**SMART GPS TRACKER USING ARDUINO**

**J Component Project Report for the course**

**CSE3009 Internet of Things**

*by*

**Rekha V(19BCE1871)**

*Submitted to*

**Priyaadharshini M**

****

**ABSTRACT**

Parents often worry about their children when they are away from them. Have they safely reached school? Are they alone at home? Or are they at the playground with friends? – these are some of the questions that bothers parents all over the world. The smart GPS tracker using Arduino helps us to keep track of a child. Besides this, the device can also be used to track your vehicle location and other objects.

Here in this system, we are using the GSM module for sending the coordinates of vehicle on mobile phone via message. GPS is sending the coordinates continuously in form of string. After reading this string using Arduino extract the required data from string and then sends it to mobile phone using GSM module via SMS. This information is called latitude and longitude. GPS used 3 or 4 satellite for tracking the location of any vehicle.

The proposed trackers make good use of a popular technology that enables communication between a Smartphone and the tracker and thus plays a significant role in the transportation system as passengers can see the real time of arriving busses or trains or airplanes at the platforms on LCD or on mobiles.

Table of contents

**Title Page No.**

1. **Abstract 2**
2. **Objective 4**
3. **Hardware and software Used 5**
4. **Modules Used 5**

1. **Architecture 6**

**5.1 Software simulation**

**5.2** **Hardware Simulation**

1. **Software Simulation 9**

1. **Hardware implementation 10**
2. **Results and Analysis 13**
3. **Coding 17**
4. **Conclusion and Future enhancement 25**

**Contribution 26**

**Reference**

**OBJECTIVES**

There is an ever-increasing need for trackers, which can be life-saving devices. We often hear people go missing and vehicles getting stolen. This project enables the third party for tracking the location of any person, vehicle or other objects. This is an effective method for tracking people as well as theft control in terms of any vehicle or other equipment and as spy equipment. It also comes handy during disasters to keep track of victims. The proposed trackers make good use of a popular technology that enables communication between a Smartphone and the tracker. It will also play a significant role in the transportation system as passengers can see the real time of arriving busses or trains or airplanes at the platforms on LCD or on mobiles. The proposed Arduino based tracking system uses global positioning system and global system using GSM modules. GSM module with a SIM card is used for purpose of communication. The project aims at building a cost-effective smart tracker that returns the real-time longitude and latitude (GPS coordinates). It aims at achieving two-way communication using GSM modem. This system can be installed or hidden in your vehicle. After installing the circuit, we can easily track the stolen vehicle using a mobile phone. Parents can also use this application to track their children by placing this circuit inside the school/college bag.

**HARDWARE/ SOFTWARE USED:**

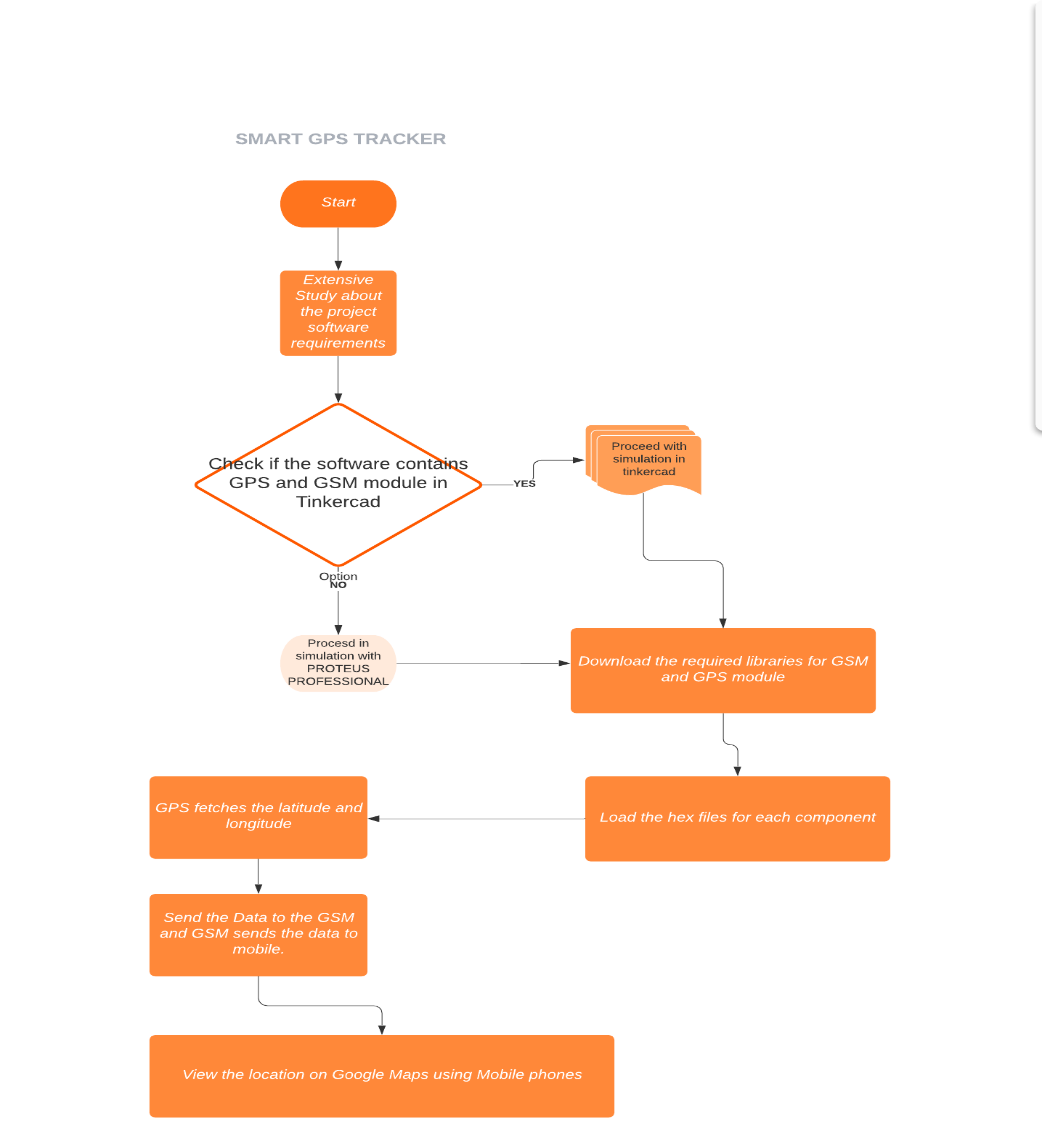
* Software Simulation:
* Proteus Professional
* Arduino Software
* Hardware Project Components Required:
* GPS Module
* GSM Module
* Arduino UNO
* Jumper Wires
* SIM card.
* Thingspeak: is an open-source Internet of Things application and API to store and retrieve data from things using the HTTP.

**MODULES USED:**

* Circuit Integration: The required circuit consists of two modules.
* Tracking: This module enables the user to keep track of the person/object with the tracker. The GPS module extracts the latitude and longitude of the current location.
* Display: GSM module helps in communication and enables us to receive the coordinates on the tracked location on the LCD as well as our mobile phone.
* Arduino Coding: Arduino acts as an interface between GPS and GSM. In this phase, the code for the Arduino is developed.
* Simulation: All the required components are connected. The libraries necessary for simulation of GPS and Arduino are downloaded. The hex files are loaded and coordinates are extracted from the virtual terminal.
* Hardware implementation: The code is loaded into the Arduino. And the GPS fetches the latitude and longitude data. Connecting to the cloud i.e. Thingspeak using cellular data of GSM.
* Execution and Testing: The integrated circuit is executed and tested for accurate output.

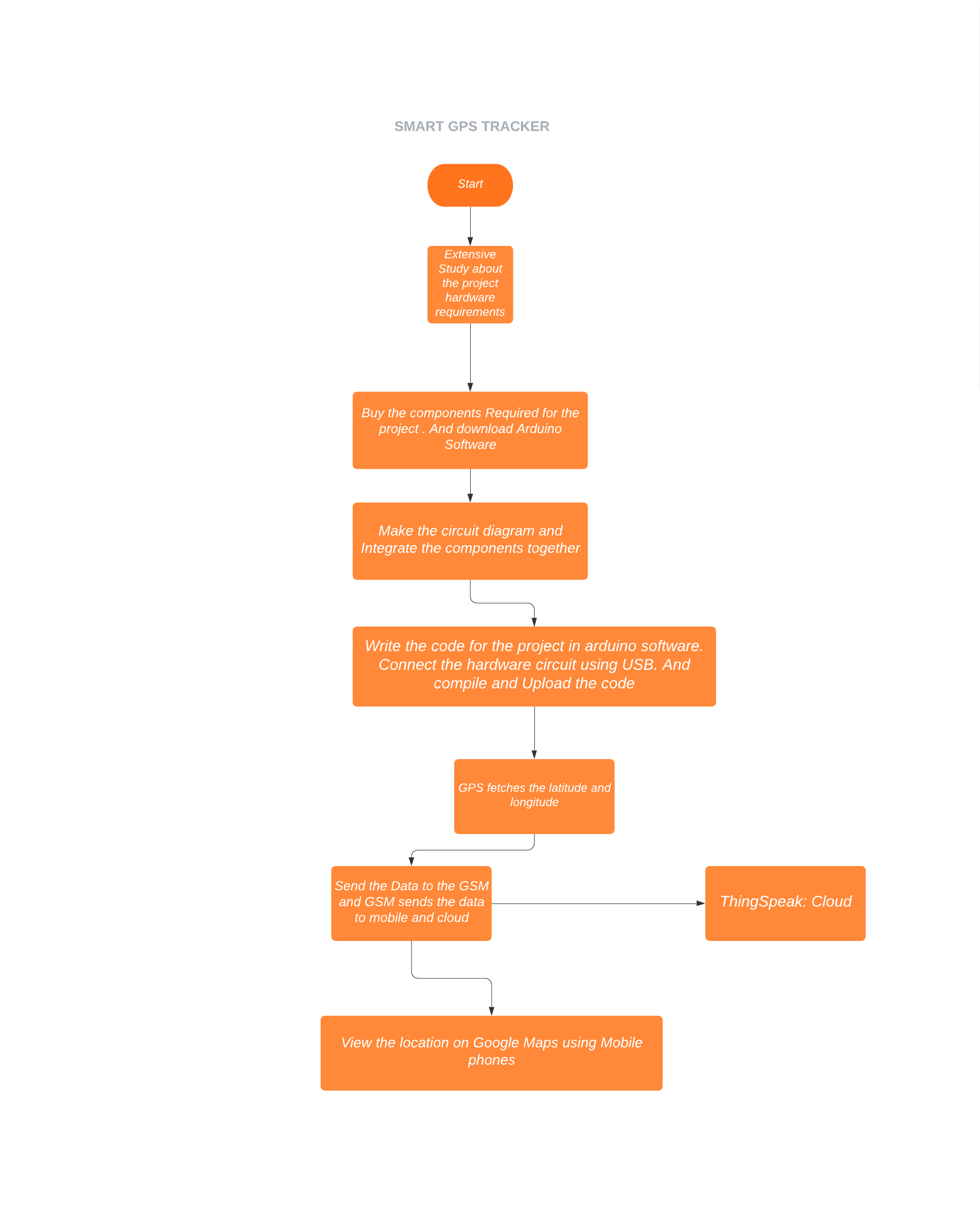
**ARCHITECTURE:**

* Software simulation:



First for this software simulation chose to work with Tinkercad online simulator but later on some of the components required for my project was not available in Tinkercad. So, decided to work with Proteus Professional 8. First for working in proteus we have to download the required library files for the components. So, the GPS and GSM library files were downloaded. Next, making the circuit with all the components and integrating them together. The important step of the simulation is writing the code in Arduino software and compiling it. Then including the hex file of the compiled Arduino code to the Arduino in Proteus. Next step, we can run the simulation and the outputs will be displayed in the virtual terminal. The output printed contains the standard latitude and longitude value. As this is a software simulation the SMS can’t be send to the mobile phones.

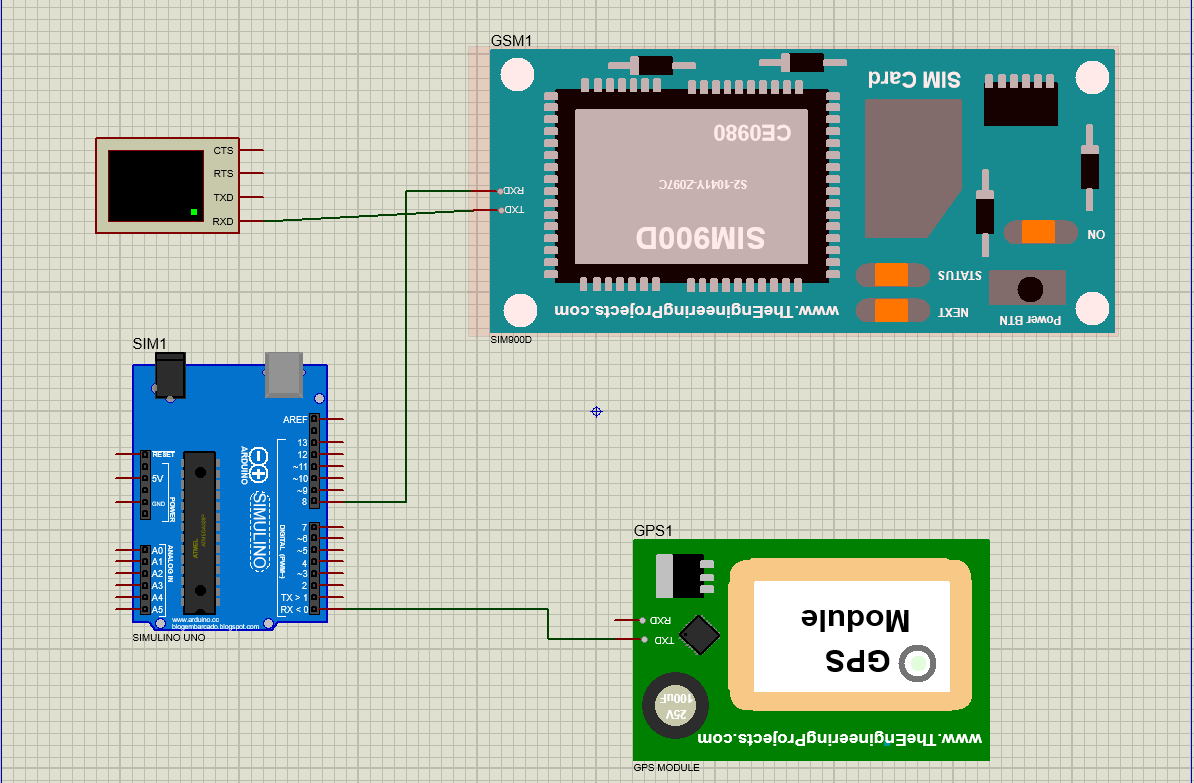
* Hardware flowchart:



For the hardware simulation all the hardware components were collected. Next the hardware circuit was built and all the components were integrated together. Next the coding was done for the built hardware circuit. Next the code is compiled and made error free. This code is uploaded into the Arduino hardware. After uploading the GPS takes few minutes to fix on the satellite and fetch the latitude and longitude data. Next the Arduino sends the coordinates to GSM and the GSM sends the data to mobile phones in the form of text and also the GSM sends the data to ThingSpeak platform using cellular data. Using the mobile phones, the location of the object can be viewed in google maps. And in the ThingSpeak platform the information about the latitude and longitude will be provided in graphical form and the location can also be viewed in maps.

**SOFTWARE SIMULATION:**

The software used for this project is Proteus Professional. Proteus Professional is used to simulate, design and drawing of electronic circuits. The software simulation of project is done using the Proteus software. The required library files of the components are downloaded and loaded into the proteus library files. The components used for building the circuit are Arduino Uno, GPS module, GSM module and virtual terminal. The figure 2.1 shows the circuit diagram for the software simulation. The coding is done using the Arduino software. The code is compiled and the hex file generated by compiling a code is loaded into the Arduino. And the hex files for the GSM and GPS modules are also uploaded. While running the simulation the output is displayed in the virtual terminal. The virtual terminal displays the latitude and longitude information of the exact location. And using the GSM module the location information is sent to the specified mobile number as SMS.



**Figure 2.1- Circuit diagram for software simulation.**

**HARDWARE IMPLEMENTATION:**

The hardware components required for project are:

* Arduino Uno
* GPS module
* GSM module
* Jumper wires
* SIM card.

The Arduino Uno technical specification is given in figure 2.2. The GPS NEO 6M technical specification is given in figure 2.3.

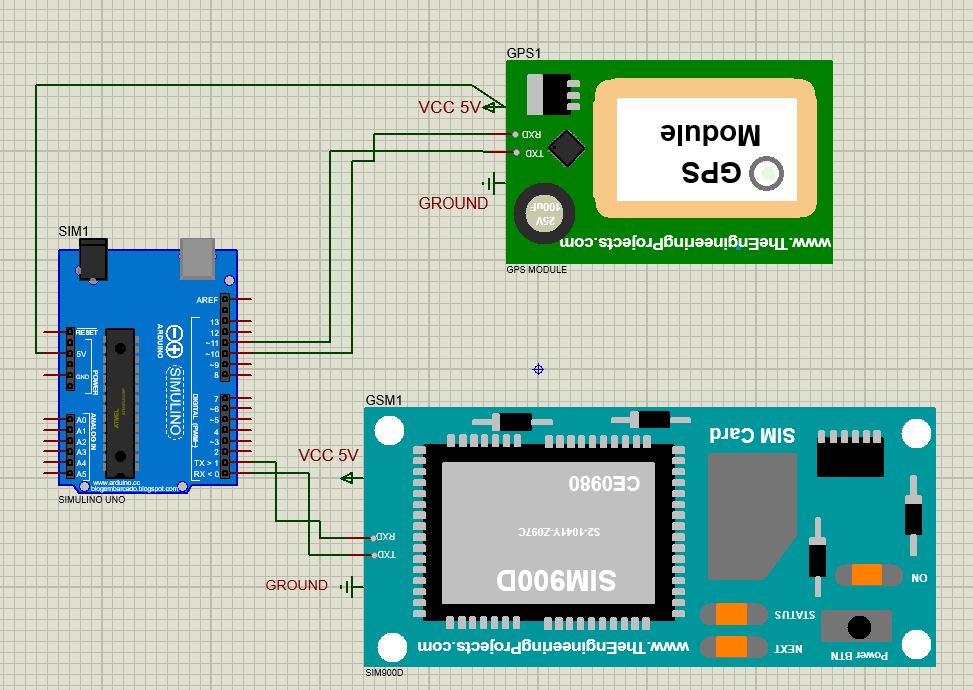


**Figure 2.2-** **Arduino Uno technical specification.**

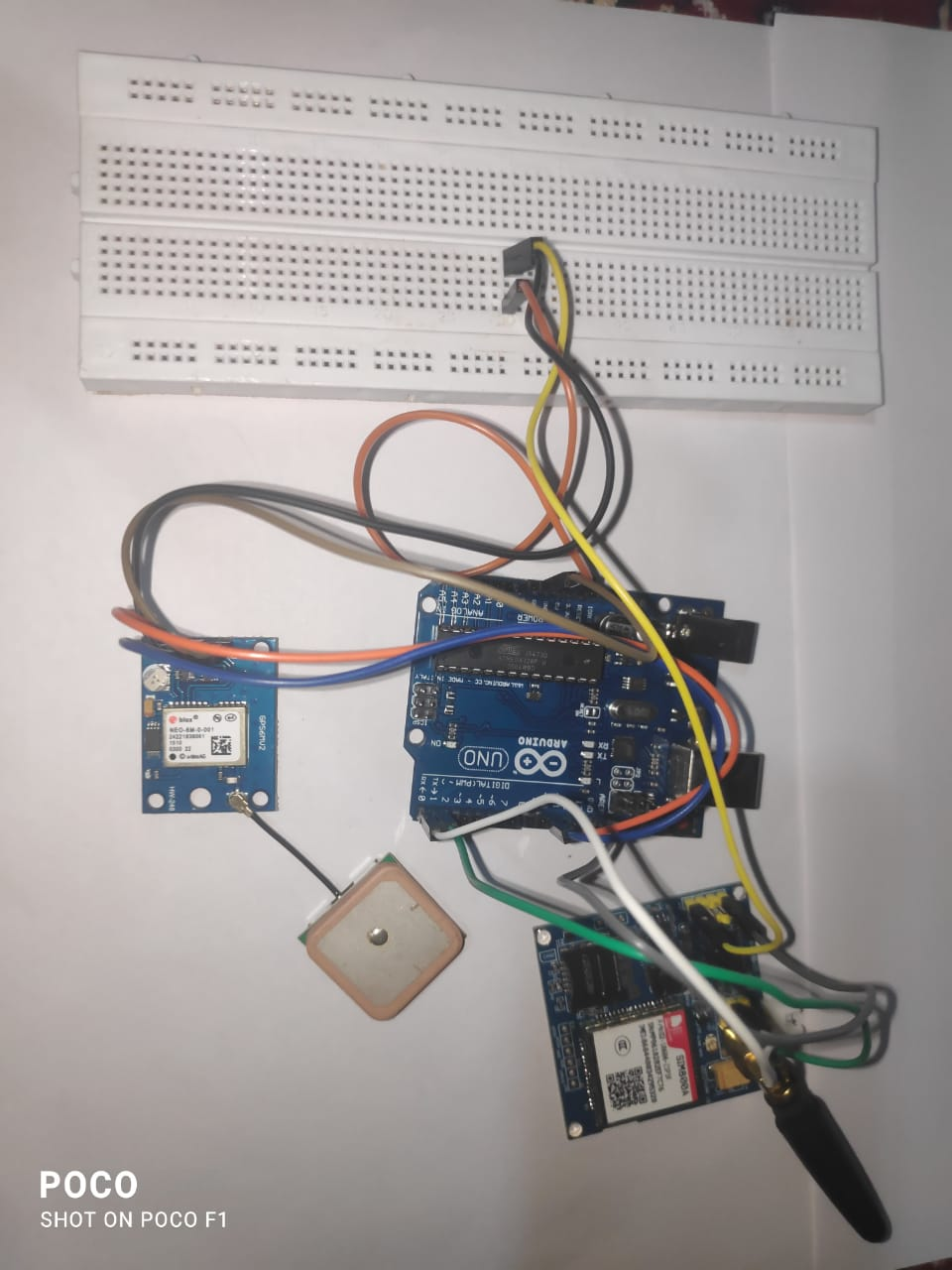


**Figure 2.3-** **GPS NEO 6M technical specification.**

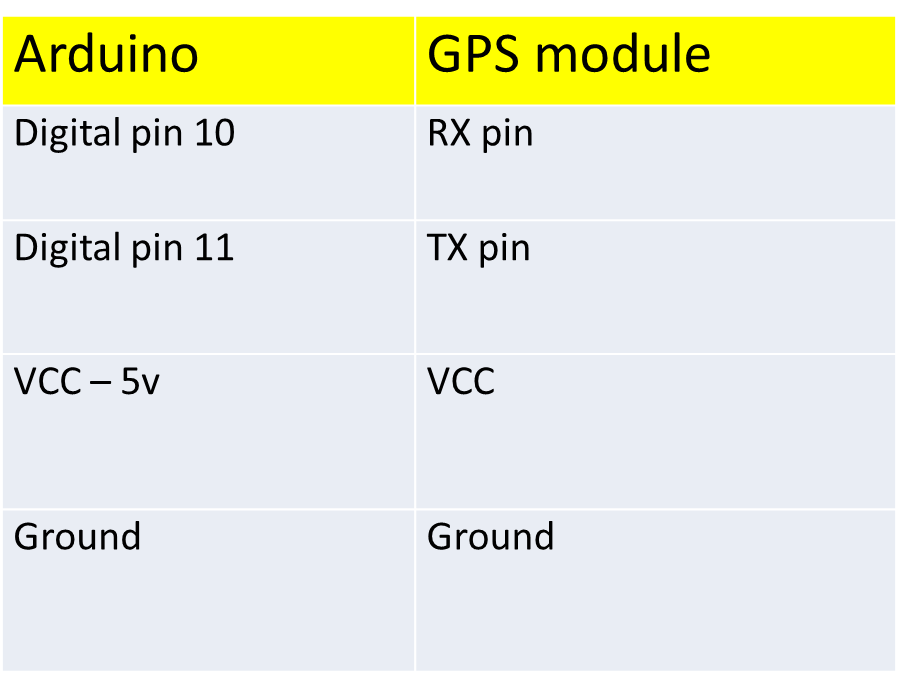
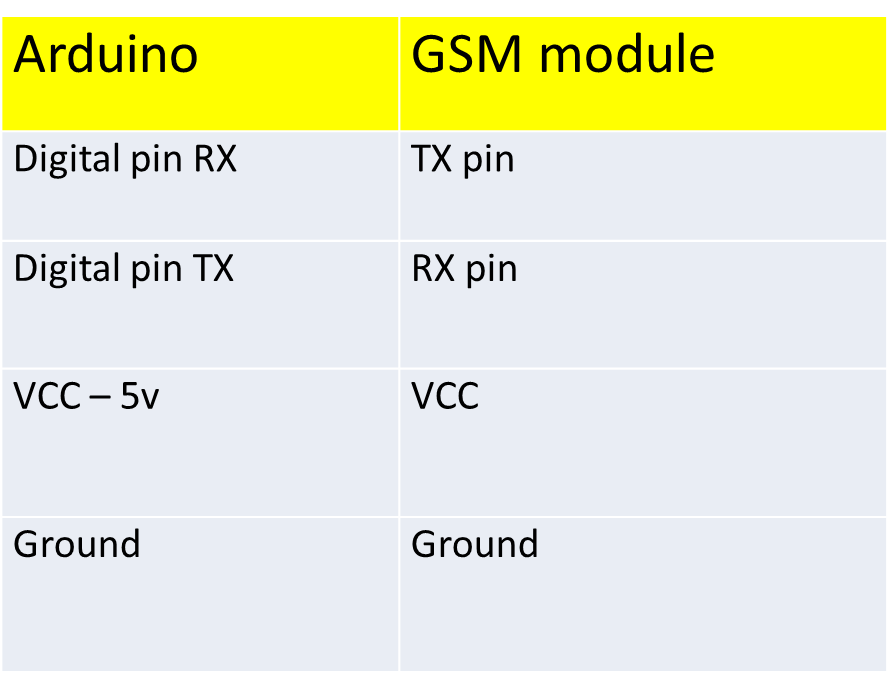
GPS is a global positioning system which is used to get the location of particular object in latitude and longitude. The Global Positioning System (GPS) is a satellite-based navigation system consists of a network of 24 satellites located into orbit. GPS works in any weather circumstances at anywhere in the world. Normally no subscription fees or system charges to use GPS. A GPS receiver must be locked on to the signal of at least three satellites to estimate2D position (latitude and longitude) and track movement. A GSM module is a basically a GSM Modem (SIM900). We are using SIM800A and then interface it with Arduino simply. SIM800A GSM means the module support communication in 800MHz bands. We are from India and most of the mobile network providers in this country operate in the 900 MHZ. The figure 2.4 shows the schematic diagram for the project. The figure 2.5 shows the wiring of the software. And the details of connection of the components are shown in the figure 2.6.



**Figure 2.4- Schematic diagram hardware project.**



**Figure 2.5- Wiring of hardware.**

****

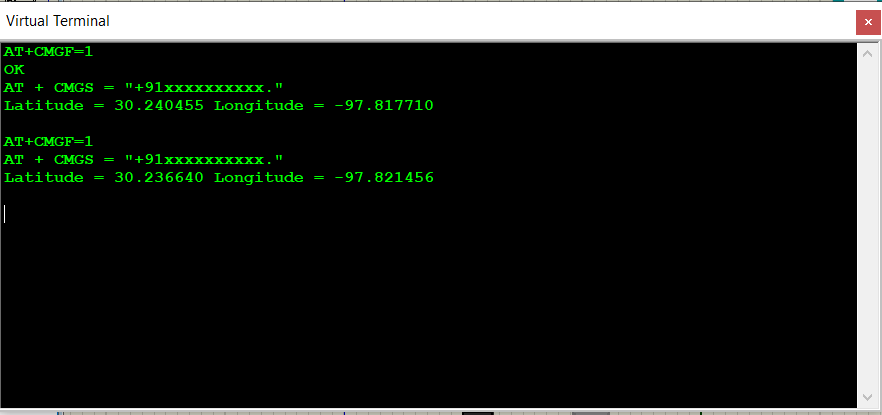
**Figure 2.6- GPS and GSM connection with Arduino.**

**RESULT AND ANALYSIS/TESTING**

The project is done in two methods, software simulation and hardware implementation.

**SOFTWARE SIMULATION:**

For the simulation Proteus Professional software was used. And the components required for software simulation the project are Arduino Uno, GPS module, GSM module and virtual terminal. Using all the components the circuit diagram is made and all the necessary libraries are included. The Arduino code is loaded into the Arduino component in form of hex file. The output will be displayed in the virtual terminal. The result obtained from the simulation is shown below.



**Figure 3.1- Output from virtual terminal.**

From the output we can infer that the latitude and longitude information is displayed. And we can tell that the GSM uses AT+CMGF command to set the GSM in SMS text mode. And the command AT+CMGS sends the latitude and longitude data to the specified mobile number as SMS.

**HARDWARE IMPLEMENTATION:**

Next the hardware implementation for the project requires an open-source Arduino software. And component required are Arduino Uno, GPS module, GSM module and jumper wires. The circuit is connected as shown in the figure 2.4. The Arduino code is compiled successfully and loaded into the Arduino Uno. The results obtained from the serial monitor are shown below.

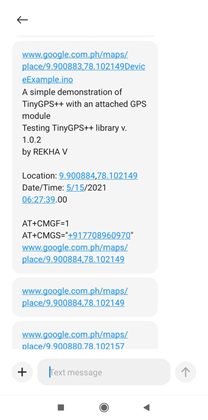
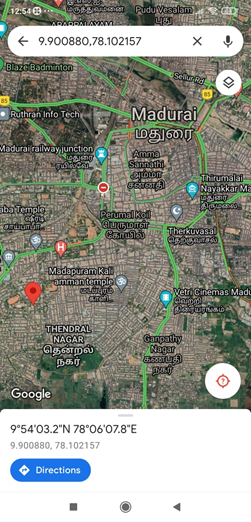


**Figure 3.2- Output from the serial monitor.**

From the output we can see that the latitude is 9.900790 and longitude is 78.102203 is printed and also the information is sent to the mobile number mentioned. The user can view the SMS as google map link. When the click the link, the exact location will be shown in google maps. The figure 3.3 and figure 3.4 shows output from the user’s mobile application.

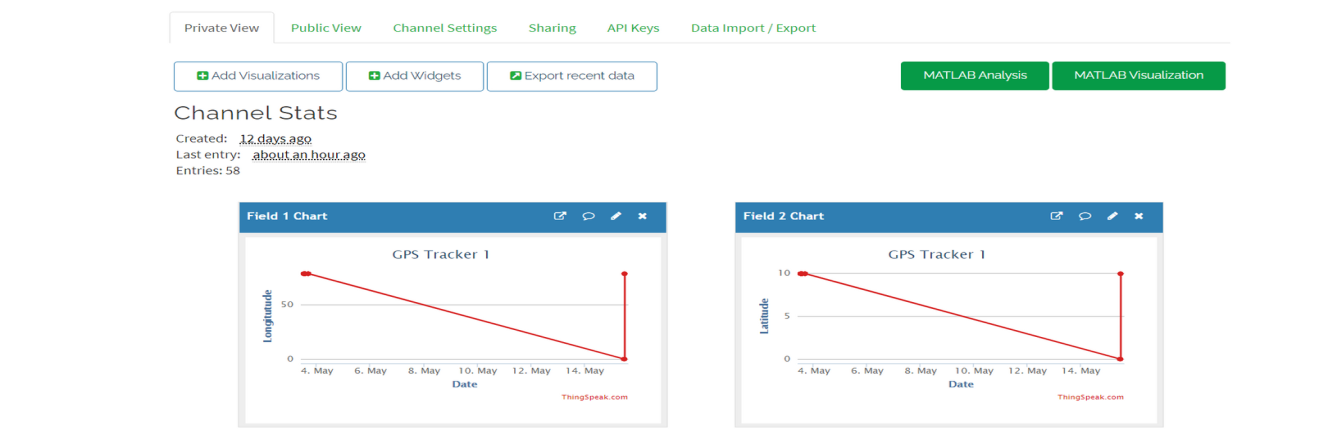
**MOBILE APPLICATION:**

From the serial monitor output we can infer that the data is also sent to ThingSpeak cloud service. ThingSpeak is an IoT analytics platform service that allows you to aggregate, visualize, and analyze live data streams in the cloud. This done by using command AT+CIPSTART which establishes a TCP connection. Then using AT+CISEND to send data to the ThingSpeak. In the ThingSpeak the latitude and longitude are displayed in chart form and also in the channel maps. The figure 3.5 shows the dashboard of the channel. And the figure 3.7 shows the latitude and longitude in maps.

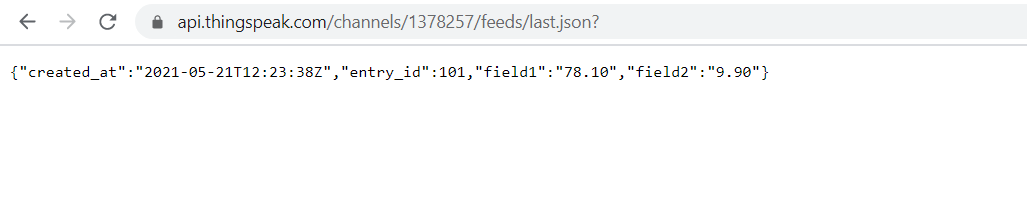
**Figure 3.3- Output from user Figure 3.4- Output from the users**

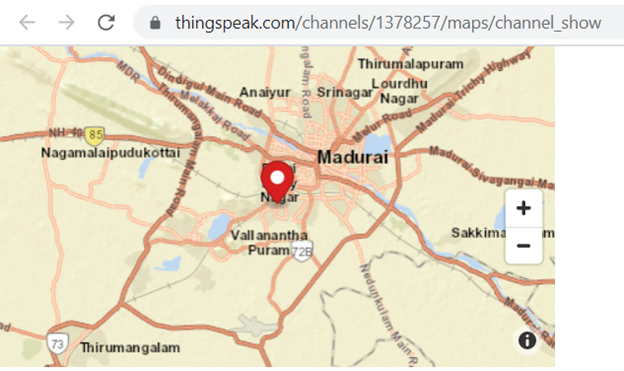
**mobile as SMS. google maps app.**



**Figure 3.5- ThingSpeak dashboard.**

The data received by the ThingSpeak platform is in the. json form. The figure below the last feed information. Here the field1 corresponds to the longitude and field2 corresponds to the latitude.

**Figure 3.6- ThingSpeak channel’s last.json**



**Figure 3.7- ThingSpeak channel maps display the location using the latitude and longitude.**

**CODING:**

**The Arduino programming code for the software simulation of the project is given below.**

#include <TinyGPS.h>

#include <SoftwareSerial.h>

SoftwareSerial SIM900(7, 8);

TinyGPS gps; //Creates a new instance of the TinyGPS object

void setup()

{

Serial.begin(9600);

SIM900.begin(9600);

}

void loop()

{

bool newData = false;

unsigned long chars;

unsigned short sentences, failed;

// For one second we parse GPS data and report some key values

for (unsigned long start = millis(); millis() - start < 1000;)

{

while (Serial.available())

{

char c = Serial.read();

//Serial.print(c);

if (gps.encode(c))

newData = true;

}

}

if (newData) //If newData is true

{

float flat, flon;

unsigned long age;

gps.f\_get\_position(&flat, &flon, &age);

SIM900.print("AT+CMGF=1\r");

delay(400);

SIM900.println("AT + CMGS = \"+91xxxxxxxxxx.\"");// recipient's mobile number with country code

delay(300);

SIM900.print("Latitude = ");

SIM900.print(flat == TinyGPS::GPS\_INVALID\_F\_ANGLE ? 0.0 : flat, 6);

SIM900.print(" Longitude = ");

SIM900.print(flon == TinyGPS::GPS\_INVALID\_F\_ANGLE ? 0.0 : flon, 6);

delay(200);

SIM900.println((char)26); // End AT command with a ^Z, ASCII code 26

delay(200);

SIM900.println();

}

Serial.println(failed);

// if (chars == 0)

// Serial.println("\*\* No characters received from GPS: check wiring \*\*");

}

**The Arduino programming code for hardware implementation of the project is given below.**

#include <TinyGPS++.h>

#include <SoftwareSerial.h>

int state = 0;

static const int RXPin = 11, TXPin = 10;

static const uint32\_t GPSBaud = 9600;

// The TinyGPS++ object

TinyGPSPlus gps;

// The serial connection to the GPS device

SoftwareSerial ss(RXPin, TXPin);

void setup()

{

Serial.begin(9600);

ss.begin(GPSBaud);

Serial.println(F("DeviceExample.ino"));

Serial.println(F("A simple demonstration of TinyGPS++ with an attached GPS module"));

Serial.print(F("Testing TinyGPS++ library v. ")); Serial.println(TinyGPSPlus::libraryVersion());

Serial.println(F("by REKHA V"));

Serial.println();

}

void loop()

{

// This sketch displays information every time a new sentence is correctly encoded.

while (ss.available() > 0)

{

if (gps.encode(ss.read()))

{ displayInfo();

Serial.print("\r");

delay(1000);

Serial.print("AT+CMGF=1\r");

delay(1000);

/\*Replace XXXXXXXXXX to 10 digit mobile number & ZZ to 2 digit country code\*/

Serial.print("AT+CMGS=\"+917708960970\"\r");

delay(1000);

//The text of the message to be sent.

Serial.print("www.google.com.ph/maps/place/");

Serial.print(gps.location.lat(), 6);

Serial.print(F(","));

Serial.print(gps.location.lng(), 6);

delay(1000);

Serial.write(0x1A);

delay(1000);

float h = gps.location.lat();

float t = gps.location.lng();

Serial.println("AT");

delay(1000);

Serial.println("AT+CPIN?");

delay(1000);

Serial.println("AT+CREG?");

delay(1000);

Serial.println("AT+CGATT?");

delay(1000);

Serial.println("AT+CIPSHUT");

delay(1000);

Serial.println("AT+CIPSTATUS");

delay(2000);

Serial.println("AT+CIPMUX=0");

delay(2000);

ShowSerialData();

Serial.println("AT+CSTT=\"airtelgprs.com\"");//start task and setting the APN,

delay(1000);

ShowSerialData();

Serial.println("AT+CIICR");//bring up wireless connection

delay(3000);

ShowSerialData();

Serial.println("AT+CIFSR");//get local IP adress

delay(2000);

ShowSerialData();

Serial.println("AT+CIPSPRT=0");

delay(3000);

ShowSerialData();

Serial.println("AT+CIPSTART=\"TCP\",\"api.thingspeak.com\",\"80\"");

//start up the connection

delay(6000);

ShowSerialData();

Serial.println("AT+CIPSEND");

//begin send data to remote server

delay(4000);

ShowSerialData();

String str="GET https://api.thingspeak.com/update?api\_key=2JF4USELJIM154YM&field1=" + String(t) +"&field2="+String(h);

Serial.println(str);

Serial.println(str);//begin send data to remote server

delay(4000);

ShowSerialData();

Serial.println((char)26);//sending

delay(5000);//waitting for reply, important! the time is base on the condition of internet

Serial.println();

ShowSerialData();

Serial.println("AT+CIPSHUT");//close the connection

delay(100);

ShowSerialData();

state = 1;

}

}

if (millis() > 5000 && gps.charsProcessed() < 10)

{

Serial.println(F("No GPS detected: check wiring."));

while(true);

}

}

void ShowSerialData()

{

while(Serial.available()!=0)

Serial.write(Serial.read());

delay(5000);

}

void displayInfo()

{

Serial.print(F("Location: "));

if (gps.location.isValid())

{

Serial.print(gps.location.lat(), 6);

Serial.print(F(","));

Serial.print(gps.location.lng(), 6);

}

else

{

Serial.print(F("INVALID"));

}

Serial.print(F(" Date/Time: "));

if (gps.date.isValid())

{

Serial.print(gps.date.month());

Serial.print(F("/"));

Serial.print(gps.date.day());

Serial.print(F("/"));

Serial.print(gps.date.year());

}

else

{

Serial.print(F("INVALID"));

}

Serial.print(F(" "));

if (gps.time.isValid())

{

if (gps.time.hour() < 10) Serial.print(F("0"));

Serial.print(gps.time.hour());

Serial.print(F(":"));

if (gps.time.minute() < 10) Serial.print(F("0"));

Serial.print(gps.time.minute());

Serial.print(F(":"));

if (gps.time.second() < 10) Serial.print(F("0"));

Serial.print(gps.time.second());

Serial.print(F("."));

if (gps.time.centisecond() < 10) Serial.print(F("0"));

Serial.print(gps.time.centisecond());

}

else

{

Serial.print(F("INVALID"));

}

Serial.println();

}

**CONCLUSION AND FUTURE ENHANCEMENT**

This project notes the increased demand for vehicle tracking systems for tracking the theft of vehicles through GPS and GSM technology. GPS is used for describing the position of the person or vehicle or object in the form of longitude & latitude and further this longitude & latitude is received by the GSM. And GSM send this information to the trackers mobile number via message. This system can be used for both personal and business purposes to improve safety and security, communication, and performance monitoring. Vehicle tracking or any object tracking systems have become increasingly important in large cities and are more secured than many other systems. Nowadays, vehicle theft is rapidly increasing. With this technology however, vehicle theft can be better controlled. This technology can also help to advance transportation systems, and can be used in many organizations for security and tracking purposes. The Google Map proves to provide main enhancement firstly, it improves the accuracy of the tracking location of vehicle or any person So it offers the additional functionality which requires high accuracy i.e., extract location of the children or person such functionality requires high, accuracy with GPS module coordinates, validity of the system 72.2% and with Google Map validity of the system 95%. For further work a high scale deployment can accomplished. In the future, this vehicle tracking and accident alert feature will play an important role in day-to-day life.

**CONTRIBUTION:**

REKHA V: Being the only member in the group I took care of everything, starting from the simulation, integration of the hardware components to taking data to ThingSpeak and performing analysis.

**REFERENCES**

* https://www.hackster.io/TechnicalEngineer/gps-and-gsm-based-vehicle-tracking-system-using-arduino-efbae1
* https://how2electronics.com/gps-gsm-based-vehicle-tracking-system-using-arduino/
* https://circuitdigest.com/microcontroller-projects/vehicle-tracking-system-using-arduino-gps-and-gsm
* https://www.electronicsforu.com/electronics-projects/smart-gps-tracker-using-arduino