Anti-theft system for vehicle using GSM & GPS with fingerprint verification

NIMALAN A/L GUNALAN

THIS THESIS REPORT IS SUBMITTED TO FULFIL THE REQUIREMENTS FOR
THE AWARD OF THE DEGREE OF
BACHELOR OF COMPUTER ENGINEERING (HONS)

SCHOOL OF SCIENCE AND ENGINEERING

FEBRUARY 2020

DECLARATION AND COPYRIGHT PAGE

Name : NIMALAN A/L GUNALAN

Matric Number : 1106161002

I hereby declare that this research is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by footnotes giving explicit references and a bibliography is appended.

Signature: ……..…….. Date: ……

APPROVAL PAGE

ANTI-THEFT SYSTEM FOR VEHICLE USING GSM & GPS WITH FINGERPRINT VERIFICATION

BY

NIMALAN A/L GUNALAN

Approved by:

"I hereby declare that I have read this thesis and in my opinion this thesis is sufficient in terms of scope and quality for the award of the degree of Bachelor of Computer Engineering (Hons)"

Supervisor

……. Date: ……

(Ms. Aparna Vyakarnam)

ACKNOWLEDGEMENT

I would like to express my deepest gratitude to my project supervisor, Ms. Aparna Vyakarnam (Sr. Lecturer, Department of Computer Engineering and Computer Science), my supervisor and Mr. Ahammad Shafeeq Battappady Mahammad (Assistant Professor, Department of Computer Engineering and Computer Science) for their guidance, enthusiastic encouragement and useful critiques that provided me with the possibility to complete this report.

ABSTRACT

In today's times, motor vehicle theft has become very rampant. And the recovery rate of these stolen vehicles is not up to the mark. There is a need for a proper security system to prevent such crimes. Currently, there are ways to recover only after the theft has happened but not much has been done to prevent it. This Anti-theft system for vehicle using GSM & GPS with fingerprint verification aims to protect any vehicle from theft. This smart key feature informs the vehicle user in the event there is a threat of theft to his vehicle by any intruder. The system also provides a means for tracking the vehicle in case of lost keys or occurrence of a threat. This system has been developed using the latest technologies which simplifies the life of common people.

ABSTRAK

Pada zaman sekarang, kecurian kenderaan telah menjadi sangat berleluasa. Dan kadar pemulihan kenderaan yang dicuri ini tidak sampai ke markah. Terdapat keperluan untuk sistem keselamatan yang betul untuk mencegah jenayah sedemikian. Pada masa ini, terdapat cara untuk pulih hanya selepas kecurian berlaku tetapi tidak banyak yang dilakukan untuk mencegahnya. Sistem Anti-kecurian ini untuk kenderaan menggunakan GSM & GPS dengan pengesahan cap jari bertujuan untuk melindungi kenderaan dari kecurian. Ciri kunci pintar ini memberitahu pengguna kenderaan sekiranya ada ancaman kecurian ke kenderaannya oleh penyerang mana pun. Sistem ini juga menyediakan cara untuk menjejaki kenderaan dalam kes kehilangan kunci atau kejadian ancaman. Sistem ini telah dibangunkan menggunakan teknologi terkini yang memudahkan kehidupan rakyat biasa.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Background of Study

The simplest way of defining a security system is, where it is a technique or methods that are set to provide protection from hostile forces with the help of a machine of interworking components and devices [1]. Theft in today's world is a common occurrence, particularly the theft of automobiles or vehicles, and hence we require a security system to protect them. The common security features that are available in a car is a key and an alarm system. A car alarm is an electronic device installed in a vehicle which identifies the attempt of theft of the vehicle itself, its component, or even both. For example, a car security system emits loud volume of sound if the necessary triggers are set [1].

Another method used to avoid car theft is by using the immobilizer. The electric immobiliser system was invented by St. George Evans and Edward Birkenbuel and patented in 1919. They invented a 3x3 grid of double-contact switches on a panel which is mounted inside the car. So when the ignition switch is activated, current from the battery goes to the spark plugs allowing the engine to start or immobilizing the vehicle and releasing its horn. Immobilisers have been mandatory in all new cars sold in Germany since

1 January 1998, in the United Kingdom since 1 October 1998, in Finland since 1998, in Australia since 2001 and in Canada since 2007 [2]. Earlier models used a static code in the ignition key which was recognised by an RFID loop around the lock barrel and checked against the vehicle's engine control unit (ECU) for a match. If the code is unrecognised, the ECU will not allow fuel to flow and ignition to take place. Later models use rolling codes or advanced cryptography to defeat copying of the code from the key or ECU [2].

Due to technology advancement it brings helpful effects as well as harmful effects. These days, there are many thieves that are aware of these technologies and are using that knowledge against the existing security system to break the system. As a result, security needs to be improved to protect the vehicle.

1.2 Problem Statement

Vehicle theft around Malaysia has become rampant. Based on Vehicle Theft Reduction Council of Malaysia (VTREC), in 2018, 12 562 vehicles were reported stolen in Malaysia [3]. Even though, there are many measures that has been taken, still vehicle theft occurs due to some system failures.

A report by The Straits Times that appeared earlier this year shows the ease with which thieves can break into a vehicle with a wireless key by using a device that costs only RM150. This device can unlock the car and start its engine by hacking its radio frequency identification (RFID) information [4].

The number of layers of a security system should be more than just verifying a key of a vehicle. The current security system does not recognize whether the car has been used or taken by an unauthorised person nor does it to inform the authorised user about such attempt.

1.3 Objectives

The aim of this project is to develop a prototype of anti-theft system for vehicle which can reduce vehicle theft and locate the vehicle.

The objectives of the project are the following:

1. To provide a biometric authentication to start the vehicle.
2. To develop a system that is capable of tracking the vehicle.
3. To develop a system which can send real-time information to the user

1.4 Significance of Study

By using this system, the security features of a vehicle will be improved with help of the biometric fingerprint scanner and a GSM receiver and Transmitter. This system will give a peace of mind to the vehicle owners about their vehicle even if their car key is stolen or lost because the thief will be requiring the owner's fingerprint.

This system also notifies the owner in case there is a theft attempt on their vehicle by sending an SMS to the registered phone number. The owner can immediately stop the vehicle's engine. This system has an additional feature which is to track the vehicle at any point of time from anywhere by just sending a text message "track" to the system.

1.5 Scope of Study

The scope of this project is to improve the security features of a vehicle. This system is mainly focussing on improving security for the car. The system is to be implemented on any type of car. This system will utilize the mobile communication system (GSM) to communicate with the user's mobile device. This system is focused to operate within Malaysia. This system concentrates on using biometrics recognition techniques to enhance the security features. This aspect will make the system unique and safe. It records fingerprints of the user which are used for verification purposes before starting the car. The developed system is very user-friendly which can be understood and operate by simple guidelines.

CHAPTER 2

LITERATURE REVIEW

This chapter will elaborate on the works by a few analysts in the relative field and also, on the key terms utilized as a part of this exploration work which are explained in the reference material (articles, journal papers, books, web and so on). This literature review will include the review of the similar systems which has been done previously. This review was done to ensure the technology being chosen is suitable for the project.

2.1 GSM-based Security for Vehicle

Sankirna Joge (2018), studied on GSM-based security for vehicle. The main aim of this project is about detecting vehicle theft and informs the owner about the theft of their vehicle. This project uses three main components which are a Microcontroller Unit- AT MEGA 16, a GSM modem, and an IR sensor. The main function of the system is to start the engine when it receives SMS saying “UNLOCK” and when it receives SMS saying “LOCK” it will demobilize the car and activates the security system. Where the security system continuously waits for the IR sensor detects any motion or if any switches are pressed, if any of the above conditions are meet it turn on the buzzer and sends an alert SMS to the user mobile. When the user sends SMS “Track”, it will send LAC (Location

Area Code) and CID (Cell Identification) no. on user's mobile. User can get latitude and longitude position from these codes by using gsm tracker website. The limitation/disadvantage of this project is the user can only start the car's engine by sending SMS to the system and this might cost more for the user [5].

2.2 Anti-Vehicle Theft System using GSM and GPS with Image Acquisition

Samuel Farayola Kolawole and Alexander Zakari (2017) studied on designing of an Anti-Vehicle Theft System using GSM and GPS with Image Acquisition. The main objective of this project is to capture the image of the driver's face if the system detects that the vehicle has been stolen. This project is done using few main components which are ATmega 2560 Microcontroller, GSM and GPS module and a camera. The system activates when the vehicle is started, the security system is powered from the vehicles battery. A five minutes time frame is given for the driver to press a secret button. When this button is pressed after the start of the engine, the alarm mode is disabled but the coordinates of the vehicle would still be transmitted at every fifty minutes to a registered phone number via SMS or E-mail to give a level of security in case the vehicle was snatched after the engine was on by the owner. If the secret button is not pressed after five minutes, the vehicle is considered stolen and immediately enters security bridge mode. At this point, the camera which is place hidden in the vehicle captures the driver facial image. Simultaneously, the GPS module picks the coordinate (location) of the vehicle. The image and coordinate are stored in the microcontroller memory before being transmitted to a

remote location. Then microcontroller sends them captured image and the coordinate acquired to an E-mail or to a server where it can be retrieved. The limitations of this project are this system only captures the drivers image if the vehicle is turned on and the sending an image to a server or an email will require a strong network connection else the image will be transmitted slow than expected [6].

2.3 Anti-Car Theft System using Android Phone

Jake M. Laguador, Moulle M. Chung, Frina Joy D. Dagon, Julie Ann M. Guevarra, Rommel J. Pureza, Jeffrey D. Sanchez, and Dan Kenneth I. Sta. Iglesia (2013) did a study on developing an Anti-Car Theft System using Android Phone. The aim of this project to detect intrusion if the driver's door is forcefully opened. This project makes use of the android platform as the user interface and uses GPS Module Shield, Gizduino Atmega328, GSM Module, Internet Protocol Camera, Android Phone and Globe Tattoo Wimax to develop this project. When this system is initialized it continuously checks for the signal form the door lock and if it receives any, the microcontroller will be triggered and sends an alert SMS saying "Intruder Alert" with the coordinates (location). As for android application, when it is launched on the phone by the user their can check for their vehicles location by clicking the navigation button and if any SMS received the phone will be alarmed. When the phone is alarmed the user can open the IP camera to view. The limitations of this project is the user has to manually connect and disconnect the power

supply to activate and disactivated the security system and the application only detects for SMS alert only if the application is running in the foreground [7].

2.4 Improving the security system of a vehicle using Raspberry Pi

Samir Rana, Ritu Mewari and Lata Nautiyal did a study on improving the security system of a vehicle. The main aim of this project was to design a security system that is capable of avoiding the vehicle from being driven away by an unauthorized user. In this project they have used multiple major hardware components to complete this system such as Raspberry Pi B+ model, a jaw gripper, wi-fi dongle, integrated circuit (L239D) and other components for device. And for the software section they have used android development kit, python and java programming language to develop the android application. This system continuously detects if there is anyone tries to steal or tamper the car once this system is activated. If there is anything detected this system will send a warning signal to the database using internet connection. And on the other side, the user will have the android application installed in their phone, where this application will notify them about the theft attempt. After receiving this warning, the user has the privilege of locking the brake and accelerator by selecting the options in the application to lock with the help of the jaw gripper installed in the vehicle. There are few limitations of this project which are, this system does not provide any information related to the location of the vehicle if the vehicle has been stolen and this system does not provide any internet connection for the system to access and send the warning message [8].

2.5 Vehicle Anti-Theft System Using Fingerprint Recognition Technique

Z.Brijet, B.Santhosh Kumar and N.Bharathi studied on developing a Vehicle Anti-Theft System Using Fingerprint Recognition Technique. The main objective of this project is to develop a system which can verify the authorize user by uniquely identifying the user's fingerprint to switch on the ignition on. As for the components used in this project are Arduino as the main processing unit, a fingerprint sensor, a relay and a voltage regulator. The system is initialized when the ignition switch is turn on where the Arduino is powered by the power supply from the ignition switch. Once the system is initialized, this system waits for authorized user's fingerprint to be read. If the system is feed with the matched fingerprint, it activates the relay that is connected to the ignition system of the car which turns on the engine of the car or else it will not allow the ignition to take place. There are few limitations of that can be stated which are, this system does not provide any information if anyone tries to steal the car and they did only study on the ignition of the system but didn't on shutting down process [9].

CHAPTER 3

METHODOLOGY

This project is to design a prototype of the vehicle anti-theft system by using GSM and GPS and fingerprint sensor, which will improve the security features that secures the vehicle and benefits the owner from vehicle thefts.

In this methodology chapter there are few important sections that will be described and explained in detail. This chapter helps to understand the technical section of this project's design and development process. The main contents of this methodology chapter are the system overview, system design, system architecture, hardware and software requirements.

3.1 System overview

This anti-theft system for vehicle using GSM (Global System for Mobile Communications) & GPS (Global Positioning System) with fingerprint verification, is to improve the existing general security system which exist in most vehicles.

This project's system is fully controlled by a microcontroller, Arduino Mega 2560. This system is proposed to use biometric fingerprint scanner to enhance this security

system. It also has few other features such as informing the authorised user about vehicle's location on demand, sending and receiving SMS (Short Message Services) as it is the main channel of communication between the user and the system. Retrieving the current location of the vehicle, when the system detects theft attempts by using the GPS (Global Positioning System) module that communicates directly with the microcontroller.

This system also provides an additional feature that allow authorized users to a switch on the ignition even when the fingerprint does not match by entering a Pin code, to provide some flexibility. This function was made possible by adding an external keypad to the system for the unauthorized user to authenticate themselves to the system.

3.2 System design

This system was developed using various technologies, which works together in order to run the anti-theft system for a vehicle smoothly. The main component of this system is the processing unit and after some background studies was carried out the most appropriate component for this processing is a microcontroller. The most suitable microcontroller for this project is the Arduino Mega 2560 R3. Out of all the available types of Arduino microcontrollers, this specific model was the only one which had the sufficient capabilities that meets the project's requirement. It has enough flash memory to store the codes and has the necessary processing power to execute the program's codes. It can be programmed by using its IDE (Integrated Development Environment) software which is

available as an open source software. By using this software this microcontroller can communicate directly with the computer for coding and uploading any programs on it.

Other than that, this system also requires other hardware components such as the GSM and GPS module, control relays, a fingerprint scanner, a 4x5 keypad a 16x2 LCD display, a 12V battery pack and other electronic components to complete this project. In this project, all these modules will be connected directly to the Arduino. Each of this component has its own important role to complete the system.

In this project, GSM SIM808 module us being used where this module has more than one function. It has three main function in it. It has the functionality of GSM communication, GPS for detecting the current location of the device and also a Bluetooth transmitter and receiver. For this project only the GSM communication and the GPS function will be used. GSM is being used to establish the connection network between the system and the owner/user. The GPS is used to retrieve the current location of the vehicle for the system to process.

A relay is used in this project to act as a switch. As the Arduino microcontroller is only capable of receiving input signal as high as five volts (5V), this situation requires a relay to detect the higher voltage flow and send a 5 volts input signal to Arduino for

processing and vice versa. This system requires two types of relay which are the 5 volts relay and the 12 volts relay. The 12 volts relay is used to detect the higher voltage flow which will be 12 volts and sends 5 volts signal to the Arduino. Likewise, the 5 volts relay is used to detect output signal of 5 volts send from the Arduino and switches on the higher voltage power supply.

The Adafruit fingerprint scanner is used to scan the fingerprints which will be used for the system. This module also comes with a flash memory which is capable of storing up to 120 images. Hence, it will reduce the memory consumption of the Arduino microcontroller. This device will be used for fingerprint enrolment and verification.

In order for the user to key in any form of input this system will require some sort of input device. For this case, this project was included with a 4x5 keypad which contains numeric values 0 to 9 and other keys such as 'F1', 'F2', '#', '*', '↑', '↓', '←', '→', 'Esc', and 'Ent'. This keypad is mainly included to authenticate unauthorized users to provide the flexibility by entering specific PIN code and also, to edit data in the menu option.

To make the system more user friendly and understandable, this system was added with a viewing purpose component. A 16x2 LCD was included to this system for the user

to know the steps to do and the processes that the system is processing. By having this display in the system, the user will be able to use the system more easily.

As a back plan a 12 volts battery pack was attached to the system in case the main power supply fails. Since the main power source connected to the system to power it up, these batteries will not be used and only be as a standby power supply. There are two relays which will always checks for the power flow. If the main power source fails or disconnected these relays will automatically switches the power source from main supply to these batteries.

Figure 3.2.1: Use Case Diagram: Vehicle Anti-theft System

There many functions that can be performed by the user on this anti-theft system based on the use case diagram in figure 3.2.1. The first function that this system provides is to start the ignition system of the vehicle. In order to start the ignition(engine), this system provides functionality to authenticate the user by verifying their fingerprints, else by entering the pin code.

When the system is initialized for the first time, the pin code and registered phone number will be initialized with a default value. This system provides a menu option for the user/owner to enter/change the default values. There are three functions in the menu option for the user to modify values in the system. The first option of the menu is the option for the user to change the phone number which will be registered to the system to send the SMS to. The second option will be for the user to change the Pin code of the system to their preferred Pin code to authenticate themselves with the system, and the last option will be for the owner/user to enroll new fingerprint to the system in order to start the ignition system.

Firstly, all required hardware needs to be set up for this system. Then the program for this system had to be coded in the Arduino IDE and upload the program on the Arduino Mega 2560 R3. This system is designed with a sequence of activities which determines the workflow of this system. As the first step, the system will be initialized by connecting the device to the power supply. Since this system is connected to the main battery of the vehicle

(in this prototype the 12V power supply used), it will be always connected to a power supply. Once the system is initialized, it will automatically go into standby mode. In the standby mode, the system will always recursively check for three different input triggers.

The first input the system checks for is the input from the keypad. The system waits for the user to enter '*' three times in a row to enter into the menu option. Once the system had detected this input, it verifies the user by requesting the user to enter the pin code of the system. The system gives up to three trials to enter the correct pin code. For each incorrect pin code entry, a long beep will be produced by a buzzer. If the user fails to enter the correct password for three continuous trial, the system will exit that option and goes back to check other inputs triggers. But if the user entered the correct pin code, the system will enter into the menu option with a short beep. In the menu option there are three different functions for the user to modify the data of the system. The functions in the menu options are to change the phone number, to change the pin code of the system and enroll a new fingerprint into the system. Once the system enters the menu option the user has to enter the option number to enter the specific option. If the user entered 'Esc' as the input the system will exit the menu option, else if the user enter '1', the system start excepting numeric values as the phone number that is going to be registered to the system, then followed by 'Ent' to save the new phone number. The second option in this menu is to change the pin code when the user entered '2'. This option is to change the default pin code which was initialized when the system is powered. The third option is to enroll a new fingerprint to the system by entering '3' from the menu option.

The second input trigger for the system is from vehicles ignition system which will be triggered by the relay switches. If there is no signal or input were received by the Arduino microcontroller, the system goes back to the standby mode and continues the checking circle again for other triggers. Likewise, if the system receives any input signal, it will turn on the fingerprint scanner for the next process. In this process, the fingerprint scanner will start to scan for fingerprint and sends the information to the system for processing and identification. Once the fingerprint is identified and it matches the registered user, a small beep will be produced and then ignition system will be supplied with power to turn on the ignition system (DC Motor is used to represent the ignition system). The system will allow the user to attempt three times in a row. Each time the system didn't find a match a long beep sound will be sound. But if the fingerprint does not have match within the given attempts of trial, the system will identify this as a theft attempt and goes into theft mode. In this mode, the system stops all processes and it will turn on the GPS to retrieve the current location of the vehicle which it has attached to. After collecting the location information, the system will send an alert message to authorized user's phone number via SMS using the GSM communication. At this point of time, the system only waits for the reply message via SMS for further action. There are two types of valid response that the authorized user can reply to the system whether it's an access allow command ("allow") or access decline command. Once the system receives commands it verifies and takes action based on it. If the received command is other than the allowed command, the system stops the ignition process and goes back to standby mode. In the same way, if the command was to allow the unauthorized user in vehicle to turn on the

ignition the system will allow the user with a condition, where they have to enter a PIN code from the authorized owner. If the entered PIN code matches, the Red LED (ignition) will be turned on. The system will allow maximum of three trial, before it block the ignition process and goes to the standby mode.

As the system checks for any unread received SMS, the last input trigger of the system will be identified. This trigger will gather the location of the vehicle and send it to the owner of the vehicle. If there is any unread SMS it will check for few conditions, firstly weather the sender's phone number matches the registered phone number. If the phone number matches, then it checks for the keyword matches to the default keyword which is "track". Only when these two conditions have been met, the system will find for the vehicles current location and sends the location to the registered phone number. Figure 3.2.2 shows the flowchart of vehicle anti-theft system. It shows the flow of activities in a briefer manner which will be easier to understand.

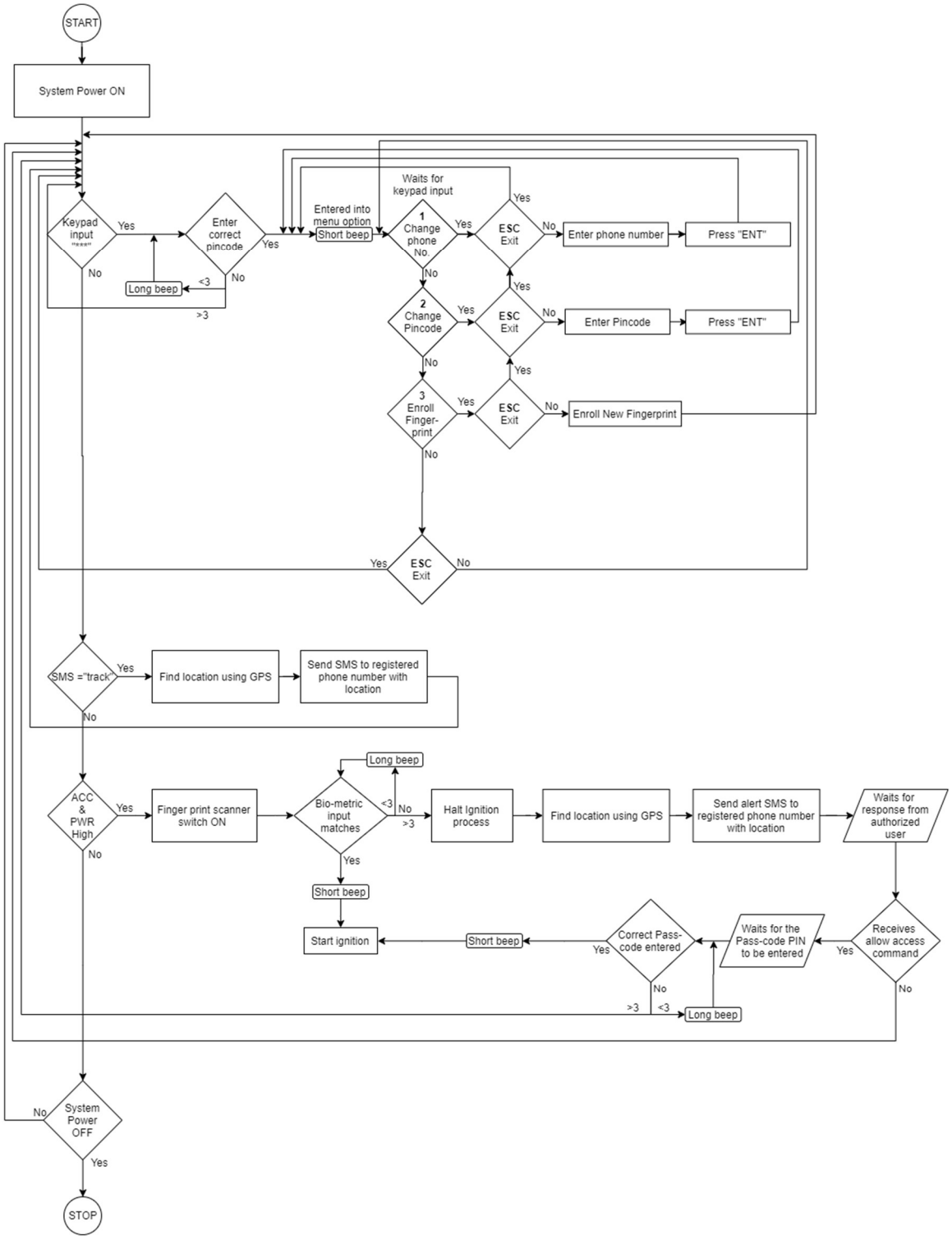

Figure 3.2.2: Flowchart: vehicle anti-theft system

3.3 System Architecture

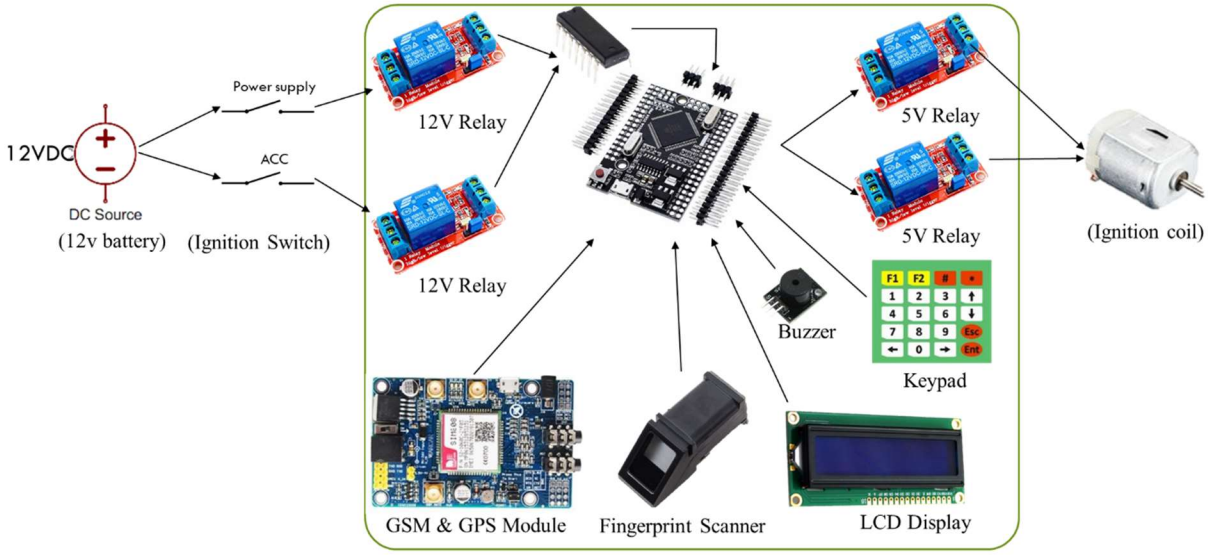

Figure 3.3.1: Vehicle Anti-theft System- Architecture

3.3.1 Required Hardware Components

- Arduino Mega
- GSM SIM808 module
- Fingerprint Scanner
- 5V Relay Module (x2)
- 12V Relay Module (x2)
- Keypad
- 16x2 Liquid Crustal LCD Display
- CD4081BE IC

- LM 7805 Voltage Regulator
- L7812CV Voltage Regulator
- AMS1117 Voltage Regulator

3.3.1.1 Arduino

Arduino is an open-source platform used to create projects for electronics. Arduino consists of both a programmable physical circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) running on your desktop, used to write and upload computer code to the physical board [10].

The Arduino platform has become quite popular with individuals just beginning with, and for excellent reason, electronics. Unlike most other programmable circuit boards, the Arduino doesn't need a different piece of hardware (called a programmer) to load new code to the board where it can just use a USB cable. The Arduino IDE also uses a simplified C++ version to make it easier to learn to program. Finally, Arduino provides a standard form factor that breaks down the microcontroller's functions into a more accessible package [10].

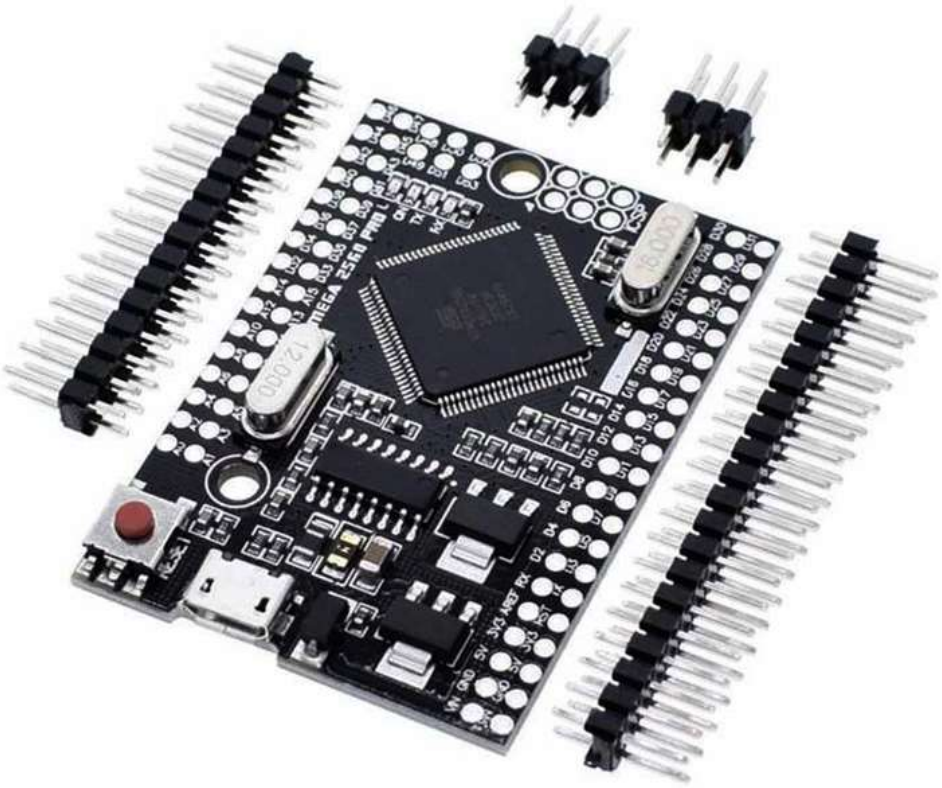

Figure 3.3.2: Arduino Mega2560

The Arduino Mega is like the UNO's advanced version. This version of Arduino has a lot of digital input and output pins and fourteen of them can be used as PWM outputs, sixteen analog inputs, a USB connection, a power jack, and a reset button. It contains everything needed to support the microcontroller. It can be powered simply by connecting it to a computer with a USB cable or connect with a DC adapter or battery ranging between 9V to 12V to get it started. The big number of pins makes this board very convenient for projects requiring a lot of digital inputs or outputs.

3.3.1.2 GSM SIM808 module

Figure 3.3.3: GSM SIM808 module

SIM808 module is a GSM and GPS two-in-one function module. This module is a complete Quad-Band GSM/GPRS module which combines GPS technology for satellite navigation.

The compact design which integrated GPRS and GPS in a SMT package will significantly save both time and costs for customers to develop GPS enabled applications. Featuring an industry-standard interface and GPS function, it allows variable assets to be tracked seamlessly at any location and anytime with signal coverage. The latest firmware

support Bluetooth function. AT commands for Bluetooth functions can be used in these latest versions.

It features an ultra-low power consumption in sleep mode and integrated with charging circuit for Li-Ion batteries, that make it get a super long standby time and convenient for projects that use rechargeable Li-Ion battery. It has high GPS receive sensitivity with 22 tracking and 66 acquisition receiver channels. Besides, it also supports A-GPS that available for indoor localization

.

There are also other features that can be stated such as Quad-band 850/ 900/ 1800/ 1900 MHz, GPRS multi-slot class12 connectivity which has maximum of 85.6kbps of download and upload capacity, a GPRS mobile station class B, supports charging control for Li-Ion battery, supports Real Time Clock, its power supply voltage ranges between 3.4V to 4.4V, an integrated GPS/CNSS and supports A-GPS ,supports 3.0V to 5.0V logic level, low power consumption as low as 1mA in sleep mode, supports GPS NMEA protocol and standard SIM Card [11].

3.3.1.3 Fingerprint Scanner

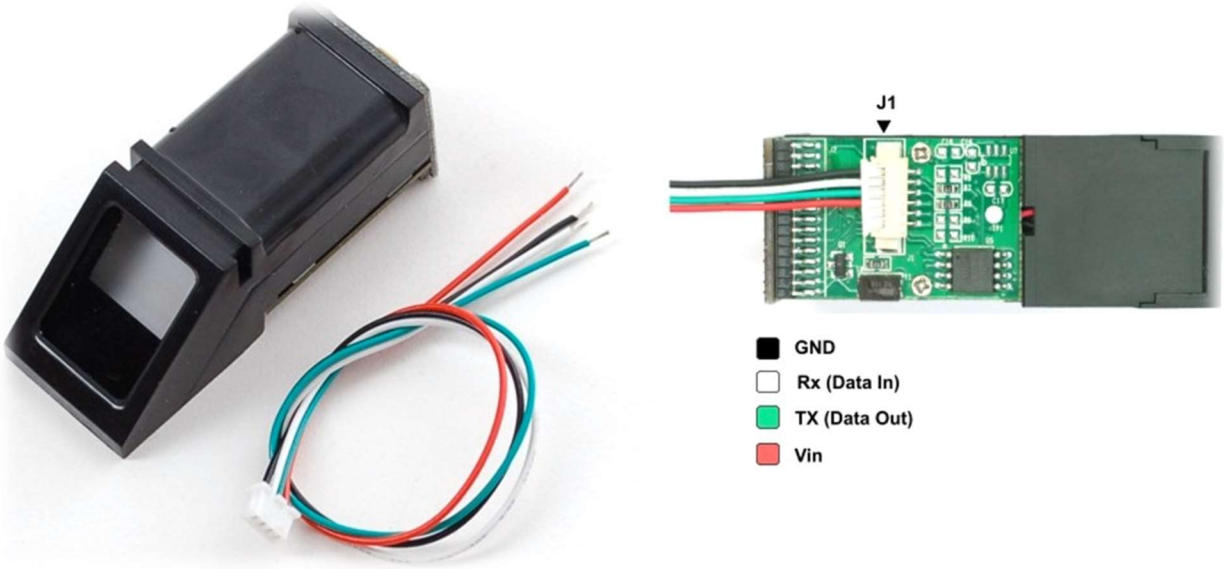

Figure 3.3.4: Adafruit Fingerprint scanner

The Fingerprint is one of the safest ways to detect and recognize the Authorized Person. We understand that fingerprint is distinctive, even if identical twins do not have identical DNA. By using this we can make pretty sure about security needs. Adding fingerprint verification in security projects. All of this can be used in one optical fingerprint sensor-scanner (R305), it makes fingerprint detection and verification very simple.

This optical fingerprint reader devices utilizes high powered DSP chip AS601 form Synochip, which is capable to do the image rendering, calculation, finding and searches for images. This also provides TTL serial out so we can connect to any microcontroller or

system to retrieve information. The DSP processor has a built-in FLASH memory which can store up to 120 fingerprints. Adafruit provides an open source fingerprint library where we can this sensor to Arduino as well. The fingerprint identification process has two steps that is enrolling fingerprint and matching fingerprint. These two steps make the system to authenticate right fingerprint.

3.3.1.4 Relay

Figure 3.3.5: 12V Relay

Figure 3.3.6: 5V Relay

A Relay is electrically operated switches, which allow low power circuits to switch a relatively high voltage or current on/off. In generally, the point of a relay is to use a small amount of power in the electromagnet to move an armature capable of switching much larger power. For a relay to operate a suitable pull in and holding current should be passed through its coil. Relay coils are designed to operate from a particular voltage often its 5V or 12V. The function of relay driver circuit is to provide the necessary current energize the relay coil, when the input is HIGH is written on the PORT PIN thus turning on the relay.

The relay is turn off by writing input back to LOW on the PORT PIN. In this system four relays are used for device control, two of 12V coil and two 5V coil relays.

3.3.1.5 4x5 Keypad Module

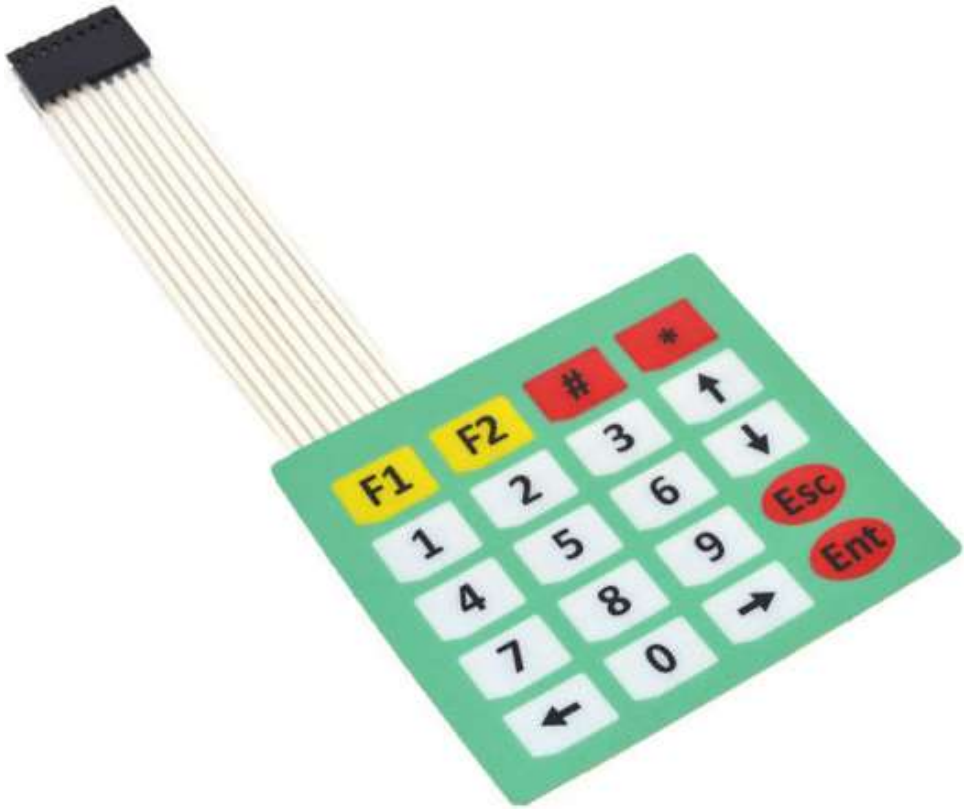

Figure 3.3.7: 4x5 Keypad Module

A Keypad Matrix is a selection of switches arranged in a grid. One side of each switch is connected with horizontal wires known as rows and one side is connected with vertical wires known as columns. By putting a signal on one side which is the rows and

reading the other side the columns, you can determine which key is pressed down. Keypads are handled by the Keypad module and this uses the Keypad.h library.

Simply supply two arrays, one of wires connected to columns, one of wires connected to rows. If a third argument (a call-back function) is supplied, watches will be set up, and the call back will be called automatically as soon as a button is pressed. If it isn't, it's up to the user to use keypad.read() to find out what key is pressed. -1 will be returned if no key is pressed. [12]

3.3.1.6 16x2 Liquid Crystal Display LCD

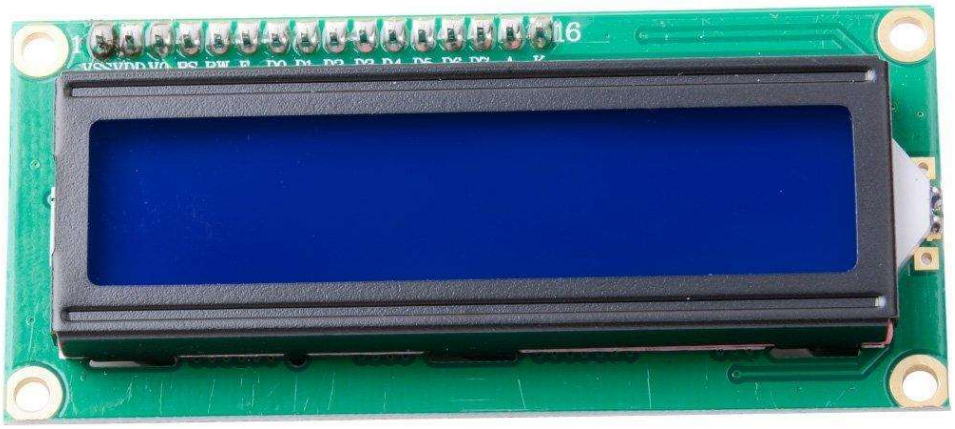

Figure 3.3.8: 16x2 Liquid Crystal Display

LCD, the electronic display unit that functions through the application of different electric voltage to a liquid crystal layer, induces changes to its optical properties. LCDs are

commonly used for portable electronics, digital and camcorder viewfinder, video screening systems, electronic billboards, computers and flat panel television. LCDs are also commonly used as screens [13]. This LCD is used in this system to display the results or instructions that needed to be shown to the user.

3.3.1.7 CD4081BE IC

Figure 3.3.9: CD4081BE IC

CD4073B, CD4081B and CD4082B AND gates, provide the system designer with direct implementation of the AND function and supplement the existing family of CMOS gates. The CD4073B, CD4081B, and CD4082B types are supplied in 14-lead hermetic dual-in-line ceramic packages (F3A suffix), 14-lead dual-in-line plastic packages (E suffix), 14-lead small-outline packages (M, MT, M96, and NSR suffixes), and 14-lead thin shrink small-outline packages (PW and PWR suffixes) [14].

3.3.1.8 LM 7805 Voltage Regulator

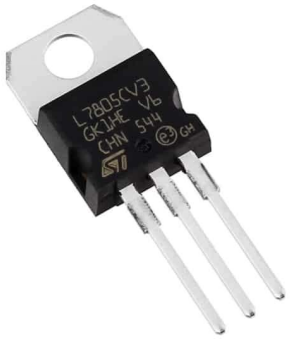

Figure 3.3.10: LM 7805 Voltage Regulator

For several electronic devices, a regulated power supply is very important because the semiconductor materials used in them have a fixed current and voltage rate. These types of devices maybe damaged by any deviation occurs from the fixed rate. This LM7805 voltage regulator controls the output voltage from the input range of 5V till 20V to a stable 5V output. The reason of using this regulator is, some components in this system uses 5V input to engage.

3.3.1.9 L7812CV Voltage Regulator

Figure 3.3.11: L7812CV Voltage Regulator

The component in the figure 3.3.11 is the voltage regulator (L7812CV) which regulates the input voltage from 12V until 20V to give the stable output voltage of 12V. In this system the regulator will provide a stable voltage to the other components from the main power supply and reduce the draining of the power from the Arduino.

3.3.1.10 AMS1117 Voltage Regulator

Figure 3.3.12: AMS1117 Voltage Regulator

The component in the figure 3.3.12 is the voltage regulator (AMS1117) which regulates the input voltage from 12V until 20V to give the stable output voltage of 3.3V. In this system the regulator is very much needed because the Arduino cannot supply current to many loads or component at the same time, hence this regulator will reduce the draining of the power from the Arduino.

3.3.2 Required Software

- Arduino IDE
- C/C++ Programming Language

3.3.2.1 Arduino IDE

The Arduino Integrated Development Environment or Arduino Software (IDE) usually includes a text editor for composing software or lines of codes, a message region, a text pad, a popular function button toolbar, and a set of menus. It binds to and communicates with Arduino and Genuino hardware to upload programs.

In Arduino Software (IDE) programs that are written are called sketches. To write these sketches it uses the text editor provides in the software itself and those are saved in the file extension. ino. This editor has also some common features for cutting and pasting

and for searching and replacing text. The Arduino Software (IDE) console shows text output, including full error messages and other data. The window's upper right corner shows the panel and serial port installed. The buttons on the toolbar enable you to check and upload programs, generate, display and save sketches, and enable the serial monitor. [15]

The Arduino Software (IDE) utilizes a sketchbook idea: a normal location for storing (or sketching) your programs. You can open the drawings in your sketchbook from the menu File > Sketchbook or from the toolbar Open button. It will automatically generate a folder for your sketchbook the first time you run the Arduino software. The location of the sketchbook can be viewed or changed later in time for user's needs. [15]

3.3.2.2 C/C++ Programming Language

C is a programming language for procedures. It was first created between 1969 and 1973 by Dennis Ritchie. It was primarily created to write operating system as a system programming language. C language's primary characteristics include low-level memory access, easy set of keywords and clean style, making C language appropriate for system programming such as operating system or compiler development. [16]

Many later languages directly or indirectly borrowed syntax / features from the C language. Syntaxes, like Java syntax, PHP, JavaScript and many other languages are predominantly C-based. C++ is almost a C language superset.

CHAPTER 4

IMPLEMENTATION AND RESULT

4.1 Overview

This Vehicle Anti-Theft System prototype is developed and integrated to provide additional security features for the user's vehicle. This prototype is fully designed based on the previous design and methodology chapter. This system is fully controlled and operated using Arduino microcontroller and the it is programmed using its Arduino IDE.

This system mainly consists of three main functions which are to start the ignition system with the fingerprint authentication, capable of tracking the vehicle's location by sending a SMS to the system and menu option for the user to modify some details in system. All these individual systems work together to make the vehicle more secure.

4.2 Arduino Setup

As the Arduino is a microcontroller, it has a fixed storage capacity in it so that there are no other additional storage disks needed to be attached such as a memory card. But

there is IDE needed to be installed in the computer which going to be used for programming the Arduino processor.

To install the IDE to computer these are the few steps to follow:

1. Search for the official Arduino software webpage to download the IDE software.
2. Download the software using this link "https://www.arduino.cc/download_handler.php".
3. Install the application file into the computer.
4. Start the Arduino IDE software for the first time.

The user interface of the Arduino IDE is shown in figure 4.2.2.

Figure 4.2.1: Software booting interface

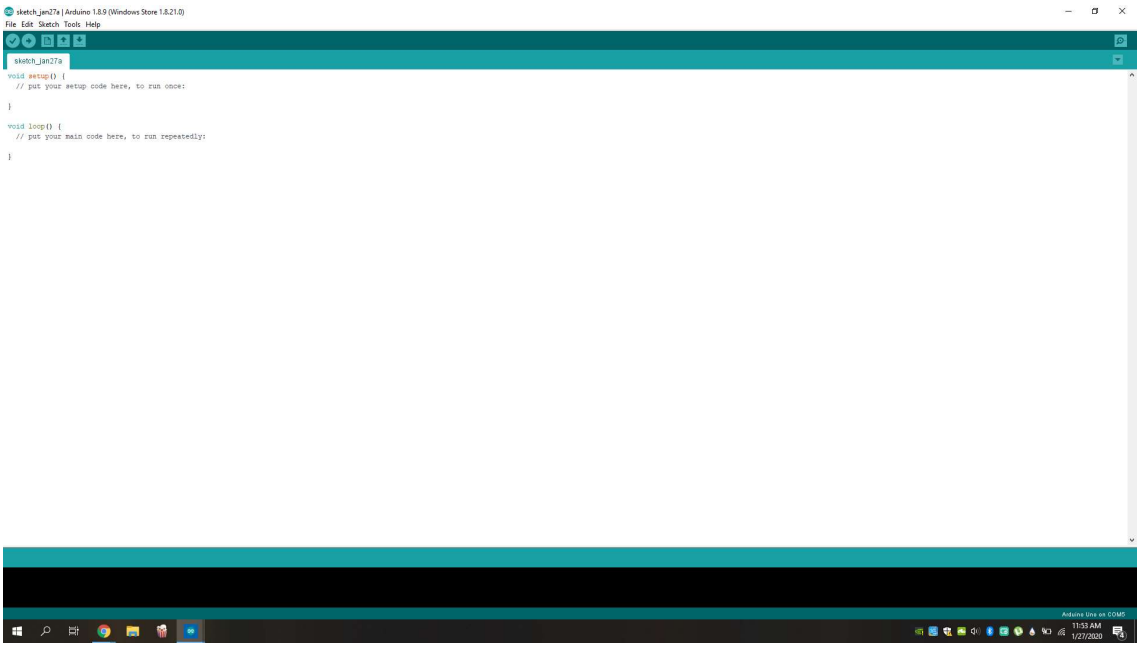

Figure 4.2.2: Arduino IDE user interface

4.3 Implementation

In this implementation step the develop process had been broken down into few sections of individual function development. This step was taken to make the development process more efficient and easier to troubleshoot if any problem arises. Those section are:

- Preparing the hardware components and its connectivity
- Power supply circuit (including back up)
- The fingerprint verification and fingerprint enrolment
- Location detector and sending its information using SIM808 Module
- Receiving physical input using Keypad
- Control relays as switches.

4.3.1 Preparing the hardware components and its connectivity

In order for any system to function it needs its hardware components to be set up beforehand to proceeding further. To set up this system all the components was gathered, and the connections was planned after researching all of its pin connectivity. Then, the circuit diagram was drawn to shows all the connectivity of these components. The figure 4.3.1.1 shows the circuit diagram of the Vehicle Anti-theft System.

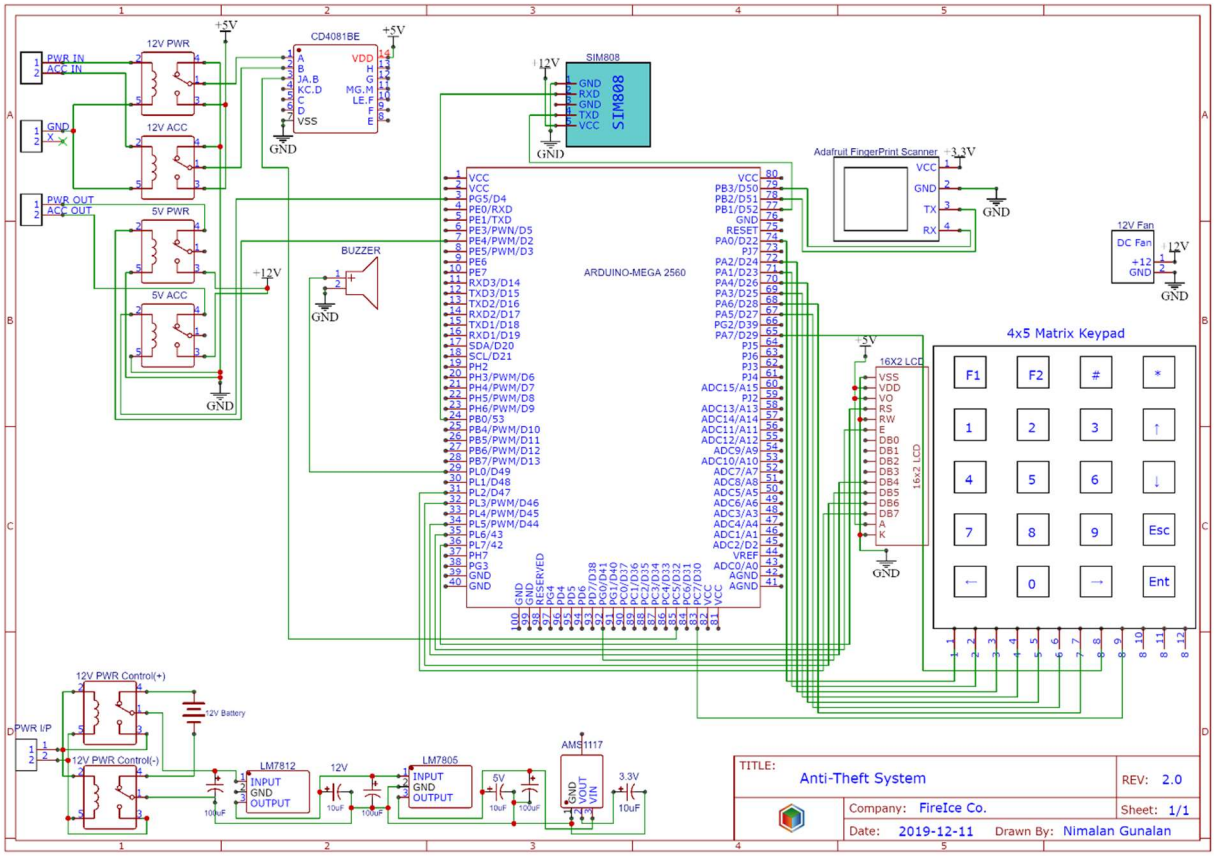

Figure 4.3.1.1: Circuit diagram of Vehicle Anti-theft System

Using this circuit diagram, a Printed Circuit Board (PCB) was developed. This PCB was developed to reduce to risk of wire failures, disconnection of circuits and this also gives a very neat finish to the system. This PCB contain all the individual components such as the relays, IC, power supply regulators, buzzer, resistors, input terminals, Arduino Mega and pin connectors for all the other modules which are the fingerprint scanner, keypad, LCD, Sim808 and back-up battery. The figure 4.3.1.2 shows the design of the PCB.

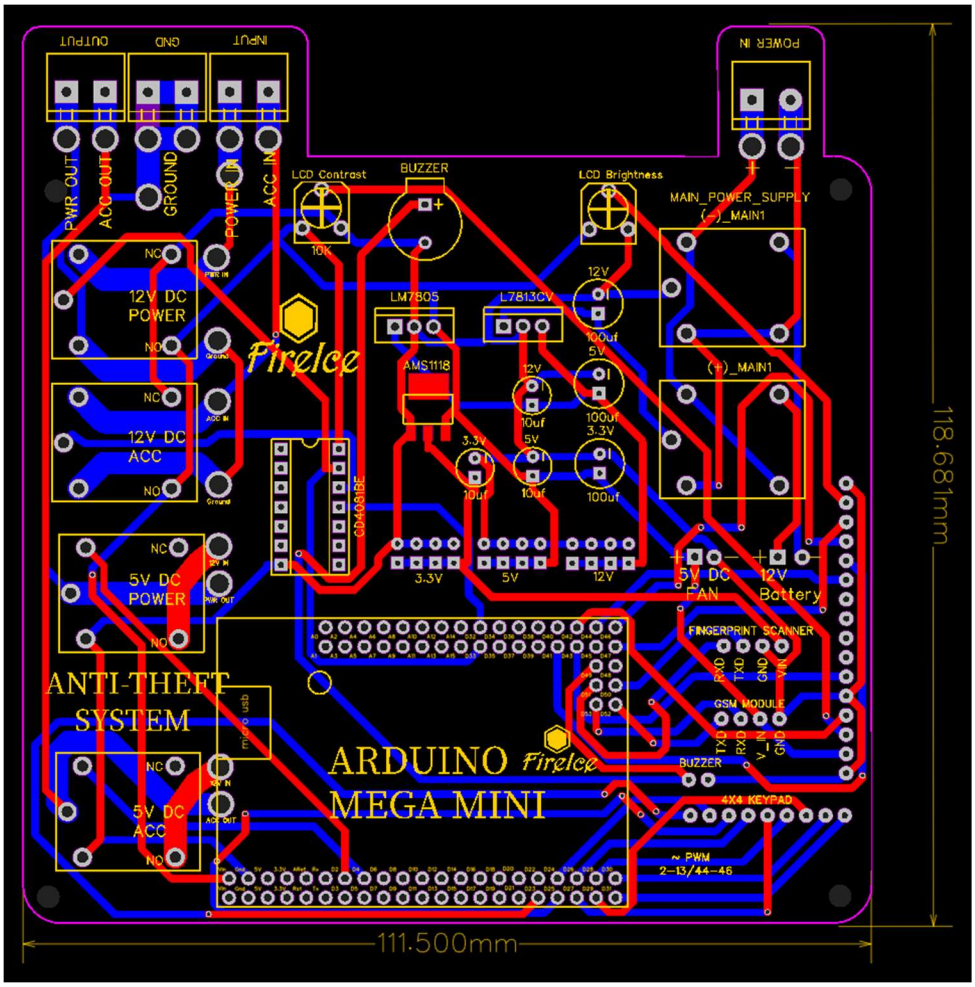

Figure 4.3.1.2: PCB Design of Vehicle Anti-theft System

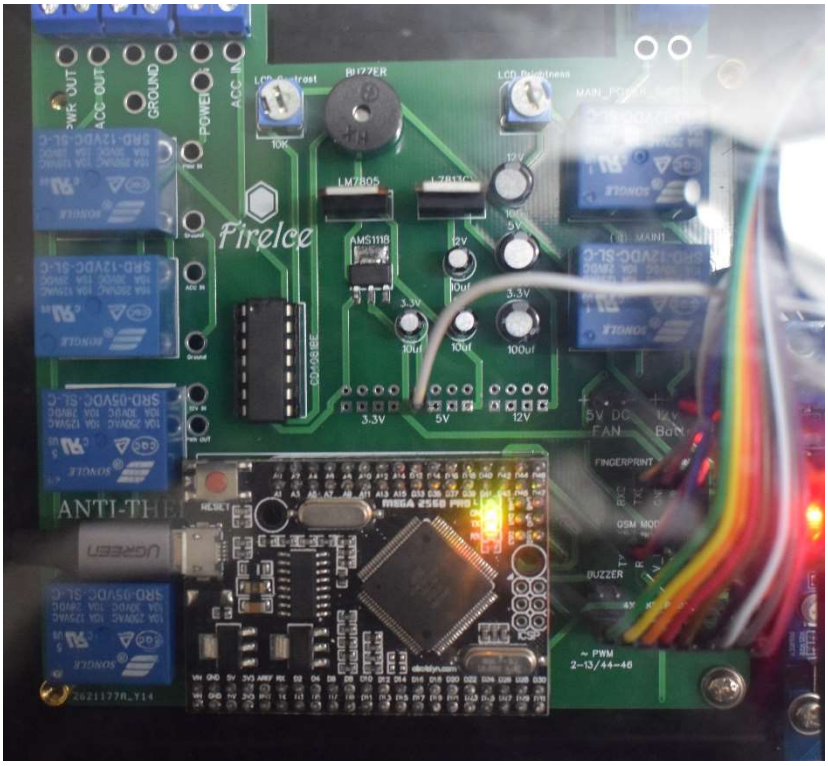

Figure 4.3.1.3: Actual PCB of Vehicle Anti-theft System

Figure 4.3.1.2 shows the actual fabricated printed circuit board with all the components soldered on it.

4.3.2 Power supply circuit (including back up)

To power up all the components for this system, a reliable and stable power source is very much need. This prototype system has its own dedicated power source to supply the necessary current/voltage to its components and modules. This prototype system is designed with this circuit is to avoid higher voltage/current flow into the other circuits since

the voltage/current from the vehicle may vary from the needed. In addition to that, it is also designed with a back-up power supply which is a 12V battery pack consist of 8 AA sized batteries connected in series. The back-up batteries only kick in when the main source is disconnected or fails. The figure below shows the circuit diagram of the power supply circuit and on the PCB.

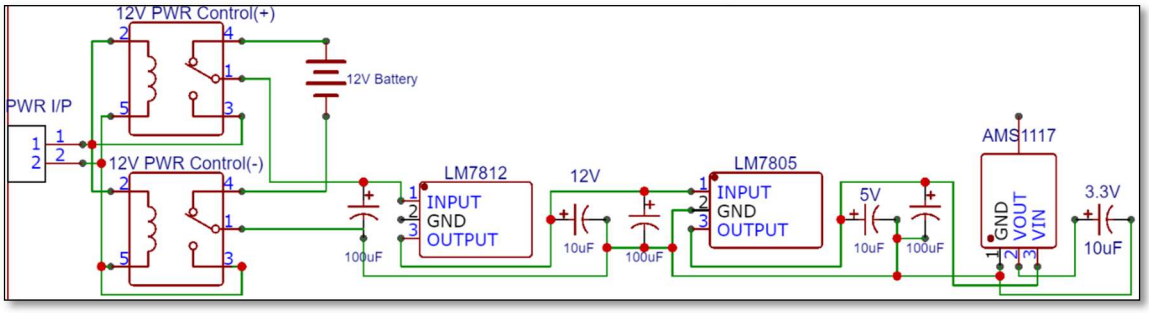

Figure 4.3.2.1: Circuit diagram of the power supply circuit

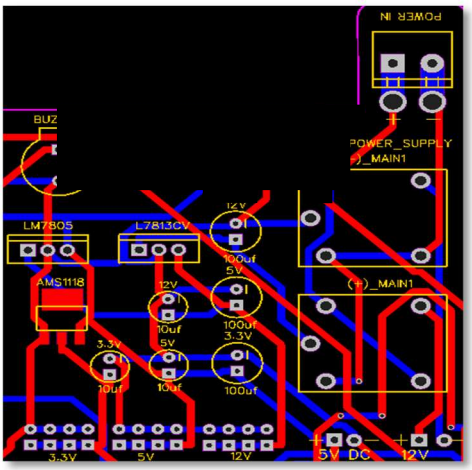

Figure 4.3.2.2: Power supply components on the PCB

This power supply circuit provides three different variation of voltage which are the 3.3V, 5V, 12V outputs. The relay in this circuit act as a switch to switch between the main power source and the batteries. The main is connected to both coils in the relays and to the Normally Open (NO) pin of the relay, the batteries are connected to the Normally Closed (NC) pin of the relay. Once the voltage is supplied from the main, the relay's coils turns ON and the current flows from the NO pins to the Common(C) pin. If the supply is been disconnected the relay automatically switches from NO to NC where the batteries will start supplying current. Voltage output from those two sources are measured in figure 4.3.2.3.

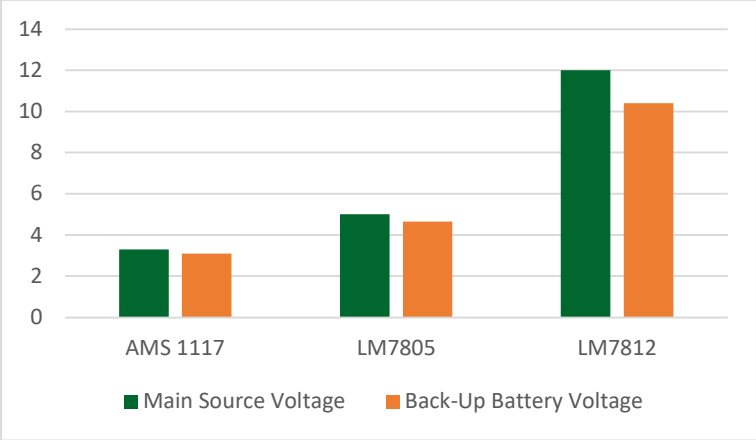

Figure 4.3.2.3: Regulated Voltage Supplies

4.3.3 The fingerprint verification and fingerprint enrolment

The fingerprint scanner is implemented in this system for two important reasons, first, to enrol new fingerprint for new user and secondly to verify the fingerprint to start the ignition system.

The first function to enrol a new fingerprint to the system has few steps for it to verify and store the fingerprint. This function falls under the menu option for the user. To access this function, the user has to enter the menu option in the system and select the third option, only then this function will be called. This function will request the user to enter the ID number which the fingerprint will be stored since this scanner is capable of storing multiple images. The system will wait for a valid finger to be placed on the scanner and store it on a temporary memory, then it asks the user to remove the finger from the scanner and place the same finger once again to get another image of the fingerprint. After comparing both images taken, if they are a match, the fingerprint will be stored to the system based on the ID given for it or not the system will reject the fingerprint.

The second function for this fingerprint scanner is to identify the given fingerprint with all the fingerprints stored in the system. This function is called when both the power and ignition give the output as high. Once the function is called, it waits for a valid finger on the scanner to get an image of the fingerprint and compare the image to get a result of found or not. The fingerprint matched is determined by the percentage of similarity between the given and the actual images. If the fingerprint does not match the system, it skips other steps with notify the user in LCD that "No FGPT Match", else the system turns ON the ignition system for 1500 milliseconds and returns to main loop.

The code for the first function (Fingerprint Enrolment) is as follows:

```
//Function 1
uint8_t getFingerprintEnroll()
{
 int p = -1;
 Serial.print("Waiting for valid finger to enroll as ID #");
 Serial.println(id);
 delay(2000);
 while (p != FINGERPRINT_OK)
 {
  p = finger.getImage();
  switch (p)
  {
   case FINGERPRINT_OK:
    Serial.println("Image taken");
    break;
   case FINGERPRINT_NOFINGER:
    Serial.print("");
    break;
   case FINGERPRINT_PACKETRECIEVEERR:
    Serial.println("Communication error");
    break;
   case FINGERPRINT_IMAGEFAIL:
    Serial.println("Imaging error");
    break;
   default:
    Serial.println("Unknown error");
    break;
  }
 }
 // OK success!

 p = finger.image2Tz(1);
 switch (p)
 {
  case FINGERPRINT_OK:
   Serial.println("Image converted");
   break;
  case FINGERPRINT_IMAGEMESS:
   Serial.println("Image too messy");
   return p;
  case FINGERPRINT_PACKETRECIEVEERR:
```

```
    Serial.println("Communication error");
    return p;
  case FINGERPRINT_FEATUREFAIL:
    Serial.println("Could not find fingerprint features");
    return p;
  case FINGERPRINT_INVALIDIMAGE:
    Serial.println("Could not find fingerprint features");
    return p;
  default:
    Serial.println("Unknown error");
    return p;
}

Serial.println("Remove finger");
delay(2000);
p = 0;
while (p != FINGERPRINT_NOFINGER)
{
  p = finger.getImage();
}
Serial.print("ID ");
Serial.println(id);
p = -1;
Serial.println("Place same finger again");
while (p != FINGERPRINT_OK)
{
  p = finger.getImage();
  switch (p)
  {
    case FINGERPRINT_OK:
      Serial.println("Image taken");
      break;
    case FINGERPRINT_NOFINGER:
      Serial.print("");
      break;
    case FINGERPRINT_PACKETRECIEVEERR:
      Serial.println("Communication error");
      break;
    case FINGERPRINT_IMAGEFAIL:
      Serial.println("Imaging error");
      break;
    default:
      Serial.println("Unknown error");
      break;
```

```
  }
}

// OK success!

p = finger.image2Tz(2);
switch (p)
{
  case FINGERPRINT_OK:
    Serial.println("Image converted");
    break;
  case FINGERPRINT_IMAGEMESS:
    Serial.println("Image too messy");
    return p;
  case FINGERPRINT_PACKETRECIEVEERR:
    Serial.println("Communication error");
    return p;
  case FINGERPRINT_FEATUREFAIL:
    Serial.println("Could not find fingerprint features");
    return p;
  case FINGERPRINT_INVALIDIMAGE:
    Serial.println("Could not find fingerprint features");
    return p;
  default:
    Serial.println("Unknown error");
    return p;
}
// OK converted!

Serial.print("Creating model for ID #");
Serial.println(id);

p = finger.createModel();
if (p == FINGERPRINT_OK)
{
  Serial.println("Prints matched!");
}
else if (p == FINGERPRINT_PACKETRECIEVEERR)
{
  Serial.println("Communication error");
  return p;
}
else if (p == FINGERPRINT_ENROLLMISMATCH)
{
```

```
    Serial.println("Fingerprints did not match");
    return p;
  }
  else
  {
    Serial.println("Unknown error");
    return p;
  }

  Serial.print("ID ");
  Serial.println(id);
  p = finger.storeModel(id);
  if (p == FINGERPRINT_OK)
  {
    Serial.println("Stored!");
  }
  else if (p == FINGERPRINT_PACKETRECIEVEERR)
  {
    Serial.println("Communication error");
    return p;
  }
  else if (p == FINGERPRINT_BADLOCATION)
  {
    Serial.println("Could not store in that location");
    return p;
  }
  else if (p == FINGERPRINT_FLASHERR)
  {
    Serial.println("Error writing to flash");
    return p;
  }
  else
  {
    Serial.println("Unknown error");
    return p;
  }
  //Serial.println(p);
  int r=321;
  return r;
}
```

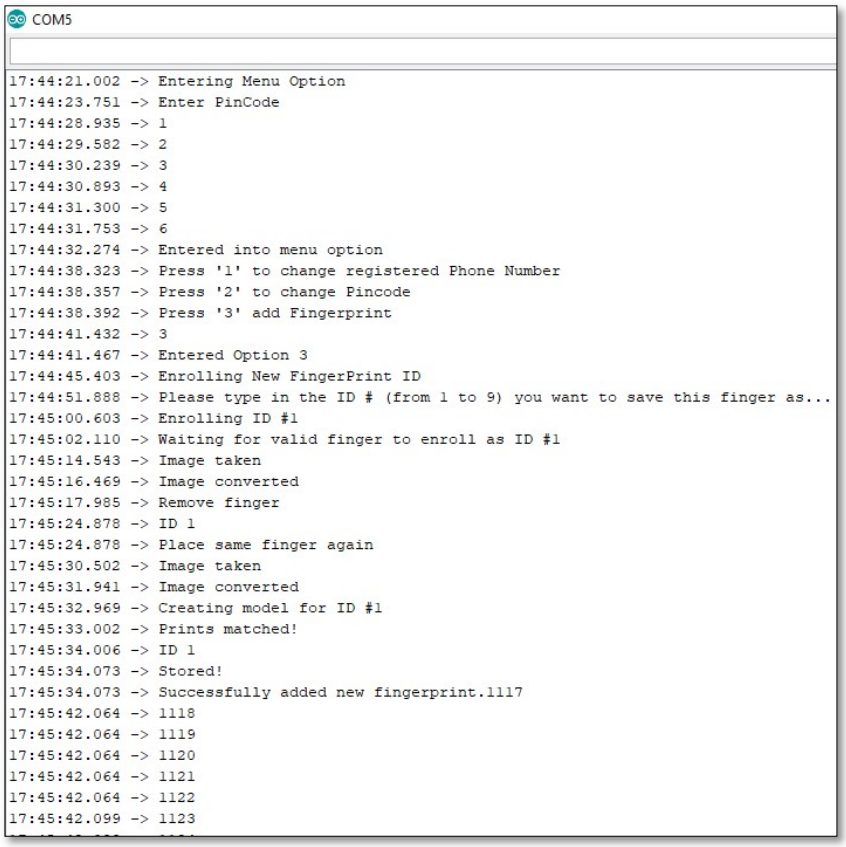

Figure 4.3.3.1: Flow of new fingerprint enrolment process

The above output screen of the serial monitor shows the workflow when the new fingerprint enrolment function is being called. As shown above, the fingerprint function is called from the menu option when the user selects option “3”. Once the fingerprint is successfully stored the function breaks and exits menu option.

The code for the second function (Fingerprint Verification) is as follows:

```
int getFingerprintIDez()
{
 //delay(3000);

 uint8_t p =finger.getImage();

 while(p!=FINGERPRINT_OK)
 {
  p = finger.getImage();
 }

 p = finger.image2Tz();
 if (p != FINGERPRINT_OK)
  return -1;

 p = finger.fingerFastSearch();
 if (p != FINGERPRINT_OK)
  return -1;

 // found a match!
 Serial.print("Found ID #");
 Serial.print(finger.fingerID);
 Serial.print(" with confidence of ");
 Serial.println(finger.confidence);

 boolean FM = true;
 return FM;
}
```

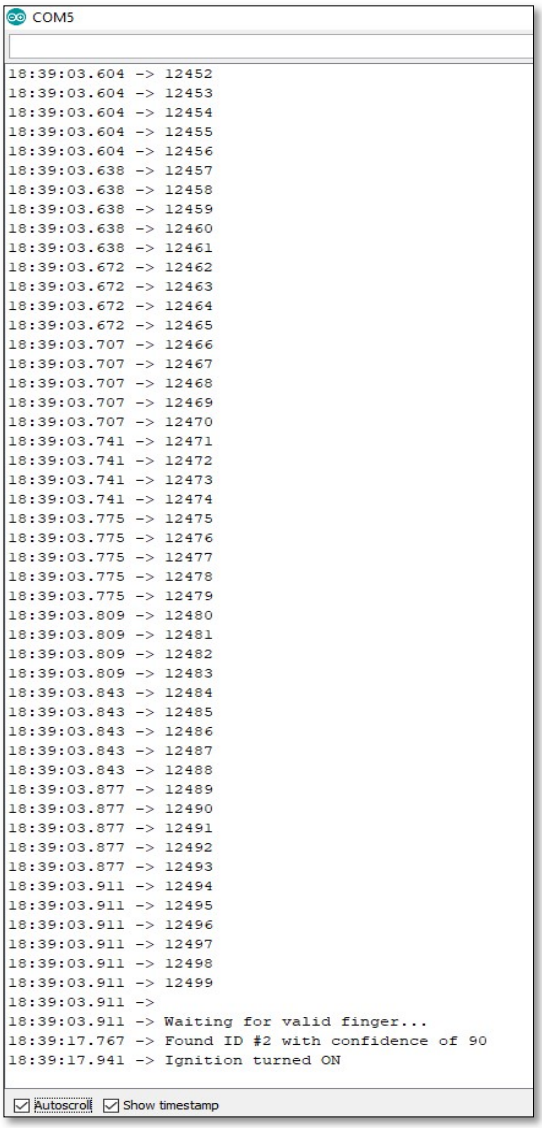

Figure 4.3.3.2: Flow of fingerprint verification process

The above output screen of the serial monitor shows the fingerprint verification process initialized when both PWR and ACC gives input signal as high. Once the function is called, it waits for a finger to be placed on the scanner to be compared with all the registered fingerprint images. If the fingerprint image matches any of the registered images

in the system, its shows found the ID number of the image with the confidence level. If the confidence level is above 80 the ignition will be turned ON, else it goes for the second and third trials.

4.3.4 Location detector and sending its information using SIM808 Module

Location detection is a crucial part of this system, where this system collects the current location's coordinates of the vehicle and sends the information to the authorized user. The required hardware for both of this functions to be executed is the SIM808 module, since this module has the functionality of GSM for telecommunication and gathering GPS information. There are two user defined function created based on this module for this system which are the "TrackingFunction" and the "AlertTrackingFunction".

The "TrackingFunction" is the function to check if any tracking command send from the user to the system. If a message is received by the system this function tests two parameters, the phone number of the sender and the keyword sent. For the phone number it compares with the registered phone in the system and for the keyword there is already a fixed keyword to be compared. Once both parameters have been satisfied, only then the system sends the location information to the registered phone number.

The “AlertTrackingFunction” will only be called when the biometric scans fail its trials for three times. Once it is called, it halts all other functions and gathers the location coordinates information to be sent to the owner or authorized user’s phone number through SMS, then it waits for the response of the owner or authorized user. There are only two type of messages that can be sent from the owner or an authorized user which are “allow” to allow the ignition system to be turned on with a condition where the user in the vehicle has to enter the PINCODE of the system and all other messages are considered as declining commands.

The code for the “TrackingFunction” is as follows:

```
void TrackingFunction(void)
{
 if(sim808.attachGPS())
 {
    Serial.println("Open the GPS power success");
 }
 else
 {
    Serial.println("Open the GPS power failure");
 }
 Serial.println("Init Success, please send SMS message to me!");
 //Serial.println(sim808.isSMSunread());

 //********** Detecting unread SMS ************************
  messageIndex = sim808.isSMSunread();


  //********** At least, there is one UNREAD SMS **********
  if (messageIndex > 0)
  {
    Serial.print("messageIndex: ");
    Serial.println(messageIndex);
```

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

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

    Serial.print("From number: ");
    Serial.println(phone);
    Serial.print("Datetime: ");
    Serial.println(datetime);
    Serial.print("Received Message: ");
    Serial.println(message);

    Serial.println(regnum);
    if(strcmp(regnum,phone) == 0)
    {
     if(strcmp(keyword,message) == 0)
     {
      GSMSerial.begin(9600);
      //******** Initialize sim808 module *************
      while(!sim808.init())
      {
         Serial.print("Sim808 initialization error\r\n");

         lcd.setCursor(0,0);lcd.print("Sim808 error");
         lcd.setCursor(0,0);lcd.print("                ");
      }
      Serial.println("GSM Setup Done!!");
      if(sim808.attachGPS())
      {
       Serial.println("Open the GPS power success");
      }
      else
      {
         Serial.println("Open the GPS power failure");
      }
      while(!sim808.getGPS())
      {
       delay(1);
      }
      Serial.print(sim808.GPSdata.year);
      Serial.print("/");
      Serial.print(sim808.GPSdata.month);
      Serial.print("/");
```

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

        float la = sim808.GPSdata.lat;
        float lo = sim808.GPSdata.lon;
        float ws = sim808.GPSdata.speed_kph;

        dtostrf(la, 6, 6, lat); //put float value of la into char array of lat. 6 = number of
digits before decimal sign. 2 = number of digits after the decimal sign.
        dtostrf(lo, 6, 6, lon); //put float value of lo into char array of lon
        dtostrf(ws, 6, 2, wspeed);  //put float value of ws into char array of wspeed

        sprintf(MESSAGE, "Latitude : %s\nLongitude : %s\nWind Speed : %s
kph\nhttp://maps.google.com/maps?q=%s,%s\n", lat, lon, wspeed, lat, lon);


        Serial.println("Sim808 init success");
        Serial.println("Start to send message ...");

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

        sim808.sendSMS(phone,MESSAGE);
        //for(i=messageIndex; i>0; i--)
        //{
          sim808.deleteSMS(messageIndex);
          //Serial.println(i);
        //}
```

```
  //************* Turn off the GPS power ************
  sim808.detachGPS();
 }
 else
 {
 sprintf(MESSAGE, "Wrong Keyword!!!");

 Serial.println("Sim808 init success");
 Serial.println("Start to send message ...");

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

 sim808.sendSMS(phone,MESSAGE);
 for(i=messageIndex; i>0; i--)
 {
  sim808.deleteSMS(i);
  Serial.println(i);
 }

 //************* Turn off the GPS power ************
 //sim808.detachGPS();
 }
}
else
{
 sprintf(MESSAGE, "Wrong phone!!!");

 Serial.println("Sim808 init success");
 Serial.println("Start to send message ...");

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

 sim808.sendSMS(phone,MESSAGE);
 for(i=messageIndex; i>0; i--)
 {
  sim808.deleteSMS(i);
  Serial.println(i);
 }

 //************* Turn off the GPS power ************
 sim808.detachGPS();
```

```
    }

  }
}
```

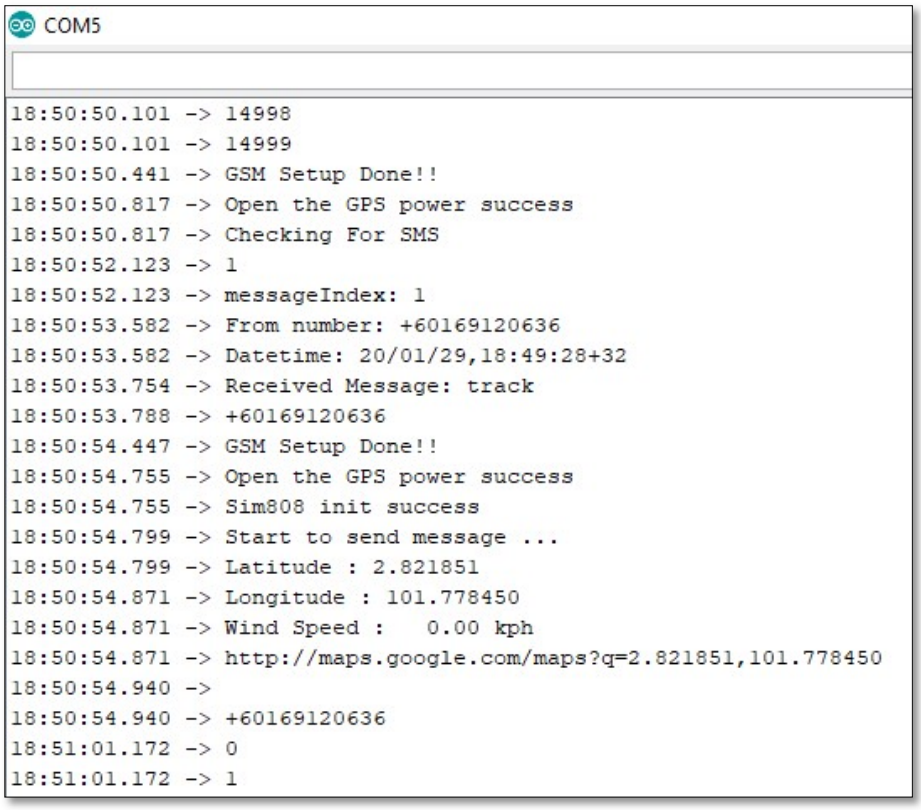

Figure 4.3.4.1: Flow of "TrackingFunction"

The above output screen of the serial monitor shows the flow of the "TrackingFunction". When this function is called, it checks for any new message, if it receives any new message is found, it checks for the keyword and phone number. If it both parameters satisfy the conditions, the system turns on the GPS power source to get the location's coordinates and sends that information to the user's phone number.

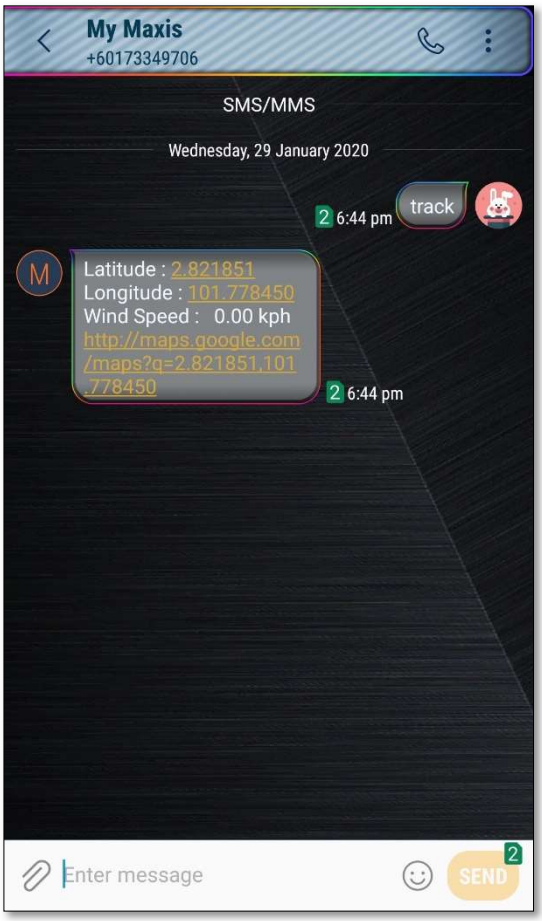

Figure 4.3.4.2: Screenshot of "TrackingFunction" function's output

The figure 4.3.4.2 shows the message sent from the system to the user's phone number through SMS when the user sends the correct keyword.

The code for the "AlertTrackingFunction" is as follows:

```
void AlertTrackingFunction(void)
{
 GSMSerial.begin(9600);
 //******** Initialize sim808 module *************
 while(!sim808.init())
```

```
{
   Serial.print("Sim808 initialization error\r\n");

   lcd.setCursor(0,0);lcd.print("Sim808 error");
   lcd.setCursor(0,0);lcd.print("                ");
}
Serial.println("GSM Setup Done!!");
//************* Turn on the GPS power ************
if (sim808.attachGPS())
{
 Serial.println("Open the GPS power success");
}
else
{
 Serial.println("Open the GPS power failure");
}

while(!sim808.getGPS())
  {
   delay(1);
  }
Serial.print(sim808.GPSdata.year);
Serial.print("/");
Serial.print(sim808.GPSdata.month);
Serial.print("/");
Serial.print(sim808.GPSdata.day);
Serial.print(" ");
Serial.print(sim808.GPSdata.hour);
Serial.print(":");
Serial.print(sim808.GPSdata.minute);
Serial.print(":");
Serial.print(sim808.GPSdata.second);
Serial.print(":");
Serial.println(sim808.GPSdata.centisecond);
Serial.print("latitude :");
Serial.println(sim808.GPSdata.lat);
Serial.print("longitude :");
Serial.println(sim808.GPSdata.lon);
Serial.print("speed_kph :");
Serial.println(sim808.GPSdata.speed_kph);
Serial.print("heading :");
Serial.println(sim808.GPSdata.heading);
Serial.println();
```

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

  dtostrf(la, 6, 6, lat); //put float value of la into char array of lat. 6 = number of digits
before decimal sign. 2 = number of digits after the decimal sign.
  dtostrf(lo, 6, 6, lon); //put float value of lo into char array of lon
  dtostrf(ws, 6, 2, wspeed);  //put float value of ws into char array of wspeed

  sprintf(MESSAGE, "ALERT MESSAGE!!!\nYour vehicle is being used by unknown
user.\nYour vehicle's current location.\nLatitude : %s\nLongitude :
%s\nhttp://maps.google.com/maps?q=%s,%s\n", lat, lon, lat, lon);

  Serial.println("Sim808 init success");
  Serial.println("Start to send message ...");
  sim808.sendSMS(regnum, MESSAGE);
  Serial.println(MESSAGE);
  Serial.println(regnum);

  //************* Turn off the GPS power ************
  sim808.detachGPS();

  delay(2000);

  if (sim808.attachGPS())
  {
   Serial.println("Open the GPS power success");
  }
  else
  {
   Serial.println("Open the GPS power failure");
  }

  Serial.println("Please send SMS message to me!");
  Serial.println(sim808.isSMSunread());

  //*********** Detecting unread SMS ************************
  messageIndex = sim808.isSMSunread();

  //*********** Waits for detecting unread SMS ************************
  while (messageIndex == -1)
  {
   messageIndex = sim808.isSMSunread();
   Serial.println(messageIndex);
```

```
  delay(1000);
}

//*********** At least, there is one UNREAD SMS ***********
if (messageIndex > 0)
{
  Serial.print("messageIndex: ");
  Serial.println(messageIndex);

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

  //*********** Reading received message from Sim**********

  Serial.print("From number: ");
  Serial.println(phone);
  Serial.print("Datetime: ");
  Serial.println(datetime);
  Serial.print("Received Message: ");
  Serial.println(message);

  int o=0;
  char AllowCmd[] = "allow";
  for(int i=0; i<sizeof(AllowCmd); i++)
  {
    if(AllowCmd[i]==message[i])
    {
     o++;
    }
  }
  Serial.println(sizeof(AllowCmd));
  Serial.println(o);
  if (o==6)
  {
    Serial.println("Allowed");
    int menuPinCounter = 0;
    char pinInput[7];
    pinInput[6] = '\0';
    Serial.println("Enter PinCode");


    int ps=0;
    while(menuPinCounter<6)
    {
      //keypad.waitForKey();
```

```
    while(true)
    {
     char input = keypad.getKey();
     pinInput[menuPinCounter]=input;
     if (input)Serial.println(input);
     if (input)
     {
      lcd.setCursor(ps++,0);
      lcd.print("*");

     }
     if (input){break;}
    }
    menuPinCounter++;
  }
  delay(500);
  int chk=0;
  for(int i = 0; i < 6; i++)
  {
   if(pincode[i]==pinInput[i])
   chk++;
  }
  if(chk==6)
  {
   lcd.setCursor(0,0);lcd.print("                ");
     tone(49,1000); //Short Beep
     delay(200);
     noTone(49);
   digitalWrite(4, HIGH);
   Serial.println("Ignition turned ON");
   lcd.setCursor(0,0);lcd.print("Ignition ON");
   delay(1500);
   lcd.setCursor(0,0);lcd.print("                ");
   digitalWrite(4, LOW);
  }
  else{
   tone(49,1000); //Short Beep
   delay(800);
   noTone(49);
   Serial.println("Wrong PINCODE");
   lcd.setCursor(0,0);lcd.print("Wrong PINCODE");
  }
 }
 else
```

```
  {
    return -1;
  }
}
 //************* Turn off the GPS power ************
 sim808.detachGPS(); }
```

Figure 4.3.4.3: Flow of the "AlertTrackingFunction"

The process flow/output of the “AlertTrackingFunction” function is shown in the figure 4.3.4.3. This is the main security feature of this system and also the alternative option to turn ON the ignition system when the fingerprint fails to read.

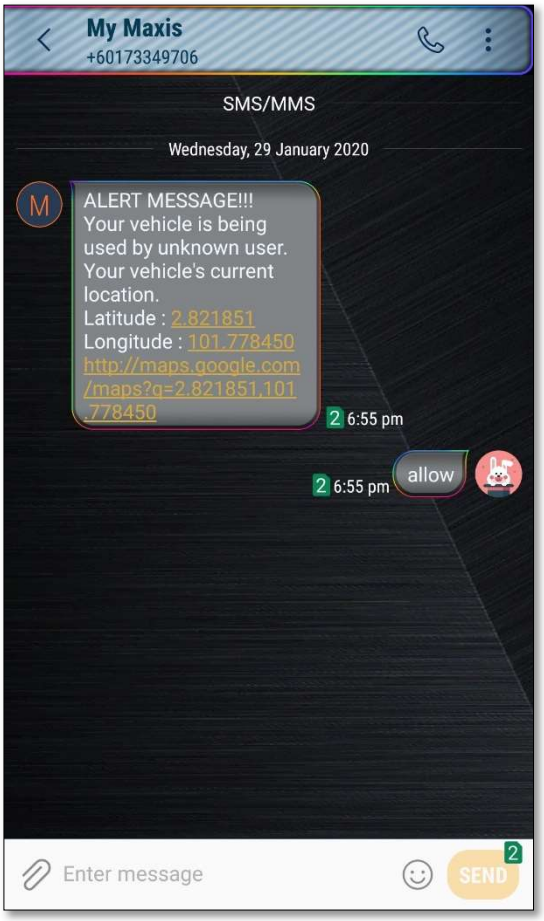

Figure 4.3.4.4: Screenshot of “AlertTrackingFunction” function’s output

The figure 4.3.4.2 shows the alert message sent from the system to the user’s phone number through SMS when the fingerprint trials exits three times and it waits for the authorized user to send a response message.

4.3.5 Receiving physical input using Keypad

It is very important to obtain input from the user, so a keypad is included as an input device with this system to make this system easier and convenient to use. To include this keypad module there are not much to define since most of the functions are included in the Keypad.h library. The keys of the keypad module are defined in the setup function. To receives any input from the user, keypad.getKey() function is called.

The keypad setup codes are as below:

```
char pincode[7];
int MenuEntries=0;
const byte ROWS = 5; //five rows
const byte COLS = 4; //four columns
char keys[ROWS][COLS] = {
  {'a','b','#','*'},
  {'1','2','3','u'},
  {'4','5','6','d'},
  {'7','8','9','Q'},
  {'l','0','r','E'}
};

byte rowPins[ROWS] = {23,24,25,26,27}; //connect to the row pinouts of the keypad
byte colPins[COLS] = {31,30,29,28}; //connect to the column pinouts of the keypad

Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );
```

4.3.6 Control relays as switches.

Relays are used as switches in this prototype system where is controlled automatically by the system without human effort. There are three sets of relays, total of six relays used, two of the sets are used to detect inputs and one is to control output.

The first set of input detection relays are to detect and manage power supply units. Both of these relay's coils are connected to main power supply. Each of the relay is dedicated for each power terminal, (+) and (-) and connected to the Normally Open (NO) pin. If the coil is activated the current flow from the main power source (NO) to the common pin (C) of the relays, else the relays switches from Normally Open (NO) to Normally Closed (NC) where the back-up batteries are connected. Table 4.3.6.1 shows the relay power input table.

Coil	Normally Open (NO)	Normally Close (NC)	Common	Power Source
0	0	1	NC	BU Battery
1	1	0	NO	Main Source

Table 4.3.6.1: Relay power input table

The second set of input detection relays are used to detect the Power (PWR) and Ignition (ACC). One of the relay's coil is connected to the PWR input and the other is connected to the ACC input. These relays are supplied with 5V to the Normally Open (NO) pin and ground (GND) is connected to the Normally Closed (NC) pin. When the coils are activated these send 5V inputs as signals to the CD4081BE IC which is an AND gate chip, else GND. Then this chip will send an input to the processor based on the signal sent from the relays. The table 4.3.6.2 show the input-output table where 0 represents GND and 1 represents 5V.

Input A (PWR)	Input B (ACC)	Output A·B
0	0	0
0	1	0
1	0	0
1	1	1

Table 4.3.6.2: Truth table of CD4081BE IC

The third set of relays are used to control the output of 12V power using 5V relays. These relay's coils are connected to the processor pin number 2 and 4. One of the relay is

to control PWR and the other is to control ACC power supply independently. Table 4.3.6.3 shows the status of the coil and the output of the relay.

	Coil (Status)	Common (V)
Power (PWR)	OFF	0
	ON	12
Ignition (ACC)	OFF	0
	ON	12

Table 4.3.6.3: Output table of ACC and PWR

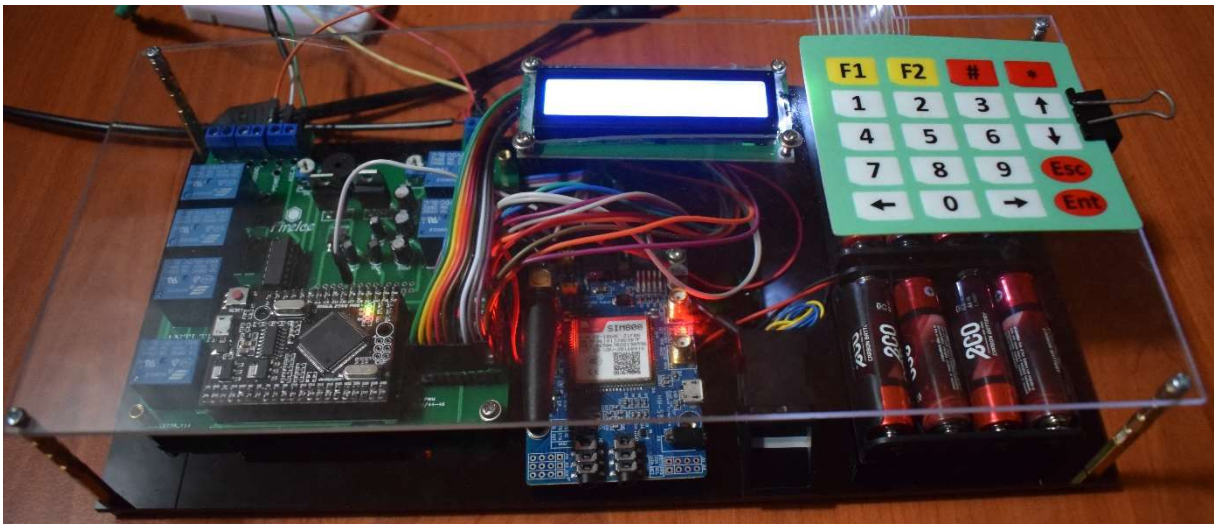

Figure 4.3.1: Full prototype system hardware setup

CHAPTER 5

DISCUSSION AND CONCLUSION

The conclusion of the project and discussion of results, progress and the completion of the project are covered in this Chapter 5.

5.1. Discussion of Results

There were three main objectives of this project as stated in CHAPTER 1 INTRODUCTION to be achieved by the end of this project. During the process of development, there were few changes were made in the workflow and functionality of the system to the proposed project earlier such as including a menu option for the user to modify the user data, including a buzzer for the user to be notified, and a LCD for the user to know the steps. These changes were made to make the system a success. At the end of this project, all three objectives were achieved and its shown with proof in the results chapter (Chapter 4) and described in section 5.2.

5.2 Achievement of the objectives

Below given table shows the achievement of the objectives as stated initially.

NO.	OBJECTIVES	REMARKS
1.	To provide a biometric authentication to start the vehicle.	This prototype system is equipped with fingerprint scanner to read the user's fingerprint and compare with the registered fingerprint images in the system.
2.	To develop a system that is capable of tracking the vehicle.	A GPS module (SIM808) has been used to gather the system's locations coordinates in order to send it to the user.
3.	To develop a system which can send real-time information to the user	Using a GSM technology (SIM808) the system was able to send real-time information to the authorized user.

Table 5.2.1: Achievement of objectives

5.3 Implication of study

Based on the result of this study, it can be clearly stated that this system can be a very useful enhancement to the security features in the automobile industry. Since this system is an additional security feature it does not needs much changes in the existing

vehicle's system. With some understanding of this system and the vehicle's system anyone will be able to install this system. Once its installed, it will reduce this risk of stolen vehicle by starting the vehicle engine. Even though if the vehicle had been hijacked, the user will be able to locate the vehicle by just sending a SMS to the system.

5.4 Future Enhancements

In the coming future, there are many other features and functionality that can be added to this system to improve its effect and efficiency to protect the vehicle. First of all, if this system was to be to produce in the mass production, the creation of the system needs to add a hashing system in order to create new default password unique for each and every product. Secondly, the system has to used lithium-ion based rechargeable batteries. By using this type of batteries, the number of time that back-up batteries needed to be changed will be reduces tremendously. Lithium-ion batteries has many advantages compared to ordinary batteries, such as, high energy density, which is potential for yet higher capacities, does not need prolonged priming when new and relatively low self-discharge, it self-discharge is less than half that of nickel-based batteries [17]. Next will be adding a function for the system to call an emergency contact number if any theft was attempted during the authorized user using the vehicle. Other than that, creating a mobile application for this system will bring this system to the current Internet of Things (IOT) era. This application can be used to monitor the vehicle's location and activities, like who are using the vehicle

and so on. Lastly, integrating more security features and functionality together with this system can improve the vehicle's and user's security. Other functions that can be included are ability to detect any forceful door breakage and glass breakage and adding external cameras surrounding the vehicle for monitoring the vehicles surroundings and internals for capture images or record videos if any theft attempt was detected.

5.5 Conclusion

Understanding the importance of security in the field in automobile industry is crucial to manufactures and also to the users. The implementation of this system in the actual world will be very wide in the coming future, according to the vehicle theft studies. Apparently, the demand for the vehicle security system will continuously increase in coming years. Although security system for vehicles exist, this project is aimed to integrate multiple security measures to increase its efficiency and accuracy. The project was successfully completed by achieving all the objectives that were listed in the beginning.

REFERENCES

[1] T. Harris, "howstuffworks," 2 August 2001. [Online]. Available: https://auto.howstuffworks.com/car-alarm.htm.

[2] Y. D. Eusebio, "autodeal," 24 April 2018. [Online]. Available: https://www.autodeal.com.ph/articles/car-features/what-engine-immobilizer-and-why-it-important.

[3] "VTREC," November 2018. [Online]. Available: http://www.vtrec.net/?page_id=455.

[4] "The Straits Times," 13 February 2018. [Online]. Available: https://www.straitstimes.com/asia/se-asia/cars-with-keyless-entry-system-are-easy-theft-targets.

[5] S. Joge, "GSM BASED SECURITY FOR VEHICLE," *International Research Journal of Engineering and Technology (IRJET),* vol. V, no. 2, pp. 915-920, 2018.

[6] F. K. Samuel and Z. Alexander, "Design of Anti-Vehicle Theft System using GSM and GPS with Image Acquisition," *Asian Journal of Engineering and Technology,* vol. 05, no. 04, pp. 82-92, 2017.

[7] J. M. Laguador, M. M. Chung, F. J. D. Dagon, J. A. M. Guevarra, R. J. Pureza, J. D. Sanchez and D. K. I. S. Iglesia, "Anti Car Theft System using Android Phone," *INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY SCIENCES AND ENGINEERING,* vol. 4, no. 5, pp. 12-14, 2013.

[8] R. Samir, M. Ritu and N. Lata, "Anti theft Security System for Vehicles," *International Journal of Engineering & Technology,* vol. 7, no. 4.12, pp. 42-46, 2018.

[9] Z. B. Kumar and N.Bharathi, "Vehicle Anti-Theft System Using Fingerprint Recognition Technique," *Open Academic Journal of Advanced Science and Technology,* vol. 1, no. 1, pp. 36-41, 2017.

[10] "BEN," sparkfun, 9 November 2011. [Online]. Available: https://learn.sparkfun.com/tutorials/what-is-an-arduino/all#whats-on-the-board.

[11] Itead, "itead.cc," itead, 2016 August 8. [Online]. Available: https://www.itead.cc/wiki/SIM808_GSM/GPRS/GPS_Module.

[12] espruino, "espruino.com," [Online]. Available: https://www.espruino.com/KeyPad.

[13] D. Dunmur and H. G. Walton, "Encyclopaedia Britannica," Britannica, 28 May 2004. [Online]. Available: https://www.britannica.com/technology/liquid-crystal-display. [Accessed 03 August 2015].

[14] Texas Instrument, [Online]. Available: https://www.ti.com/store/ti/en/p/product/?p=CD4081BE.

[15] "arduino.cc," Arduino, 7 September 2015. [Online]. Available: https://www.arduino.cc/en/Guide/Environment.

[16] RishabhPrabhu, "geeksforgeeks," [Online]. Available: https://www.geeksforgeeks.org/c-language-set-1-introduction/.

[17] "Safewise," [Online]. Available: https://www.safewise.com/home-security-faq/how-do-security-systems-work/.

[18] M. Bullock, "howstuffworks.com," [Online]. Available: https://electronics.howstuffworks.com/relay2.htm.

APPENDIX

File Name: Anti_Theft_System.ino

Full code for this system is as below:

```
#include <DFRobot_sim808.h>
#include <sim808.h>
#include <Key.h>
#include <Keypad.h>
#include <SoftwareSerial.h>
#include <Adafruit_Fingerprint.h>
#include <LiquidCrystal.h>

//********Global Declaration for LCD
Crystal**********************************
//const int rs = 42, en = 43, d4 = 47, d5 = 46, d6 = 41, d7 = 44;

const int rs = 42, en = 43, d4 = 44, d5 = 41, d6 = 46, d7 = 47;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
//**********************************************************************
******

//********Global Declaration for
Keypad**************************************
char pincode[7];
int MenuEntries=0;
const byte ROWS = 5; //five rows
const byte COLS = 4; //four columns
char keys[ROWS][COLS] = {
  {'a','b','#','*'},
  {'1','2','3','u'},
  {'4','5','6','d'},
  {'7','8','9','Q'},
  {'l','0','r','E'}
};

byte rowPins[ROWS] = {23,24,25,26,27}; //connect to the row pinouts of the keypad
```

```
byte colPins[COLS] = {31,30,29,28}; //connect to the column pinouts of the keypad

Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );
//*************************************************************************
******

//*****Global declaration for
Tracking*****************************************
#define MESSAGE_LENGTH 160
char message[MESSAGE_LENGTH];
int messageIndex = 0;
char phone[16];
char datetime[24];
char regnum[] = "+60160000000";
char keyword[] = "track";
char MESSAGE[300];
char lat[12];
char lon[12];
char wspeed[12];
int i;

#define PIN_TX 52
#define PIN_RX 53

SoftwareSerial GSMSerial(PIN_TX,PIN_RX);
DFRobot_SIM808 sim808(&GSMSerial);
//*************************************************************************
******

//*****Global declaration for
FingerPrint*****************************************
//uint8_t id;
boolean FM = false;
boolean StoreChk;
char id="";

SoftwareSerial mySerial(50,51);

Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);
//*************************************************************************
******

int IgnitionChecker = 0;
int PowerState = 0;
```

```
void setup() {
  Serial.begin(9600);
  pinMode(2, OUTPUT);
  pinMode(4, OUTPUT);

  //*****for LCD
Crystal*****************************************************
  lcd.begin(16,2);
  lcd.setCursor(0,0);
  Serial.println("LCD Crystal Setup Done!!");

  int TN1=1;
  char displayWords1[]="LCD Crystal Setup Completed!!!";
  for(int d=0 ; d < sizeof(displayWords1)-1 ; d++ )
  {
   if(d<16)
   {
    lcd.setCursor(d,0);lcd.print(displayWords1[d]);
   }
   else
   {
    lcd.setCursor(0,0);
    lcd.print("                ");
    lcd.setCursor(0,0);
    for(int d1=TN1 ; d1 < sizeof(displayWords1) ; d1++ )
    {
     lcd.print(displayWords1[d1]);
    }
    TN1++;
   }
   delay(200);
  }
  delay(700);
  lcd.setCursor(0,0);
  lcd.print("                ");

//*****************************************************************
******

  //*****for
Keypad*******************************************************
  pincode[0]='1';
  pincode[1]='2';
```

```
  pincode[2]='3';
  pincode[3]='4';
  pincode[4]='5';
  pincode[5]='6';
  pincode[6]='\0';
  Serial.println("Keypad Setup Done!!");

  /*lcd.setCursor(0,0);lcd.print("Keypad Setup Done!!");
  delay(500);
  lcd.setCursor(0,0);lcd.print("                ");
  delay(500);*/

  int TN2=1;
  char displayWords2[]="Keypad Setup Completed!!!";
  for(int d=0 ; d < sizeof(displayWords2)-1 ; d++ )
  {
   if(d<16)
   {
    lcd.setCursor(d,0);lcd.print(displayWords2[d]);
   }
   else
   {
    lcd.setCursor(0,0);
    lcd.print("                ");
    lcd.setCursor(0,0);
    for(int d1=TN2 ; d1 < sizeof(displayWords2) ; d1++ )
    {
     lcd.print(displayWords2[d1]);
    }
    TN2++;
   }
   delay(200);
  }
  delay(700);
  lcd.setCursor(0,0);
  lcd.print("                ");

//*****************************************************************************
******


  //*****for
Tracking*************************************************************
```

```
  GSMSerial.begin(9600);
  //******** Initialize sim808 module *************
  while(!sim808.init())
  {
     Serial.print("Sim808 initialization error\r\n");

     lcd.setCursor(0,0);lcd.print("Sim808 error");
     lcd.setCursor(0,0);lcd.print("                ");
  }
  Serial.println("GSM Setup Done!!");

  int TN3=1;
  char displayWords3[]="GSM Setup Completed!!!";
  for(int d=0 ; d < sizeof(displayWords3)-1 ; d++ )
  {
   if(d<16)
   {
    lcd.setCursor(d,0);lcd.print(displayWords3[d]);
   }
   else
   {
    lcd.setCursor(0,0);
    lcd.print("                ");
    lcd.setCursor(0,0);
    for(int d1=TN3 ; d1 < sizeof(displayWords3) ; d1++ )
    {
     lcd.print(displayWords3[d1]);
    }
    TN3++;
   }
   delay(200);
  }
  delay(700);
  lcd.setCursor(0,0);
  lcd.print("                ");


//*****************************************************************************
*****

  //***for FingerPrint
Scanner*************************************************
  while (!Serial);
  //delay(100);
```

```
Serial.println("\nAdafruit Fingerprint sensor enrollment");

// set the data rate for the sensor serial port
finger.begin(57600);

if (finger.verifyPassword())
{
  Serial.println("Found fingerprint sensor!");
}
else
{
  Serial.println("Did not find fingerprint sensor :(");
  while(1)
  {
    delay(1);
  }
}

finger.getTemplateCount();
Serial.print("Sensor contains ");
Serial.print(finger.templateCount);
Serial.println(" templates");
Serial.println("FingerPrint Scanner Setup Done!!");

int TN4=1;
char displayWords4[]="FingerPrint Scanner Setup Completed!!!";
for(int d=0 ; d < sizeof(displayWords4)-1 ; d++ )
{
  if(d<16)
  {
    lcd.setCursor(d,0);lcd.print(displayWords4[d]);
  }
  else
  {
    lcd.setCursor(0,0);
    lcd.print("                ");
    lcd.setCursor(0,0);
    for(int d1=TN4 ; d1 < sizeof(displayWords4) ; d1++ )
    {
      lcd.print(displayWords4[d1]);
    }
    TN4++;
  }
  delay(200);
```

```
  }
  delay(700);
  lcd.setCursor(0,0);
  lcd.print("                ");


//*****************************************************************************
******
}

void loop()
{
  PowerState = digitalRead(33);

  if(PowerState == HIGH)
  { digitalWrite(2, HIGH); }
  else
  { digitalWrite(2, LOW); }

  for(int x=0;x<21000;x++)
  {
    MenuOption();
    IgnitionCheck();
    Serial.println(x);

  }

  lcd.setCursor(0,0);lcd.print("Wait...");
  TrackingFunction();
  lcd.setCursor(0,0);lcd.print("                ");
}

//***Function
Declarations*********************************************************

//***Ignition
Check*********************************************************

void IgnitionCheck(void)
{
  IgnitionChecker = digitalRead(32);
  if (IgnitionChecker != LOW)
  {
    for(int i = 1; i <= 3; i++)
```

```
  {
   Serial.println("\nWaiting for valid finger...");
   lcd.setCursor(0,0);lcd.print("Waiting for FGPT");



   if(getFingerprintIDez() == true && PowerState == HIGH)
   {
    lcd.setCursor(0,0);lcd.print("                ");
      tone(49,1000);
      delay(200);
      noTone(49);
    digitalWrite(4, HIGH);
    Serial.println("Ignition turned ON");
    lcd.setCursor(0,0);lcd.print("Ignition ON");
    delay(1500);
    lcd.setCursor(0,0);lcd.print("                ");
    digitalWrite(4, LOW);
    i=4;
   }
   else
   {
    lcd.setCursor(0,0);lcd.print("                ");
    lcd.setCursor(0,0);lcd.print("No FGPT Match");
    tone(49,1000);
    delay(800);
    noTone(49);
    delay(200);
    lcd.setCursor(0,0);lcd.print("                ");
    Serial.println("Fingerprint did not match");
    Serial.print("Failed Trail #");
    Serial.print(i);
    lcd.setCursor(0,0);lcd.print("Failed Trial #");
    lcd.setCursor(14,0);lcd.print(i);
    delay(1200);
    lcd.setCursor(0,0);lcd.print("                ");
    if(i==3)
    {
     AlertTrackingFunction();
     break;
    }
   }
  }
}
```

```
}
//*****************************************************************************
******

//***MenuOption*************************************************************
******

void MenuOption(void)
{
 char key = keypad.getKey();

 if (key){Serial.println(key);} // just print the pressed key

 if (key =='*'){
  if (MenuEntries <3){
    MenuEntries++;
    if(MenuEntries == 3) //Entering into Menu option
    {
     Serial.println("Entering Menu Option");
     int TN4=1;
     char displayWords4[]="Entering Menu Option";
     for(int d=0 ; d < sizeof(displayWords4)-1 ; d++ )
     {
      if(d<16)
      {
       lcd.setCursor(d,0);lcd.print(displayWords4[d]);
      }
      else
      {
       lcd.setCursor(0,0);
       lcd.print("                ");
       lcd.setCursor(0,0);
       for(int d1=TN4 ; d1 < sizeof(displayWords4) ; d1++ )
       {
        lcd.print(displayWords4[d1]);
       }
       TN4++;
      }
      delay(100);
     }
     delay(700);
     lcd.setCursor(0,0);
     lcd.print("                ");
     MenuEntries = 0; // Reset the counter value to 0
```

```
//Validating Pincode
int menuPinCounter=0;
char pinInput[7];
pinInput[6]='\0';
Serial.println("Enter PinCode");
int TN5=1;
char displayWords5[]="Enter PinCode";
for(int d=0 ; d < sizeof(displayWords5)-1 ; d++ )
{
 if(d<16)
 {
  lcd.setCursor(d,0);lcd.print(displayWords5[d]);
 }
 else
 {
  lcd.setCursor(0,0);
  lcd.print("                ");
  lcd.setCursor(0,0);
  for(int d1=TN5 ; d1 < sizeof(displayWords5) ; d1++ )
  {
   lcd.print(displayWords5[d1]);
  }
  TN5++;
 }
 delay(200);
}
delay(700);
lcd.setCursor(0,0);
lcd.print("                ");
int ps=0;
while(menuPinCounter<6)
{
 //keypad.waitForKey();
 while(true)
 {
  char input = keypad.getKey();
  pinInput[menuPinCounter]=input;
  if (input)Serial.println(input);
  if (input)
  {
   lcd.setCursor(ps++,0);
   lcd.print("*");
```

```
    }
    if (input){break;}
   }
   menuPinCounter++;
 }
 delay(500);
 int chk=0;
 for(int i = 0; i < 6; i++)
 {
  if(pincode[i]==pinInput[i])
  chk++;
 }
 if(chk==6)
 {
  //Short Beep
  Serial.println("Entered into menu option");
    tone(49,1000);
    delay(200);
    noTone(49);
  int TN5=1;
  char displayWords5[]="Entered into menu option ";
  for(int d=0 ; d < sizeof(displayWords5)-1 ; d++ )
  {
   if(d<16)
   {
    lcd.setCursor(d,0);lcd.print(displayWords5[d]);
   }
   else
   {
    lcd.setCursor(0,0);
    lcd.print("                ");
    lcd.setCursor(0,0);
    for(int d1=TN5 ; d1 < sizeof(displayWords5) ; d1++ )
    {
     lcd.print(displayWords5[d1]);
    }
    TN5++;
   }
   delay(200);
  }
  delay(700);
  lcd.setCursor(0,0);
  lcd.print("                ");
```

```
Serial.println("Press '1' to change registered Phone Number");
Serial.println("Press '2' to change Pincode");
Serial.println("Press '3' add Fingerprint");
for(int i=0; i<1; i++)
{
 lcd.setCursor(0,0);lcd.print("1.Change Number");
 delay(1000);
 lcd.setCursor(0,0);lcd.print("1.Change Number");
 lcd.setCursor(0,1);lcd.print("2.Change Pincode");
 delay(1000);
 lcd.setCursor(0,0);lcd.print("                ");
 lcd.setCursor(0,1);lcd.print("                ");

 lcd.setCursor(0,0);lcd.print("2.Change Pincode");
 lcd.setCursor(0,1);lcd.print("3.Add FingerPrint");
 delay(1000);
}
while(true)
{
 char input = keypad.getKey();
 if(input)Serial.println(input);
 //***Option 1**************************************
 if (input=='1')
 {
  lcd.setCursor(0,0);lcd.print("                ");
  lcd.setCursor(0,1);lcd.print("                ");
  Serial.println("Entered Option 1");
  int TN5=1;
  char displayWords5[]="Entered Option 1";
  for(int d=0 ; d < sizeof(displayWords5)-1 ; d++ )
  {
   if(d<16)
   {
    lcd.setCursor(d,0);lcd.print(displayWords5[d]);
   }
   else
   {
    lcd.setCursor(0,0);
    lcd.print("                ");
    lcd.setCursor(0,0);
    for(int d1=TN5 ; d1 < sizeof(displayWords5) ; d1++ )
    {
     lcd.print(displayWords5[d1]);
    }
```

```
    TN5++;
   }
   delay(200);
  }
  delay(700);
  lcd.setCursor(0,0);
  lcd.print("                ");

  Serial.println("Enter Phone Number then 'Enter' ");
  int TN51=1;
  char displayWords51[]="Enter Phone Number then 'Enter' ";
  for(int d=0 ; d < sizeof(displayWords51)-1 ; d++ )
  {
   if(d<16)
   {
    lcd.setCursor(d,0);lcd.print(displayWords51[d]);
   }
   else
   {
    lcd.setCursor(0,0);
    lcd.print("                ");
    lcd.setCursor(0,0);
    for(int d1=TN51 ; d1 < sizeof(displayWords51) ; d1++ )
    {
     lcd.print(displayWords51[d1]);
    }
    TN51++;
   }
   delay(200);
  }
  delay(700);
  lcd.setCursor(0,0);
  lcd.print("                ");

  int t=0;
  char opt1input='\0';
  while(opt1input=='\0')
  {
   opt1input = keypad.getKey();

   if(opt1input!='\0')
   {
    if(opt1input=='Q')
    {
```

```
                Serial.println("Exiting Option 1");
                break;
               }
               else
               {
                char tempregnum[12];

                if (opt1input!='E' && opt1input!='d' && opt1input!='u' && opt1input!='*'
&& opt1input!='#' && opt1input!='a' && opt1input!='b')
                {
                 Serial.println(opt1input);
                 tempregnum[t]=opt1input;
                 lcd.setCursor(t,0);
                 lcd.print(tempregnum[t]);
                 //Serial.println(tempregnum[t]);
                 t++;
                }

                else
                {
                 for(int i=0;i<12;i++)
                 {
                  regnum[i+2]=tempregnum[i];
                 }
                 Serial.println("New Number Registered");
                 Serial.print(tempregnum);
                 Serial.print(regnum);

                 int TN11=1;
                 char displayWords11[]="New Number Registered";
                 for(int d=0 ; d < sizeof(displayWords11)-1 ; d++ )
                 {
                  if(d<16)
                  {
                   lcd.setCursor(d,0);lcd.print(displayWords11[d]);
                  }
                  else
                  {
                   lcd.setCursor(0,0);
                   lcd.print("                ");
                   lcd.setCursor(0,0);
                   for(int d1=TN11 ; d1 < sizeof(displayWords11) ; d1++ )
                   {
                    lcd.print(displayWords11[d1]);
```

```
                }
                TN11++;
               }
               delay(200);
             }
             delay(700);
             lcd.setCursor(0,0);
             lcd.print("                ");
             lcd.setCursor(0,0);lcd.print("Exiting Option 1");
             delay(1000);
             lcd.setCursor(0,0);
             lcd.print("                ");
             break;
            }
           }
          }
          opt1input='\0';
         }break;
        }

        //***Option 2**************************************
        if (input=='2')
        {
         lcd.setCursor(0,0);lcd.print("                ");
         lcd.setCursor(0,1);lcd.print("                ");
         Serial.println("Entered Option 2");
         int TN5=1;
         char displayWords5[]="Entered Option 2";
         for(int d=0 ; d < sizeof(displayWords5)-1 ; d++ )
         {
          if(d<16)
          {
           lcd.setCursor(d,0);lcd.print(displayWords5[d]);
          }
          else
          {
           lcd.setCursor(0,0);
           lcd.print("                ");
           lcd.setCursor(0,0);
           for(int d1=TN5 ; d1 < sizeof(displayWords5) ; d1++ )
           {
            lcd.print(displayWords5[d1]);
           }
           TN5++;
```

```
  }
  delay(200);
}
delay(700);
lcd.setCursor(0,0);
lcd.print("                ");

Serial.println("Enter New Pincode then 'Enter' ");
int TN51=1;
char displayWords51[]="Enter New Pincode then 'Enter' ";
for(int d=0 ; d < sizeof(displayWords51)-1 ; d++ )
{
  if(d<16)
  {
    lcd.setCursor(d,0);lcd.print(displayWords51[d]);
  }
  else
  {
    lcd.setCursor(0,0);
    lcd.print("                ");
    lcd.setCursor(0,0);
    for(int d1=TN51 ; d1 < sizeof(displayWords51) ; d1++ )
    {
      lcd.print(displayWords51[d1]);
    }
    TN51++;
  }
  delay(200);
}
delay(700);
lcd.setCursor(0,0);
lcd.print("                ");

int t2=0;
char opt2input='\0';
while(opt2input=='\0')
{
  opt2input = keypad.getKey();

  if(opt2input!='\0')
  {
    if(opt2input=='Q')
    {
      Serial.println("Exiting Option 2");
```

```
  break;
 }
 else
 {
  char tempPincode[6];

  if (opt2input!='E')
  {
   Serial.println(opt2input);
   tempPincode[t2]=opt2input;

   lcd.setCursor(t2,0);lcd.print(tempPincode[t2]);
   //Serial.println(tempPincode[t2]);
   t2++;
  }

  else
  {
   for(int i=0;i<6;i++)
   {
    pincode[i]=tempPincode[i];
   }
   Serial.println("New PINCODE Registered");
   Serial.print(tempPincode);
   Serial.print(pincode);

   int TN21=1;
   char displayWords21[]="New PINCODE Registered";
   for(int d=0 ; d < sizeof(displayWords21)-1 ; d++ )
   {
    if(d<16)
    {
     lcd.setCursor(d,0);lcd.print(displayWords21[d]);
    }
    else
    {
     lcd.setCursor(0,0);
     lcd.print("                ");
     lcd.setCursor(0,0);
     for(int d1=TN21 ; d1 < sizeof(displayWords21) ; d1++ )
     {
      lcd.print(displayWords21[d1]);
     }
     TN21++;
```

```
            }
            delay(200);
          }
          delay(700);
          lcd.setCursor(0,0);
          lcd.print("                ");
          lcd.setCursor(0,0);lcd.print("Exiting Option 2");
          delay(1000);
          lcd.setCursor(0,0);
          lcd.print("                ");
          break;
        }
      }
    }
    opt2input='\0';
  }break;
}

//***Option 3**************************************
if (input=='3')
{
 lcd.setCursor(0,0);lcd.print("                ");
 lcd.setCursor(0,1);lcd.print("                ");
 Serial.println("Entered Option 3");

 int TN5=1;
 char displayWords5[]="Entered Option 3";
 for(int d=0 ; d < sizeof(displayWords5)-1 ; d++ )
 {
  if(d<16)
  {
   lcd.setCursor(d,0);lcd.print(displayWords5[d]);
  }
  else
  {
   lcd.setCursor(0,0);
   lcd.print("                ");
   lcd.setCursor(0,0);
   for(int d1=TN5 ; d1 < sizeof(displayWords5) ; d1++ )
   {
    lcd.print(displayWords5[d1]);
   }
   TN5++;
  }
```

```
            delay(200);
           }
           delay(700);
           lcd.setCursor(0,0);
           lcd.print("                ");

           Serial.println("Enrolling New FingerPrint ID");

           int TN52=1;
           char displayWords6[]="Enrolling New FingerPrint ID";
           for(int d=0 ; d < sizeof(displayWords6)-1 ; d++ )
           {
            if(d<16)
            {
             lcd.setCursor(d,0);lcd.print(displayWords6[d]);
            }
            else
            {
             lcd.setCursor(0,0);
             lcd.print("                ");
             lcd.setCursor(0,0);
             for(int d1=TN52 ; d1 < sizeof(displayWords6) ; d1++ )
             {
              lcd.print(displayWords6[d1]);
             }
             TN52++;
            }
            delay(200);
           }
           delay(700);
           lcd.setCursor(0,0);
           lcd.print("                ");

           Serial.println("Please type in the ID # (from 1 to 9) you want to save this
finger as...");

           int TN53=1;
           char displayWords7[]="Please type in the ID #";
           for(int d=0 ; d < sizeof(displayWords7)-1 ; d++ )
           {
            if(d<16)
            {
             lcd.setCursor(d,0);lcd.print(displayWords7[d]);
            }
```

```
  else
  {
   lcd.setCursor(0,0);
   lcd.print("                ");
   //lcd.setCursor(1,0);
   //lcd.print("                ");
   lcd.setCursor(0,0);
   for(int d1=TN53 ; d1 < sizeof(displayWords7) ; d1++ )
   {
    lcd.print(displayWords7[d1]);
   }
   TN53++;
  }
  delay(200);
}
delay(700);
lcd.setCursor(0,0);
lcd.print("                ");

//char id="";
char checker='F';
while(checker=='F')
{
 id=keypad.getKey();
 if(id)checker='T';
}

if(id!='0' && id!='Q')
{
 Serial.print("Enrolling ID #");
 Serial.println(id);

 lcd.setCursor(0,0);
 lcd.print("Enrolling ID #");
 lcd.setCursor(14,0);
 lcd.print(id);
 delay(1500);

 while (getFingerprintEnroll()==321);

 if(StoreChk==true)
 {
  Serial.print("Successfully added new fingerprint.");
```

```
  int TN61=1;
  char displayWords11[]="Successfully added new fingerprint.";
  for(int d=0 ; d < sizeof(displayWords11)-1 ; d++ )
  {
   if(d<16)
   {
    lcd.setCursor(d,0);lcd.print(displayWords11[d]);
   }
   else
   {
    lcd.setCursor(0,0);
    lcd.print("                ");
    lcd.setCursor(0,0);
    for(int d1=TN61 ; d1 < sizeof(displayWords11) ; d1++ )
    {
     lcd.print(displayWords11[d1]);
    }
    TN61++;
   }
   delay(200);
  }
  delay(700);
  lcd.setCursor(0,0);
  lcd.print("                ");
 }
 else if(StoreChk==false)
 {
  Serial.print("Fail to added new fingerprint. Try again...");

  int TN33=1;
  char displayWords9[]="Fail to added new fingerprint. Try again...";
  for(int d=0 ; d < sizeof(displayWords9)-1 ; d++ )
  {
   if(d<16)
   {
    lcd.setCursor(d,0);lcd.print(displayWords9[d]);
   }
   else
   {
    lcd.setCursor(0,0);
    lcd.print("                ");
    lcd.setCursor(0,0);
    for(int d1=TN33 ; d1 < sizeof(displayWords9) ; d1++ )
    {
```

```
          lcd.print(displayWords9[d1]);
         }
         TN33++;
        }
        delay(200);
      }
      delay(700);
      lcd.setCursor(0,0);
      lcd.print("                ");
     }

     break;
    }
    else if(id==0)
    {
     Serial.print("ID #0 not allowed, try again!");

     int TN54=1;
     char displayWords8[]="ID #0 not allowed, try again!";
     for(int d=0 ; d < sizeof(displayWords8)-1 ; d++ )
     {
      if(d<16)
      {
       lcd.setCursor(d,0);lcd.print(displayWords8[d]);
      }
      else
      {
       lcd.setCursor(0,0);
       lcd.print("                ");
       lcd.setCursor(0,0);
       for(int d1=TN54 ; d1 < sizeof(displayWords8) ; d1++ )
       {
        lcd.print(displayWords8[d1]);
       }
       TN54++;
      }
      delay(200);
     }
     delay(700);
     lcd.setCursor(0,0);
     lcd.print("                ");

     return;
    }
```

```
          else if(id=='Q')
          {
            Serial.println("Exiting Option 3");
          }


        }

        if (input=='Q')
        {
          lcd.setCursor(0,0);lcd.print("                ");
          lcd.setCursor(0,1);lcd.print("                ");
          Serial.println("Exited Menu Option");
          break;
        }

       }
       chk=0;
      }
      else
      {
         //Long beep
         tone(49,1000);
         delay(800);
         noTone(49);

       Serial.println(pinInput);
       Serial.println(pincode);
       lcd.setCursor(0,0);lcd.print("Wrong PINCODE");
       delay(1500);
       Serial.println("Failed to enter into menu option");
       lcd.setCursor(0,0);lcd.print("                ");

      }


    }
   }
}
else if(key!='\0' && key!='*')
{
  Serial.println(key);
  MenuEntries=0;
}
```

```
}
//*****For
tracking************************************************************
void TrackingFunction(void)
{
 if(sim808.attachGPS())
 {
    Serial.println("Open the GPS power success");
 }
 else
 {
    Serial.println("Open the GPS power failure");
 }
 Serial.println("Init Success, please send SMS message to me!");
 //Serial.println(sim808.isSMSunread());

 //********** Detecting unread SMS ************************
  messageIndex = sim808.isSMSunread();


  //********** At least, there is one UNREAD SMS ***********
  if (messageIndex > 0)
  {
    Serial.print("messageIndex: ");
    Serial.println(messageIndex);

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

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

    Serial.print("From number: ");
    Serial.println(phone);
    Serial.print("Datetime: ");
    Serial.println(datetime);
    Serial.print("Received Message: ");
    Serial.println(message);

    Serial.println(regnum);
    if(strcmp(regnum,phone) == 0)
    {
     if(strcmp(keyword,message) == 0)
     {
       GSMSerial.begin(9600);
```

```
//******** Initialize sim808 module *************
while(!sim808.init())
{
   Serial.print("Sim808 initialization error\r\n");

   lcd.setCursor(0,0);lcd.print("Sim808 error");
   lcd.setCursor(0,0);lcd.print("                ");
}
Serial.println("GSM Setup Done!!");
if(sim808.attachGPS())
{
 Serial.println("Open the GPS power success");
}
else
{
   Serial.println("Open the GPS power failure");
}
while(!sim808.getGPS())
{
 delay(1);
}
Serial.print(sim808.GPSdata.year);
Serial.print("/");
Serial.print(sim808.GPSdata.month);
Serial.print("/");
Serial.print(sim808.GPSdata.day);
Serial.print(" ");
Serial.print(sim808.GPSdata.hour);
Serial.print(":");
Serial.print(sim808.GPSdata.minute);
Serial.print(":");
Serial.print(sim808.GPSdata.second);
Serial.print(":");
Serial.println(sim808.GPSdata.centisecond);
Serial.print("latitude :");
Serial.println(sim808.GPSdata.lat);
Serial.print("longitude :");
Serial.println(sim808.GPSdata.lon);
Serial.print("speed_kph :");
Serial.println(sim808.GPSdata.speed_kph);
Serial.print("heading :");
Serial.println(sim808.GPSdata.heading);
Serial.println();
```

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

        dtostrf(la, 6, 6, lat); //put float value of la into char array of lat. 6 = number of
digits before decimal sign. 2 = number of digits after the decimal sign.
        dtostrf(lo, 6, 6, lon); //put float value of lo into char array of lon
        dtostrf(ws, 6, 2, wspeed);  //put float value of ws into char array of wspeed

        sprintf(MESSAGE, "Latitude : %s\nLongitude : %s\nWind Speed : %s
kph\nhttp://maps.google.com/maps?q=%s,%s\n", lat, lon, wspeed, lat, lon);


        Serial.println("Sim808 init success");
        Serial.println("Start to send message ...");

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

        sim808.sendSMS(phone,MESSAGE);
        //for(i=messageIndex; i>0; i--)
        //{
          sim808.deleteSMS(messageIndex);
          //Serial.println(i);
        //}

        //************* Turn off the GPS power ************
        sim808.detachGPS();
      }
      else
      {
      sprintf(MESSAGE, "Wrong Keyword!!!");

      Serial.println("Sim808 init success");
      Serial.println("Start to send message ...");

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

      sim808.sendSMS(phone,MESSAGE);
      for(i=messageIndex; i>0; i--)
      {
        sim808.deleteSMS(i);
        Serial.println(i);
```

```
      }

      //************* Turn off the GPS power ************
      //sim808.detachGPS();
      }
    }
    else
    {
      sprintf(MESSAGE, "Wrong phone!!!");

      Serial.println("Sim808 init success");
      Serial.println("Start to send message ...");

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

      sim808.sendSMS(phone,MESSAGE);
      for(i=messageIndex; i>0; i--)
      {
       sim808.deleteSMS(i);
       Serial.println(i);
      }

      //************* Turn off the GPS power ************
      sim808.detachGPS();
    }

  }
}
//*****************************************************************************
******

//***FingerPrintOption**************************************************
******

//Function 1
uint8_t readnumber(void)
{
 uint8_t num = 0;

 while (num == 0)
 {
  while (! Serial.available());
  num = Serial.parseInt();
```

```
  }
  return num;
}

//Function 2
uint8_t getFingerprintEnroll()
{
  int p = -1;
  Serial.print("Waiting for valid finger to enroll as ID #");
  Serial.println(id);

     int TN321=1;
     char displayWords321[]="Waiting for valid finger to enroll";
     for(int d=0 ; d < sizeof(displayWords321)-1 ; d++ )
     {
       if(d<16)
       {
         lcd.setCursor(d,0);lcd.print(displayWords321[d]);
       }
       else
       {
         lcd.setCursor(0,0);
         lcd.print("                ");
         lcd.setCursor(0,0);
         for(int d1=TN321 ; d1 < sizeof(displayWords321) ; d1++ )
         {
           lcd.print(displayWords321[d1]);
         }
         TN321++;
       }
       delay(200);
     }
     delay(3000);
     //x`lcd.setCursor(0,0);lcd.print("                ");

  //delay(2000);

  while (p != FINGERPRINT_OK)
  {
    p = finger.getImage();
    switch (p)
    {
      case FINGERPRINT_OK:
        Serial.println("Image taken");
```

```
      lcd.setCursor(0,0);lcd.print("                ");
      lcd.setCursor(0,0);lcd.print("Image taken");
      delay(1500);
      lcd.setCursor(0,0);lcd.print("                ");
      break;
    case FINGERPRINT_NOFINGER:
      Serial.print("");
      break;
    case FINGERPRINT_PACKETRECIEVEERR:
      Serial.println("Communication error");
      lcd.setCursor(0,0);lcd.print("                ");
      lcd.setCursor(0,0);lcd.print("Comm error");
      delay(1500);
      lcd.setCursor(0,0);lcd.print("                ");
      break;
    case FINGERPRINT_IMAGEFAIL:
      Serial.println("Imaging error");
      lcd.setCursor(0,0);lcd.print("                ");
      lcd.setCursor(0,0);lcd.print("Imaging error");
      delay(1500);
      lcd.setCursor(0,0);lcd.print("                ");
      break;
    default:
      Serial.println("Unknown error");
      lcd.setCursor(0,0);lcd.print("                ");
      lcd.setCursor(0,0);lcd.print("Unknown error");
      delay(1500);
      lcd.setCursor(0,0);lcd.print("                ");
      break;
  }
}
// OK success!

p = finger.image2Tz(1);
switch (p)
{
  case FINGERPRINT_OK:
    Serial.println("Image converted");
    lcd.setCursor(0,0);lcd.print("                ");
    lcd.setCursor(0,0);lcd.print("Image converted");
    delay(1500);
    lcd.setCursor(0,0);lcd.print("                ");
    break;
  case FINGERPRINT_IMAGEMESS:
```

```
    Serial.println("Image too messy");
    lcd.setCursor(0,0);lcd.print("                ");
    lcd.setCursor(0,0);lcd.print("Image too messy");
    delay(1500);
    lcd.setCursor(0,0);lcd.print("                ");
    return p;
  case FINGERPRINT_PACKETRECIEVEERR:
    Serial.println("Communication error");
    lcd.setCursor(0,0);lcd.print("                ");
    lcd.setCursor(0,0);lcd.print("Com error");
    delay(1500);
    lcd.setCursor(0,0);lcd.print("                ");
    return p;
  case FINGERPRINT_FEATUREFAIL:
    Serial.println("Could not find fingerprint features");
    lcd.setCursor(0,0);lcd.print("                ");
    lcd.setCursor(0,0);lcd.print("Could not find fingerprint features");
    delay(1500);
    lcd.setCursor(0,0);lcd.print("                ");
    return p;
  case FINGERPRINT_INVALIDIMAGE:
    Serial.println("Could not find fingerprint features");
    lcd.setCursor(0,0);lcd.print("                ");
    lcd.setCursor(0,0);lcd.print("Could not find");
    delay(1500);
    lcd.setCursor(0,0);lcd.print("                ");
    return p;
  default:
    Serial.println("Unknown error");
    lcd.setCursor(0,0);lcd.print("                ");
    lcd.setCursor(0,0);lcd.print("Unknown error");
    delay(1500);
    lcd.setCursor(0,0);lcd.print("                ");
    return p;
}

Serial.println("Remove finger");
lcd.setCursor(0,0);lcd.print("                ");
lcd.setCursor(0,0);lcd.print("Remove finger");
delay(2000);
lcd.setCursor(0,0);lcd.print("                ");
p = 0;
while (p != FINGERPRINT_NOFINGER)
{
```

```
  p = finger.getImage();
}
Serial.print("ID ");
Serial.println(id);
p = -1;
Serial.println("Place same finger again");

int TN332=1;
          char displayWords322[]="Place same finger again";
          for(int d=0 ; d < sizeof(displayWords322)-1 ; d++ )
          {
           if(d<16)
           {
            lcd.setCursor(d,0);lcd.print(displayWords322[d]);
           }
           else
           {
            lcd.setCursor(0,0);
            lcd.print("                ");
            lcd.setCursor(0,0);
            for(int d1=TN332 ; d1 < sizeof(displayWords322) ; d1++ )
            {
             lcd.print(displayWords322[d1]);
            }
            TN332++;
           }
           delay(200);
          }
          delay(700);
          lcd.setCursor(0,0);
          lcd.print("                ");

while (p != FINGERPRINT_OK)
{
 p = finger.getImage();
 switch (p)
 {
  case FINGERPRINT_OK:
    Serial.println("Image taken");
    lcd.setCursor(0,0);lcd.print("                ");
    lcd.setCursor(0,0);lcd.print("Image taken");
    delay(1000);
    lcd.setCursor(0,0);lcd.print("                ");
    break;
```

```
    case FINGERPRINT_NOFINGER:
      Serial.print("");
      break;
    case FINGERPRINT_PACKETRECIEVEERR:
      Serial.println("Communication error");
      lcd.setCursor(0,0);lcd.print("                ");
      lcd.setCursor(0,0);lcd.print("Comm error");
      delay(1000);
      lcd.setCursor(0,0);lcd.print("                ");
      break;
    case FINGERPRINT_IMAGEFAIL:
      Serial.println("Imaging error");
      lcd.setCursor(0,0);lcd.print("                ");
      lcd.setCursor(0,0);lcd.print("Imaging error");
      delay(1000);
      lcd.setCursor(0,0);lcd.print("                ");
      break;
    default:
      Serial.println("Unknown error");
      lcd.setCursor(0,0);lcd.print("                ");
      lcd.setCursor(0,0);lcd.print("Unknown error");
      delay(1000);
      lcd.setCursor(0,0);lcd.print("                ");
      break;
  }
}

// OK success!

p = finger.image2Tz(2);
switch (p)
{
  case FINGERPRINT_OK:
    Serial.println("Image converted");
      lcd.setCursor(0,0);lcd.print("                ");
      lcd.setCursor(0,0);lcd.print("Image converted");
      delay(1000);
      lcd.setCursor(0,0);lcd.print("                ");
    break;
  case FINGERPRINT_IMAGEMESS:
      Serial.println("Image too messy");
      lcd.setCursor(0,0);lcd.print("                ");
      lcd.setCursor(0,0);lcd.print("Image too messy");
      delay(1000);
```

```
      lcd.setCursor(0,0);lcd.print("                ");
    return p;
  case FINGERPRINT_PACKETRECIEVEERR:
    Serial.println("Communication error");
      lcd.setCursor(0,0);lcd.print("                ");
      lcd.setCursor(0,0);lcd.print("Comm error");
      delay(1000);
      lcd.setCursor(0,0);lcd.print("                ");
    return p;
  case FINGERPRINT_FEATUREFAIL:
    Serial.println("Could not find fingerprint features");
      lcd.setCursor(0,0);lcd.print("                ");
      lcd.setCursor(0,0);lcd.print("Could not find fingerprint features");
      delay(1000);
      lcd.setCursor(0,0);lcd.print("                ");
    return p;
  case FINGERPRINT_INVALIDIMAGE:
    Serial.println("Could not find fingerprint features");
      lcd.setCursor(0,0);lcd.print("                ");
      lcd.setCursor(0,0);lcd.print("Could not find fingerprint features");
      delay(1000);
      lcd.setCursor(0,0);lcd.print("                ");
    return p;
  default:
    Serial.println("Unknown error");
      lcd.setCursor(0,0);lcd.print("                ");
      Serial.println("Unknown error");
      delay(1000);
      lcd.setCursor(0,0);lcd.print("                ");
    return p;
}
// OK converted!

Serial.print("Creating model for ID #");
Serial.println(id);

p = finger.createModel();
if (p == FINGERPRINT_OK)
{
  Serial.println("Prints matched!");
  StoreChk = true;
  lcd.setCursor(0,0);lcd.print("                ");
  lcd.setCursor(0,0);lcd.print("Prints matched!");
  delay(1000);
```

```
  lcd.setCursor(0,0);lcd.print("                ");

 }
 else if (p == FINGERPRINT_PACKETRECIEVEERR)
 {
  Serial.println("Communication error");
  StoreChk = false;
  lcd.setCursor(0,0);lcd.print("                ");
  lcd.setCursor(0,0);lcd.print("Comm error");
  delay(1000);
  lcd.setCursor(0,0);lcd.print("                ");
  return p;
 }
 else if (p == FINGERPRINT_ENROLLMISMATCH)
 {
  Serial.println("Fingerprints did not match");
  StoreChk = false;
  lcd.setCursor(0,0);lcd.print("                ");
  lcd.setCursor(0,0);lcd.print("Did not match");
  delay(1000);
  lcd.setCursor(0,0);lcd.print("                ");
  return p;
 }
 else
 {
  Serial.println("Unknown error");
  StoreChk = false;
  lcd.setCursor(0,0);lcd.print("                ");
  lcd.setCursor(0,0);lcd.print("Unknown error");
  delay(1000);
  lcd.setCursor(0,0);lcd.print("                ");
  return p;
  return p;
 }

 Serial.print("ID ");
 Serial.println(id);
 p = finger.storeModel(id);
 if (p == FINGERPRINT_OK)
 {
  Serial.println("Stored!");
  StoreChk = true;
 }
 else if (p == FINGERPRINT_PACKETRECIEVEERR)
```

```
  {
   Serial.println("Communication error");
   StoreChk = false;
   return p;
  }
  else if (p == FINGERPRINT_BADLOCATION)
  {
   Serial.println("Could not store in that location");
   StoreChk = false;
   return p;
  }
  else if (p == FINGERPRINT_FLASHERR)
  {
   Serial.println("Error writing to flash");
   StoreChk = false;
   return p;
  }
  else
  {
   Serial.println("Unknown error");
   StoreChk = false;
   return p;
  }
  //Serial.println(p);
  int r=321;
  return r;
}

int getFingerprintIDez()
{
 //delay(3000);

 uint8_t p =finger.getImage();

 while(p!=FINGERPRINT_OK)
 {
  p = finger.getImage();
 }

 p = finger.image2Tz();
 if (p != FINGERPRINT_OK)
  return -1;

 p = finger.fingerFastSearch();
```

```
  if (p != FINGERPRINT_OK)
    return -1;

  // found a match!
  Serial.print("Found ID #");
  Serial.print(finger.fingerID);
  Serial.print(" with confidence of ");
  Serial.println(finger.confidence);

  boolean FM = true;
  return FM;
}
//*****************************************************************************
******

//***For Alert Tracking
Function*************************************************

void AlertTrackingFunction(void)
{
  GSMSerial.begin(9600);
  //******** Initialize sim808 module *************
  while(!sim808.init())
  {
     Serial.print("Sim808 initialization error\r\n");

     lcd.setCursor(0,0);lcd.print("Sim808 error");
     lcd.setCursor(0,0);lcd.print("                ");
  }
  Serial.println("GSM Setup Done!!");
  //************* Turn on the GPS power ************
  if (sim808.attachGPS())
  {
   Serial.println("Open the GPS power success");
  }
  else
  {
   Serial.println("Open the GPS power failure");
  }

  while(!sim808.getGPS())
    {
     delay(1);
    }
```

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

  float la = sim808.GPSdata.lat;
  float lo = sim808.GPSdata.lon;
  float ws = sim808.GPSdata.speed_kph;

  dtostrf(la, 6, 6, lat); //put float value of la into char array of lat. 6 = number of digits
before decimal sign. 2 = number of digits after the decimal sign.
  dtostrf(lo, 6, 6, lon); //put float value of lo into char array of lon
  dtostrf(ws, 6, 2, wspeed);  //put float value of ws into char array of wspeed

  sprintf(MESSAGE, "ALERT MESSAGE!!!\nYour vehicle is being used by unknown
user.\nYour vehicle's current
location.\nLatitude : %s\nLongitude : %s\nhttp://maps.google.com/maps?q=%s,%s\n",
lat, lon, lat, lon);

  Serial.println("Sim808 init success");
  Serial.println("Start to send message ...");
  sim808.sendSMS(regnum, MESSAGE);
  Serial.println(MESSAGE);
  Serial.println(regnum);

  //************* Turn off the GPS power ************
```

```
sim808.detachGPS();

delay(2000);

if (sim808.attachGPS())
{
  Serial.println("Open the GPS power success");
}
else
{
  Serial.println("Open the GPS power failure");
}

Serial.println("Please send SMS message to me!");
Serial.println(sim808.isSMSunread());

//********** Detecting unread SMS ************************
messageIndex = sim808.isSMSunread();

//********** Waits for detecting unread SMS ************************
while (messageIndex == -1)
{
  messageIndex = sim808.isSMSunread();
  Serial.println(messageIndex);
  delay(1000);
}

//********** At least, there is one UNREAD SMS ***********
if (messageIndex > 0)
{
  Serial.print("messageIndex: ");
  Serial.println(messageIndex);

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

  //********** Reading received message from Sim**********

  Serial.print("From number: ");
  Serial.println(phone);
  Serial.print("Datetime: ");
  Serial.println(datetime);
  Serial.print("Received Message: ");
  Serial.println(message);
```

```
int o=0;
char AllowCmd[] = "allow";
for(int i=0; i<sizeof(AllowCmd); i++)
{
 if(AllowCmd[i]==message[i])
 {
  o++;
 }
}
Serial.println(sizeof(AllowCmd));
Serial.println(o);
if (o==6)
{
 Serial.println("Allowed");
 int menuPinCounter = 0;
 char pinInput[7];
 pinInput[6] = '\0';
 Serial.println("Enter PinCode");


 int ps=0;
 while(menuPinCounter<6)
 {
  //keypad.waitForKey();
  while(true)
  {
    char input = keypad.getKey();
    pinInput[menuPinCounter]=input;
    if (input)Serial.println(input);
    if (input)
    {
      lcd.setCursor(ps++,0);
      lcd.print("*");

    }
    if (input){break;}
  }
  menuPinCounter++;
 }
 delay(500);
 int chk=0;
 for(int i = 0; i < 6; i++)
 {
  if(pincode[i]==pinInput[i])
```

```
      chk++;
    }
    if(chk==6)
    {
     lcd.setCursor(0,0);lcd.print("                ");
       tone(49,1000); //Short Beep
       delay(200);
       noTone(49);
     digitalWrite(4, HIGH);
     Serial.println("Ignition turned ON");
     lcd.setCursor(0,0);lcd.print("Ignition ON");
     delay(1500);
     lcd.setCursor(0,0);lcd.print("                ");
     digitalWrite(4, LOW);
    }
    else{
     tone(49,1000); //Short Beep
     delay(800);
     noTone(49);
     Serial.println("Wrong PINCODE");
     lcd.setCursor(0,0);lcd.print("Wrong PINCODE");
    }
   }
   else
   {
    return -1;
   }
  }

  //************* Turn off the GPS power ************
  sim808.detachGPS();
}

//*****************************************************************************
******
```