("Sim808 init error\r\n");
    delay(1000);
  }
  delay(3000);

  Serial.println("SIM Init success");

  Serial.println("Init Success, please send SMS message to me!");
}
```

Void setup() explained:

1. `Serial.begin(9600)`

----The code starts with initialising the serial communication, so that the Arduino can send out

commands through the USB connection. The value 9600 is called the 'baud rate' of the connection.

This is how fast the data is to be sent.

2. `while(!sim808.init())`

```
{  
  Serial.print("Sim808 init error\r\n");  
  delay(1000);  
}
```

----This while block checks whether or not the sim808 module has initialised. Till the module starts, the loop will keep on

printing "Sim808 init error" on the serial monitor with a delay of 1 seconds.

3. `Serial.println("SIM Init success");`

```
Serial.println("Init Success, please send SMS message to me!");
```

----once the sim808 module gets initialised, "SIM Init success", "Init Success, please send SMS message to me!"

gets printed on the serial monitor.

```
-----  
void loop()  
{  
  //***** Detecting unread SMS *****  
  messageIndex = sim808.isSMSunread();  
  
  //***** At least, there is one UNREAD SMS *****  
  if (messageIndex > 0)  
  {  
  
    readSMS();  
    getGPS();  
    sendSMS();  
  
    //***** Turn off the GPS power *****  
    sim808.detachGPS();  
  
    Serial.println("Please send SMS message to me!");  
  }  
}
```

Void loop() Explained:

1. messageIndex = sim808.isSMSUnread()

----This line of code checks for any unread messages, and sets the 'messageIndex' value greater than 0 depending on the number of unread messages.

2. if (messageIndex > 0)

----This if block gets executed only if the 'messageIndex' value is greater than 0, which means there are unread messages.

3. sim808.detachGPS()

----Once all the functions ["readSMS()", "getGPS()", "sendSMS()"] get executed successfully, the GPS inside the sim808

module will turn off and the module will be on standby for the next message.

```
void readSMS()
{
  Serial.print("messageIndex: ");
  Serial.println(messageIndex);
}
```



```
sim808.readSMS(messageIndex, message, MESSAGE_LENGTH, phone,
datetime);

//*****In order not to full SIM Memory, is better to delete it*****

sim808.deleteSMS(messageIndex);
Serial.print("From number: ");
Serial.println(phone);
Serial.print("Datetime: ");
Serial.println(datetime);
Serial.print("Recieved Message: ");
Serial.println(message);
}
```

void readSMS() Explained:

1. sim808.readSMS(messageIndex, message, MESSAGE_LENGTH, phone, datetime)

----This line will get all the detail of the message such as index, the actual message, message length, phone number, datetime of the message into variables.

2. sim808.deleteSMS(messageIndex);

----This line deletes the sms from the SIM memory in order not to fill the SIM memory.

3. Serial.print("From number: ");

```
Serial.println(phone);
```

```
Serial.print("Datetime: ");
```

```
Serial.println(datetime);
```

```
Serial.print("Recieved Message: ");
```

```
Serial.println(message);
```

----This part will print phone, datetime and message on the serial port.

```
void getGPS()
{
  while(!sim808.attachGPS())
  {
    Serial.println("GPS power failure");
    delay(1000);
  }
  delay(3000);
```

```
Serial.println("GPS power success");

while(!sim808.getGPS())
{

}

Serial.print(sim808.GPSdata.year);
Serial.print("/");
Serial.print(sim808.GPSdata.month);
Serial.print("/");
Serial.print(sim808.GPSdata.day);
Serial.print(" ");
Serial.print(sim808.GPSdata.hour);
Serial.print(":");
Serial.print(sim808.GPSdata.minute);
Serial.print(":");
Serial.print(sim808.GPSdata.second);
Serial.print(":");
Serial.println(sim808.GPSdata.centisecond);
Serial.print("latitude :");
Serial.println(sim808.GPSdata.lat);
Serial.print("longitude :");
Serial.println(sim808.GPSdata.lon);
```

```
Serial.print("speed_kph :");  
Serial.println(sim808.GPSdata.speed_kph);  
Serial.print("heading :");  
Serial.println(sim808.GPSdata.heading);  
Serial.println();  
  
float la = sim808.GPSdata.lat;  
float lo = sim808.GPSdata.lon;  
float ws = sim808.GPSdata.speed_kph;  
  
dtostrf(la, 4, 6, lat); //put float value of la into char array of lat. 4 = number of  
digits before decimal sign. 6 = number of digits after the decimal sign.  
dtostrf(lo, 4, 6, lon); //put float value of lo into char array of lon  
dtostrf(ws, 6, 2, wspeed); //put float value of ws into char array of wspeed  
  
sprintf(MESSAGE, "Latitude : %s\nLongitude : %s\nWind Speed : %s kph\nMy  
Module Is Working. Try With This Link.\nhttp://www.latlong.net/Show-Latitude-  
Longitude.html\nhttp://maps.google.com/maps?q=%s,%s\n", lat, lon, wspeed, lat,  
lon);  
}
```

void getGPS() Explained:

1. while(!sim808.attachGPS())

```
{  
  Serial.println("GPS power failure");  
  delay(1000);  
}
```

----This while block will check for GPS power and until it turns on successfully it will keep on printing

"GPS power failure" on the serial monitor with a delay of 1 second

2. Serial.print(sim808.GPSdata.year);

```
Serial.print("/");  
Serial.print(sim808.GPSdata.month);  
Serial.print("/");  
Serial.print(sim808.GPSdata.day);  
Serial.print(" ");  
Serial.print(sim808.GPSdata.hour);  
Serial.print(":");  
Serial.print(sim808.GPSdata.minute);  
Serial.print(":");  
Serial.print(sim808.GPSdata.second);  
Serial.print(":");  
Serial.println(sim808.GPSdata.centisecond);  
Serial.print("latitude :");  
Serial.println(sim808.GPSdata.lat);
```

```
Serial.print("longitude :");  
Serial.println(sim808.GPSdata.lon);  
Serial.print("speed_kph :");  
Serial.println(sim808.GPSdata.speed_kph);  
Serial.print("heading :");  
Serial.println(sim808.GPSdata.heading);  
Serial.println();
```

----This will print datetime, latitude, longitude, speed on the serial monitor.

3. float la = sim808.GPSdata.lat;

```
float lo = sim808.GPSdata.lon;  
float ws = sim808.GPSdata.speed_kph;
```

----This will store the latitude, longitude and speed from the GPS data into float type variables.

4. dtostrf(la, 4, 6, lat);

```
dtostrf(lo, 4, 6, lon);  
dtostrf(ws, 6, 2, wspeed);
```

----This will format the latitude, longitude and speed data so that it has 4 digits before the decimal point and 6 digits

after the decimal point and store them into new variables.


```
-----  
  
void sendSMS()  
{  
  Serial.println("Start to send message ...");  
  
  Serial.println(MESSAGE);  
  Serial.println(phone);  
  
  sim808.sendSMS(phone,MESSAGE);  
}
```

void **sendSMS()** Explained:

1. **Serial.println("Start to send message ...")**

---Prints "Start to send message ..." on the serial monitor.

2. **Serial.println(MESSAGE);**

Serial.println(phone);

----Prints the message with coordinates and the google maps link and the phone number on the serial monitor.

3. sim808.sendSMS(phone, MESSAGE);

----This will send the message containing the coordinates and the google maps link to the phone number from which the message was received.

HARDWARE CONNECTION TEST

This test is conducted to check whether all the assembled elements configured correctly. Two LED lights, LED ON and LED L blinked on Arduino UNO after a 9V source applied along with successful power-up on a display screen. Red LED in the modem flashed on after it got the voltage. Work and NET status LED light switched on when the power key triggered for 2 seconds. The GPS status LED light (blue) started functioning after the modem connected to the satellite signal. The blinking nature of the network led provided the following information for visual confirmation. 64ms on and 800ms off- the network was not connected 64ms on, 3 seconds off-cellular connection was successful and could send/receive voice and SMS 64ms on, 300ms off - the GPRS data connection was working.

Serial Monitor

Arduino IDE has a separate pop-up window called serial monitor, in which the computer interface with Arduino to communicate with GPRS/GPS/GSM module. The serial monitor interacts by sending and receiving serial data. It shows the status of the component. The serial monitor makes it easier to debug Arduino sketches.

Serial Connections

Software serial port selection from jumper cap was a crucial part of serial communication. SIM808 hooked up to Arduino through software UART connecting the RX and TX terminal of the shield to the board, so-called a cross-linked between ports.

AT commands, also known as Hayes Smart modem commands (by Hayes Microcom puter Products, Inc.), controls any popular modems through the series of instructions for various operations. AT stands for attention. The AT command performs different actions on supporting modems like dialing phone, sending SMS, powering module, inquiring the quality of GSM, fax, information and configuration on SIM, module and so on. The command line starts with the prefix AT which tells modem about the beginning of the instruction or code. The device uses those commands to communicate with computer.

The table 1 below shows a few required samples AT command and their description. Used in the project.

<u>AT Command</u>	<u>Description</u>
AT	Match baud rate/Check interface
ATI	Recognition of shield
AT+CCID	SIM confirmation

Track!t-IoT Based Tracker

AT+CREG	Check network registration
AT+CSQ	Provide signal quality
AT+CMGF	Text message format
AT+CNMI	To specify how to handle the newly arrived message
AT+CMGS	Send message
AT+CMGR	Read message
AT+CGNSTST	Send GNSS data to AT UART
AT+CGNSPWR	Power control of GNSS

Track!t-IoT Based Tracker

AT+CGNSSEQ	Define the last NMEA sentences that parsed
AT+CGNSINF	Read NMEA sentences

BIBLIOGRAPHY

- <https://alselectro.wordpress.com/2018/05/04/sim808-gps-gsm-module-sending-gps-location-sms/>
- <https://www.arduino.cc/en/software>
- <https://circuitdigest.com/microcontroller-projects/vehicle-tracking-system-using-arduino-gps-and-gsm>
- <https://simcom.ee/modules/gsm-gprs-gnss/sim808/>
- <https://www.arduino.cc>