TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION:-

Vehicle tracking system main aim is to give security to all Vehicles. This is improved security system for vehicles. The latest like GPS are highly useful now-a-days, this system enables the owner to observe and track his vehicle movement and its past activities of vehicle.

This new technology, popularly called vehicle Tracking Systems which created many wonders in the security of the vehicle. This hardware is fitted on to the vehicle in such a manner that is not visible to anyone who is inside or outside of the vehicle. Thus it is used as a covert unit which continuously or by any interrupt to the system, sends the location data to the monitoring unit.

When the vehicle is stolen, the location data from tracking system can be used to find the location and can be informed to police for further action. Some Vehicle tracking System can even detect unauthorized movements of the vehicle and then alert the owner. This gives an edge over other pieces of technology for the same purpose.

The system automatically sends the position of the vehicle in terms of latitude and longitude. A program has been developed which is used to locate the exact position of the vehicle and also to navigate track f the moving vehicle on Map.

1.2 Proposed System:-

The proposed system is used for positioning and navigating the vehicle with an accuracy of 10m. The exact location is indicated in the form of latitude and longitude along with the exact Navigated track on map.
The system tracks the location of particular vehicle and updates the data on the thingspeak server. The arrived data, in the form of latitude and longitude is used to locate the vehicle on the map.

1.3 VEHICLE TRACKING FEATURES:-

It is mainly benefit for the companies which are based on transport system. Since it can show the position of all vehicles in real time, so that they can create expected data accordingly.
These tracking system can store the whole data where the vehicle had gone, where did it stop, how much time it take at every stop and can create whole data analysis. It is also used in buses and trains, to estimate how far are they, how much time it takes for them to come to a particular stop. These systems are used to data capture, data storage, data analysis and finally data transfer

1.4 USAGE OF TRACKING IN INDIA:-

Tracking in India is mainly used by transport systems, taxi companies, traffic operators, Taxi operators use this to estimate how far the vehicle is from a particular are and send this information to call centers and they can inform general public about the distance of the taxi location and time it takes to come to them. Another use is for traffic police if this system is located in every vehicle they can estimate the traffic by looking on the map and if any accident is detected then they can route the traffic in to another way. This is how tracking is useful because India is one of the busy traffic countries and this system can control many of the traffic problems.

1.5 APPLICATION:-

The project that has been introduced here can be used for variety of applications –

1. Car navigation

2. Fleet management/tracking

3. Palmtop, Laptop, PDA, and Handheld

4. Location based Services enabled Devices

CHAPTER 2

METHODOLOGY

2.1 WORKING:-

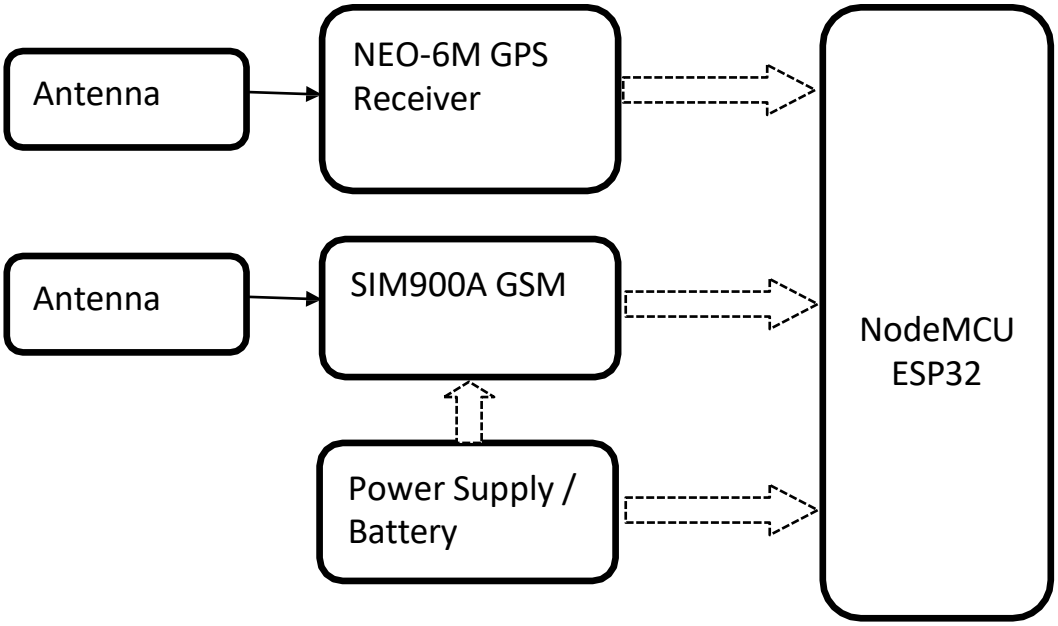
The project consists of GPS receiver and GSM modem with a micro controller. The whole system is attached to the vehicle. So the GPS will send the longitudinal and altitude values corresponding to the position of the vehicle to the ThingSpeak server with GPRS Connection.

Imagine the bus has left the Bangalore at 6 o clock in the morning. If the officer in charge for that vehicle wants to know where the vehicle is, he will come to the computer and login on the application. The application will show the real-time location on Mapbox maps.

The location coordinates come through GSM service provider and then reach the ThingSpeak server, because the vehicle has a GSM device with SIM card. The device will read the current location of the vehicle in real-time and will update the coordinates on the ThingSpeak server with GSM modem.

2.2 BLOCK DIAGRAM:-

The block diagram of the Vehicle tracking system is shown below. The block diagram shows the overall view of the system. The block that are connected here are Microcontroller, GPS, GSM, Power Supply.



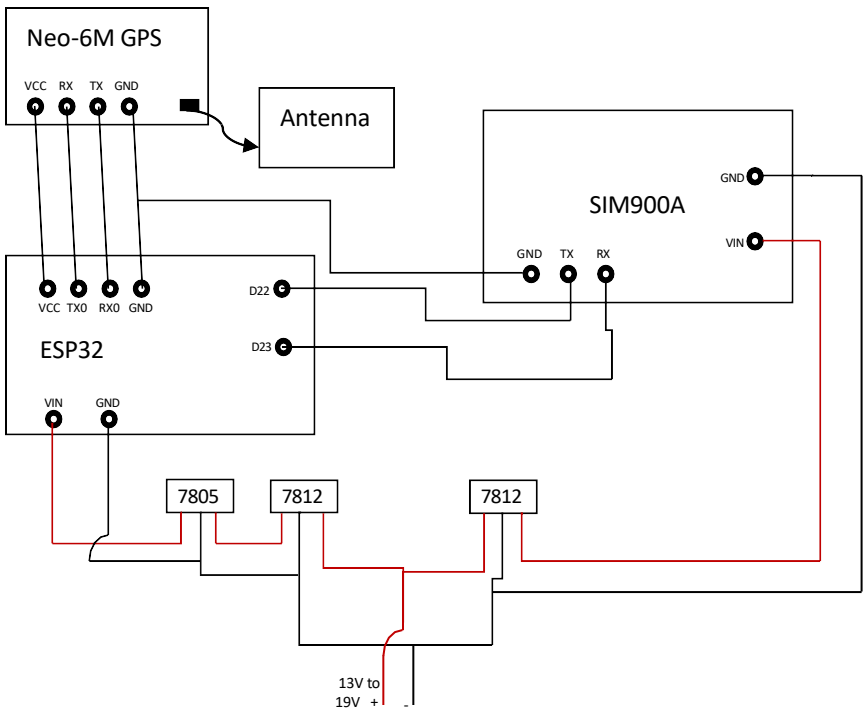
2.2.1 BLOCK DIAGRAM DISCRIPTION:-

In this Project it is proposed to design an embedded system which is used for tacking and positioning of any vehicle by using Global Positioning System (GPS) and Global System for mobile Communication (GSM).
In this project esp32 microcontroller is used for interfacing to various hardware peripherals. The current design is an embedded application, which will continuously monitor a moving vehicle and update the status of vehicle on the server on real-time basis.
For doing so an esp32 microcontroller is interfaced serially to a GSM Modem and GPS Receiver. A GSM modem is used to send the position (Latitude and Longitude) of the vehicle from a remote place. The GPS modem will continuously monitor the data i.e. latitude and longitude indicating the position of the vehicle, and will update on servers with GSM Modem.

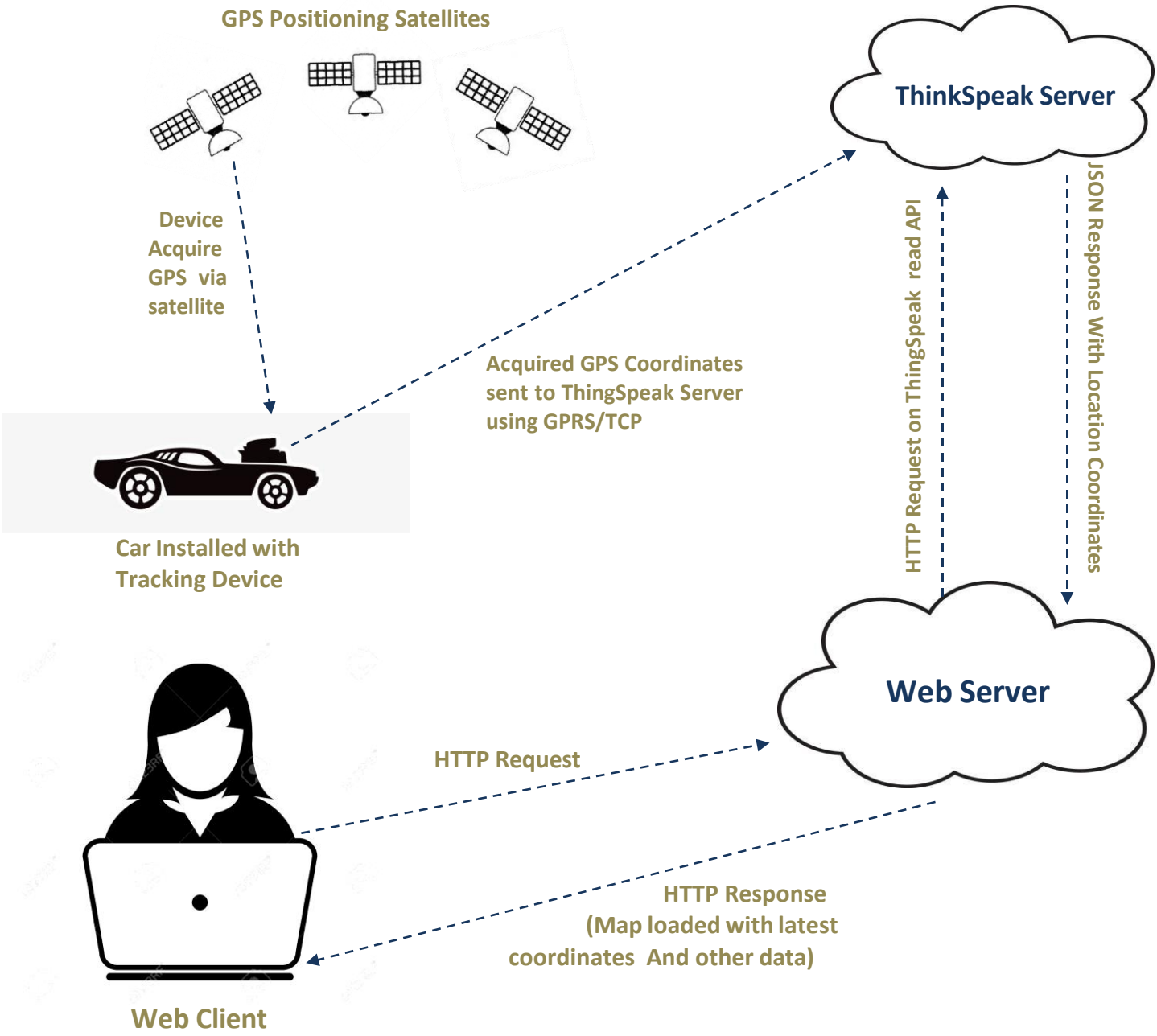
2.2.2 CONCEPT AND OVERVIEW:-

This vehicle tracking system takes input from GPS and send it through GSM module to the ThingSpeak servers. Vehicle Tracking System is one of the biggest technological advancements to track the activities of the vehicle. The security system uses Global Positioning System GPS, to find the location of the monitored or tracked vehicle and then uses satellite or radio systems to send the coordinates and the location data to the monitoring station (servers). At monitoring center various software's are used to plot the vehicle on a map, In this way the Vehicle owners are able to track their vehicle on a real-time basis. Due to real-time tracking facility, vehicle tracking systems are being increasingly popular among owners of expensive vehicles.

2.3 CIRCUIT DIAGRAM:-



2.4 WORKING DIAGRAM:-



CHAPTER 3

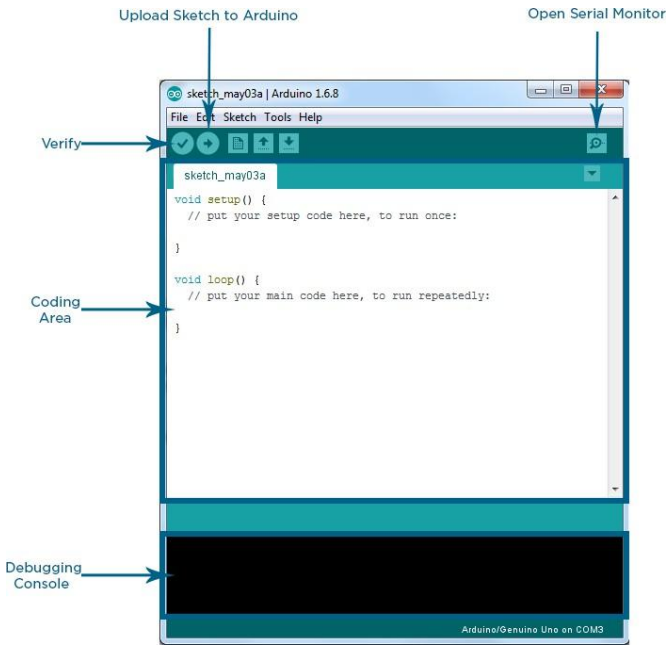
PROCESS AND REQUIREMENTS

3.1 SOFTWARE REQUIREMENTS:-

3.1.1 Arduino IDE:-

The Arduino IDE is a cross-platform application written in Java, and is derived from the IDE for the Processing programming language and the Wiring project. It is designed to introduce programming to artists and other newcomers unfamiliar with software development. It includes a code editor with features such as syntax highlighting, brace matching, and automatic indentation, and is also capable of compiling and uploading programs to the board with a single click. There is typically no need to edit make files or run programs on a command-line interface. Although building on command-line is possible if required with some third party tools.

The Arduino IDE comes with a C/C++ library called “Wiring” (from the project of the same name), which makes many common input/ouput operations much easier. Arduino programs are written in C/C++.



3.1.2 Mapbox Maps:-

Mapbox is one of the largest providers of custom designed maps for websites and mobile apps. This service is used by well-known delivery and transportation companies like DHL, DPDgroup, Grubhub, Instacart.

The majority of data Mapbox uses is openly available, and Mapbox supports a community of volunteer mappers. They often provide the freshest updates, including fast-changing location data. Mapbox data sources include OpenStreetMap (OSM), USGS, Landsat, Natural Earth, and OpenAddresses.
The platform uses OpenStreetMap as the base map and lets developers add different markers, lines, polylines, and polygons as well as layers from external sources (in GeoJSON, GPX, and other formats).
Mapbox offers many tools to help you integrate maps and other Mapbox online web mapping services – such as Directions, Geocoding, and Static Images – into a mobile app. In order to keep the Maps SDKs for iOS and Android small, Mapbox provides different mapping APIs for interfacing with Mapbox web services: the Mapbox Directions API, Geocoding API, and Static Images API.

3.1.3 000webhost server:-

000webhost.com is free of cost and an amazing quality web hosting service provider that provides hosting services to the users on the Internet.
For those who are starting their first project of website building then this is something that might help you out. 000webhost is the ultimate catch when it comes to trying out some ideas about a free website. Absolutely free of any expenses, it is a fairly good way of getting dedicated and impressive services of hosting.

3.1.4 ThingSpeak server:-

ThingSpeak is an open source “Internet of Things” application and API to store and retrieve data from things using HTTP over the Internet or via a Local Area Network. With ThingSpeak, you can create sensor logging applications, location tracking applications, and a social network of things with status updates.

In addition to storing and retrieving numeric and alphanumeric data, the ThingSpeak API allows for numeric data processing such as timescaling, averaging, median, summing, and rounding. Each ThingSpeak Channel supports data entries of up to 8 data fields, latitude, longitude, elevation, and status. The channel feeds support JSON, XML, and CSV formats for integration into applications.

3.2 COMPONENTS DESCRIPTION:-

For designing this hardware many types of devices are used to make it perfectly working. All the devices are purchased from different manufacturers. These components are soldered on a soldering board. The following list of hardware are required for this system.

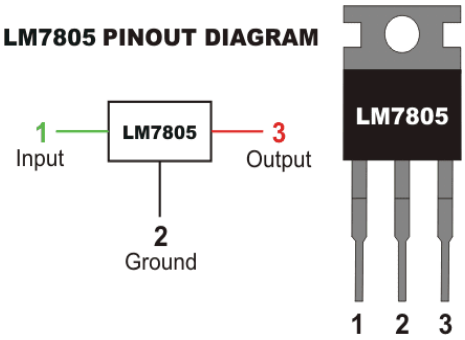
- 0.1µf Cap (x3)
- IC LM7805
- IC LM7812 (x2)
- ESP32 Microcontroller
- SIM900A GSM/GPRS Module
- Neo-6m GPS Module

3.2.1 CAPACITORS:-

A capacitor (originally known as a condenser) is a passive twoterminal electrical component used to store energy electrostaticallyin
an electric field. The forms of practical capacitors vary widely, but all contain at least two electrical conductors (plates) separated by
a dielectric (i.e. insulator). The conductors can be thin films, foils or sintered
beads of metal or conductive electrolyte, etc. The nonconducting dielectric
acts to increase the capacitor's charge capacity. A dielectric can be glass, ceramic, plastic film, air, vacuum, paper, mica, oxide layer etc. Capacitors are widely used as parts of electrical circuits in many common electrical devices. Unlike aresistor, an ideal capacitor does not dissipate energy. Instead, a capacitor stores energy in the form of an electrostatic
field between its plates

3.2.2 IC LM7805:-

7805 is a voltage regulator integrated circuit. It is a member of 78xx series of fixed linear voltage regulator ICs. The voltage source in a circuit may have fluctuations and would not give the fixed voltage output. The voltage regulator IC maintains the output voltage at a constant value. The xx in 78xx indicates the fixed output voltage it is designed to provide. 7805 provides +5V regulated power supply. Capacitors of suitable values can be connected at input and output pins depending upon the respective voltage levels.

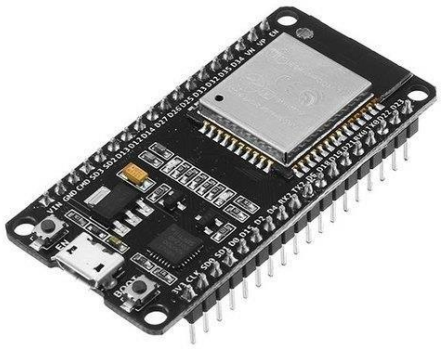


3.2.2 IC LM7812:-

7812 is also a voltage regulator integrated circuit just like 7805. It is a member of 78xx series of fixed linear voltage regulator ICs. 7812 provides +12V regulated power supply. Capacitors of suitable values can be connected at input and output pins depending upon the respective voltage levels.

3.2.3 ESP32 DevKit – The ESP32 Development Board:-

ESP32 is a low-cost System on Chip (SoC) Microcontroller from Espressif Systems, the developers of the famous ESP8266 SoC. It is a successor to ESP8266 SoC and comes in both single-core and dual-core variations of the Tensilica's 32-bit Xtensa LX6 Microprocessor with integrated Wi-Fi and Bluetooth.



The good thing about ESP32, like ESP8266 is its integrated RF components like Power Amplifier, Low-Noise Receive Amplifier, Antenna Switch, Filters and RF Balun. This makes designing hardware around ESP32 very easy as you require very few external components.

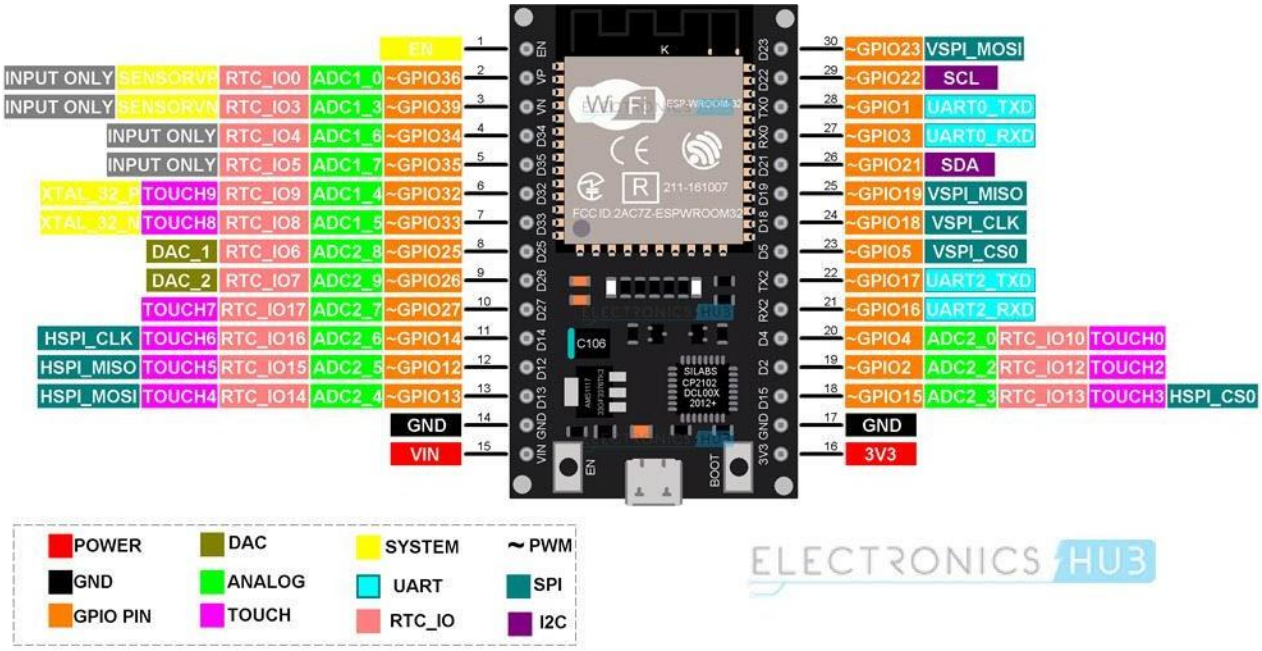
Another important thing to know about ESP32 is that it is manufactured using TSMC's ultra-low-power 40 nm technology. So, designing battery operated applications like wearables, audio equipment, baby monitors, smart watches, etc., using ESP32 should be very easy.

Specifications of ESP32

ESP32 has a lot more features than ESP8266 and it is difficult to include all the specifications in this Getting Started with ESP32 guide. So, I made a list of some of the important specifications of ESP32 here. But for complete set of specifications, I strongly suggest you to refer to the Datasheet.

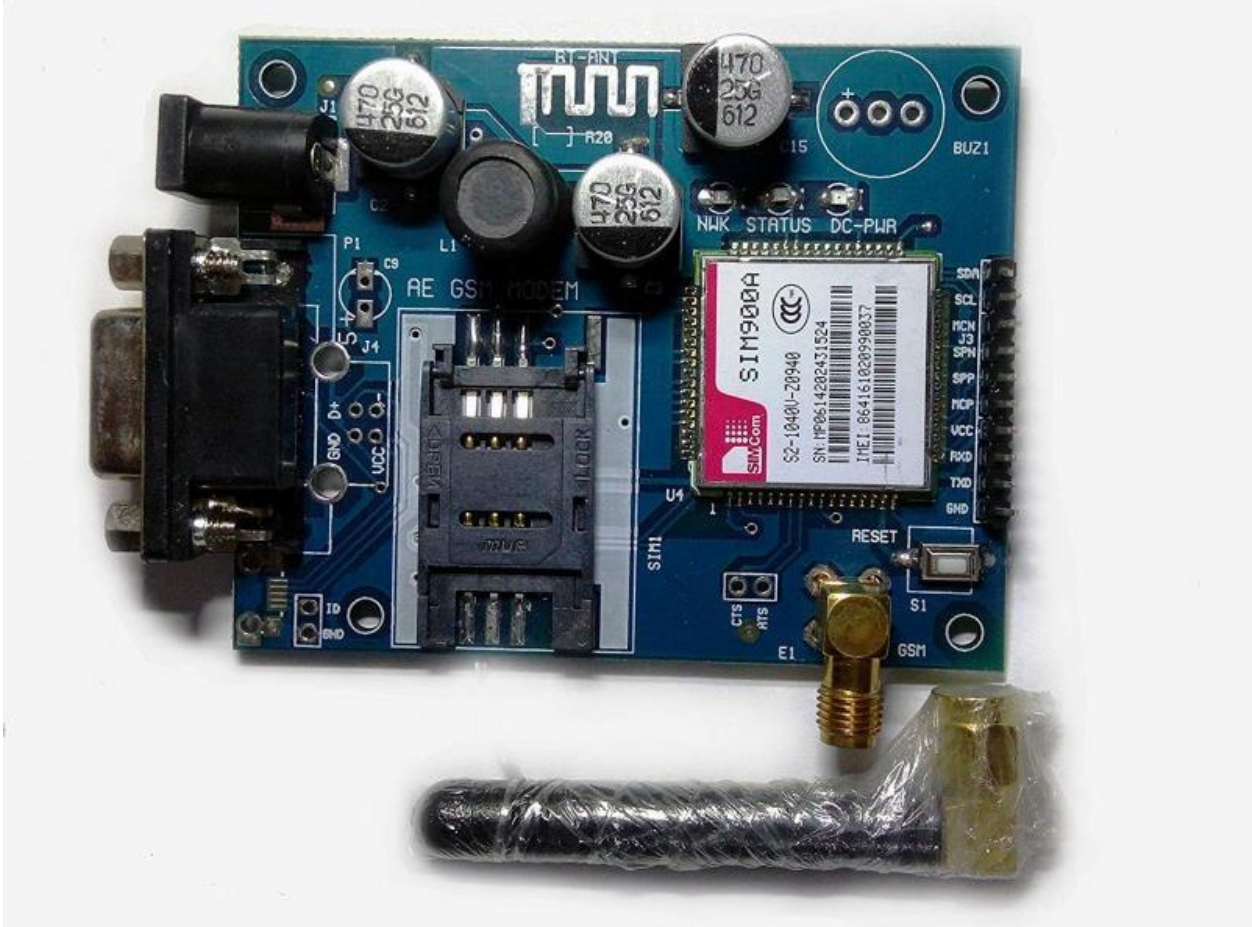
- Single or Dual-Core 32-bit LX6 Microprocessor with clock frequency up to 240 MHz.
- 520 KB of SRAM, 448 KB of ROM and 16 KB of RTC SRAM.
- Supports 802.11 b/g/n Wi-Fi connectivity with speeds up to 150 Mbps.
- Support for both Classic Bluetooth v4.2 and BLE specifications.
- 34 Programmable GPIOs.
- Up to 18 channels of 12-bit SAR ADC and 2 channels of 8-bit DAC
- Serial Connectivity include 4 x SPI, 2 x I2C, 2 x I2S, 3 x UART.
- Ethernet MAC for physical LAN Communication (requires external PHY).
- 1 Host controller for SD/SDIO/MMC and 1 Slave controller for SDIO/SPI.
- Motor PWM and up to 16-channels of LED PWM.
- Secure Boot and Flash Encryption.
- Cryptographic Hardware Acceleration for AES, Hash (SHA-2), RSA, ECC and RNG.

Pinout of ESP32 Board



This pinout is for the 30 - pin version of the ESP Board.

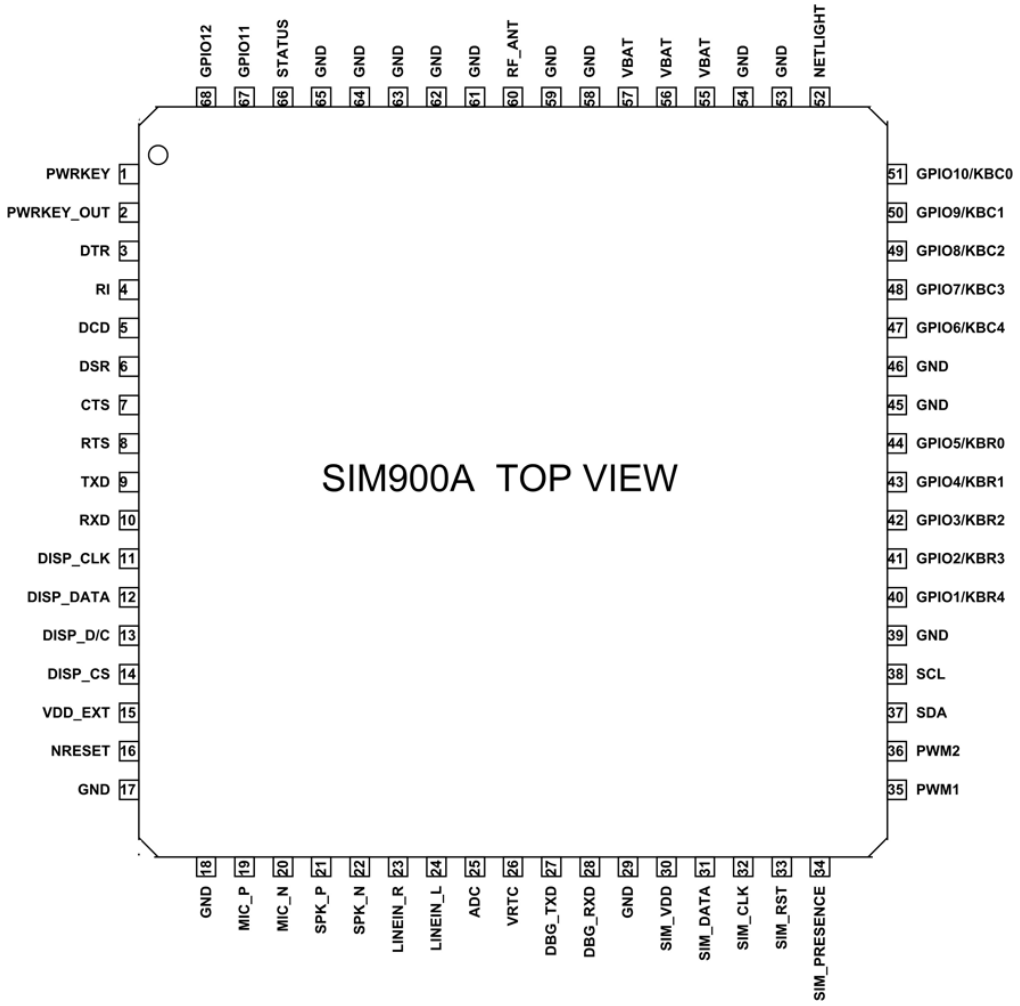
3.2.4 SIM900A GSM Module:-



SIM900A GSM Module is the smallest and cheapest module for GPRS/GSM communication. It is common with Arduino and microcontroller in most of embedded application. The module offers GPRS/GSM technology for communication with the uses of a mobile sim. It uses a 900 and 1800MHz frequency band and allows users to receive/send mobile calls and SMS. The keypad and display interface allows the developers to make the customize application with it. Furthermore, it also has modes, command mode and data mode. In every country the GPRS/GSM and different protocols/frequencies to operate. Command mode helps the developers to change the default setting according to their requirements.

3.2.4.1 SIM900A Pin Configuration:-

The Module SIM900A looks like a single chip but it has a bunch of features that can help to build almost many commercial applications. Although, there are a total of 68 pins on SIM900A and using these pins helps to build the applications. But we will need few pins if you we use a module for interfacing with Arduino. We lists details of pinout diagram in next section.



3.2.4.2 SIM900A Pin Configuration Description:-

GPIO Pins

The GPIO pins help to perform the simple and advance I/O function. All pins give the maximum output equal to the power supply which is useable to control most of the devices like sensors and other modules. All GPIO pins in SIM900A are:

- GPIO1 – Pin40
- GPIO2 – Pin41
- GPIO3 – Pin42
- GPIO4 – Pin43
- GPIO5 – Pin44
- GPIO6 – Pin47
- GPIO7 – Pin48
- GPIO8 – Pin49
- GPIO9 – Pin50
- GPIO10 – Pin51
- GPIO11 – Pin67
- GPIO12 – Pin68

Status Pins

The module has two status pins which help to indicate two different kinds of status. The first one is the working status of the module and the second for communication status. Net status means either the module is connecting to the network or other network functions, etc. Both these pins can't operate LED directly. They always act with a combination of a transistor.

- STATUS – Pin52
- NIGHTLIGHT – Pin66

SIM900A Display Interface Pins

The device offers a 4 pin display interface with itself. The display isn't necessary, it is only in case of requirement. The use of interface helps to get the visualization with the module and make it an application. All display pins are:

- DISP_DATA – Pin12 – For Display Data
- DISP_CLK – Pin11 – For Clock Input
- DISP_CS – Pin14 – To enable the display
- DISP_D/C – Pin13 – To select between data and command

I2C Pins

SIM900A has multiple kinds of communication and I2C is one of them due to is popularity. The module has a single I2C protocol pin, which helps to build the application with any module with that communication.

- SCL – Pin38
- SDA – Pin37

SDA for data and SCL for clock pulse.

SIM900A GSM Module Keypad interface Pins

The two-pin keypad is interfaceable with the module. The module will take the keypad data as a 2D matrix value from the KCB pins for each value. The keypad interface pins in the module are:

- KBR0~KBR4 (ROWS) – Pin40~Pin44
- KBC0~KBC4 (COLUMN) – Pin47~Pin51

Serial Port

The UART serial interface uses the two pins for proper data communication, which are RX and TX. Both pins have no independence on any other pins or modules. In SIM900A these pins are available but it also has some other pins for status/indication of data. By combining these pins, the serial port helps to generate the RS-232 connector too. All the serial pins are:

- RXD – Pin10 – To receive the data
- TXD – Pin 9- To send the data
- RTS – Pin8 – To send the request of data transmission
- CTS – Pin7 – To clear the send request
- RI – Pin4 – Ring indicator
- DSR – Pin6 – To indicate that data set ready
- DCD – Pin5 – To indicate data carry detect
- DTR – Pin3 – To indicate data terminal ready

Debug Interface

Debugging helps the developers to debug the module and update its firmware. In this module, there are sperate serial interface pins for debugging. Both pins are:

- DBG_TXD – Pin27 – For Data Transmission
- DBG_RXD – Pin28 – For Data receiving

SIM Interface

As we know that module SIM900A is a GPRS/GSM module. The module is dependent on some devices for some of its features. The most important one is the SIM. The SIM needs to connect with the module for GPRS/GSM functions to fully operate. All the sim interface of the module is:

- SIM_VDD – Pin30 – Power Supply of the SIM
- SIM_DATA – Pin31 – For data output
- SIM_CLK – Pin32 – For clock pulse
- SIM_RST – Pin33 – For reset
- SIM_PRESENCE – Pin34 – To detect the SIM

SIM900A Analog to Digital converter Pins

The module has only a single pin to detect and convert the analog signal to digital for SIM900A. The voltage range on the ADC pin is from 0 to 3 only.

- ADC – Pin25

PWM Pins

The PWM is mostly in microcontrollers for industrial applications but due to IoT, the module offers two PWM pins which helps to make the IoT and PWM based device without using any third interface.

- PWM1 – Pin35
- PWM2 – Pin36

Audio Interface

The audio interface will help to connect the mic and speaker with SIM900A. The connection of Line, Audio and Speaker will help to make the calls through the modules.

- MIC_P – Pin19
- MIC_N – Pin20
- SPK_P – Pin21
- SPK_N – Pin22
- LINEIN_R – Pin23

- LINE_L – Pin24

Control Pin

There is power on pins on the device, which helps to turn it on using external signals. There is two power on pins. The first one is PWRKEY which requires a LOW signal to power on/off the system. To do that, the pins require an input signal for a little bit long time. The second pin is PWRKEY_OUT, which gets short with the PWRKEY pin and turn on/off the device.

- PWRKEY – Pin1
- PWRKEY_OUT – Pin2

Reset pins

The device has an external LOW input signal reset pin to reset the device with the use of an external signal.

- NRESET – Pin16

SIM900A GSM Module RF Antenna

To extend the range of the SIM900A the antenna pin needs to connect with an external wire. The official antenna is also available for the module.

- RF_ANT – Pin60

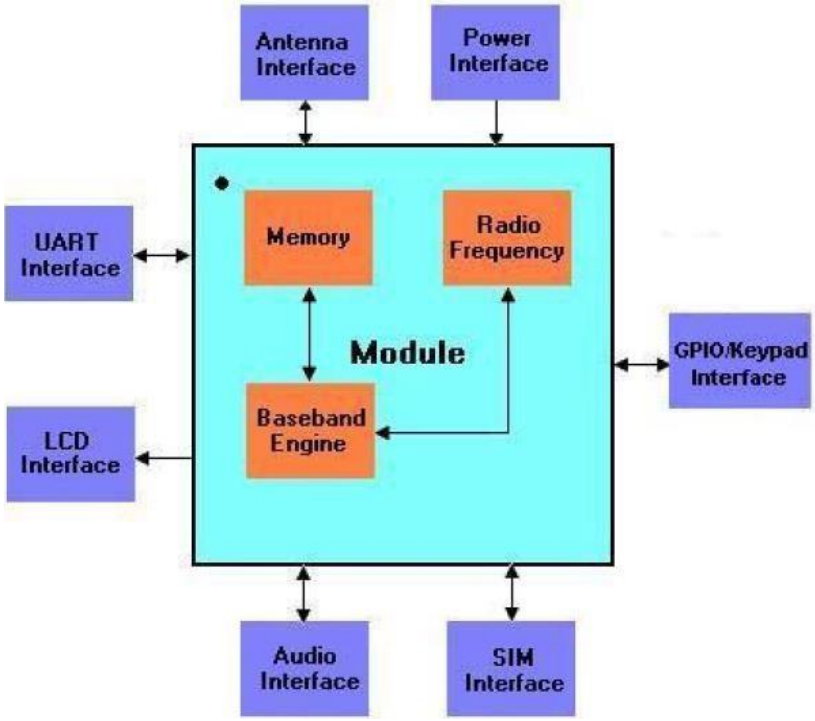
Power Pins

The module SIM900A has multiple types of power pin. Some works as input and some as output. The most important one to understand is VRTC, which acts as a backup for the internal RTC of the device. All power and ground pins of the module are:

- VBAT(Input) – Pin55, Pin56, Pin57
- VRTC (Input/Output) – Pin26
- VDD_EXT(OUTPUT) – Pin15
- GND – Pin17, Pin18, Pin29, Pin39, Pin45, Pin46, Pin53, Pin54, Pin58, Pin59, Pin61, Pin62, Pin63, Pin64, Pin65

SIM900A GSM Module Block Diagram

The following diagram is describing the SIM900A internal structure of the module.



Features of SIM900A

- Quad-Band GSM/ 850/900/1800/1900MHz
- Compatible with arduino, raspberry pi, arm, avr, pic, 8051
- power supply 12v 1amp to 2 amps max
- Use in the area of full signal strength.
- Perfect suited for GSM based Microcontroller Projects (better than SIM300 and other GSM Modems)
- Option for connecting MIC and SPEAKER directly to GSM MODEM for calls (LINE IN also available)
- Supports communication through RS232 with DB9 Connector, TTL Pins and I2C Pins
- CALL SMS GPRS facility - MIC input, LINE input and SPEAKER output pins

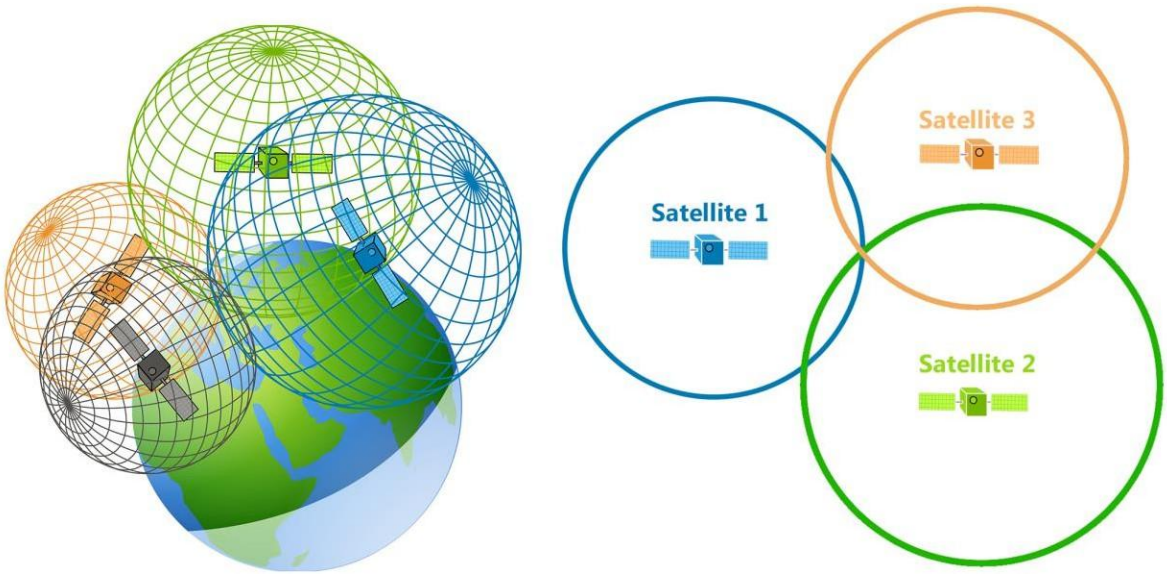
3.2.5 NEO-6M GPS Module:-

To understand the working of the NEW-6M GPS module, we first need to understand the working of a GPS system.

How GPS works?

Any instant of time, there are at least 4 GPS satellites in line of sight to a receiver on the earth. Each of these GPS satellites sends information about its position and the current time to the GPS receiver at fixed regular instants of time. This information is transmitted to the receiver in the form of signal which is then intercepted by the receiver devices. These signals are radio signals that travel with the speed of light. The distance between a GPS receiver and the satellite is calculated by finding the difference between the time the signal was sent from GPS satellite and the time the GPS receiver received the signal.

Once the receiver receives the signal from at least three satellites, the receiver then points its location using trilateration process. A GPS requires at least 3 satellites to calculate 2-D position(latitude and longitude on a map). In this case, the GPS receiver assumes that it is located at mean sea level. However, it requires at least 4 satellites to find receivers 3-D position(latitude, longitude, and altitude).



What is trilateration?

Trilateration is the process of determining your position based on the intersection of spheres. When a receiver receives a signal from one of the satellite, it calculates its distance from the satellite considering a 3-D sphere with the satellite located at the center of the sphere. Once the receiver does the same with 3 other GPS satellites, the receiver then proceeds to find the intersection point of the 3 spheres to calculate it's location.

Once the position of a receiver is calculated, the GPS device can then easily calculate:

- Time of sunrise and sunset
- Speed
- Track
- distance to destination of the GPS receiver.

Technical challenges face by GPS:

- Time synchronization between individual satellites and the GPS receiver
- Real time update of the exact location of the GPS satellite
- Precise measurement of time of flight
- Interference with other signals

NEO-6M GPS Chip

At the heart of the module is a NEO-6M GPS chip from u-blox. The chip measures less than the size of a postage stamp but packs a surprising amount of features into its little frame.

It can track up to 22 satellites on 50 channels and achieves the industry's highest level of sensitivity i.e. -161 dB tracking, while consuming only 45mA supply current.

Unlike other GPS modules, it can do up to 5 location updates a second with 2.5m Horizontal position accuracy. The u-blox 6 positioning engine also boasts a Time-To-First-Fix (TTFF) of under 1 second.

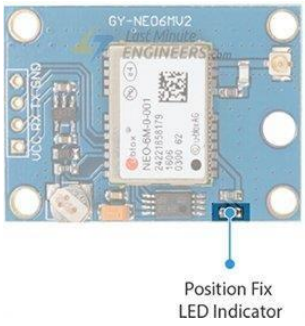
The necessary data pins of NEO-6M GPS chip are broken out to a 0.1″ pitch headers. This includes pins required for communication with a microcontroller over UART. The module supports baud rate from 4800bps to 230400bps with default baud of 9600.

Here are complete specifications:

Receiver Type	50 channels, GPS L1(1575.42Mhz)
Horizontal Position Accuracy	2.5m
Navigation Update Rate	1HZ (5Hz maximum)
Capture Time	Cool start: 27sHot start: 1s
Navigation Sensitivity	-161dBm
Communication Protocol	NMEA, UBX Binary, RTCM
Serial Baud Rate	4800-230400 (default 9600)
Operating Temperature	-40°C ~ 85°C
Operating Voltage	2.7V ~ 3.6V
Operating Current	45mA
TXD/RXD Impedance	510Ω

Position fix LED Indicator

There is an LED on the NEO-6M GPS Module which indicates the status of Position Fix. It'll blink at various rates depending on what state it's in:



No Blinking – It's searching for satellites.
Blink every 1s – Position Fix is found (The module can see enough satellites).

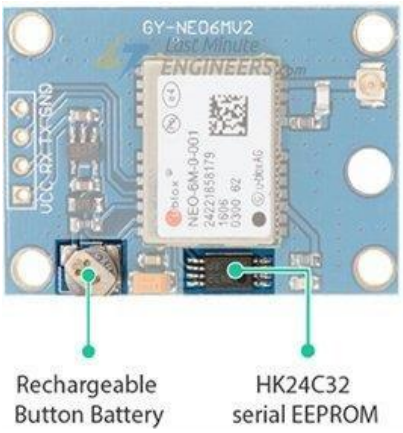
3.3 V LDO Regulator

The operating voltage of the NEO-6M chip is from 2.7 to 3.6V. But the good news is that, the module comes with MIC5205 ultra-low dropout 3V3 regulator from MICREL.

The logic pins are also 5-volt tolerant, so we can easily connect it to an Arduino or any 5V logic microcontroller without using any logic level converter.

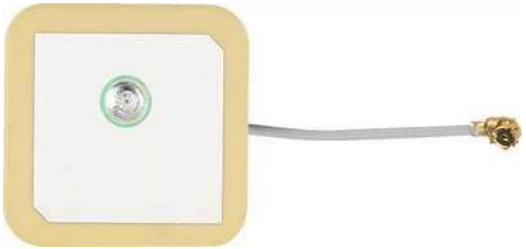
BATTERY AND EEPROM

The module is equipped with an HK24C32 two wire serial EEPROM. It is 4KB in size and connected to the NEO-6M chip via I2C. The module also contains a rechargeable button battery which acts as a super-capacitor. An EEPROM together with battery helps retain the battery backed RAM (BBR). The BBR contains clock data, latest position data(GNSS orbit data) and module configuration. But it's not meant for permanent data storage. As the battery retains clock and last position, time to first fix (TTFF) significantly reduces to 1s. This allows much faster position locks. Without the battery the GPS always cold-start so the initial GPS lock takes more time. The battery is automatically charged when power is applied and maintains data for up to two weeks without power.

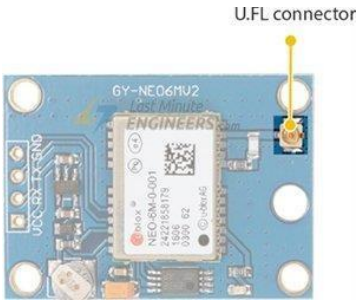


Antenna

An antenna is required to use the module for any kind of communication. So, the module comes with a patch antenna having -161 dBm sensitivity.

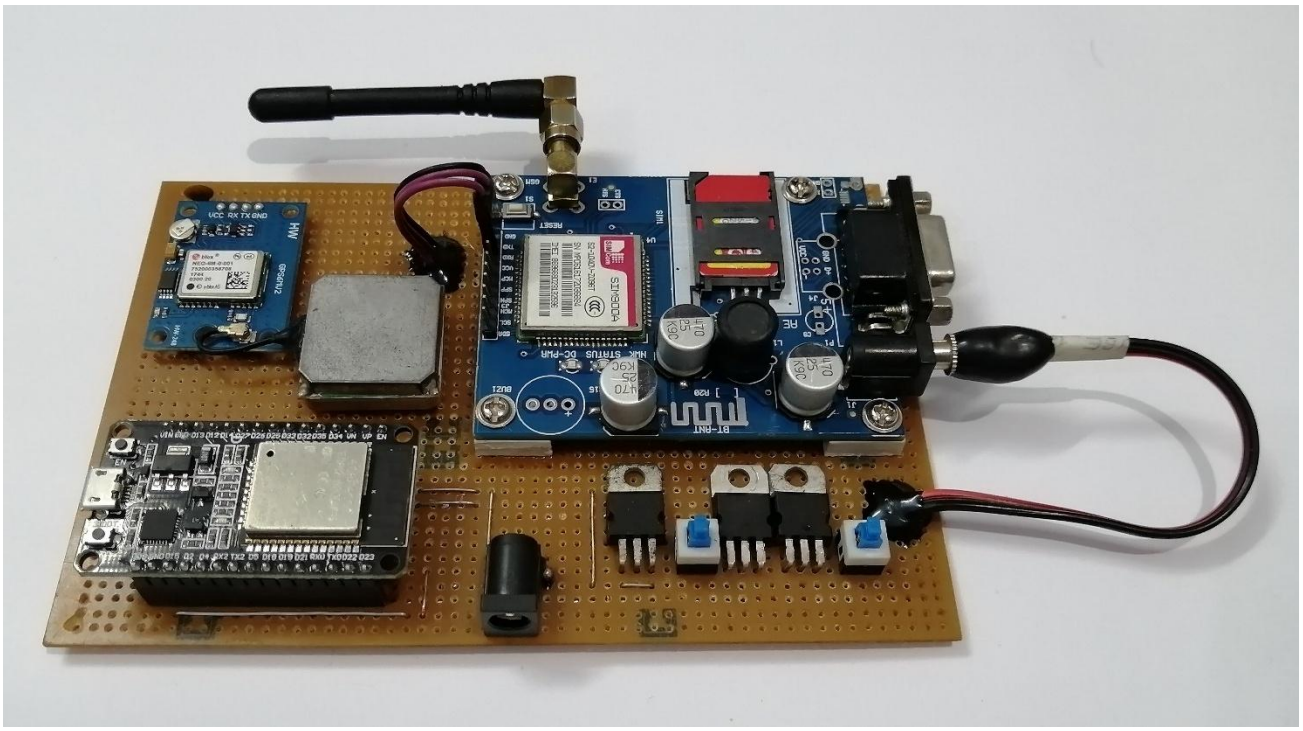


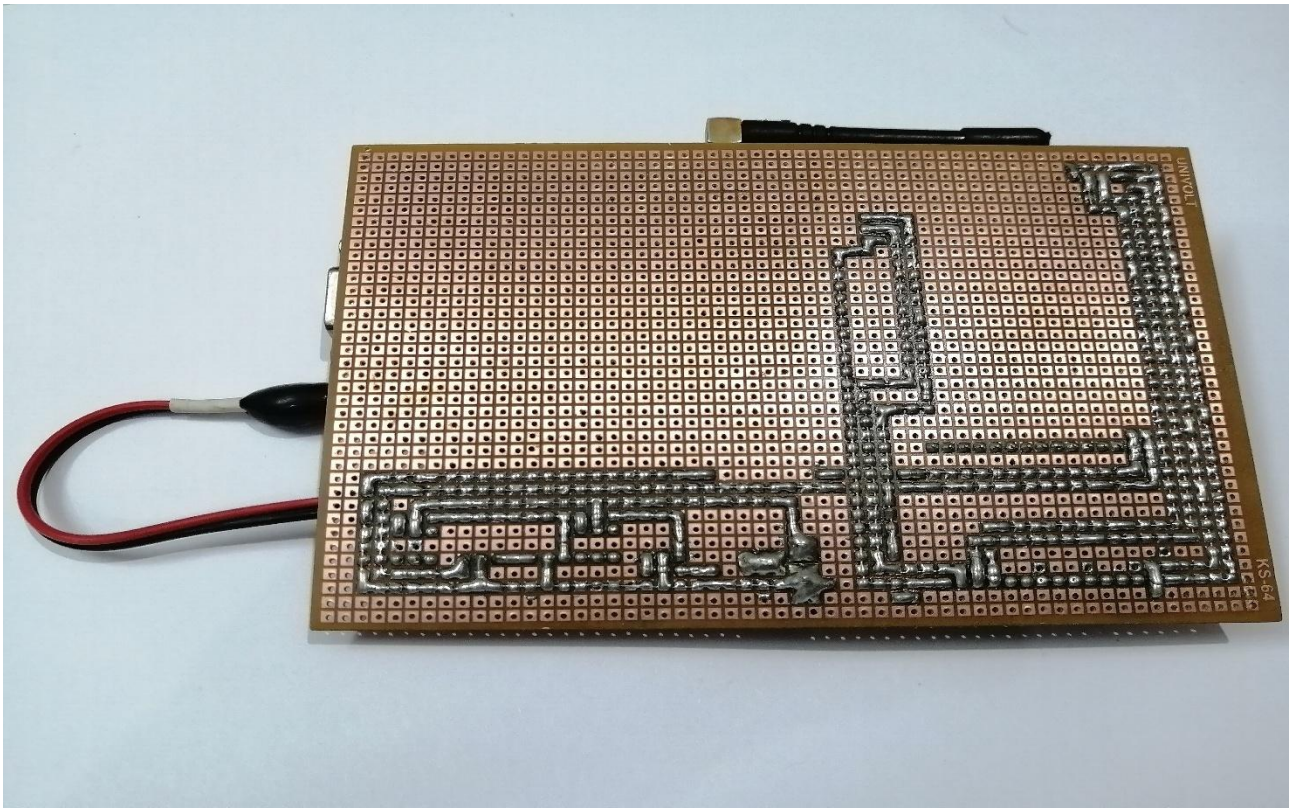
You can snap-fit this antenna to small U.FL connector located on the module.



Patch antenna is great for most projects. But if you want to achieve more sensitivity or put your module inside a metal case, you can also snap on any 3V active GPS antenna via the U.FL connector.

Hardware Capture:





CHAPTER 4

CODING

Coding is as follows….

CODE for Hardware:

```
#include <TinyGPS++.h>
#include <SoftwareSerial.h>
/* Create object named SIM900 of the class SoftwareSerial */

static const int RXPin1 = 22, TXPin1 = 23, RXPin2 = 03, TXPin2 = 01;
static const uint32_t DEFAULTBaud = 9600;

// The TinyGPS++ object
TinyGPSPlus gps;


// The serial connection to the GPS device
SoftwareSerial ss(RXPin2, TXPin2);
SoftwareSerial gprsSerial(RXPin1, TXPin1);

bool gsm_f = false;

void setup()
{
  gprsSerial.begin(DEFAULTBaud); // the GPRS baud rate
  Serial.begin(DEFAULTBaud); // the GPRS baud rate
  ss.begin(DEFAULTBaud);
  delay(1000);
  print("\nInitializing...\n\n");
  print("\nConecting to the GPS Satellites...\n");
}

void loop()
{
```

```
  // This sketch displays information every time a new sentence is cor
rectly encoded.
  while (ss.available() > 0){
    gps.encode(ss.read());
    if (gps.location.isUpdated()){

      print("\n Reading the location data\n\n");

      double latitude, longitude;

      latitude = gps.location.lat(); // Latitude in degrees (double)
      longitude = gps.location.lng(); // Longitude in degrees (double)

      Serial.print("Latitude= ");
      Serial.print(latitude);
      Serial.print(" Longitude= ");
      Serial.print(longitude);


      //  Connect wit sim900a GPRS Connection
      if(!gsm_f){
        gsm_f=true;
        Serial.print("\nInitiating GPRS Connection...\n");
        start_gsm_connection();
      }

      send_data_to_server(latitude, longitude);

    }
  }
}


void send_data_to_server(float h, float t)
```

```
{
  gprsSerial.println("AT+CIPSPRT=0");
  delay(3000);
  ShowSerialData();

  gprsSerial.println("AT+CIPSTART=\"TCP\",\"api.thingspeak.com\",\"80\
"");//start up the connection
  delay(6000);
  ShowSerialData();

  gprsSerial.println("AT+CIPSEND");//begin send data to remote server
  delay(4000);
  ShowSerialData();

  String str="GET https://api.thingspeak.com/update?api_key=your_api_k
ey&field1=" + String(h) +"&field2="+String(t);
  Serial.println(str);
  gprsSerial.println(str);//begin send data to remote server

  delay(4000);
  ShowSerialData();

  gprsSerial.println((char)26);//sending
  delay(5000);//waitting for reply, important! the time is base on the
 condition of internet
  gprsSerial.println();

  ShowSerialData();

  gprsSerial.println("AT+CIPSHUT");//close the connection
  delay(100);
  ShowSerialData();
```

```
}

void start_gsm_connection()
{
  gprsSerial.println("AT");
  delay(1000);

  gprsSerial.println("AT+CPIN?");
  delay(1000);

  gprsSerial.println("AT+CREG?");
  delay(2000);

  gprsSerial.println("AT+CGATT?");
  delay(1000);

  gprsSerial.println("AT+CIPSHUT");
  delay(1000);

  gprsSerial.println("AT+CIPSTATUS");
  delay(2000);

  gprsSerial.println("AT+CIPMUX=0");
  delay(2000);

  ShowSerialData();

  gprsSerial.println("AT+CSTT=\"www\"");//start task and setting the A
PN,
  delay(1000);

  ShowSerialData();
```

```
  gprsSerial.println("AT+CIICR");//bring up wireless connection
  delay(3000);

  ShowSerialData();

  gprsSerial.println("AT+CIFSR");//get local IP adress
  delay(2000);

  ShowSerialData();
}

void ShowSerialData()
{
  while(gprsSerial.available()!=0)
  Serial.write(gprsSerial.read());
  delay(5000);
}


void print(char *str){
  Serial.print(str);
}
```

CODE for Web App (geolocation.php):

```
<?php

session_start();

if(isset($_SESSION['login'])){
}else
  $_SESSION['login'] = false;


$loggedin = $_SESSION['login'];

if(!$loggedin){ header('location:http://adrakchai.000webhostapp.com/');
}

?>

<!DOCTYPE html>
<html>
<head>
<style type="text/css">

@import url('https://fonts.googleapis.com/css2?family=Work+Sans&displa
y=swap');
@import url('https://fonts.googleapis.com/css2?family=Ubuntu&display=s
wap');

*{
    margin: 0;
    padding: 0;
    font-family: 'Work Sans', sans-serif;
}
```

```css
html, body {
  background: #FDFDFD;
}

input{
    border: none;
    padding: 1px 7px;
    border-radius: 5px;
    width: 65px;
    background: none;
}

.width{
    width: 300px;
}

.width-180{
  width: 180PX;
}

.width-100{
  width: 100PX;
}

.button {
    color: #a19fa5;
    font-size: 17px;
    background: white;
    padding: 9px 14px;
    box-shadow: 3px 3px 30px rgb(118 96 168 / 20%);
    display: inline-block;
    cursor: pointer;
```

```
    transition: transform 0.2s ease, box-
shadow 0.2s ease, color 0.2s ease;
    margin-bottom: 15px;
}

.button:hover {
  color: #DBD7E0;
  transform: translateY(3px);
  box-shadow: 3px 3px 30px rgba(118, 96, 168, 0.17);
}

.arrow {
  width: 13px;
  transition: transform 0.3s ease;
  float: right;
  margin-top: 9px;
}

.arrow.open {
  transform: rotate(180deg);
}

.button p {
  display: inline;
  margin-right: 75px;
}

.dropdown {
  position: relative;
  font-size: 17px;
  background: white;
  padding-top: 10px;
  padding-bottom: 10px;
```

```
  display: block;
  cursor: pointer;
  transform: scale(0.01);
  opacity: 0;
  transition: transform 0.3s ease, opacity 0.3s ease, box-
shadow 0.3s ease 0.15s;
  transform-origin: center top;
  overflow: hidden;
}

.dropdown.open {
  transform: scale(1);
    opacity: 0.8;
    box-shadow: 3px 3px 30px rgb(118 96 168 / 15%);
    background: black;
    color: white !important;
    border: 1px solid #681414;
    padding: 0 13px;
}

.dropdown a {
  position: relative;
  color: #6C6185;
  text-decoration: none;
  display: block;
  padding: 12.5px 30px;
  transition: color 0.2s ease, background-color 0.2s ease, padding-
left 0.2s ease;
  overflow: hidden;
}

.dropdown a.clicked {
  padding-left: 35px;
```

```
  color: #987CD9;
}

.dropdown a:hover {
  color: #987CD9;
  padding-left: 35px;
}

span {
  z-index: -1;
  top: 0;
  left: 0;
  bottom: 0;
  width: 0px;
  position: absolute;
  background: #987CD9;
  transition: width 0.4s ease;
  border-radius: 1px;
}

span.clicked {
  width: 5px;
}

.button2{
  box-
shadow: 0 8px 16px 0 rgb(0 0 0 / 20%), 0 6px 20px 0 rgb(0 0 0 / 19%);
  background-color: #4CAF50;
  border: none;
  color: white;
  padding-top: 11px;
  padding-left: 5px;
  padding-right: 5px;
```

```
  text-align: center;
  text-decoration: none;
  display: inline-block;
  font-size: 13px;
  margin: 8px 2px;
  cursor: pointer;
  -webkit-transition-duration: 0.4s;
  transition-duration: 0.4s;
}


.btn-group .button2 {
  background-color: #4CAF50; /* Green */
  border: 1px solid green;
  box-
shadow: 0 8px 16px 0 rgba(0,0,0,0.2), 0 6px 20px 0 rgba(0,0,0,0.19);
  color: white;
  padding: 15px 32px;
  text-align: center;
  text-decoration: none;
  font-size: 16px;
  cursor: pointer;
  width: 150px;
  display: block;
}

i{
  color: #eef1f3;
}

.btn-group .button2:not(:last-child) {
  border-bottom: none; /* Prevent double borders */
}
```

```
.created_at i{
  align-self: flex-end;
  font-size: 12px;
  margin-bottom: 10px;
}

.btn-group .button2:hover {
  background-color: #3e8e41;
}

.history_tr{
  padding: 10px 0;
}

.logout_btn{
  border-radius: 25px;
  padding: 5px 6px;
}

.logout_btn:hover{
  background-color: #ab2734 !important;
}


</style>
    <title>Vehical Tracking System</title>

    <meta charset="utf-8">
    <meta name="viewport" content="initial-scale=1,maximum-
scale=1,user-scalable=no">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jq
uery.min.js"></script>
    <link href="https://api.mapbox.com/mapbox-gl-js/v2.2.0/mapbox-
gl.css" rel="stylesheet">
```

```
    <script src="https://api.mapbox.com/mapbox-gl-js/v2.2.0/mapbox-gl.js"></script>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/css/bootstrap.min.css" rel="stylesheet">
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">

  <link rel="stylesheet" type="text/css" href="fonts/font-awesome-4.7.0/css/font-awesome.min.css">
<!--
================================================================================-->
  <link rel="stylesheet" type="text/css" href="fonts/Linearicons-Free-v1.0.0/icon-font.min.css">
<!--
================================================================================-->
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>

    <style>
        body { margin: 0; padding: 0; }
        #map { position: absolute; top: 0; bottom: 0; width: 100%; transition: all 0.3s; }
        #left_menu{ z-index: 5; position: absolute; width: fit-content; padding-left: 11px; margin-top: 10px; height: 230px; }
    </style>

</head>
<body>

<div id="left_menu">
    <div style="background-color: #fff;" class="width pt-1">
```

```
        <div class="ms-1 mt-1 pb-2" style="width: fit-content;">
            <font class="fs-4 fw-
bold"><a href="http://rohitkrtiwari.000webhostapp.com/logout.php" clas
s="btn fa fa-power-off text-white bg-danger ms-1 me-
1 logout_btn" style="    border-
radius:25px;padding: 5px 6px;"></a>Live Vehical Tracking</font><br>
        </div>
    </div>

    <button id="fly" class="button2">RELOCATE
      <div class="container-fluid-0 mt-1 pb-1" style="font-
size: 10px;">
        <div class="d-flex">
          <i class="lat_lang_title_style">Lat:</i> <input type="number
" class="ms-2 m-0 p-0 text-
white" name="latitude_val" disabled  id="latitude_val">
        </div>
        <div class="d-flex">
          <i class="lat_lang_title_style">Long:</i> <input type="text"
 class="ms-2 p-0 m-0 text-
white" name="longitude_val" disabled id="longitude_val">
        </div>
      </div>
    </button>


      <div class="button mt-1 width-180"> <p>History</p> <img
      src="https://svgshare.com/i/8u6.svg" alt="" class="arrow"> </div
>

      <div class="dropdown width" id="history"></div>
```

```
</div>
<script type="text/javascript">
    $(document).ready(function(){
  $(".button").click(function() {
    $(".dropdown a").removeClass("clicked");
    $(".dropdown a").children("span").removeClass("clicked");
    $(".arrow").toggleClass("open");
    $(".dropdown").toggleClass("open");
  });

  $(".dropdown a").click(function() {
    $(".dropdown a").removeClass("clicked");
    $(".dropdown a").children("span").removeClass("clicked");
    $(this).toggleClass("clicked"); $(this).children("span").toggleCl
ass("clicked");
  });
});
</script>
<div id="map"></div>
<br>


<script>

    mapboxgl.accessToken = 'your_API_token';
    var map = new mapboxgl.Map({
        container: 'map',
        style: 'mapbox://styles/vehical-
tracking/ckn7qx70r11jm17o0t1oo3w53',
        zoom: 8
    });

    var popup =  new mapboxgl.Popup({
        closeButton: false,
```

```
        closeOnClick: false
    });

    var field2;
    var field1;

    // Add zoom and rotation controls to the map.
    map.addControl(new mapboxgl.NavigationControl(), 'top-right');


    var url = 'read_api_thingspeak';
    map.on('load', function () {
        var request = new XMLHttpRequest();
        window.setInterval(function () {
            // make a GET request to parse the GeoJSON at the url
            request.open('GET', url, true);
            request.onload = function () {
                if (this.status >= 200 && this.status < 400) {
                    // retrieve the JSON from the response
                     var json = JSON.parse(this.response);

                    if( parseFloat(json.feeds[0].field2) != field2 ||
parseFloat(json.feeds[0].field2) != field2 )
                    {



                        field2 = parseFloat(json.feeds[0].field2);
                        field1 = parseFloat(json.feeds[0].field1);



                        var geojson = {
                            "name":"NewFeatureType",
```

```
                        "type":"FeatureCollection",
                        "features":[{
                            "type":"Feature",
                            "geometry":{
                                "type":"Point",
                                "coordinates":[field2, field1]
                            },
                            "properties":{
                                'description':json.feeds[0].create
d_at
                            }
                        }]
                    };

                    var created_at = json.feeds[0].created_at;


                    // update the drone symbol's location on the m
ap

                    // fly the map to the drone's current location
                    map.getSource('drone').setData(geojson);
                    console.log(geojson.features[0].geometry.coord
inates);
                    map.flyTo({
                        center: geojson.features[0].geometry.coord
inates,
                        speed: 0.5,
                    });


                    update_fields(geojson.features[0].geometry.coo
rdinates[0], geojson.features[0].geometry.coordinates[1]);
```

```
                        map.on("mouseenter", "drone", () => {
                            popup.setHTML(created_at)
                            .setLngLat(geojson.features[0].geometry.co
ordinates)
                                .addTo(map);
                            map.getCanvas().style.cursor = "pointer";
                        });
                        map.on("mouseleave", "drone", () => {
                          map.getCanvas().style.cursor = "";
                          popup.remove();
                        });

                    }

                }
            };
            request.send();
        }, 2000);

        map.addSource('drone', { type: 'geojson', data: url });
        map.addLayer({
            'id': 'drone',
            'type': 'symbol',
            'source': 'drone',
            'layout': {
                'icon-image': 'rocket-15',
                'icon-size': 2
            }
        });
    });
    document.getElementById('fly').addEventListener('click', function
() {
      // Fly to a random location by offsetting the point -74.50, 40
```

```
      // by up to 5 degrees.
      map.flyTo({
        center: [
          field2,
          field1
        ],
        essential: true // this animation is considered essential with
respect to prefers-reduced-motion
      });
      var geojson = {
          "name":"NewFeatureType",
          "type":"FeatureCollection",
          "features":[{
              "type":"Feature",
              "geometry":{
                  "type":"Point",
                  "coordinates":[field2, field1]
              },
              "properties":{
                  'description':0
              }
          }]
      };
    map.getSource('drone').setData(geojson);
    });


    var xmlhttp = new XMLHttpRequest();
    xmlhttp.onreadystatechange = function() {
      if (this.readyState == 4 && this.status == 200) {
        var myObj = JSON.parse(this.responseText);
        var html="";

        var pre_field1, pre_field2 = 0;
```

```
        for(var i in myObj.feeds){

          if( myObj.feeds[i].field2 != '' && myObj.feeds[i].field2 !=
0 && myObj.feeds[i].field2 != null && myObj.feeds[i].field2 != 'null'
&& myObj.feeds[i].field1 != '' && myObj.feeds[i].field1 != 0 && myObj.
feeds[i].field1 != null && myObj.feeds[i].field1 != 'null' )
          {

            if(myObj.feeds[i].field2 != pre_field2 && myObj.feeds[i].f
ield2 != pre_field2)
            {
              pre_field2 = myObj.feeds[i].field2;
              pre_field1 = myObj.feeds[i].field1;

              temp= '<div class="d-
flex history_tr" onclick="fly('+myObj.feeds[i].field2+', '+myObj.feeds
[i].field1+')""><div class="m-0 width-100" style="font-
size: 13px;">Lat: '+myObj.feeds[i].field2+' <br> Long: '+myObj.feeds[i
].field1+'</div> <div class="m-0 created_at d-
flex"> <i>At : '+myObj.feeds[i].created_at+' </i></div></div>';
              html = temp.concat(html);
            }
          }
        }
        console.log(html)
        $("#history").append(html);
      }
    };
    xmlhttp.open("GET", "thingspeak_read_api", true);
    xmlhttp.send();


function update_fields(lat, long){
```

```
    $("#latitude_val").val(lat);
    $("#longitude_val").val(long);
}

function fly(temp_lat, temp_long){
  map.flyTo({
    center: [
      temp_lat,
      temp_long
    ],
    essential: true // this animation is considered essential with res
pect to prefers-reduced-motion
  });
  var geojson = {
        "name":"NewFeatureType",
        "type":"FeatureCollection",
        "features":[{
            "type":"Feature",
            "geometry":{
                "type":"Point",
                "coordinates":[temp_lat, temp_long]
            },
            "properties":{
                'description':0
            }
        }]
    };
  map.getSource('drone').setData(geojson);
}

</script>
</body>
</html>
```

(login.php):

```
<?php
session_start();
$msg='';

if(isset($_SESSION['login'])){
}else
    $_SESSION['login'] = false;


$loggedin = $_SESSION['login'];

if($loggedin){
   header('location:http://adrakchai.000webhostapp.com/geolocation
.php') ;
}

if(isset($_POST['submit']))
{
    $username=$_POST['username'];
    $password=$_POST['pass'];
    if($username == "adrakchai@gmail.com" && $password == 'adm in123')
    {
        $_SESSION['login'] = true;
        $loggedin = $_SESSION['login'];
        header('location:http://adrakchai.000webhostapp.com/geoloc
ation.php') ;
    }else{
         $msg = "Please Enter Correct Login Details";
    }
}
?>
```

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Login V12</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
<!--
===============================================================
============================-->
    <link rel="icon" type="image/png" href="images/icons/favicon.ico"/
>
<!--
===============================================================
============================-->
    <link rel="stylesheet" type="text/css" href="vendor/bootstrap/css/
bootstrap.min.css">
<!--
===============================================================
============================-->
    <link rel="stylesheet" type="text/css" href="fonts/font-awesome-
4.7.0/css/font-awesome.min.css">
<!--
===============================================================
============================-->
    <link rel="stylesheet" type="text/css" href="fonts/Linearicons-
Free-v1.0.0/icon-font.min.css">
<!--
===============================================================
============================-->
    <link rel="stylesheet" type="text/css" href="vendor/animate/animat
e.css">
```

```
<!--
===============================================================================
============================-->
    <link rel="stylesheet" type="text/css" href="vendor/css-
hamburgers/hamburgers.min.css">
<!--
===============================================================================
============================-->
    <link rel="stylesheet" type="text/css" href="vendor/select2/select
2.min.css">
<!--
===============================================================================
============================-->
    <link rel="stylesheet" type="text/css" href="css/util.css">
    <link rel="stylesheet" type="text/css" href="css/main.css">
<!--
===============================================================================
============================-->

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
beta1/dist/css/bootstrap.min.css" rel="stylesheet">
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/fo
nt-awesome/4.7.0/css/font-awesome.min.css">

</head>
<body>

    <div class="limiter">
        <div class="container-login100" style="background-
image: url('images/img-01.jpg');">
            <div class="wrap-login100 p-t-190 p-b-30">
                <form class="login100-form validate-
form" method="POST">
```

```
                    <div class="login100-form-avatar">
                        <img src="images/123.jpg" alt="AVATAR">
                    </div>

                    <span class="login100-form-title p-t-20 p-b-45">
                        Rohit Tiwari
                    </span>

                    <div class="wrap-input100 validate-input m-b-
10" data-validate = "Username is required">
                        <input class="input100" type="text" name="user
name" placeholder="Username">
                        <span class="focus-input100"></span>
                        <span class="symbol-input100">
                            <i class="fa fa-user"></i>
                        </span>
                    </div>

                    <div class="wrap-input100 validate-input m-b-
10" data-validate = "Password is required">
                        <input class="input100" type="password" name="
pass" placeholder="Password">
                        <span class="focus-input100"></span>
                        <span class="symbol-input100">
                            <i class="fa fa-lock"></i>
                        </span>
                    </div>

                    <div class="container-login100-form-btn p-t-10">
                        <button class="login100-form-
btn" type="submit" name="submit">
                            Login
                        </button>
```

```
                    </div>

                    <?php if($msg!='') echo "<span style='color:red'>*
*".$msg."</span>"; ?>


                </form>
            </div>
        </div>
    </div>

<!--
============================================================================
==========================-->
    <script src="vendor/jquery/jquery-3.2.1.min.js"></script>
<!--
============================================================================
==========================-->
    <script src="vendor/bootstrap/js/popper.js"></script>
    <script src="vendor/bootstrap/js/bootstrap.min.js"></script>
<!--
============================================================================
==========================-->
    <script src="vendor/select2/select2.min.js"></script>
<!--
============================================================================
==========================-->
    <script src="js/main.js"></script>

</body>
</html>
```

(logout.php):

```
<?php
session_start();
session_unset();

session_destroy(); header('location:http://adrakchai.000webhostapp.com/');
?>
```

Interface:

CHAPTER 5

ADVANTAGES AND FUTURE SCOPE

5.1 ADVANTAGES:

Commercial fleet operators are by far the largest users of vehicle tracking systems. These systems are used for operational functions such as routing, security, dispatch and collecting on-board information.
These are also used for fire detector in large vehicles like train, bus etc. because the vehicle like train contains large number of people and the sending alert of fire accident can save many lives.
The applications for this project are in military, navigation, automobiles, aircrafts, fleet management, remote monitoring, remote control, security systems, tele services, etc.

- Fleet monitoring
- Vehicle scheduling
- Route monitoring
- Driver monitoring
- Accident analysis
- Geo-fencing geo-coding

These are just a few advantages of the project that has been introduced in this report . We can interface more number of sensors in order to serve multiple purposes. The microcontroller that has been used in this project have inbuit ADCs and hence the controller is capable of accepting analog inputs, which is the biggest advantage. Since all real world signals are analog in nature, by incorporating different sensors required purpose can be served.

5.2 FUTURE SCOPE:

- We can use the EEPROM to store the previous Navigating positions up to 256 locations and we can navigate up to N number of locations by increasing its memory.
- We can reduce the size of the kit by using GPS+GSM on the same module.
- We can increase the accuracy up to 3m by increasing the cost of the GPS receivers.
- We can use our kit for detection of bomb by connecting to the bomb detector.
- With the help of high sensitivity vibration sensors we can detect the accident. whenever vehicle unexpectedly had an accident on the road with help of vibration sensor we can detect the accident and we can send the location to the owner, hospital and police.
- We can use our kit to assist the traffic. By keeping the kits in the entire vehicles and byknowing the locations of all the vehicles.

CHAPTER 6

CONCLUSION

6.1 CONCLUSION :

Vehicle tracking system makes better fleet management and which in turn brings large profits. Better scheduling or route planning can enable you handle larger jobs loads within a particular time. Vehicle tracking both in case of personal as well as business purpose improves safety and security, communication medium, performance monitoring and increases productivity. So in the coming year, it is going to play a major role in our day-to-day living.

Main motto of the project is to incorporate different types of sensors so that they help in decrease the chances of losing life in such accident which we can't stop from occurring. Whenever accident is alerted the paramedics are reached to the particular location to increase the chances of life. This device invention is much more useful for the accidents occurred in deserted places and midnights. This vehicle tracking and accident alert feature plays much more important role in day to day life in future.