Vehicle Tracking System using Internet of Things

Thesis · May 2018

DOI: 10.13140/RG.2.2.12437.96482

Argha Ghosh
Alagappa University

VEHICLE TRACKING SYSTEM USING IoT

SCHOOL OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE
Master of Computer Applications

PROJECT REPORT

NAME	**:**	**ARGHA GHOSH**
REG. NO.	**:**	**15352095**
BATCH	**:**	**II**
SEMESTER	**:**	**VI**
YEAR	**:**	**III**

PONDICHERRY UNIVERSITY

(A Central University)
R.Venkataraman Nagar, Kalapet
PUDUCHERRY-605014

PONDICHERRY UNIVERSITY

(A Central University)

R.Venkataraman Nagar, Kalapet

PUDUCHERRY-605014

SCHOOL OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE

Master of Computer Application

BONAFIDE CERTIFICATE

Certified that this is a bonafide report of project work done by **ARGHA GHOSH** with register number **15352095** of **M.C.A** in semester **VI** during the academic year 2017-18.

Project Guide

R.P.SEENIVASAN

Assistant Professor,

Dept. of Computer Science

Head of the Department

Dr. T. Chithralekha

Associate Professor/ HOD,

Dept. of Computer Science

Submitted for the viva-voce examination held on 2nd May, 2018 at the Department of Computer Science, Pondicherry University

Internal Examiner

External Examiner

ACKNOWLEDGEMENT

An idea is the first step for generation & development of anything. The initial idea may have been of any one person, but the development & improvement on it have always been a result of hard work, help, inspiration and blessings of several different people.

I would also like to thank my project Guide **R.P.SEENIVASAN**, Assistant Professor, Department of Computer Science School of Engineering Technology, Pondicherry University who never hesitated in extending their sincere co-operation from time to time to prepare an interesting and challenging project.

It is a pleasure acknowledgement with deep sense of gratitude the invaluable help by **Dr. T. Chithralekha, Head (i/c)**, Department of Computer Science, School of Engineering

& Technology, Pondicherry University for the successful completion of this project report.

The guidance and support received from all the teaching and non-teaching staff of the Department of Computer Science was vital for the success of the project. I am also thankful to my family and friends for their continuous support and encouragement throughout the course. I take this opportunity to express my gratitude to the people who have been instrumental in the completion of this project.

Argha Ghosh
15352095
MCA-3rd Year

ABSTRACT

In this project we are going one step ahead with GPS and going to track a vehicle using GPS and GSM. This Vehicle Tracking System can also be used for Accident Detection Alert System, Soldier Tracking System and many more, by just making few changes in hardware and software.

Tracking of vehicle is a process in which we track the vehicle location in form of Latitude and Longitude (GPS coordinates). GPS Coordinates are the value of a location. This system is very efficient for outdoor application purpose.

This kind of Vehicle Tracking System Project is widely in tracking Cabs/Taxis, stolen vehicles, school/colleges buses etc.

This system is basically an embedded one. Embedded stands for hardware controlled by software. Here, the software using a Arduino controls all the hardware components. Arduino plays an important role in the system.

Currently almost of the public having an own vehicle, theft is happening on parking and sometimes driving insecurity places. The safe of vehicles is extremely essential for public vehicles. Vehicle tracking and locking system installed in the vehicle, to track the place.

The place of the vehicle identified using Global Positioning system (GPS) and Global system mobile communication (GSM). These systems constantly watch a moving Vehicle and report the status on demand. When the theft identified, the responsible person send SMS to the microcontroller, then microcontroller issue the control signals to stop the engine motor. Authorized person need to send the password to controller to restart the vehicle and open the door. This is more secured, reliable and low cost.

TABLE OF CONTENTS

1.INTRODUCTION

1.1 Research background

Vehicle Tracking system is getting popular and widely used in a lot of countries worldwide. It has tons of advantages to users even more to the vehicle users in which it will make it easier for them to track their vehicles.

Nowadays, everyone cannot be separated from their smartphones. a number of five thousands individuals from USA, UK, South Korea, India, China, South Africa, Indonesia and Brazil took a survey regarding which was done by Time magazine. The result proved most of them is inseparable from their smartphones, eighty four per cent allegedly claimed that survive without their smartphones.

Another study shows that seventy five per cent of the market share is smartphone and a total of one hundred and six million smartphone were shipped in the second half of 2012.Smartphone became the top telecommunication medium in the market in the present time worldwide and it became the most popular used telecommunication medium known to man.

So, from the above mentioned survey now it’s clear that how smartphones became important and integral part of our modern day life, that’s the reason to make this vehicle tracking system text message oriented so that we can take care of our own vehicle in just one touch of our hand. Through smart phone we can track real time location of our vehicle with the help of internet connection. In such a manner, this tracking system designed so that users can have easy and user friendly interface to fetch their vehicle.

1.2 Problem statement

In the present day vehicle tracking is becoming essential for the purpose of improving our life condition. Convenience and ease of using vehicle is what home vehicle tracking is offering. Vehicle tracking offers a futuristic way of life in which an individual gets to control his vehicle using a smart phone, from tracking a vehicle /detecting accidental place of a vehicle; it also offers an efficient use of technology.

But to get or acquire such system installed will cost a lot of money and that is the major reason of why vehicle tracking has not received much demand and attention, adding to that also the complexity of installing it and configuring it. Thus it is essential to make it cost effective and easy to configure, if this is granted to people then they will be willing to acquire it in their personal vehicles, school buses and taxes/cabs etc. In other words, a system modification for the vehicle tracking is required in order to lower the price of applying it to vehicles. Also this tracking project can be used to purpose of women safety as well as parents can be used to take care of their child/kid for the safety or missing purpose or to track their activities for their future.

Even more realistically this project can be used to track airline baggage because as we know every year almost 13% airline baggage used to get missing by a worldwide survey.

1.3 Objective of the study

To construct a vehicle tracking system controlled by a smartphone specifically an embedded device. To design and implement cost effective vehicle tracking system yet an efficient one.

To design a user friendly and a safe system to control vehicles especially aimed to aid the all aged people. As well as it's designed such a way so that it can be used for multi-purpose like detecting the accident place of the vehicle or the accident alert system as well as it's useful to track soldiers or to track child/ kid for their safety or missing purpose and also can be useful for women safety purpose. So when this embedded system design take care of every aspect of its purpose as it mentioned above.

1.4 Scope of study

In order to fulfil the stated objectives several steps must be taken. These steps involve both software programming and hardware implementation.

These steps are as follows:

- Establishing a wireless network communication between the GSM module and the smart phone, using a microcontroller (Arduino-Uno).
- Create a simple yet reliable vehicle tracking system using Arduino-Uno as a microcontroller that will be the medium between the GPS and the GSM module so that embed system works efficiently.
- To find a suitable place locator app (in this project we are using Google Maps) that will work efficiently with the internet connection (online as well as offline) in order to track the vehicles.
- Program the Arduino-Uno board in a way that will let it interact with the GPS and GSM module directly and easily.

1.5 Significance of the study

This study will be undertaken to create a vehicle tracking system at low cost and easy to create, this will benefit both the manufacturer and the client. It will help the manufacturer by making it easy and cheaper to apply it, and it will also benefit the clients by making it cost effective and the most important advantage is that it will make the vehicle a much more safer than its actually for the clients.

2. MODULES

Vehicle Tracking System

Tracking of vehicle is a process in which we track the vehicle location in form of Latitude and Longitude (GPS coordinates). GPS Coordinates are the value of a location. This system is very efficient for outdoor application purpose.

This kind of Vehicle Tracking System Project is widely in tracking Cabs/Taxis, stolen vehicles, school/colleges buses etc.

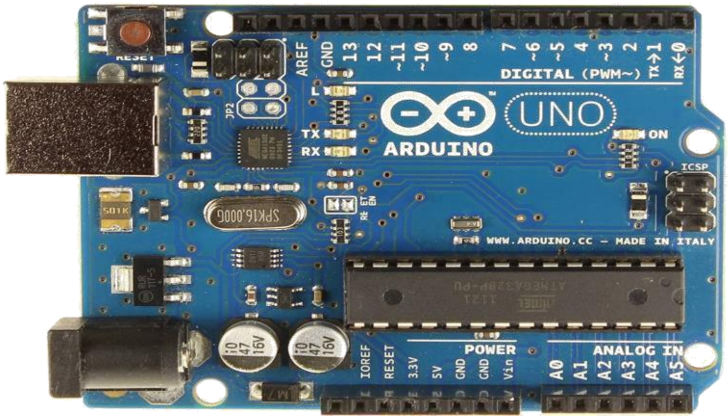
This circuit is designed for tracking the location of vehicles. Most of tracking systems are made by using GPS. This is very simple and cheap. Tracking systems are mostly used by fleet operators for tracking a vehicle location, routing and others. This is a very good method for preventing our vehicles from stolen. This tracking system sends us the geographical coordinates and by using these coordinates we can track our vehicle position on electronic maps using internet. By using these tacking systems we can share real time information about transportations. And also can be share real time information or position bus/taxi/cab with passengers. Means passengers can see the real time of arriving bus/taxi/cab on Mobiles.

2.1 Arduino

Arduino is open source physical processing which is base on a microcontroller board and an incorporated development environment for the board to be programmed. Arduino gains a few inputs, for example, switches or sensors and control a few multiple outputs, for example, lights, engine and others. Arduino program can run on Windows, Macintosh and Linux operating systems (OS) opposite to most microcontrollers’ frameworks which run only on Windows. Arduino programming is easy to learn and apply to beginners and amateurs.

Arduino is an instrument used to build a better version of a computer which can control, interact and sense more than a normal desktop computer. It's an open-source physical processing stage focused around a straightforward microcontroller board, and an environment for composing programs for the board. Arduino can be utilized to create interactive items, taking inputs from a diverse collection of switches or sensors, and controlling an assortment of lights, engines, and other physical outputs. Arduino activities can be remaining solitary, or they can be associated with programs running on your machine (e.g. Flash, Processing and Maxmsp.) The board can be amassed by hand or bought preassembled; the open-source IDE can be downloaded free of charge.

Focused around the Processing media programming environment, the Arduino programming language is an execution of Wiring, a comparative physical computing platform.



2.1.1 Why choosing Arduino

There are numerous different microcontrollers and microcontroller platforms accessible for physical computing. Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard, and numerous others offer comparative usefulness. These apparatuses take the chaotic subtle elements of microcontroller programming and wrap it up in a simple to-utilize bundle. Arduino additionally rearranges the methodology of working with microcontrollers; moreover it offers some advantages for instructors, students, and intrigued individuals:

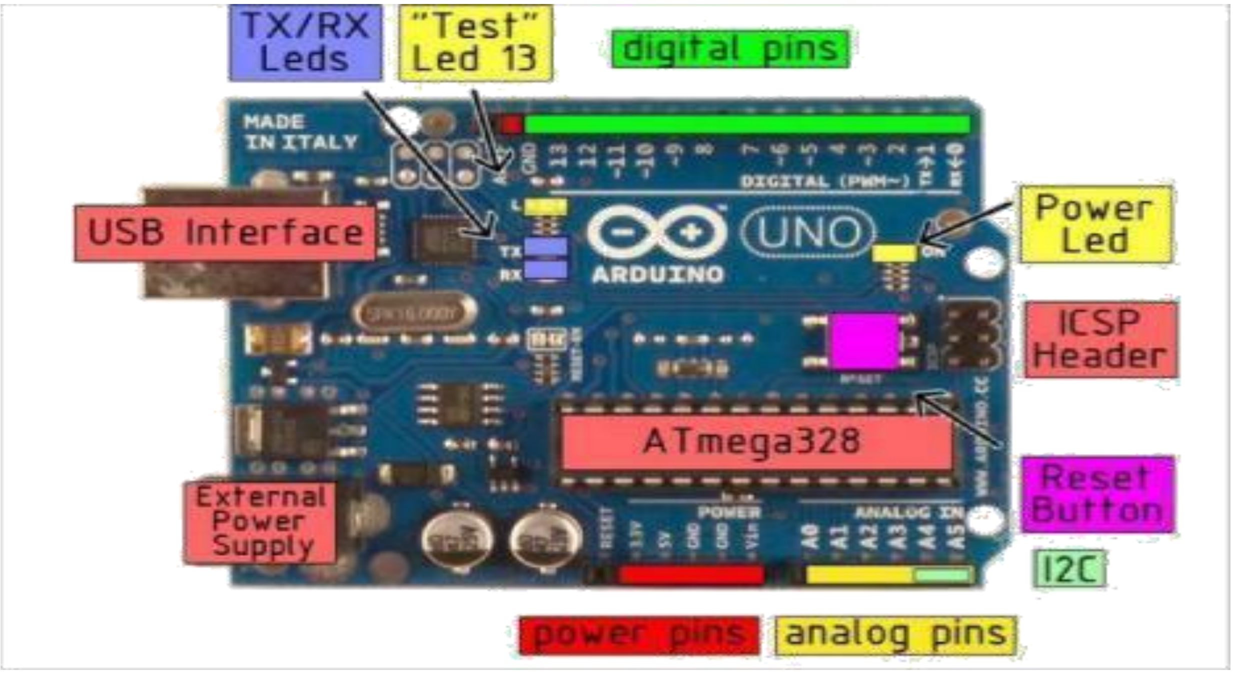
• Inexpensive - Arduino boards are moderately cheap compared with other microcontroller boards. The cheapest version of the Arduino module can be amassed by hand, and even the preassembled Arduino modules cost short of 450 bucks.

• Cross-platform - The Arduino programming runs multiple operating systems Windows, Macintosh OSX, and Linux working frameworks. So we conclude that Arduino has an advantage as most microcontroller frameworks are constrained to Windows.

• Straightforward, clear programming method - The Arduino programming environment is easy to use for novices, yet sufficiently versatile for cutting edge customers to adventure as well. For educators, its favourably engaged around the Processing programming environment, so understudies finding ways to understand how to program in that environment will be familiar with the nature of Arduino.

2.1.2 Little bit more about Arduino

The Arduino Uno is a microcontroller board based on the ATmega328. It has a 16 MHz ceramic resonator, 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a USB connection, a power jack, an In-Circuit Serial Programming header, and a reset button. This board is very simple and can be easily used, everything you need to support the microcontroller is in this board, just plug it in a computer via USB cable and power using an AC-to-DC adapter or battery to get started. "Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0.



2.2. GPS Module

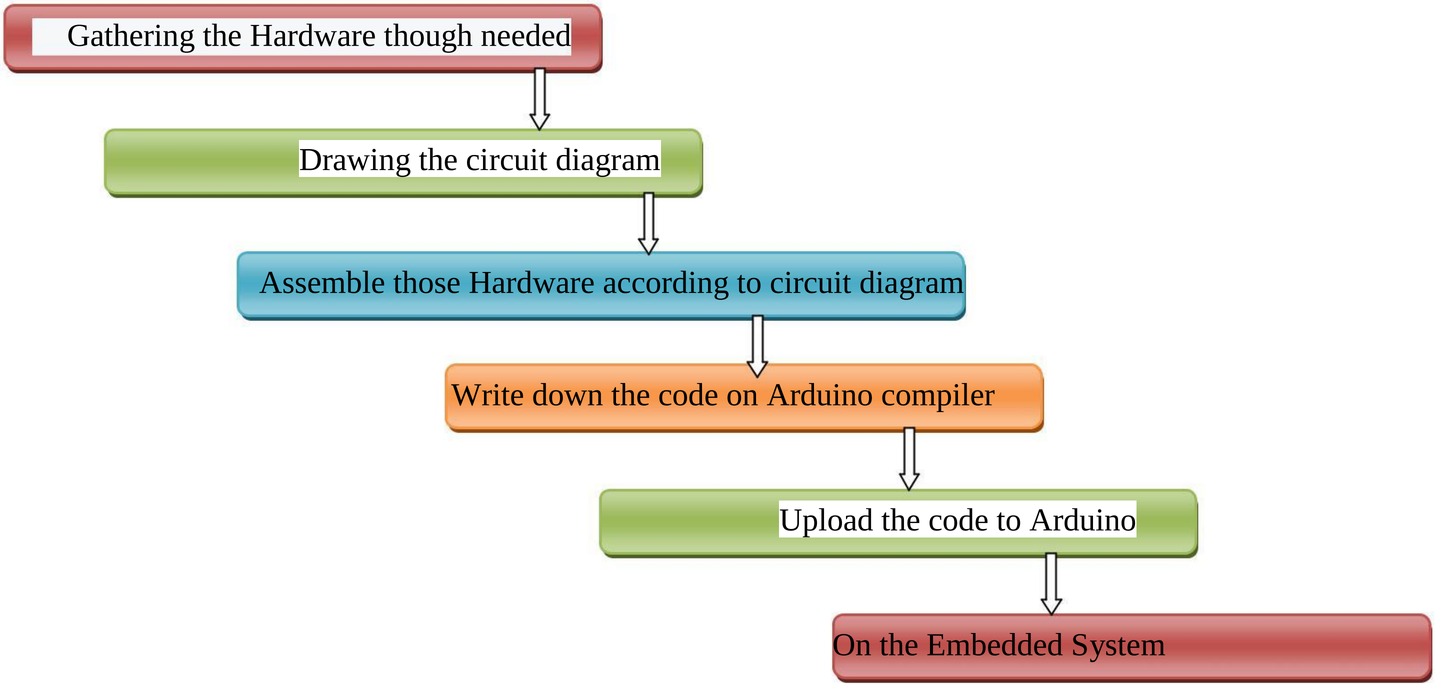
GPS stands for Global Positioning System and used to detect the Latitude and Longitude of any location on the Earth, with exact Universal Time Co-ordinate time. This device receives the coordinates from the satellite for each and every second, with time and date. GPS offers great accuracy and also provides other data besides position coordinates. The module connects to the Arduino through a 9600 bps Transistor-Transistor Logic-level interface. Only four wires are needed to read the module's GPS data. This module Compatible with 3.3V-5V interface. It also has an Electrically Erasable Programmable Read-Only Memory and a Flash which is useful to save configuration data.

2.3 GSM Module

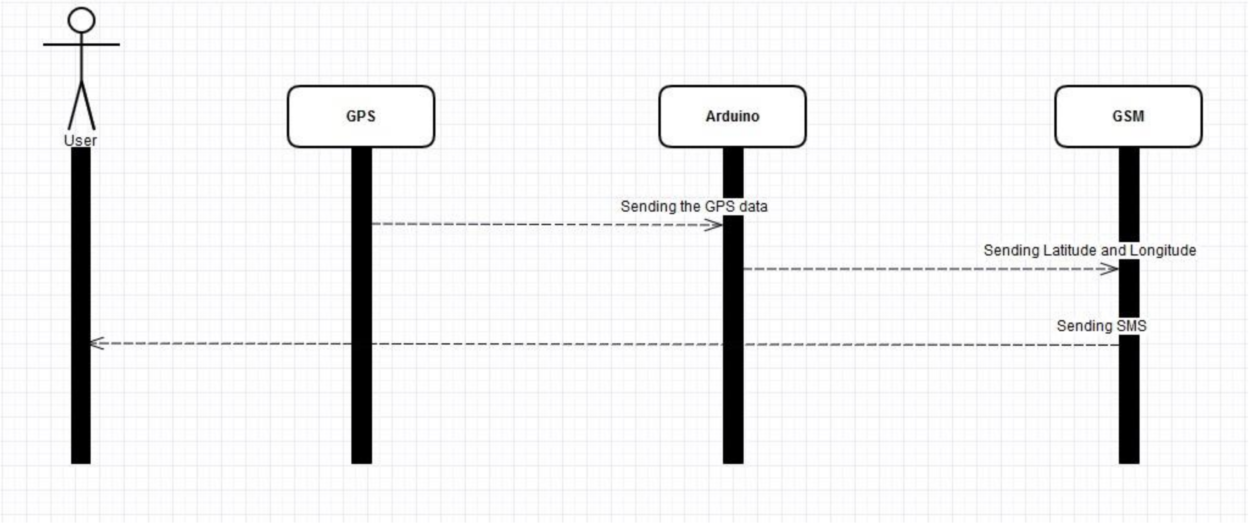
It has On board Micro SIM card holder. Its Working frequencies are quad-band network, 850 / 900 / 1800 / 1900MHZ.It can communicate with TTL(Transistor– transistor logic) serial port. It can communicate with controllers via AT commands. This module support software power on and reset. It able to make and answer phone calls ,Sending and receiving SMS. It can send and receive GPRS data through TCP/IP, HTTP protocol.

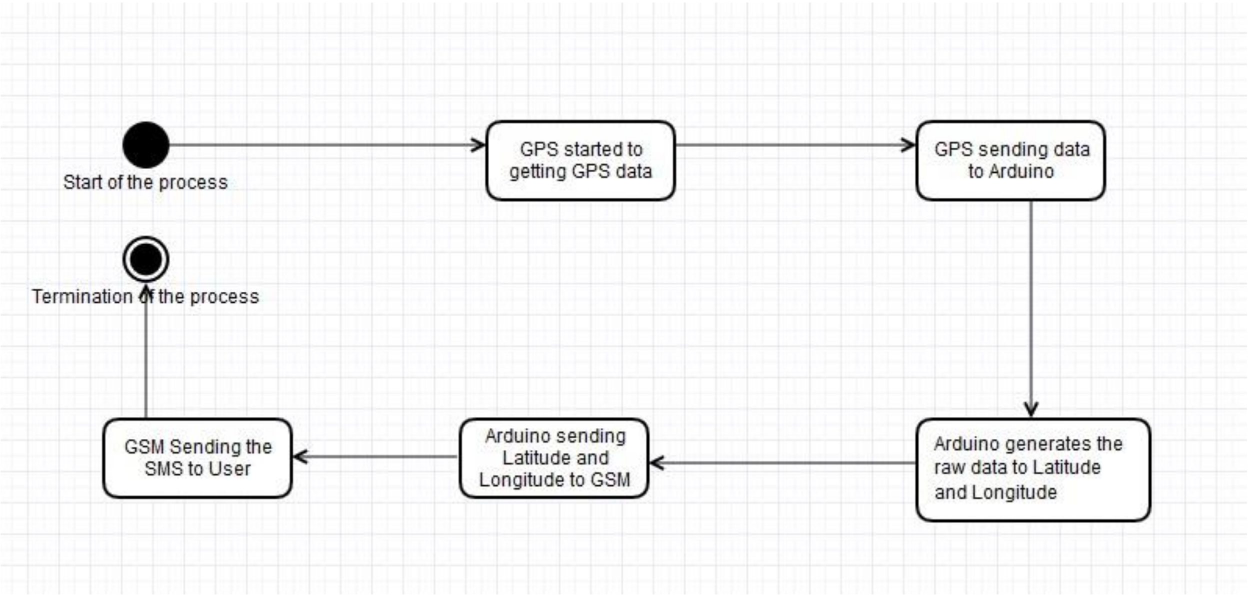
3. DESIGN

3.1. Project Flow


Gathering the Hardware though needed
Drawing the circuit diagram
Assemble those Hardware according to circuit diagram
Write down the code on Arduino compiler
Upload the code to Arduino
On the Embedded System

3.2 Sequence and Activity Diagram



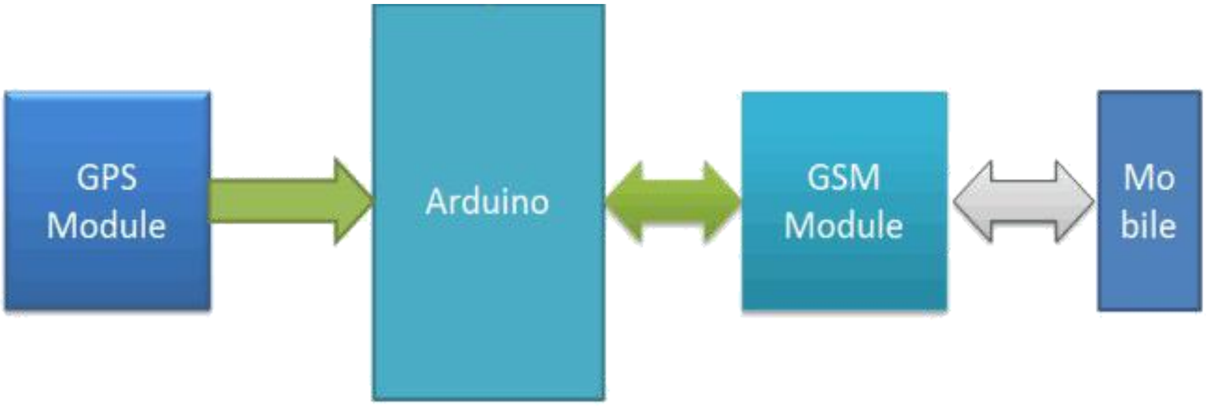

Start of the process
GPS started to getting GPS data
GPS sending data to Arduino
Arduino generates the raw data to Latitude and Longitude
Arduino sending Latitude and Longitude to GSM
GSM Sending the SMS to User
Termination of the process

3.3 Components were used

- ❑ Arduino Uno R3
- ❑ GPS Module NEO6MV2 NEO-6M
- ❑ Thinklets™ A6 GSM GPRS Module
- ❑ Power Supply (9V Batteries)
- ❑ Connecting Wires(Jumper Wires)

3.4 Working Principal

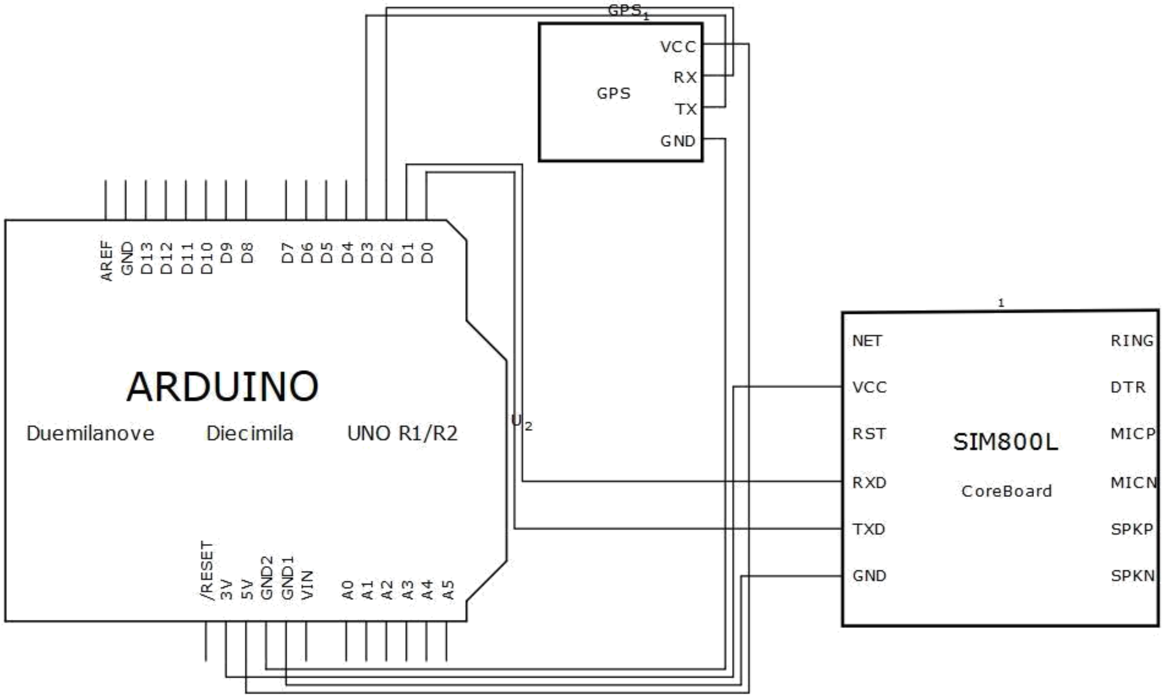
In this project, Arduino will use for controlling whole the process with a GPS Receiver and GSM module.GPS Receiver is used for detecting coordinates of the embedded system, and GSM module is used for sending the coordinates to user by SMS.When we turn on embedded system first GPS module will collect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



3.5 Circuit Explanation

GPS’s RX and TX pins are going respectively to the Arduino’s D2 and D3 pins.GPS’s GND pin common to the Arduino’s GND pins and GPS’s VCC pin goes to arduino’s 5V power source pin.

GSM’s RX and TX pins are going to Arduino’s D1 and D2 pins respectively and GND pin of GSM Module are common to the Arduino’s GND pin.

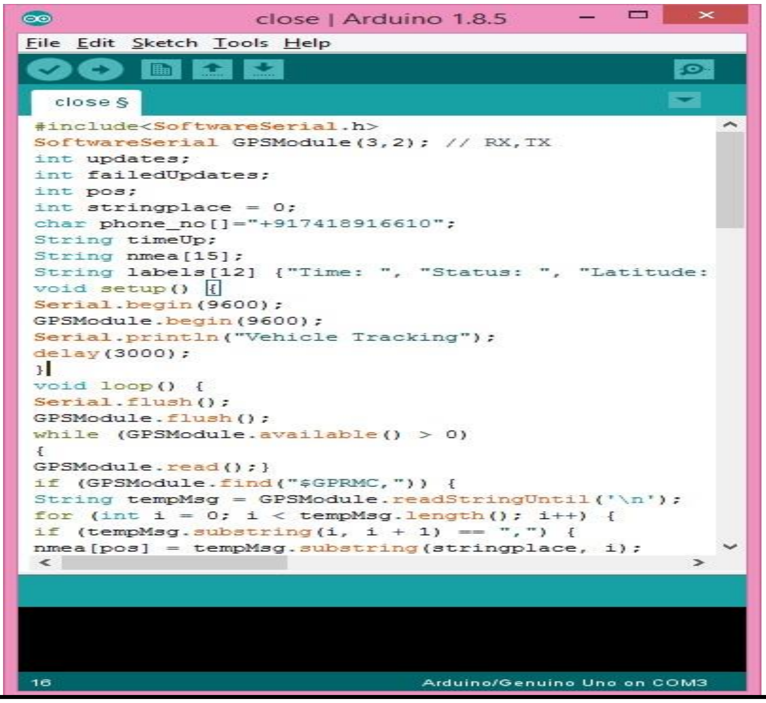


3.6 Software and Technology Specification

- Software
  - Arduino compiler version 1.8.5
  - Programming Language: C
- Technology
  - GSM
  - GPS
  - IoT - Internet of Things

3.6 Programming the Arduino-Uno

In order for the Arduino-Uno board to be able to interact with the application used in this project certain program (code) needs to be uploaded to the Arduino-Uno. Arduino Company provides user friendly software which allows writing any code for any function wanted to be performed by the Arduino-Uno and upload it to the board. Refer to chapter 4 for the full source code of the Arduino-Uno board.



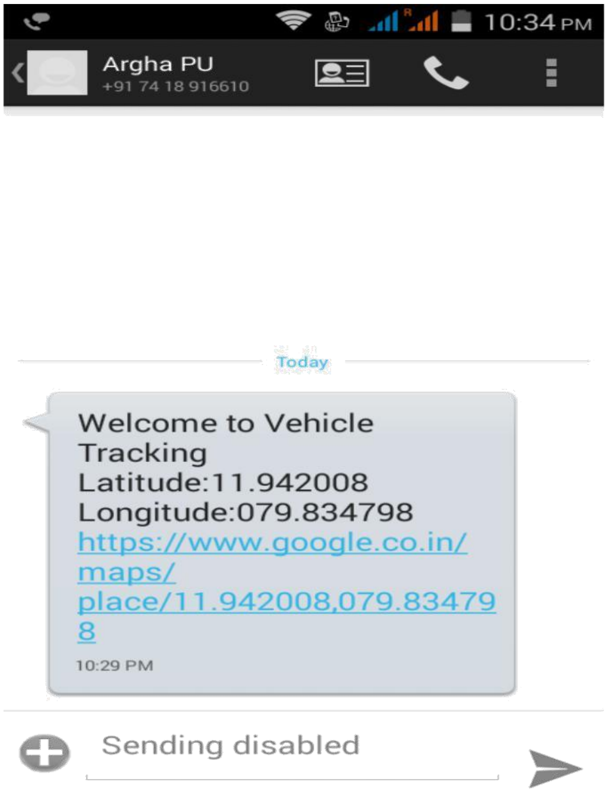
```
#include<SoftwareSerial.h>
SoftwareSerial GPSModule(3,2); // RX,TX
int updates;
int failedUpdates;
int pos;
int stringplace = 0;
char phone_no[]="+917418916610";
String timeUp;
String nmea[15];
String labels[12] {"Time: ", "Status: ", "Latitude:
void setup() {
Serial.begin(9600);
GPSModule.begin(9600);
Serial.println("Vehicle Tracking");
delay(3000);
}
void loop() {
Serial.flush();
GPSModule.flush();
while (GPSModule.available() > 0)
{
GPSModule.read();}
if (GPSModule.find("$GPRMC,")) {
String tempMsg = GPSModule.readStringUntil('\n');
for (int i = 0; i < tempMsg.length(); i++) {
if (tempMsg.substring(i, i + 1) == ",") {
nmea[pos] = tempMsg.substring(stringplace, i);
```

4. Source Code

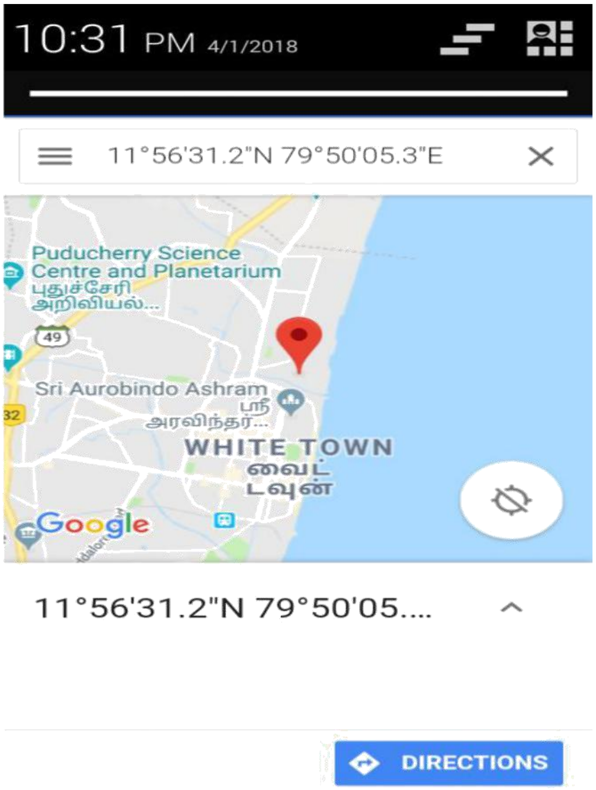
```
#include<SoftwareSerial.h>

SoftwareSerial GPSModule(3,2); /* TX,RX Pins of GPS Module*/

 int updates;

int failedUpdates;

int pos;

int stringplace = 0;

char phone_no[]="7418916610"; //User’s Phone Number

 String timeUp;

String nmea[15]; /* Character Array to store GPS data */

String labels[12] {"Time: ", "Status: ", "Latitude: ", "Hemisphere: ", "Longitude: ", "Hemisphere:
", "Speed: ", "Track Angle: ", "Date: "}; /*Respective Labels for GPS data*/

/*Main() Function */

void setup() { Serial.begin(9600); /* Setting the baud-rate of the program */

GPSModule.begin(9600); /* Starting as well as setting the baud-rate of GPS Module */

Serial.println("Vehicle Tracking");

delay(3000);} /* End of Main() Function */

//Loop() Function for performing the task again and again

void loop() { Serial.flush();

GPSModule.flush(); //Tuning on the GPS flush LED

/* Checking GPS data available or not */

while (GPSModule.available() > 0){

GPSModule.read();} /* if GPS data available then reading the GPS data */

if (GPSModule.find("$GPRMC,")) {/*Storing and retrieving GPS data in their respective order*/

String tempMsg = GPSModule.readStringUntil('\n');

for (int i = 0; i < tempMsg.length(); i++) { if(tempMsg.substring(i, i + 1) == ",") {

nmea[pos] = tempMsg.substring(stringplace, i);
stringplace = i + 1;
pos++;} //Those GPS data are in their GPS string format, not yet refine in normal form
if (i == tempMsg.length() - 1) { /* checking all the respective GPS data store or not */
nmea[pos] = tempMsg.substring(stringplace, i);}}
updates++;
/* Calling user define function to convert Latitude and Longitude respectively */
nmea[2] = ConvertLat();
nmea[4] = ConvertLng();
for (int i = 0; i < 9; i++) {
Serial.print(labels[i]);
Serial.print(nmea[i]);
Serial.println("");}
delay(300);
/* Sending SMS */
Serial.println("AT+CMGF=1");      /* AT Command for Sending SMS */
delay(2000);
Serial.print("AT+CMGS=\"");
Serial.print(phone_no); /* User’s Phone Number */
Serial.write(0x22);
Serial.write(0x0D); /* Hex Equivalent of Carriage Return
*/ Serial.write(0x0A); /* Hex Equivalent of Newline */
delay(2000);
/* SMS String or Message */
Serial.println("Welcome to Vehicle Tracking");
Serial.print("Latitude:");
Serial.println(nmea[2]);
Serial.print("Longitude:");

Serial.println(nmea[4]);

/* Generating The URL */

Serial.print("https://www.google.co.in/maps/place/");

Serial.print(nmea[2]);

Serial.print(",");

Serial.print(nmea[4]);

delay(500);

Serial.println (char(26)); /* The ASCII code of the ctrl+z is 26 */

delay(120000);} /* Creating the 2 minutes Interval between SMS’s */
else {

failedUpdates++;}

stringplace = 0;

pos = 0;}

/* Converting the Latitude from core GPS’s Latitude data */

String ConvertLat() {

String posneg = "";

if (nmea[3] == "S") {

posneg = "-";}

String latfirst;
float latsecond;

for (int i = 0; i < nmea[2].length(); i++) {

if (nmea[2].substring(i, i + 1) == ".") {

latfirst = nmea[2].substring(0, i - 2);

latsecond = nmea[2].substring(i - 2).toFloat();}}

latsecond = latsecond / 60;

String CalcLat = "";
```

```
char charVal[9];
dtostrf(latsecond, 4, 6, charVal);
for (int i = 0; i < sizeof(charVal); i++){
CalcLat += charVal[i];}
latfirst += CalcLat.substring(1);
latfirst = posneg += latfirst;
return latfirst;}
/* Converting the Longitude from core GPS's Longitude data */
String ConvertLng() {
String posneg = "";
if (nmea[5] == "W") {
posneg = "-";}
String lngfirst;
float lngsecond;
for (int i = 0; i < nmea[4].length(); i++) {
if (nmea[4].substring(i, i + 1) == ".") {
lngfirst = nmea[4].substring(0, i - 2);
lngsecond = nmea[4].substring(i - 2).toFloat();}}
lngsecond = lngsecond / 60;
String CalcLng = "";
char charVal[9];
dtostrf(lngsecond, 4, 6, charVal);
for (int i = 0; i < sizeof(charVal); i++){
CalcLng += charVal[i];}
lngfirst += CalcLng.substring(1);
lngfirst = posneg += lngfirst;
return lngfirst;}
```

5. Output

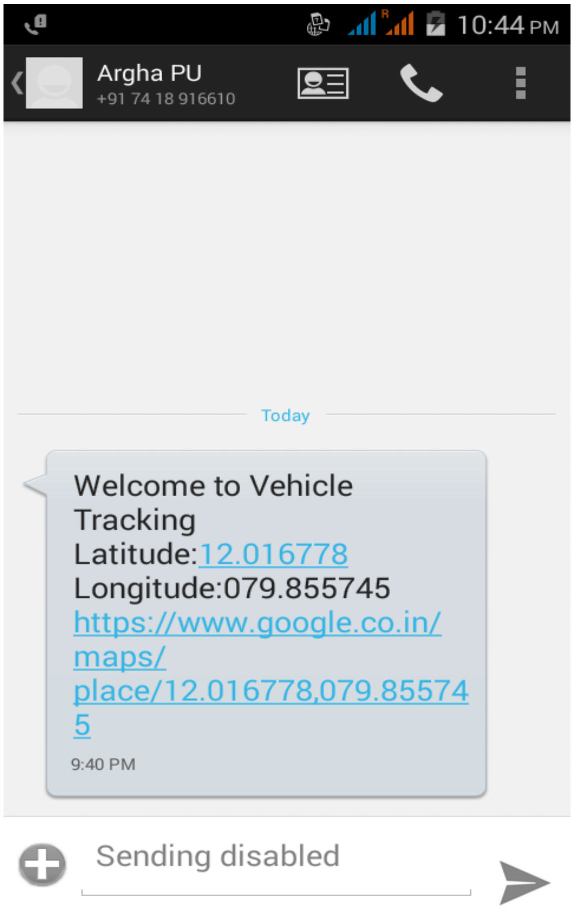


Welcome to Vehicle Tracking
Latitude:11.942008
Longitude:079.834798
https://www.google.co.in/maps/place/11.942008,079.834798
10:29 PM

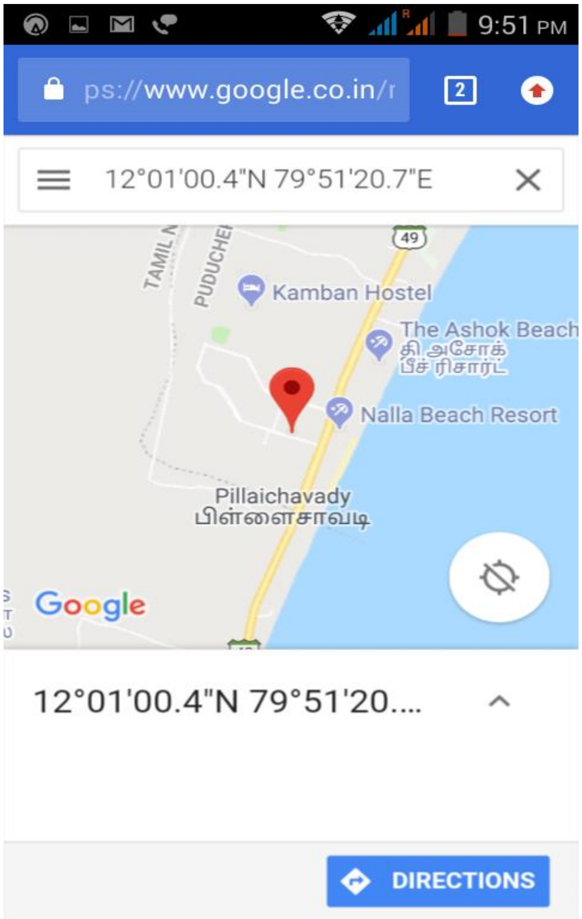


11°56'31.2"N 79°50'05.3"E

11°56'31.2"N 79°50'05....



Welcome to Vehicle Tracking
Latitude:12.016778
Longitude:079.855745
https://www.google.co.in/maps/place/12.016778,079.855745
9:40 PM

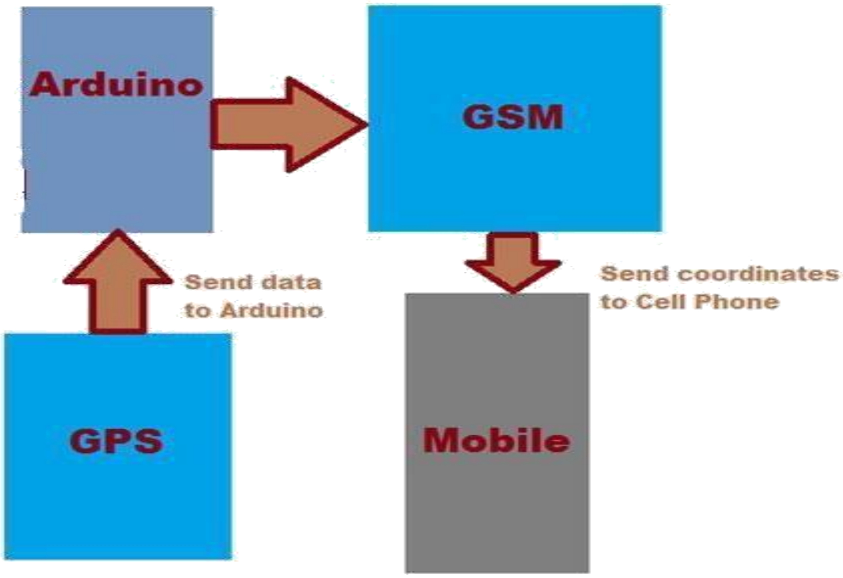


12°01'00.4"N 79°51'20.7"E

12°01'00.4"N 79°51'20....

6. RESULTS

This chapter discusses the results and any limitation and problems encountered during the period of the project. Managed to successfully apply the VEHICLE TRACKING SYSTEM USING IoT and it was user friendly and cost effective. User friendly as in anyone can use just a click of a URL on a smart phone screen and everything works. And it is cost effective as in it will cost exactly as the project requires (optimum price). Figure shows the prototype of the system.



7. CONCLUSION AND FUTURE WORKS

This chapter confers on the conclusion of Vehicle Tracking System Using IoT and discusses some future recommendation.

7.1 Conclusion

It can be concluded that VEHICLE TRACKING SYSTEM USING IoT was a success. This system consists of an Arduino-Uno board, a GPS Module, a GSM Module and a Google Map Application. It is user friendly and it is cost effective. Also it can be concluded that the objectives of this project has been successfully met and they are as follows:

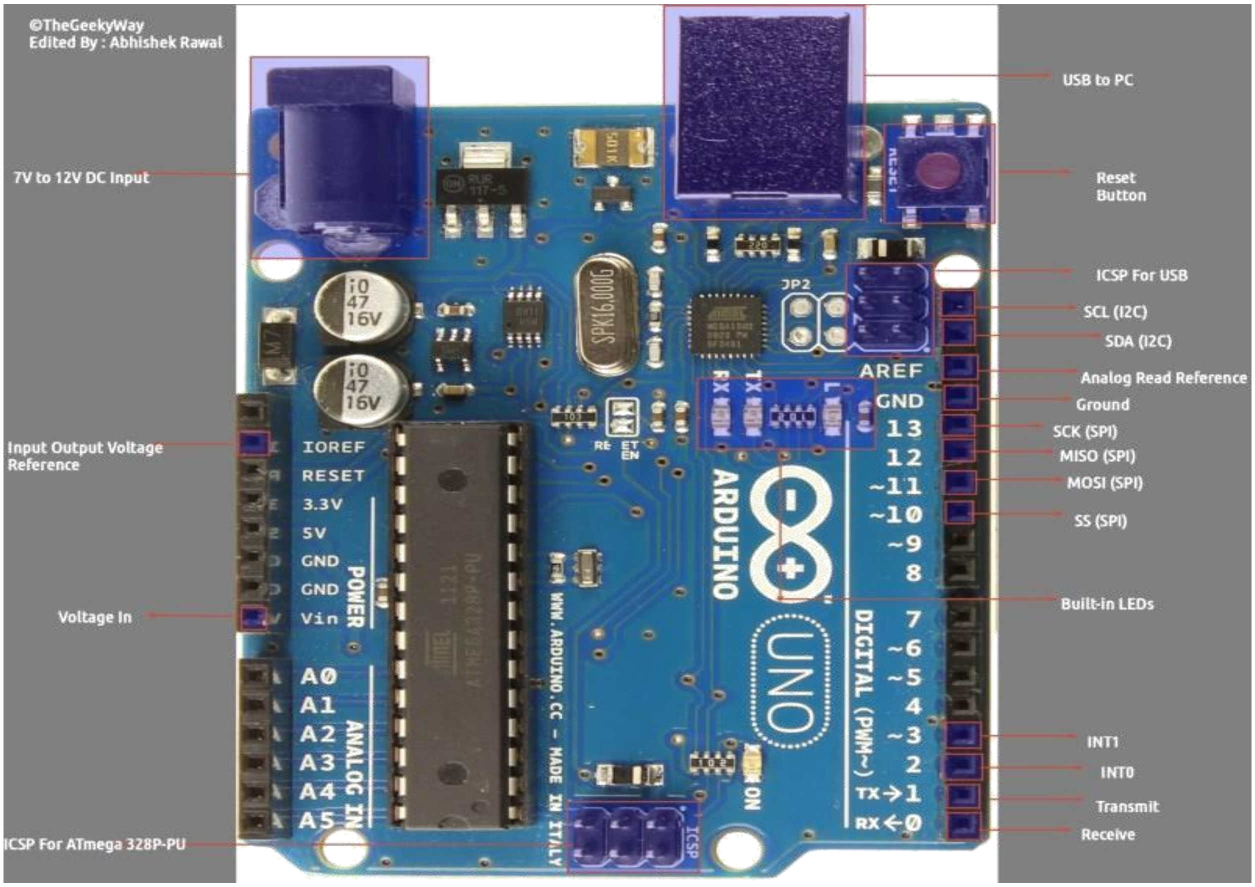
- Constructed a vehicle tracking system controlled by a smartphone specifically an embedded device.
- Designed and implement cost effective vehicle tracking system yet an efficient one.
- Designed a user friendly and a safe system to control vehicles especially aimed to aid the elders.

7.2 Future Recommendations

There are some recommendations for Future works. Some of them are:

1. Better to use breadboard because without using any resistors connecting directly those modules to arduino can goes wrong in the matter of voltage up-down.
2. Test each and every component before using them especially the relays for safety purposes.

8. ABBREVIATION AND ARDUINO UNO PIN FUNCTIONS



USB To PC : It is used to communicate Arduino via Universal Serial Bus to PC for programming/sending data to Arduino serially.

7V to 12V DC Input : For external supply, the voltage range of 7V to 12V DC is recommended. The 9V battery can be used to power your Arduino Uno board.

Reset Button : It Resets the Arduino board if pressed.

ICSP : Abbreviated as In Circuit Serial Programming which consists of MOSI, MISO, SCK, RESET, VCC, GND. It is either used to program USB or Microcontroller (For UNO, ATmega 328P-PU). ICSP allows the user to program the microcontroller when it is in circuits, In Arduino UNO it allows to program ATmega328P-PU directly with AVR instructions without using Arduino IDE.

SDA : Serial Data, It is the bidirectional data line that is used by I2C.

SCL : Serial Clock, It is used to indicate that data is ready on bidirectional data line that is used by I2C.

AREF : Analog Read Reference, It is mainly used for analogReference() function calls, as default it is not required but to use it you have to add some voltage source between 0V to 5V in AREF Pin which will be considered as accurate reference voltage.

GND : Ground.

SCK : Serial Clock, which is used by SPI (Serial Peripheral Interface). It is clock generated by 'Master' which is used to clock the data to the 'Slave'.

MOSI : Master Out Slave In, The data is transmitted from Master to Slave. (Master -> Slave)

MISO : Master In Slave Out, The data is transmitted to Master from Slave. (Slave - > Master)

SS : Slave Select, It is used to select the 'Slave'. Make high to SS pin to deactivate & make low to activate it.

INT1 & INT0 : These are hardware interrupts, it calls the ISR (Interrupt Service Routine) when the pin change occurs.

TX : Transmit, It is used to transmit TTL serial data. It is also referred as 'outwards' since it 'transmits' data from Arduino to other connected peripheral device.

RX : Receive, It is used to receive TTL serial data. IT is also referred as 'inwards' since it 'receives' data from external hardware to Arduino.

Vin : Voltage In, If you're powering your Arduino board from USB nothing is obtained from Vin pin. But, if you're powering the board with external supply then that supply is directly obtained from Vin pin. However, the supply obtained at Vin pin is usually lesser by 1V than voltage supplied to Power pin due to reverse polarity protection diode.

5V Pin : It is used to power external components connected to Arduino which needs 5V.

3.3V Pin **:** It is used to power external components connected to Arduino which needs 3.3V

IORef : Input Output Voltage Reference, It allows shields connected to Arduino board to check whether the board is running at 3.3V or 5V.