Faculty of Electrical Technology and Engineering

DEVELOPMENT OF SMART VEHICLE IGNITION SYSTEM USING FINGERPRINT

NURUL HASYA BINTI MASLIZAM

Bachelor of Electrical Engineering Technology with Honours

2023

DEVELOPMENT OF SMART VEHICLE IGNITION SYSTEM USING FINGERPRINT

NURUL HASYA BINTI MASLIZAM

A project report submitted
in partial fulfillment of the requirements for the degree of
Bachelor of Electrical Engineering Technology with Honours

Faculty of Electrical and Electronic Engineering Technology

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2023

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

FAKULTI TEKNOLOGI KEJUTERAAN ELEKTRIK DAN ELEKTRONIK

BORANG PENGESAHAN STATUS LAPORAN

PROJEK SARJANA MUDA II

Tajuk Projek : Development of Smart Vehicle Ignition System using Fingerprint

Sesi Pengajian : 2023/2024

Saya Nurul Hasya Binti Maslizam mengaku membenarkan laporan Projek Sarjana Muda ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Laporan adalah hakmilik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan laporan ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. Sila tandakan (9):

□ **SULIT*** (Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)

□ **TERHAD*** (Mengandungi maklumat terhad yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)

☑ **TIDAK TERHAD**

Disahkan oleh:

(TANDATANGAN PENULIS)

Alamat Tetap: No 122 Laluan Kelebang Ria 2, Taman Klebang Ria, 31200 Chemor, Perak.

(COP DAN TANDATANGAN PENYELIA)

Dr. Nor Hafizah Binti Hussin
Pensyarah
Fakulti Teknologi Kejuruteraan
Elektrik Dan Elektronik
Universiti Teknikal Malaysia Melaka

Tarikh: 14/1/2024

Tarikh:

*CATATAN: Jika laporan ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh laporan ini perlu dikelaskan sebagai SULIT atau TERHAD.

DECLARATION

I declare that this project report entitled “Development of Smart Vehicle Ignition System using Fingerprint” is the result of my own research except as cited in the references. The project report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature : Hasya.

Student : Nurul Hasya Binti Maslizam

Name Date : 14/1/2024

APPROVAL

I hereby declare that I have checked this project report and in my opinion, this project report is adequate in terms of scope and quality for the award of the degree of Bachelor of Electrical Engineering Technology with Honours.

Signature :

Supervisor Name : Dr. Nor Hafizah Binti Hussin

Date :

Signature :

Co-Supervisor :

Name (if any)

Date :

DEDICATION

I am thankful to Universiti Teknikal Malaysia Melaka for providing me with an opportunity to demonstrate my abilities in a bachelor's degree project under the guidance of my supervisor, Madam Nor Hafizah Binti Hussin.

Our sincere gratitude goes out to the University for providing the funding necessary to carry out the research. Not to mention, I want to express my gratitude to my friends for helping me along the way and providing me with guidance and encouragement.

Finally, but just as importantly, I want to express my gratitude to my family for their unwavering emotional support while I finished my project.

ABSTRACT

A project to stop vehicles thefts is the fingerprint system. Therefore, this effort or this initiative will lessen the probability of thefts occurring in Malaysia. This concept was developed based on the issue statement we gathered, which is that thieves may now easily steal motorcycles and cars. The goal of this project is to improve the safety system, make it simpler to start the vehicles, and lower the current rate of motorcycle theft. In essence, we are creating a new starter for a bike and cars that was previously utilized. They can start their vehicles with just their fingerprint. The owner can only use their own vehicle, making this method safer. The initial step in the methodology for this study was to create some questionnaires and deliver them to bikers and drivers in order to collect their feedback on the safety aspects of their vehicles. This assignment demands us to learn about electronic devices and circuits in our preliminary research. Therefore, research is necessary to create a circuit that will allow the fingerprint sensor to function properly. In this project, GPS and GSM will both be used to track a vehicle. With a few adjustments, this vehicle tracking system can also serve as an accident detection and alert system. Tracking is a technique in which we use GPS coordinates (latitude and longitude) to track the location of the vehicle. Finally, by introducing a different and intriguing method of starting your vehicles, this concept will help to reduce the number of stolen cars and motorcycles.

ABSTRAK

Satu projek untuk menghentikan pencurian kenderaan adalah sistem cap jari. Oleh itu, usaha atau inisiatif ni dapat akan mengurangkan kemungkinan berlakunya pencurian di Malaysia. Kami membangunkan konsep ini berdasarkan kenyataan masalah yang kami kumpulkan, iaitu pencuri sekarang mudah mencuri motosikal dan kereta. Tujuan projek ini adalah untuk meningkatkan sistem keselamatan, menjadikan penghidupan kenderaan lebih mudah, dan mengurangkan kadar pencurian motosikal semasa. Pada dasarnya, projek ini ingin penghidup baru untuk motosikal dan kereta yang sebelum ini digunakan. Mereka hanya boleh menghidupkan kenderaan mereka dengan cap jari sendiri, menjadikan kaedah ini lebih selamat. Langkah pertama dalam metodologi kajian ini adalah untuk membuat beberapa soal selidik dan mengedarkannya kepada pemandu motosikal dan pemandu kereta untuk mengumpulkan maklum balas mereka tentang aspek keselamatan kenderaan mereka. Tugasan ini memerlukan kita untuk mempelajari tentang peranti elektronik dan litar dalam penyelidikan awal kami. Oleh itu, penyelidikan adalah penting untuk mencipta litar yang membolehkan pengesan cap jari berfungsi dengan baik. Dalam projek ini, GPS dan GSM akan digunakan untuk mengesan kenderaan. Dengan beberapa penyesuaian, sistem pengesanan kenderaan ini juga boleh berfungsi sebagai sistem pengesanan kemalangan dan penghantaran isyarat. Pengesanan adalah teknik di mana kami menggunakan koordinat GPS (latitud dan longitud) untuk mengesan lokasi kenderaan. Akhirnya, dengan memperkenalkan kaedah yang berbeza dan menarik untuk menghidupkan kenderaan, konsep ini diharap akan membantu mengurangkan bilangan kenderaan dan motosikal yang dicuri.

TABLE OF CONTENT

LIST OF TABLE

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Background

Development of smart vehicle starter using fingerprint project. This concept draws inspiration from the common fingerprint sensor found in smartphones nowadays. This method is safer than utilizing the outdated pin lock or pattern lock methods, and it prevents people from being able to unlock phone if the sensor cannot read the fingerprint. If the phone is not unlock by using fingerprint, no one else can use it. Due to the recent surge in vehicles theft instances, this application for vehicles that can be use to motorcycles or car has been submitted. This situation is getting worse and only occurs with super bikes and cars; it does not happen with moped bikes.

Because it affects the customers' money, the bike and car maker shouldn't ignore this issue. People should to attempt to address the issue with their safety features as soon as they become aware of it. Therefore, this project are moving through with the plan to create a fingerprint sensor for motorcycles and cars. People would not need to use the key is anymore because this fingerprint sensor will start the vehicles. This approach is more practical and up to date, similar to the "key-less entry" technology that is now used in cars. An alarm sound will be produced if someone tries to damage the sensor, alerting everyone close that the bike or car is in danger. Consequently, this method is superior to the current safety features. Aside from that, an alarm will go off, which can make the would-be thief feel terrified and panicked and ensure that the vehicles is in safe hands.

An electronic device installed in a car that allows the owner or a third party to follow the location of the car is called a vehicle tracking system. This study suggested developing a GPS and GSM-based car tracking system, which would be the most affordable method of tracking vehicles and serve as an anti-theft system. It is an embedded technology that uses the Global Positioning technology (GPS) to track and position any vehicle.

1.2 Problem Statement

Nowadays, there is a lot of emphasis placed on car security because of the rise in vehicle thefts. Handling the keys to the automobiles is still another problem. Keys must be carried, and losing or misplacing them will be a severe problem. Here, this project offer the use of a fingerprint-authenticated vehicle starter system as a solution to this issue. The technology offers a safe and convenient way to start and stop a vehicle's engine. To start the vehicle, the user only needs to scan their finger; a key is not required. The car can only be started by people who are authorized by the system. By scanning their fingerprints, users can first register on the system. Multiple people can register as authorized users on the system. The system looks for users to scan when in monitoring mode. When a user is scanned, the system determines whether they are an authorized user and only allows them to start the vehicle. An AT Mega 32 microcontroller is used in this case. The microcontroller is attached to the fingerprint sensor, and there is also an buzzer, rocket switch, and a LED as an indicator for vehicle engine. The motor is used to show how to start a car. This system uses a fingerprint-based approach to automate both vehicle security and access control. Therefore, fingerprint security is essential, and it cannot be the same for any two people. Therefore, our car will be locked up.

The company's vehicles and equipment are some of its most valuable assets. So it should come as no surprise that recovering stolen property is one of the top benefits of GPS monitoring. When GPS monitoring is installed on the cars and other assets, users can easily watch the locations of their vehicles and other assets, build calendar templates for intended use, and immediately notice strange or unauthorized use. Get prompt alerts when a vehicle or piece of equipment leaves the designated space or the designated operating window. The recovery of a stolen car with the use of location monitoring can save money on insurance and expensive replacement charges.

The safety of its drivers is without a doubt the organization's top priority. The initial stage in this process is having well-maintained cars, but it should also focus on recording driver behaviour and ensuring that safe driving practise are being followed.

Most fleet tracking businesses provide digital maintenance programme that enable customers to schedule maintenance with automated notifications based on odometer readings or scheduled repairs. It can also create digital processes for inspections and maintenance jobs that enable drivers and maintenance staff to submit issues that are urgently fixable.

The majority of GPS trackers also have an accelerometer, which may warn users when a driver exhibits risky driving behaviour including speeding, abrupt braking, harsh acceleration, or hard turning. When dash cams are AI-enabled, fleet managers may look into inattentive driving, tailgating, and red-light violations even more thoroughly. With the aid of incentive and recognition programme that fleets may develop utilizing this data, the top drivers can be recognized and other drivers can be motivated to do better.

1.3 Project Objective

The following lists of objectives:

i. To design the fingerprint based vehicles starting system.

ii. To develop an operating prototype of a fingerprint based vehicle starting system.

iii. To analyze the performance of the developed prototype.

1.4 Scope of Project

1.4.1 Finger Print Image Extraction

The library includes a technique for extracting fingerprint images from the image buffer of the module. If the process begins, it issues the module the proper command (based on the datasheet) and, if successful, receives an acknowledgement packet. The module then starts transmitting the data packets that make up the fingerprint image. An 8-bit, 256 x 288 grayscale bitmap represents the image. Accordingly, each pixel is represented by an 8-bit value (a byte) between 0 and 255, where 0 stands for "black," and 255 for "white." There are different shades of grey between these values. The image is 73728 pixels or bytes in size and has a width of 256 pixels and a height of 288 pixels.The host computer/MCU has to receive all 73728 bytes (one byte for each pixel) from the module in order to put the image back together. Instead, the module sends 2 pixels each byte to speed up transmission. In other words, the module creates an 8-bit value that comprises the data for both pixels by combining the high nibble (4 bits) of one pixel and the high nibble of an adjacent pixel in the same row. Normally, each pixel equals one byte. By doing this, the module loses portion of the image data and communicates just 73728/2 = 36864 bytes to the MCU.

1.4.2 Arduino

One of the platforms used for this project is Arduino. An advanced programming designer can use the Arduino code to converge with the present programming language libraries thanks to a software feature that allows for library expansion and modification. It is a fantastic tool for people of all skill levels. In Arduino, there is a physical programmable circuit board as well as programming.It continues to run on the computer that is used to create and transmit computer code to the physical world. A finger on a button, running a motor, turning on an LED, and spreading anything online are just a few of the capabilities of Arduino. Additionally, because it can use it with a USB cable, Arduino doesn't require any additional hardware to put fresh code onto the board. The Arduino Uno and Arduino Mega are the two that are used the most. The Arduino IDE, which employs a simple version of C++, is used to programme an Arduino. This makes learning the programme less difficult.

1.4.3 GSM Network

GSM provides recommendations rather than requests. The functionality and interface requirements are outlined in great detail in the GSM standards, although hardware is not included. The three primary systems that make up the GSM network are the switching system (SS), base station system (BSS), and the GSM system. The time division multiple access (TDMA) approach was used in the development of the GSM system as a digital mechanism for communication. Before being broadcast across a channel system that can handle data speeds of 64 kbps to 120 Mbps, the data is digitalized and compressed using a GSM. Using two different client data streams, each in a different temporal window.

1.4.4 GPS Detector

The global positioning system is built on the 'trilateration' mathematical idea. Utilising satellite distance information, the location is computed. The four satellites seen in the image can be used to locate the receiver on Earth. The space segment, the control segment, and the user segment are the three components that make up the GPS system. GPS satellites, military users, and civilian users, in that order, make use of the space segment, control segment, and user segment. This is used by the vast majority of the population for mapping, agriculture, transportation, and the management of natural resources.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

The term "biometric" refers to the use of measurable biological characteristics, such as fingerprints or iris patterns, to identify a person to an electronic system. After they have been made, these steps can be utilized to authenticate a person or user. By comparing the collected biometric data to a previous template, this is achieved. For economic operations, it was customary in ancient Egypt to identify persons by their physical traits or qualities, such as scars, eye and hair colour, height, etc. Biometric is a rapidly evolving field of technology that has been used extensively in forensics, such as criminal identification and prison security, and has the potential to be applied widely across a wide range of fields.

2.2 Evolution of Car Key

The government put pressure on automakers to increase vehicle security in the 1990s. Hotwiring, duplicating the key, and other methods made it incredibly simple for thieves to steal automobiles. The installation of immobiliser chips based on RFID technology in key fobs is the first application of cryptography in automobiles. St. George Evans and Edward Birkenbeuel created and patented the first immobiliser alarm system in 1919 [1]. In 1995, many automakers began to produce vehicles with immobiliser chips. Since January 1, 1998, immobilisers have been required in all new vehicles sold in Germany, and since January 2007, in Canada [2]. Car theft significantly decreased once immobilisers were installed. As automakers add more software and hardware to cars for convenience and security, criminals figure out ways to take advantage of technological flaws to steal cars without a key.

Table 2.1 Types of Key System [3]

Denomination	**Entry**	**Start engine**
Physical Key	Physical Key	Physical Key
Physical Key with immobilizer	Physical Key	Physical Key + RFID
Remote Keyless Entry System	Remote active (press button)	Physical Key + RFID
Remote Keyless Ignition System	Remote passive	Remote passive

2.2.1 Physical keys

The Chrysler Corporation debuted the key as an ignition key for cars in 1949, according to a Popular Science article [4]. In the past, starting an automobile required pressing both the starter and ignition buttons separately, as shown in the illustration. In addition to being convenient for the driver, the key served as a safeguard against kids starting cars. Although the key improved safety a little, automobiles could still be readily hot-wired and hijacked. Additionally, metallic keys were simple to copy, giving an attacker who had previously touched the key access to the car.

Figure 2.1 Car Key

2.2.2 Physical keys with imobilizers

A metal key and an RFID transponder are both included in an immobilizer-equipped key's plastic body. Communication between the immobiliser and steering column enables the fuel injection system.

The scanner sends out a signal that the immobiliser, a passive device, uses as electromagnetic induction. Since the car cannot start until the RFID chip has properly verified it, this technique was created to avoid car thefts like hot wire.

Immobilizers come in two varieties: electronic and cryptographic [5].

The first generation of transponders with a static signature was called electronic immobilisers. Despite lacking encryption, they significantly reduced automobile theft, as shown in Figure X. The subsequent immobilizer employs cryptographic methods in order to deter attackers from quickly replicating the electronic immobiliser.

Figure 2.2 Car Key with immobilizer

2.2.3 Remote Keyless Entry (RKE) Systems

When a door is locked or unlocked, the trunk is opened, or the car alarm is disarmed, radio waves are sent to the vehicle through the Remote Keyless Entry System, or RKE. While more recent devices employ radio waves, older generations utilised the infrared spectrum.

The standard operating frequency of RKE systems is 315 MHz for North America and 433.92 MHz for Europe and Asia, with a transmission range of 10 to 100 metres [6]. This is an active system since the gadget has a power supply and transmits signals to the receiver within the car. The first automobile with a central locking system was the 1982 Renault Fuego [7].

A typical RKE system (Figure 5) includes a microcontroller in the key or key fob. The microcontroller is activated by a pushbutton on the key. It then sends a stream of 64 or 128 bits to the key's RF transmitter, modulates the carrier, and radiates the signal through a simple printed-circuit loop antenna. The car is unlocked by this procedure. A loop antenna is inefficient despite being easily made and widely used [8].

A microprocessor decodes the data and sends the appropriate signal to start the engine or open the door when it is picked up by an RF receiver in the automobile. The digital data stream normally consists of a data preamble, a common code, a few check bits, and a "rolling code" that is changed after each use to ensure the security of the vehicle. It is transmitted at a speed of between 2.4 and 20 kbps. Because of this, a hacker cannot intercept the signal simply once and get access several times [8].

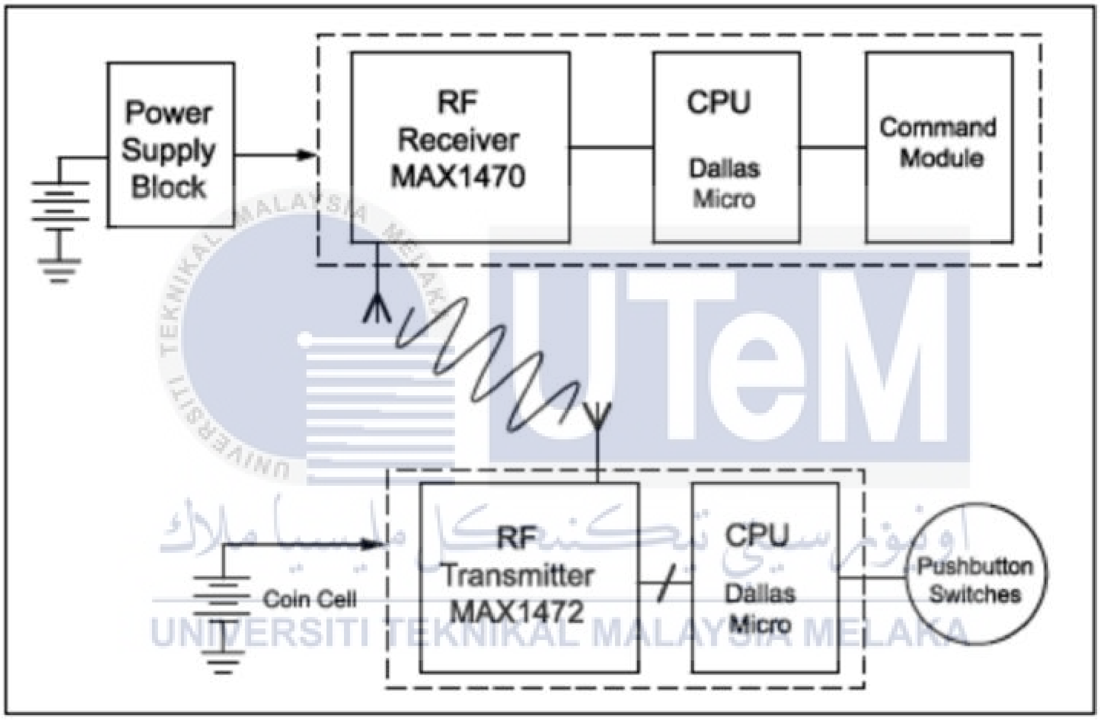


Figure 2.3 A key fob circuit (lower figure) transmits to a receiver in the car (upper diagram) in an RKE system [7].

2.2.4 RKI, or Remote Keyless Ignition Systems

The same features of RKEs are provided by passive keyless entry and start systems (PKES), commonly referred to as smart keys or remote keyless ignition systems (RKI), but they do not need a metal item to start the car. Most vehicles with RKI systems allow the owner to unlock doors without pressing any buttons on the key by touching a sensor on the door handle while keeping the key in their pocket. While some automobiles merely require the key fob to be inside the vehicle, others require it to be put into the ignition slot.

Security issues might arise from a car's "automatic" unlocking or ignition since an attacker could take the car while the owner is nearby (for instance, when the owner is filling up the petrol tank or loading the trunk). The default setting of the key uses two channels. After connecting with the car through inductive coupling on the LF channel (120-135 kHz) in the vicinity of 1 to 2 metres, the key will replay back on the second UHF channel (315-433 MHz), even in the neighbourhood of 50 to 100 metres [4].

A car initially sends out LF signals regularly until the key sends out a UHF signal admitting its presence; after that, the car sends out an LF signal with its Id number and a challenge; and finally, the key sends out a UHF signal with a response. The battery-depleted mode, which works both ways, uses the passive component on the key. A metallic key must be placed into the key fob to start the car, and the passive component has to be close to the RFID reader.

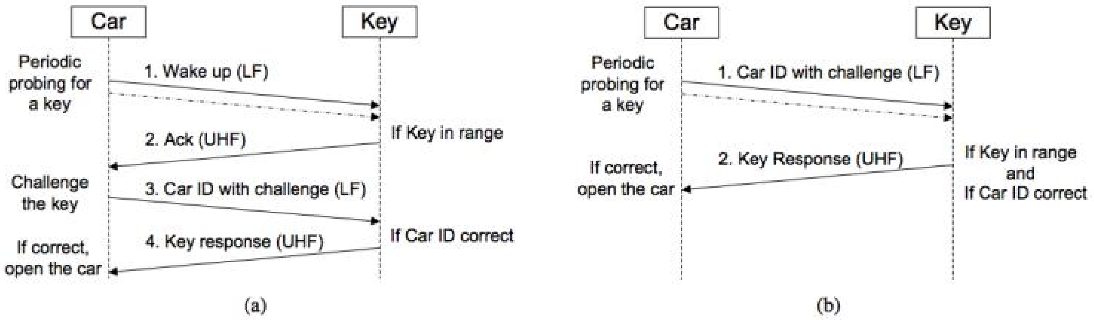


Figure 2.4 Examples of implementations of the passive keyless entry system protocol.

a) In a typical implementation, the car regularly scans the channel for the presence of the key by sending out short signals. If the key is within range, a challenge-response protocol is employed between the car and the key to grant or deny access. Since the key only emits very short beacons, energy is conserved.

b) In a second version, the vehicle routinely tests the channel directly using larger challenge beacons that are integrated with the car identification. If the key is nearby, it quickly responds to the challenge.

2.3 Fingerprint Analysis

The friction ridges and furrows that form on the pads of the fingers and thumbs to create fingerprints are distinctive designs. Although the prints from the palms, toes, and feet are equally distinctive and can be used to identify a person, this tutorial concentrates on the prints from the fingers and thumbs [9]. The friction ridges on that specific finger create the fingerprint pattern, such as the impression made when an inked finger is rubbed against paper. Depending on the arrangement and connection of the ridges, three different types of friction ridge patterns—loops, whorls, and arches—are distinguished:

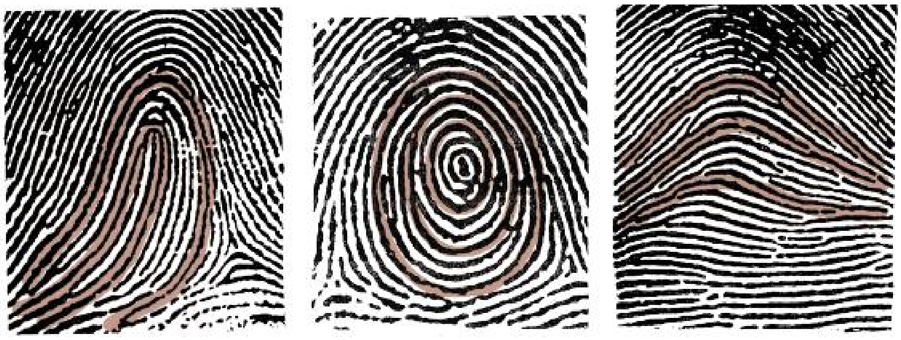


Figure 2.5 Loop, whorl & arch pattern of fingerprints.

Loops are prints that fold back on themselves to form a shape like a loop.Loops make up over 60% of pattern types, and are further divided into radial loops (pointing towards the radius bone, or thumb), and ulnar loops (pointing towards the ulna bone, or pinky) [10].

Whorls, which resemble tiny whirlpools, create patterns in the form of spirals or circles.There are four categories of whorls: the plain whorls (concentric circles), the centre pocket whorls (a loop with a whorl at the end), the double whorls (two whorls that form an S-shaped pattern), and the accidental whorls (irregular shaped). About 35% of all pattern types are whorls [10].

Tented and plain arches are combined to create an arch-like pattern.Compared to simple arches, tented arches rise to a sharper point. Five percent of all pattern kinds are arches [10].

2.4 Fingerprint Module

With its TTL UART interface, this fingerprint sensor module may be connected directly to a microcontroller's UART or to a computer via a MAX232 or USB-Serial converter. It may be configured to identify a person in 1:1 or 1:N mode and has the capacity to store fingerprint data in the module. Direct connections to 3V3 or 5V microcontrollers are possible with the FP module. A level converter (like MAX232) is required to connect to a PC serial port. Using visible light, optical fingerprint imaging involves taking a digital picture of the print. This type of sensor essentially serves as a customised digital camera. The top layer of the sensor, which is where the finger is placed, is referred to as the touch surface. Under this one, a layer of light-emitting phosphor illuminates the finger's surface. The light reflected from the finger forms a picture of the fingerprint through the phosphor layer and an array of solid state pixels (a charge-coupled device). The look of the fingerprint might be altered by a damaged or filthy touch surface. This sensor's imaging powers are constrained by the finger's skin quality, which is a disadvantage. For instance, it might be difficult to see a finger that is dirty or marked.

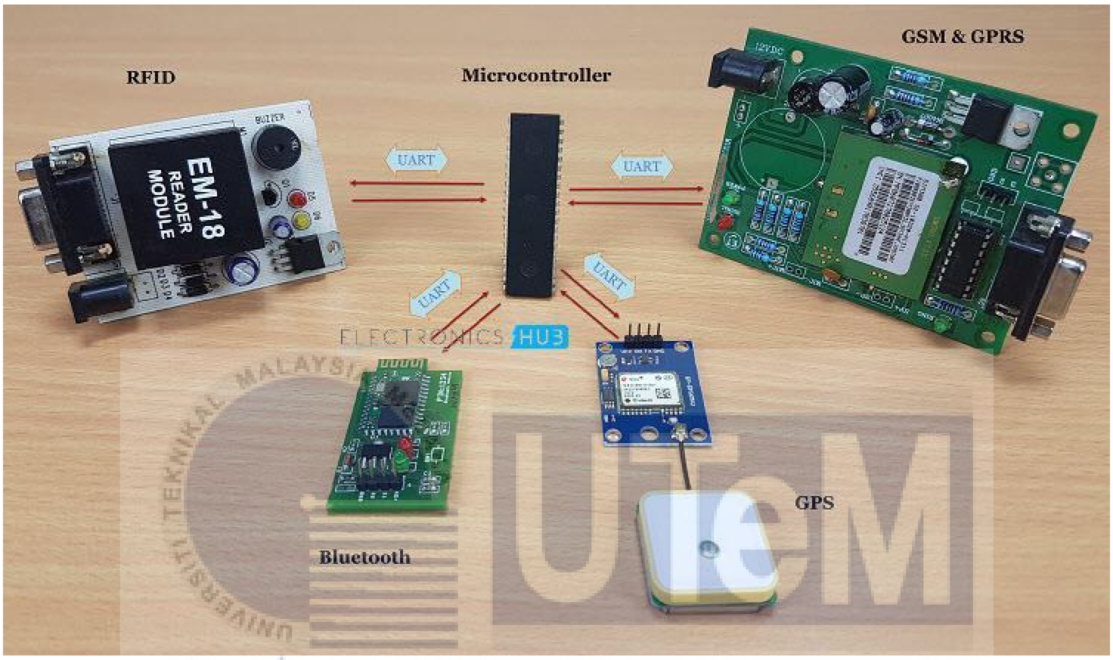


Figure 2.6 Bluetooth, GPS, GSM & GPRS, RFID module and Microcontroller.

Features

- A fingerprint reader may be further developed and integrated into a range of finished goods. Chip with an integrated algorithm for collecting images.
- Low price, minuscule size, and excellent performance. little electricity use.
- Professional optical technology, precision module production procedures, and excellent image processing abilities allow for the successful capture of 500 dpi photos.

2.5 Biometric Security Using Fingerprint Recognition

The biometric fingerprint-based security mechanism was introduced by Subhran and Venkata. The PXA27x DVK platform's fingerprint recognition is the major goal of this security system. The writers of this paper discuss several fingerprint analysis types. Among the features are fingerprint patterns that were distinctive based on aggregate ridges, which are a special attribute of the pattern. These three fundamental fingerprint patterns were circles, loops, and arches. Additionally, the author made advantage of the fingerprint sensor. The fingerprint sensor was created with the intention of taking scans and transmitting fingerprint patterns as digital pictures. A biometric template was created from this live scan using digital processing, and it was archived and utilised for matching. There were three types of fingerprint sensors used: optical, ultrasonic, and capacitive. The author utilises a Siemens ID Mouse for this project. PXA27x platform support for Linux kernel versions up to 2.6.9 bond, for drivers, Siemens ID Mouse was a gadget that used a capacitive fingerprint scanner USB2.0. Siemens ID Mouse is available in a number of versions, including 2.6.10 and later.

2.6 Security System Using Password

S. R. Khan constructed the security system using passwords. To lock password-based microcontrollers, the bulk of them used peripheral interface controllers (PIC), however some also used Atmel chips and FPGA. An electronic key that was based on a simple, inexpensive PIC was shown. Relay served as the key, and the input was a 4x4 keypad. A alternative key-based password technique is the Office

Access Control scheme.It is a low-cost device that is used to keep unauthorized individuals out of designated regions. The system employs the PIC as its primary controller and includes a 4x4 keypad. In general, this system was fairly unusual since it had an alert feature. [11]. Another electronic key with a similar design was also introduced.Similar to the key before it, it uses a PIC microcontroller and a keyboard for input.The option to add backspace and reset the password were among the added features, though. The programme was approved without any issues, thus the system must likewise work wonderfully. FPGA-based electronic keys have been made available. Because it can include the most recent design into an FPGA without requiring hardware modifications, the author claimed that it is easy to modify and unstable as a hardware-based technique. The main disadvantage is that it could cost more than other locks that employ a microcontroller chip. [11]. Lighting, temperature control, locking systems, and fire detection systems are just a few of the many subsystems found in Atmel-based systems. In this project, input was accepted using a keyboard input device that was nine times regarded a lock and used password authentication. A lock was installed in the door where it was concealed from view from the outside in order to reduce the likelihood of damage from criminals.

2.7 Vehicle starting system using fingerprint.

Utilizing a biometric system can help to lessen the problem of automobile theft or hijacking caused by simple access to the functioning system of the vehicle.The starting of a vehicle's engine is increasingly crucial since many expensive goods require protection and access restrictions.Biometric technologies have long been used as a reliable security solution in a variety of industries, and the automotive sector will soon use by people.A biometric system is a technical tool that identifies a person by using information about that individual.It needs specialized information on a certain biological feature in order to function properly.This method includes processing data using algorithms to produce a certain outcome, often a successful identification of a user or other people.The Arduino generates the signals and sends them to the suitable module.

2.8 GPS and GSM technology real time biometric vehicle security system.

Kiruthiga et al. have investigated the subject of automobile theft prevention. This system, which is simple to use, constantly performs well, is quick and simple to use, and it includes a respectable fingerprint recognition method, genuinely serves to guard the automobile from any unauthorised entrance. This innovation notifies the permitted user of the vehicle's position via GSM technology. If a permitted person tries to enter the automobile, a notification will be issued to the owner and the engine will be turned off. On the other hand, it is acceptable if an unapproved individual tries to enter the car. A GPS system is fitted to track the vehicle's position and current location. The location is established if a car is towed or a theft is thought to have occurred. If an engine is shut off but the GPS shifts significantly, an alarm message will be delivered to the authorised user. Additionally, the PIC16F877a is used as the security system's main platform, which monitors all of the system's input and output.

2.9 Design and development of fingerprint based car starting system

This article's goal is to develop and construct a fingerprint-based car ignition system with the goal of reducing auto theft and preventing unauthorised users. Car hijacking has increased recently as armed criminals focus on robbing cars, particularly brand-new ones. Therefore, it is believed to be essential to protect the vehicles against hijackers. In this trial, the automobile can only be started by individuals who have received authorization using a system designed expressly to collect their fingerprints and other distinctive pattern features. This is accomplished using the fingerprint module, PIC18F4620 microcontroller, and LCD module.

2.10 Development of Fingerprint Engine Starter

In order to enhance and create high security on the motorbikes and scooters that Batangas State University ARASOF Nasugbu students commonly used, their research's primary objective was to understand how an electric engine starter was converted into a fingerprint-based engine starter. In order to improve the safety of motorcycles and scooters, the researchers focused their attention on the design and adaption of the fingerprint engine starter. The analysis as a whole revealed substantial differences between the present electric engine starter technology and the fingerprint engine starter for motorcycles and scooters. The upgraded Fingerprint Engine Starter for Motorcycles and Scooters provided a greater level of security compared to the existing Electric Engine Starting System. to create a brand-new, improved technique for starting bikes and scooters. Additionally, the built-in fingerprint engine starter system showed its best performance in terms of precision, effectiveness, and security, and this study has the potential to change the current electric engine starter system. demonstrating the significant differences between the two systems' acceptance scores in terms of reliability, user-friendliness, and the aforementioned criteria.

CHAPTER 3

METHODOLOGY

3.1 Introduction

The methodology is the comprehensive research strategy, which among other things outlines the procedures to be used and details how research will be conducted.These processes, which are described in the methodology, detail the methods or strategies for data collection or, occasionally, the formula utilized to arrive at a certain result.Although the nature and types of processes to be used in a certain operation or to reach an objective are heavily emphasized, methodology does not identify specific techniques.

These stages can be broken down into smaller ones, combined, or have their sequence changed when they are appropriate for a methodological research's overall framework.

Both a technique and a paradigm serve as constructive frameworks. The development of paradigms in theoretical research satisfies the majority or all of the methodological criteria. An algorithm is an example of a constructive framework, which consists of linked logical connections as opposed to physical parts.

3.2 Software design method

The system's software development comprises a flowchart, algorithm, circuit simulation, and programming language selection.

3.2.1 Simulation method

Proteus ver8.0 was used to simulate the fingerprint vehicle starting system. The procedure was followed in accordance with the block diagram in Figure 3.1. The phases of implementation comprise a microcontroller, LCD display, motor output driver, fingerprint module, buzzer unit, and regulated power supply.

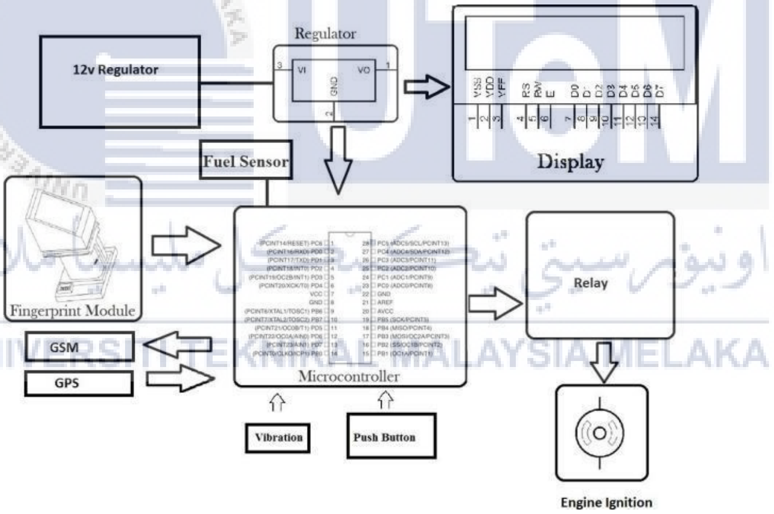


Figure 3.1 Block Diagram [12]

3.2.1.1 Power Supply Unit

An electrical device called a power supply delivers electricity to a load of other electrical devices. Although several components of the circuit require 5V DC, 220V AC is the primary power source. In order to create 12V DC from 220V AC, a rectifier must first rectify 12V AC using a step-down transformer. The rectifier's output is known as fluctuating DC since it still contains certain aberrations despite being a DC signal. The ripples will be removed and a smoothed signal will be produced by using DC power filter circuits [13]. The microcontroller is powered by converting the 12V from the relay/motor unit to 5V. The power supply unit as in Figure 3.2.

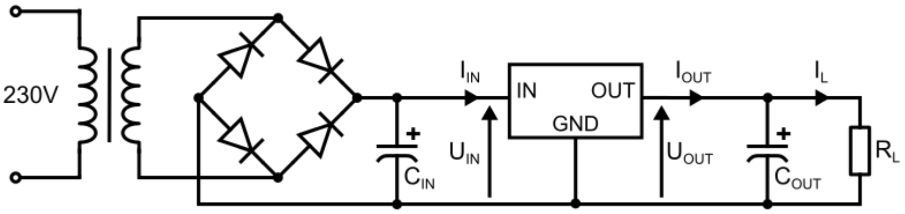


Figure 3.2 The regulated power supply unit [13]

3.2.1.2 The Fingerprint Module Unit

A digital image of the fingerprint pattern is captured using an electronic fingerprint sensor or module [14]. There are several different types, including optical, capacitive, mechanical, thermal, and dynamic output built in either stationary or moving form. This study made use of an optical biometric fingerprint reader (R305) module with a TTL UART interface for direct connections to a 3.3v or 5v microcontroller [15]. With the use of a strong Digital Serial Port (DSP) at its heart, the user may save fingerprint data in the module, which can be set up to identify a person in 1:1 or 1: N mode [16]. The hex codes we use to communicate with it must follow a specified format, but the commands used to operate each module might differ. Pins 1 through 4 are used for serial transmission, with a 57600bps default baud rate. For USB transmission, pins 5 through 8 are used [17]. Table I displays the function of the module pins. The module has power DC 3.6V- 6.0V, working current 100mA, character file size: 256 bytes, image acquiring time: <0.5s, storage capacity: 256, security level: 5 (low to high: 1, 2, 3, 4, 5), False Accept Rate (FAR): <0.001%, False

Reject Rate (FRR): <0.1%, average searching time: < 0.8s, window dimension: 18mm*22mm, baud rate is (9600*N) bps, N = 1 - 12 (default N = 6). Because of this, the user may select the baud rate between 9600 and 115200bps [17]. In Figure 3.3, the fingerprint module unit is displayed.

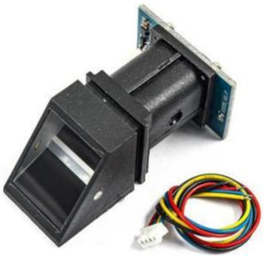
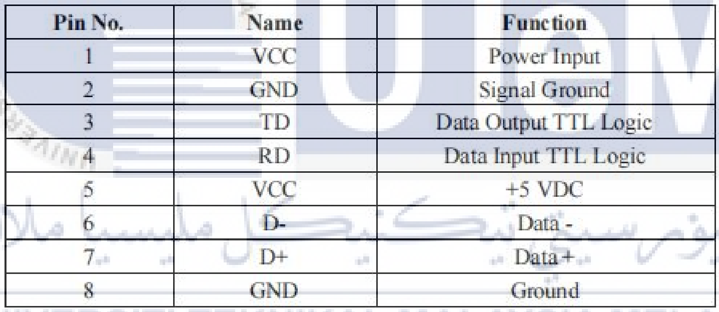


Figure 3.3 The fingerprint module

Table 3.1 The R305 Module Pins Function



Pin No.	Name	Function
1	VCC	Power Input
2	GND	Signal Ground
3	TD	Data Output TTL Logic
4	RD	Data Input TTL Logic
5	VCC	+5 VDC
6	D-	Data -
7	D+	Data +
8	GND	Ground

3.2.1.3 The Microcontroller Unit

The Arduino is a platform for making electronic projects that is open-source and free. It is made comprised of a physical, programmable circuit board (often known as a microcontroller) and IDE (Integrated Development Environment) software, which runs on your computer and is used to write and upload computer code to the physical board. The Arduino IDE's use of C++ makes learning the software easier. This project used the Arduino Uno, which is based on the Atmel ATmega328P. Based on the AVR architecture, it is a 32K 8-bit microcontroller with excellent performance and minimal power. This unit has 131 instructions that are executed once per 20 clock cycles for a throughput of over 20 MIPS @ 20 MHz. In order to function with our 28-pin AVR development board, it is packed in a 28-pin PDIP configuration. It has 32kB of programmable flash, 1kB of EEPROM, 2kB of SRAM, 10,000 write and erase cycles, and 100,000 write and erase cycles for the EEPROM. Data retention is 20 years at 85°C and 100 years at 25°C. The microcontroller unit is shown in Figure 3.5.

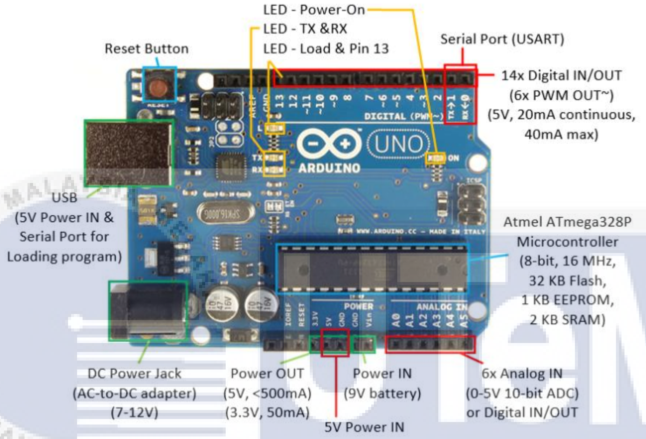


Figure 3.4 Atmel ATmega328P Microcontroller unit on Arduino UNO [18]

3.2.1.4The buzzer unit

An auditory signalling tool that can be electromechanical, piezoelectric, magnetic, electromagnetic, or electroacoustic. Since its element may be driven by an oscillating electronic circuit or another audio signal source that is driven with a piezoelectric audio amplifier, an active buzzer of the piezoelectric type that will buzz at a predefined frequency (2300 tolerance of 300Hz) on its own when a steady DC power is applied is used in this work. An unregistered, unauthenticated, or unauthorised fingerprint will beep. Figure 3.6 depicts the buzzer circuit.

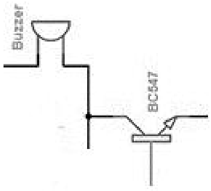


Figure 3.5 The buzzer unit

3.2.1.5 GPS and GSM Module Unit

To establish a communication link between the vehicle's user and security system, a GSM (Global System for Mobile Communication) module is necessary. The SIM 900 module has been put in [19]. This module may be controlled using AT commands.Serial devices may send SMS and data via the GSM network with a GSM modem, which offers full functionality. GPRS service is offered with this SIM 300. In sleep mode, the current consumption is as low as 2.5mA. Information and communications are kept in SIMs (Subscriber Identity Modules). It operates between 3.2 and 4.5 volts and talks with the ARM controller via asynchronous serial communication at a baud rate of 9600. Sending an SMS instructs you to lock the bike in case it is stolen or taken. A network of 24 satellites makes up the satellite-based Global Positioning System (GPS). It is employed to track the vehicle. The GPS receiver utilised is the Media Tek GPS MT3329 [20], which has a maximum update rate of 10Hz and supports up to 66 channels of satellite searching with a sensitivity of -165dBm. We are able to pinpoint the exact location of the bike using GPS. GPS satellites relay signal data to the ground by making two accurate orbits around the planet each day. This data is used by the GPS receiver to triangulate the user's precise location.

Figure 3.6 GPS Module

Figure 3.7 GSM Module

3.2.2 Flowchart

Figure 3.9 shows the flowchart for the fingerprint car starting system.

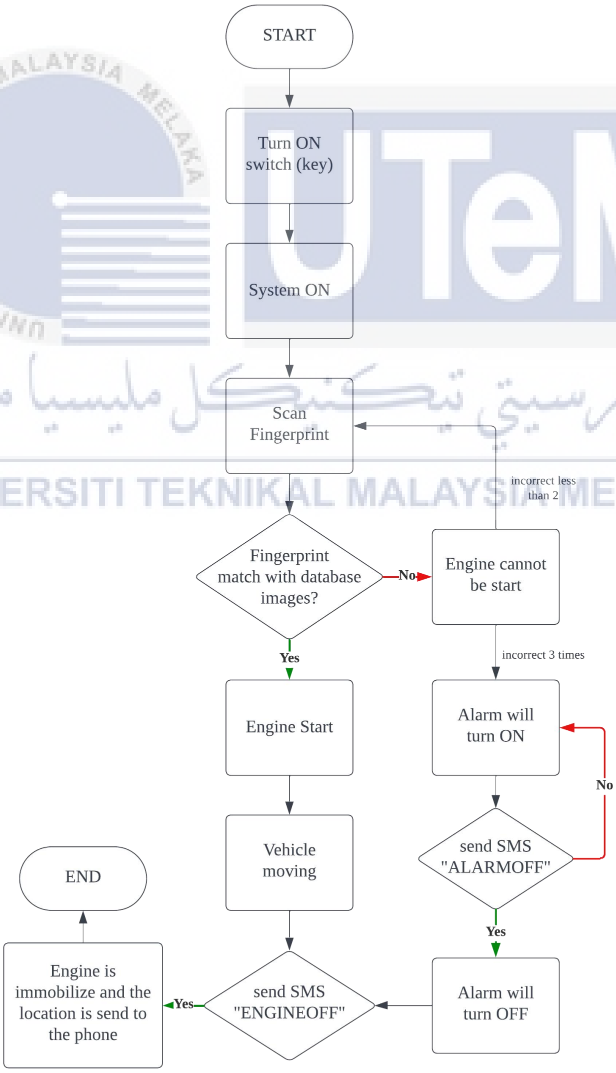


Figure 3.8 Flowchart of the proposed system

3.2.3 Algorithm

The following is the algorithm that describes the fingerprint car starting system's flowchart:

1. Start

2. Press the Start key.

3. Position finger for fingerprint scanning.

4. If the image matches the one that was previously loaded, start the engine.

5. If not, return to step one and skip the output.

6. Stop

3.2.4 Choice of Programming Language

Embedded C and the Arduino programming language are used to develop the core software application. C language programming was employed since it is more compatible with the Arduino programming platform. This is widely used for programming embedded microcontrollers. The C programming language is widely used to create code portions where timing accuracy and code efficiency are essential criteria.

3.2.5 Registration of fingerprints

The fingerprint input must be set up by the same finger registering (entering) a fingerprint twice. The two are preserved as a template to improve the system's failure tolerance. If the contact variation in the fingers occurs and the input slot enlarges, the system will reject registration and consider them as a separate fingerprint in the two entry method. Additionally, the system may record two finger input points as having the exact same fingerprint and deny registration if there is even the tiniest change between them.

3.3 Hardware Manufacturing Process

The circuit construction process followed the block architecture in Figure 3.1 in stages. The components were first constructed on an electronics breadboard in order to ensure proper terminal connections. Then, using a soldering iron and MBO 1mm wire lead solder, which has a melting point of +183°C, they were transferred to a PCB board and permanently soldered. To decrease awkwardness and component bridging, excessive Lead was avoided.

3.3.1 Soldering process

Figure 3.9 Soldering Process

Soldering Irons

Iron tips are necessary for every iron. The iron's tip warms up, causing solder to flow around the two components being linked. It's a widespread fallacy that the tip transmits the solder, even though it will adhere to the tip when applied. The solder melts as a result of the tip's real heat transmission, which raises the temperature of the metal parts to solder's melting point. If you need to replace an outdated tip or move to a new kind of tip, most irons allow you to change the tip. To fit any component, tips are available in a range of sizes and forms.

The portion of the iron that retains the tip is known as the "wand." Additionally, the user is in charge of this portion. Wands commonly have wires and metal contacts that transmit heat from the base or outlet to the tip while also preventing the heat from the tip from passing to the outside of the wand. Wands are typically composed of a variety of insulating materials (such as rubber). A high-quality wand is much valued since it serves the twin purposes of heating and avoiding burns.

The rubber to soldering's pencil is called Solder Wick. Solder wick is a highly useful tool for addressing problems like jumpers or desoldering components. Copper wire that has been braided together is the basis of desoldering braid, also known as solder wick. It is possible to "erase" excess solder globs because the copper soaks (wicks) up the solder.

When desoldering components, a solder vacuum (also known as a solder sucker) is a fantastic tool for removing solder that is left in through-holes.

Preparing to Solder

Tinning The Soldering Tip

A brand-new soldering tip or one that is extremely unclean has to be tinned before use. A thin layer of solder is applied to a soldering tip during the "tinning" procedure. This provides a basis for the solder to flow from and helps heat transmission between the tip and the component you are soldering.

Step 1: Preheating the Iron

Thoroughly warm up the soldering iron or gun. As you are about to melt a lot of solder on it, be sure it has reached its full temperature. This is crucial if the iron is brand-new since it could have been packaged with a coating to stop corrosion.

Step 2: Make A Small Space Ready

Prepare a small workspace while the soldering iron is heating up. Place a moistened sponge in the base of your soldering iron stand or a nearby dish. Make sure you have enough space to work comfortably and lay down a piece of cardboard in case you drop solder, which you probably will.

Step 3: Completely Solder The Tip

Apply a thick layer of solder on the soldering tip. It is crucial to completely enclose the tip. Be prepared since you'll need a lot of solder and it will leak during this operation. Run the solder up and down the tip and all the way around it to thoroughly cover it in molten solder. If you leave any portion of the tip exposed, it will prone to accumulate flux residue and won't transfer heat very effectively.

Step 4: Clean Up After Soldering

Once the tip has been well covered with solder, wash out any remaining flux by wiping the tip on a moist sponge. Do this right away to prevent the flux from drying out and solidifying.

Step 5: Complete

Your soldering tip has just been tinned. To ensure that the iron maintains efficient heat transmission, do this whenever you replace or clean the tip.

Safety in Soldering

Even while soldering is typically a safe pastime, there are a few things to be aware of. The first and most evident is that high temperatures are involved. Burns from soldering irons will occur fast and at a temperature of 350F or higher.

Use a stand to support the iron and keep the cord out of the path of foot traffic. It makes reasonable to avoid soldering over exposed body parts since solder itself can drop. Always work in a well-lit place with room to spread out your components and move about.

As the fumes from the flux and other coatings will irritate your respiratory system and eyes, it is best to avoid soldering with your face immediately above the junction. Since most solders contain lead, you should always wash your hands before eating and refrain from touching your face when working with solder.

3.4 Method for Testing Circuits

Before mounting components to the donut board, testing of the components was done. Additionally, continuity tests and power-on tests were performed during construction to verify appropriate circuit operation, to make sure no circuit components heat up when the device is in use, and to prevent loading and impedance mismatch between one stage and another.

3.4.1 Test of Continuity

After the soldering, this test is run to look for any electrical open routes in the circuit. To ensure that no cable or line was blocked and that all lines had a free flow of electrons, a continuity test was performed. The researcher can examine to see if the built-in circuit has current flowing through it or if the lines are continuous. A simple light bulb that illuminates when current flows or even a piezoelectric speaker may be used as a continuity tester, as can multimeters that measure current and more affordable specialised continuity testers. The circuit is "open" if there are any wires that are broken, any components that are damaged, or any resistance that is too high. This test was conducted with a multimeter in this test.

3.4.2 On-Board Test

This test is run to see if the voltage at the various terminals is in compliance with the specifications or not. To prevent the microcontroller from being harmed by any high voltage or heat, it was performed without one. This test was performed in this investigation using a multimeter.

3.5 Software Development

Software development is required for this project to make sure the system is fully functional. The necessary software included circuit diagram drafting, scripting microcontroller commands, and creating an application to manage the operation of the device. The linked software functionalities are developed using a variety of programmes.

3.5.1 Proteus 8

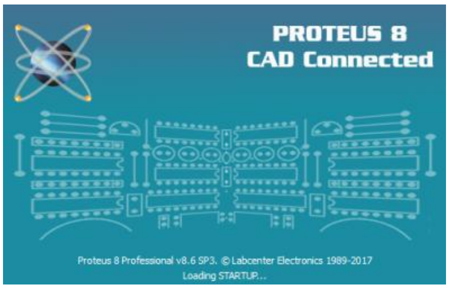

PROTEUS 8
CAD Connected
Proteus 8 Professional v8.6 SP3. © Labcenter Electronics 1989-2017
Loading STARTUP...

Figure 3.10 Proteus 8 Software

An programme called Proteus 8 is used to create circuits for microcontrollers like Arduino. It includes a number of libraries for various Arduino devices and sensors. Additionally, it works with the Arduino IDE software, which is used to programme Arduinos. As a result, in this project, this programme is utilised to create the circuit design for the Arduino simulation.

3.5.2 Arduino IDE

Figure 3.11 Arduino IDE Software

A well-known programme for programming Arduino is called Arduino IDE. This software was created to allow you to programme Arduino commands. This programme will code the commands necessary for the particular function, which will then be built and uploaded to the Arduino [21]. The programme will then send the command to the Arduino, which will subsequently carry out the instructions. The Proteus 8 programme may also be used to upload the commands and simulate how the Arduino works.

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Introduction

The next part summaries the outcomes derived from the experimental or analytical inquiry undertaken in this study. It covers the main conclusions, data analysis, and the consequences of these findings in relation to the goals of the study. To make it easier to fully understand the results, the data is provided in an organized way.

After the results are displayed, the discussion section critically evaluates and analyses the data, making links between the more general research questions and the observed results. This section also examines the data' importance in the context of current theories, hypotheses, and literature. The objective is to cultivate an advanced knowledge of the topic and offer readers a better appreciation of the significance of the study's findings.

The goal of the talks and results taken together is to make a significant contribution to the body of knowledge already available in the subject, providing knowledge that may open up opportunities for investigation and study.

4.2 Circuit Testing

Unit testing, another name for component testing, is the process of evaluating each software system components or units separately. The aim is to confirm that every software component operates as intended. A unit in Arduino programming might be a function, a class, or a particular code module.

4.2.1 GSM Module SIM900A

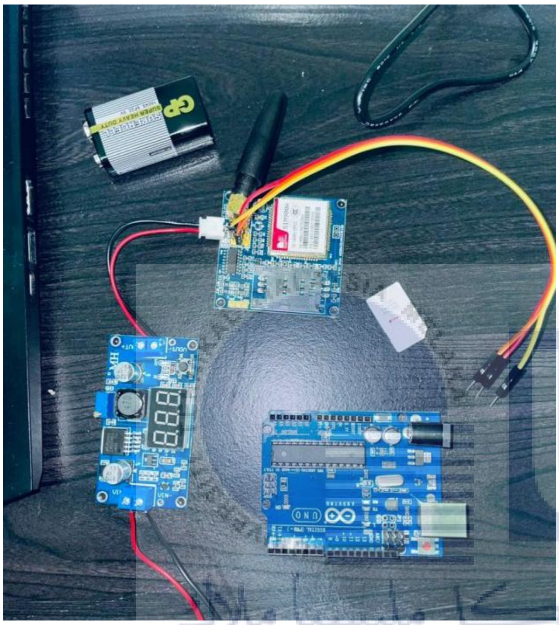


Figure 4.1 Before testing GSM module

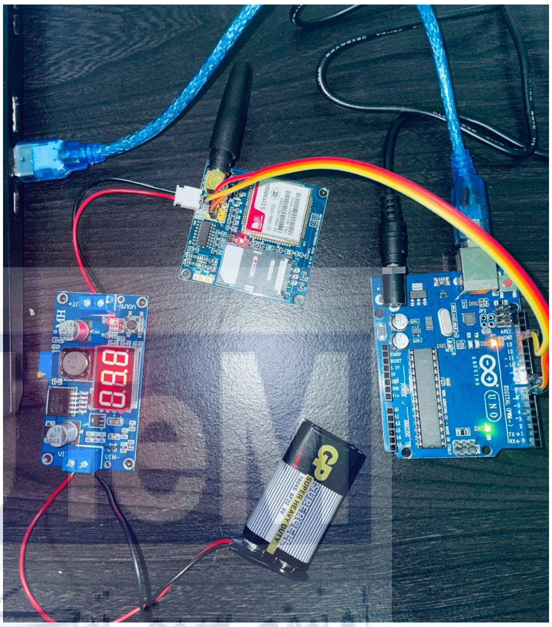
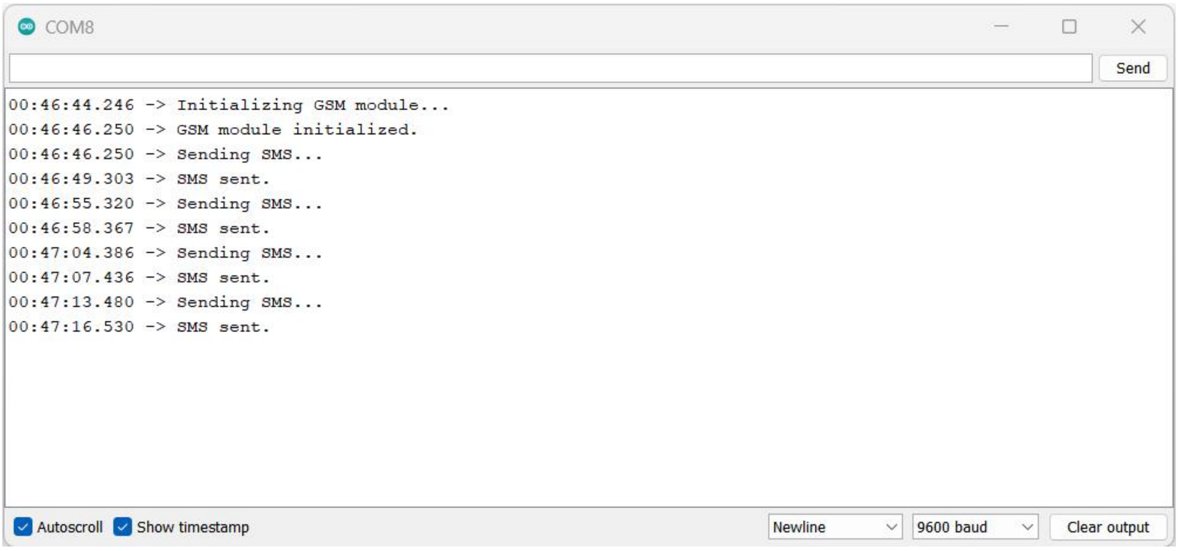


Figure 4.2 Testing GSM module

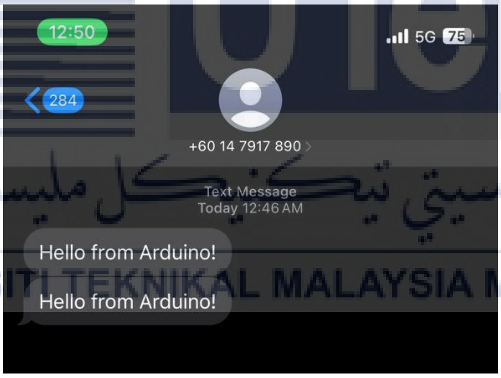
Figure 4.1 shown before testing the GSM module. To testing the module, Arduino UNO, GSM module SIM900A, battery 9V, Step down module, and sim card is needed. In Figure 4.2 shown the component is connected and GSM module is been testing.



```
00:46:44.246 -> Initializing GSM module...
00:46:46.250 -> GSM module initialized.
00:46:46.250 -> Sending SMS...
00:46:49.303 -> SMS sent.
00:46:55.320 -> Sending SMS...
00:46:58.367 -> SMS sent.
00:47:04.386 -> Sending SMS...
00:47:07.436 -> SMS sent.
00:47:13.480 -> Sending SMS...
00:47:16.530 -> SMS sent.
```

Figure 4.3 Serial monitor output

Figure 4.3 shown above is serial monitor output based on GSM module. Based on the results, the SMS has been sent to the recipient's phone.



Hello from Arduino!

Hello from Arduino!

Figure 4.4 SMS that received from GSM module

Figure 4.4 above shown the SMS that been received on the recipient’s phone from the GSM module. The testing for the GSM module is successful.

Based on the results, the testing of GSM module was successful and in conclude the GSM module was in good condition because it can functioning well.

4.3 Circuit Design

This section shows the circuit design of the project. The key element of the project is the circuit design, which provides a structure for the interactions between electronic parts to provide the functionality that is expected. Power supply, arduino UNO, fingerprint sensor, GSM SIM800L modules, GPS module, rocket switch, buzzer, LED (an engine indicator) and safety mechanisms are some of the circuit's essential components.

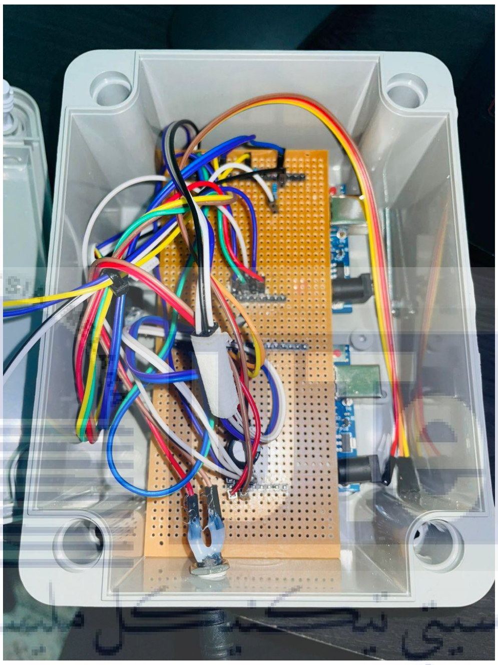


Figure 4.5 Condition inside the casing

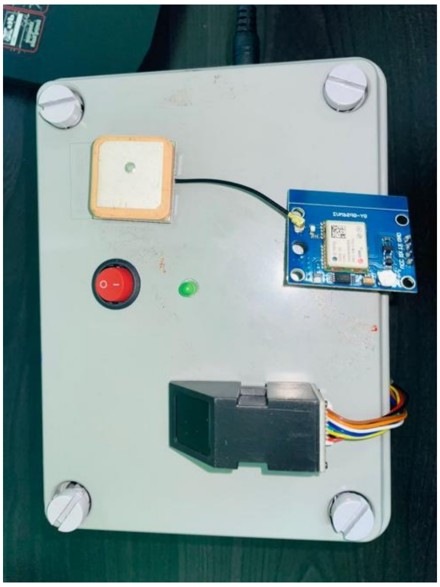
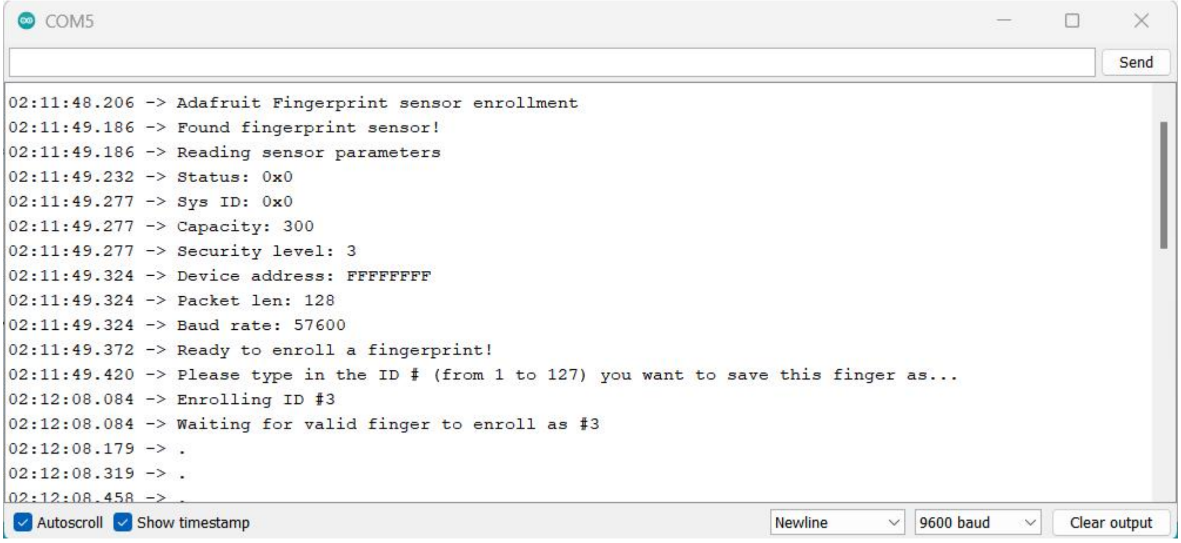


Figure 4.6 Condition outside the casing

4.4 Results

4.4.1 Register User's Fingerprint

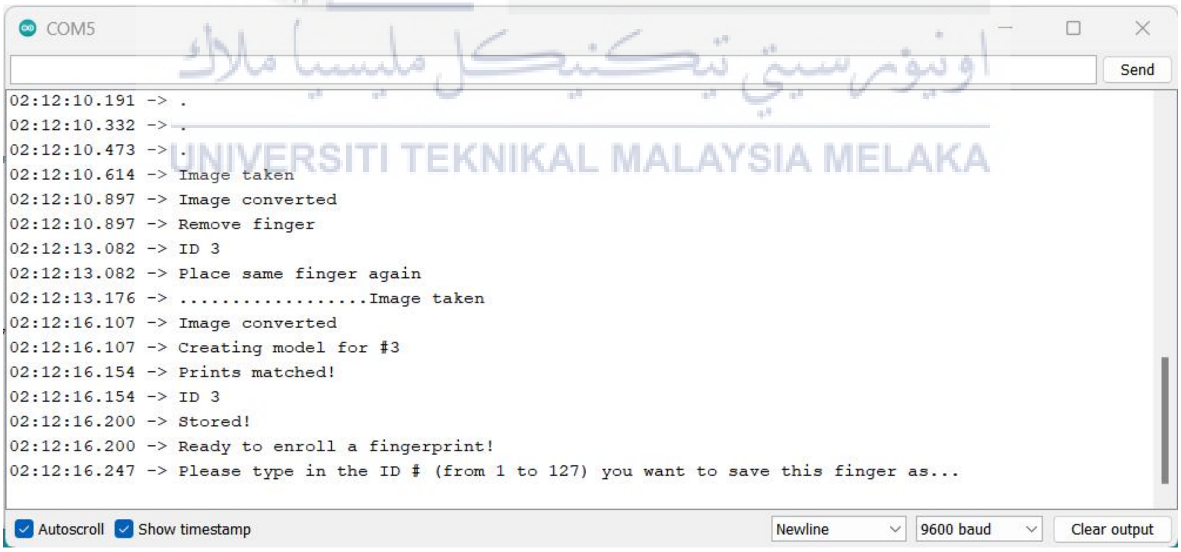


```
02:11:48.206 -> Adafruit Fingerprint sensor enrollment
02:11:49.186 -> Found fingerprint sensor!
02:11:49.186 -> Reading sensor parameters
02:11:49.232 -> Status: 0x0
02:11:49.277 -> Sys ID: 0x0
02:11:49.277 -> Capacity: 300
02:11:49.277 -> Security level: 3
02:11:49.324 -> Device address: FFFFFFFF
02:11:49.324 -> Packet len: 128
02:11:49.324 -> Baud rate: 57600
02:11:49.372 -> Ready to enroll a fingerprint!
02:11:49.420 -> Please type in the ID # (from 1 to 127) you want to save this finger as...
02:12:08.084 -> Enrolling ID #3
02:12:08.084 -> Waiting for valid finger to enroll as #3
02:12:08.179 -> .
02:12:08.319 -> .
02:12:08.458 -> .
```

☑ Autoscroll ☑ Show timestamp Newline 9600 baud

Figure 4.7 Enrollment fingerprint ID #3

The figure 4.7 above shown the output of serial monitor in Arduino IDE, the fingerprint sensor enrollment for ID #3. This output shown before the fingerprint was place on the fingerprint sensor.

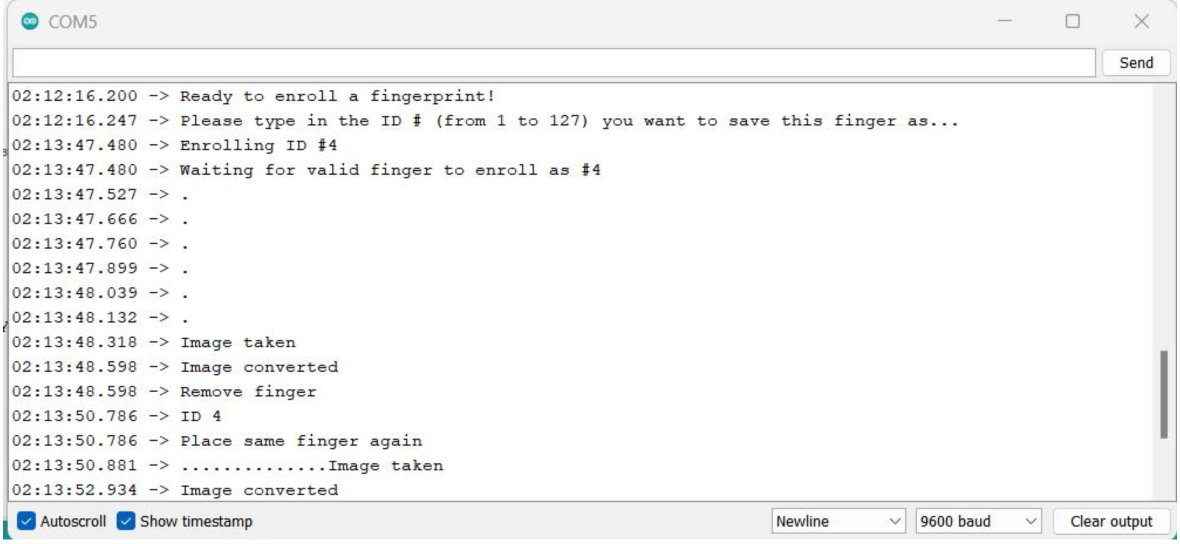


```
02:12:10.191 -> .
02:12:10.332 -> .
02:12:10.473 -> .
02:12:10.614 -> Image taken
02:12:10.897 -> Image converted
02:12:10.897 -> Remove finger
02:12:13.082 -> ID 3
02:12:13.082 -> Place same finger again
02:12:13.176 -> ..................Image taken
02:12:16.107 -> Image converted
02:12:16.107 -> Creating model for #3
02:12:16.154 -> Prints matched!
02:12:16.154 -> ID 3
02:12:16.200 -> Stored!
02:12:16.200 -> Ready to enroll a fingerprint!
02:12:16.247 -> Please type in the ID # (from 1 to 127) you want to save this finger as...
```

☑ Autoscroll ☑ Show timestamp Newline 9600 baud

Figure 4.8 Successful enrollment fingerprint ID #3

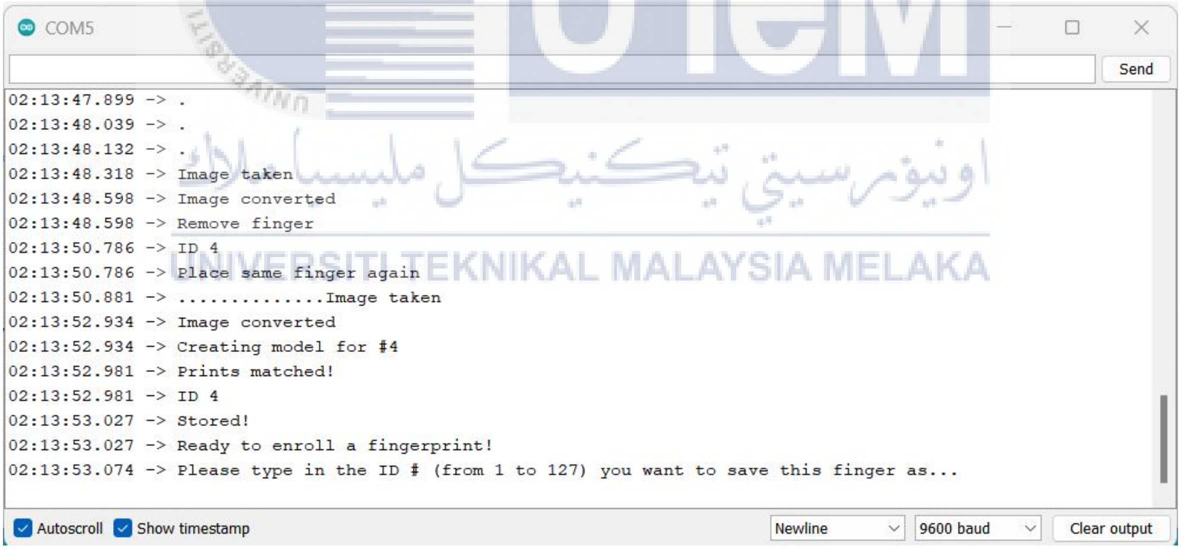
The figure 4.8 above shown the output of serial monitor in Arduino IDE, the successful fingerprint sensor enrollment for ID #3. This output shown after the fingerprint was place on the fingerprint sensor and the ID #3 was stored.



```
02:12:16.200 -> Ready to enroll a fingerprint!
02:12:16.247 -> Please type in the ID # (from 1 to 127) you want to save this finger as...
02:13:47.480 -> Enrolling ID #4
02:13:47.480 -> Waiting for valid finger to enroll as #4
02:13:47.527 -> .
02:13:47.666 -> .
02:13:47.760 -> .
02:13:47.899 -> .
02:13:48.039 -> .
02:13:48.132 -> .
02:13:48.318 -> Image taken
02:13:48.598 -> Image converted
02:13:48.598 -> Remove finger
02:13:50.786 -> ID 4
02:13:50.786 -> Place same finger again
02:13:50.881 -> .............Image taken
02:13:52.934 -> Image converted
```

Figure 4.9 Enrollment fingerprint ID #4

The figure 4.9 above shown the output of serial monitor in Arduino IDE, the fingerprint sensor enrollment for ID #4. This output shown before the fingerprint was place on the fingerprint sensor.



```
02:13:47.899 -> .
02:13:48.039 -> .
02:13:48.132 -> .
02:13:48.318 -> Image taken
02:13:48.598 -> Image converted
02:13:48.598 -> Remove finger
02:13:50.786 -> ID 4
02:13:50.786 -> Place same finger again
02:13:50.881 -> .............Image taken
02:13:52.934 -> Image converted
02:13:52.934 -> Creating model for #4
02:13:52.981 -> Prints matched!
02:13:52.981 -> ID 4
02:13:53.027 -> Stored!
02:13:53.027 -> Ready to enroll a fingerprint!
02:13:53.074 -> Please type in the ID # (from 1 to 127) you want to save this finger as...
```

Figure 4.10 Successful enrollment fingerprint ID #4

The figure 4.10 above shown the output of serial monitor in Arduino IDE, the successful fingerprint sensor enrollment for ID #4. This output shown after the fingerprint was place on the fingerprint sensor and the ID #4 was stored.

	Time Taken for fingerprint to be stored (s)	
Number of ID	Trial 1	Trial 2
#1	3.662	3.612
#2	3.862	3.517
#3	3.466	3.287
#4	3.326	3.291
#5	3.791	3.143
#6	4.412	3.194
#7	3.985	3.380
#8	3.664	3.276
#9	3.194	3.327
#10	3.654	3.377

Table 4.1 Time Taken for fingerprint to be stored

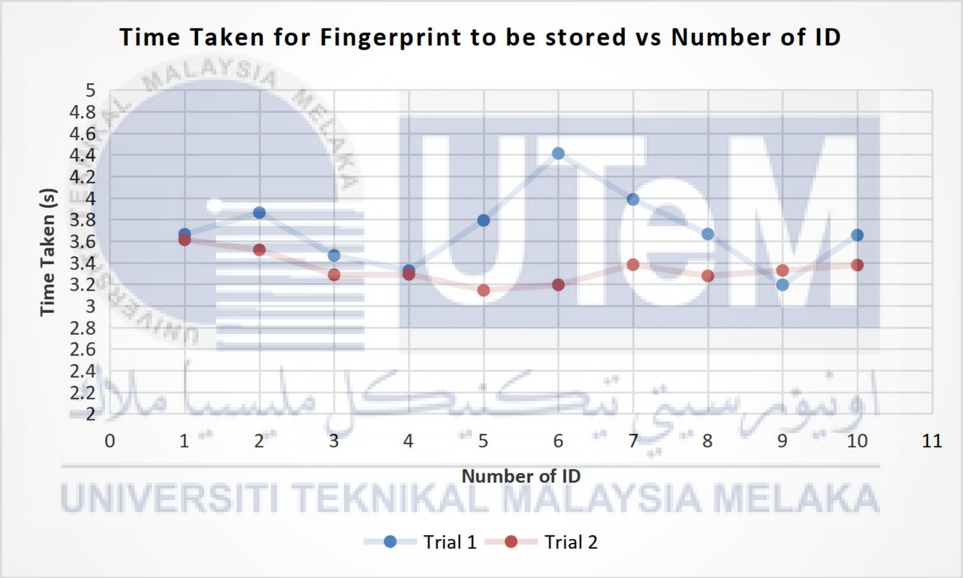
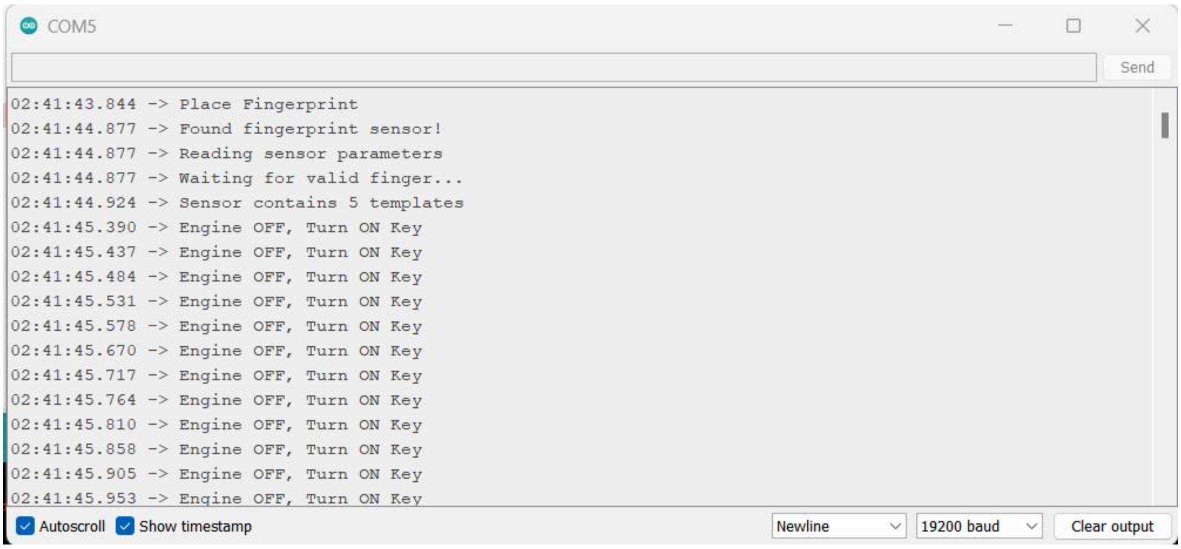


Figure 4.11 Graph of time taken for fingerprint to be stored

The results shown, the time taken for fingerprint to be stored for each number of ID, it is between a range of 3.0s to 4.5s. Based on trial 1, the highest time taken that recorded was 4.412s and the lowest timer is 3.194s. Based on trial 2, the highest time taken that recorded was 3.612s and the lowest is 3.143s. In summary, this project has different time taken because of the effectiveness fingerprint sensor. A few processes are involved in storing a fingerprint in a fingerprint sensor, which adds to the process's total duration. A fingerprint sensor may take some time to store a fingerprint maybe because of image capture, image processing, template extraction, template storage, quality checks, and user interaction.

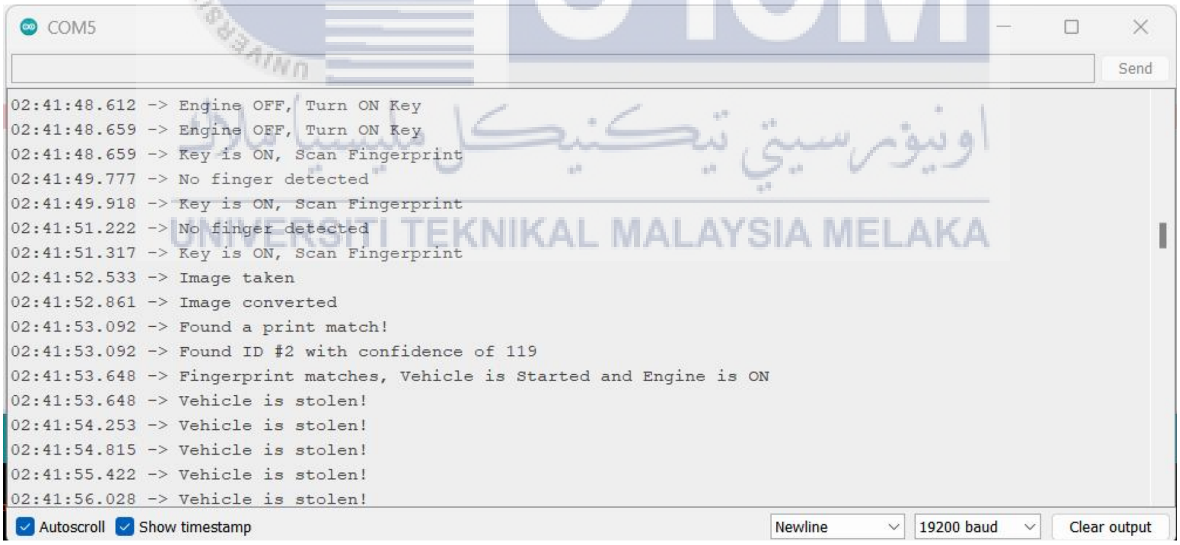
4.4.2 Vehicle Condition After Placing The Fingerprint



```
02:41:43.844 -> Place Fingerprint
02:41:44.877 -> Found fingerprint sensor!
02:41:44.877 -> Reading sensor parameters
02:41:44.877 -> Waiting for valid finger...
02:41:44.924 -> Sensor contains 5 templates
02:41:45.390 -> Engine OFF, Turn ON Key
02:41:45.437 -> Engine OFF, Turn ON Key
02:41:45.484 -> Engine OFF, Turn ON Key
02:41:45.531 -> Engine OFF, Turn ON Key
02:41:45.578 -> Engine OFF, Turn ON Key
02:41:45.670 -> Engine OFF, Turn ON Key
02:41:45.717 -> Engine OFF, Turn ON Key
02:41:45.764 -> Engine OFF, Turn ON Key
02:41:45.810 -> Engine OFF, Turn ON Key
02:41:45.858 -> Engine OFF, Turn ON Key
02:41:45.905 -> Engine OFF, Turn ON Key
02:41:45.953 -> Engine OFF, Turn ON Key
```

Figure 4.12 Output for serial monitor

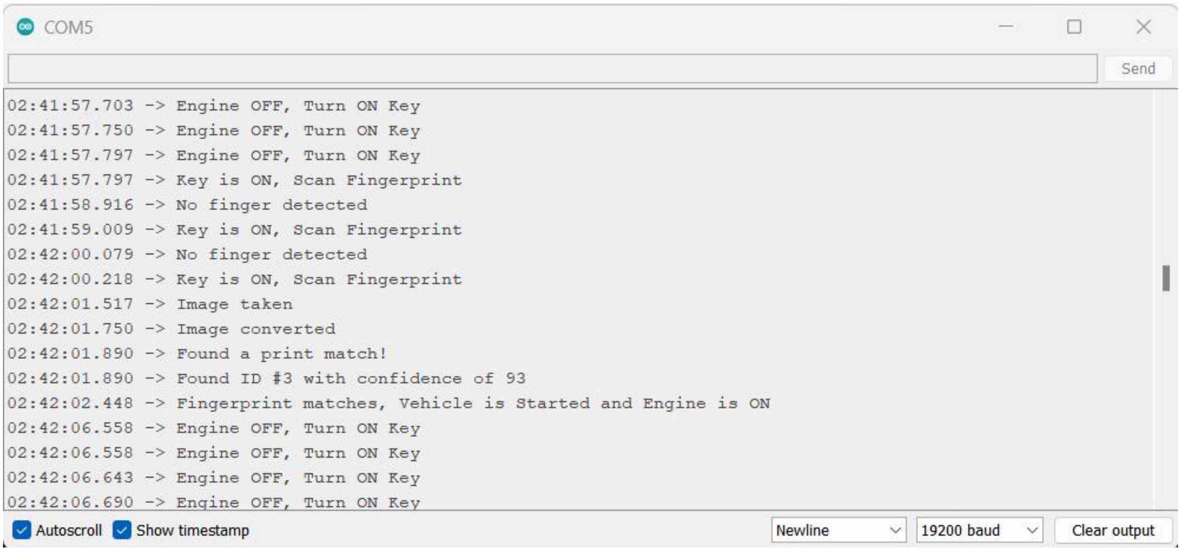
The figure 4.12 above shown the output of serial monitor in Arduino IDE. The figure shown the output that the Engine is still in OFF condition, and need to turn ON the switch to ON the system.



```
02:41:48.612 -> Engine OFF, Turn ON Key
02:41:48.659 -> Engine OFF, Turn ON Key
02:41:48.659 -> Key is ON, Scan Fingerprint
02:41:49.777 -> No finger detected
02:41:49.918 -> Key is ON, Scan Fingerprint
02:41:51.222 -> No finger detected
02:41:51.317 -> Key is ON, Scan Fingerprint
02:41:52.533 -> Image taken
02:41:52.861 -> Image converted
02:41:53.092 -> Found a print match!
02:41:53.092 -> Found ID #2 with confidence of 119
02:41:53.648 -> Fingerprint matches, Vehicle is Started and Engine is ON
02:41:53.648 -> Vehicle is stolen!
02:41:54.253 -> Vehicle is stolen!
02:41:54.815 -> Vehicle is stolen!
02:41:55.422 -> Vehicle is stolen!
02:41:56.028 -> Vehicle is stolen!
```

Figure 4.13 Output for serial monitor

The figure 4.13 above shown the output of serial monitor in Arduino IDE. The figure shown the key is ON, and the user can place the fingerprint on the fingerprint sensor. The results shown, the fingerprint is matched with the fingerprint ID #2 and the vehicle is started and the engine is ON, but unfortunately the error of the results shown the vehicle is stolen even the fingerprint is matched.

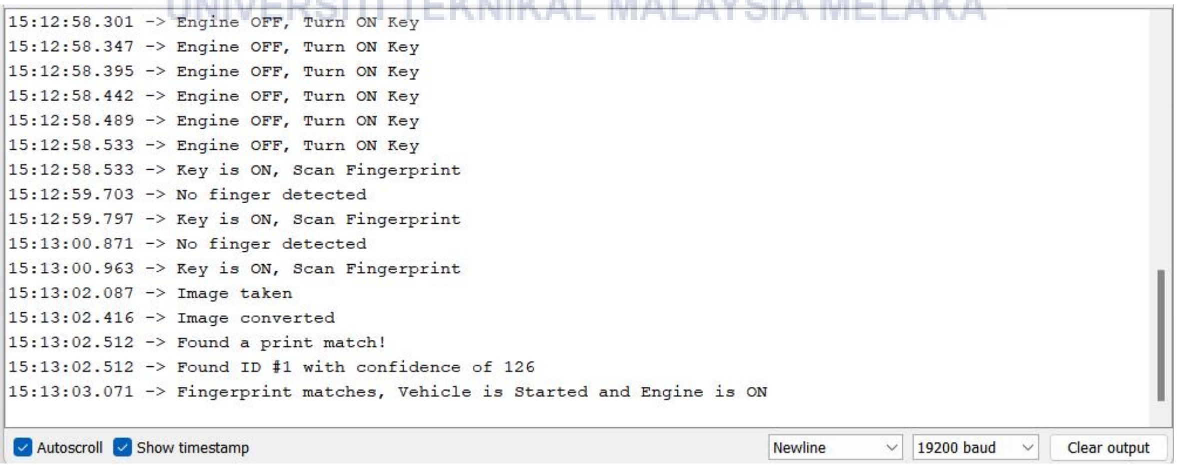


```
02:41:57.703 -> Engine OFF, Turn ON Key
02:41:57.750 -> Engine OFF, Turn ON Key
02:41:57.797 -> Engine OFF, Turn ON Key
02:41:57.797 -> Key is ON, Scan Fingerprint
02:41:58.916 -> No finger detected
02:41:59.009 -> Key is ON, Scan Fingerprint
02:42:00.079 -> No finger detected
02:42:00.218 -> Key is ON, Scan Fingerprint
02:42:01.517 -> Image taken
02:42:01.750 -> Image converted
02:42:01.890 -> Found a print match!
02:42:01.890 -> Found ID #3 with confidence of 93
02:42:02.448 -> Fingerprint matches, Vehicle is Started and Engine is ON
02:42:06.558 -> Engine OFF, Turn ON Key
02:42:06.558 -> Engine OFF, Turn ON Key
02:42:06.643 -> Engine OFF, Turn ON Key
02:42:06.690 -> Engine OFF, Turn ON Key
```

☑ Autoscroll ☑ Show timestamp — Newline — 19200 baud — Clear output

Figure 4.14 Output for serial monitor

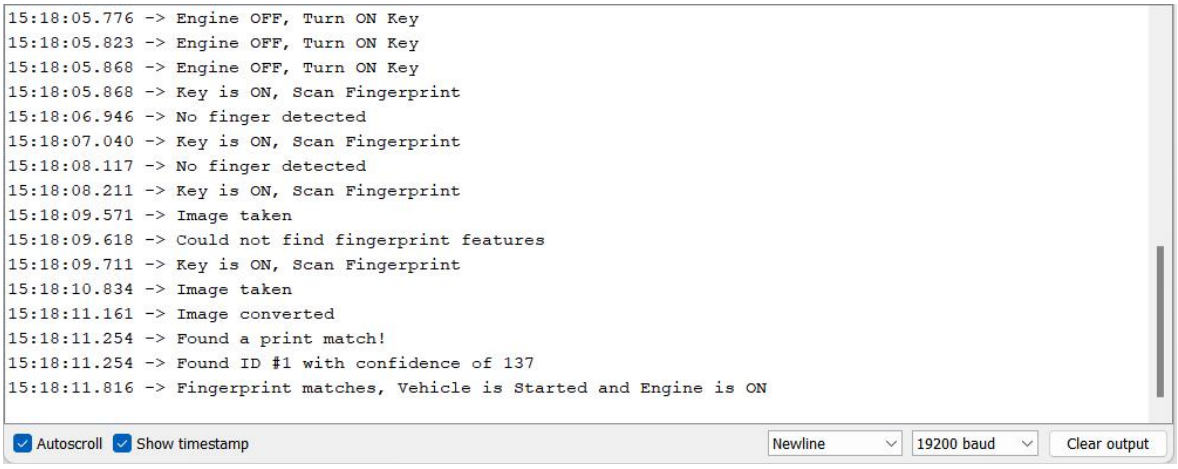
The figure 4.14 above shown the output of serial monitor in Arduino IDE. The figure shown the output that the Engine is in OFF condition, and need to turn ON the switch to ON the system. Then, the user need to place the fingerprint on the fingerprint sensor. The results shown, the fingerprint is matched with the fingerprint ID #3 and the vehicle is started and the engine is ON, but unfortunately the error of the results shown the vehicle is stolen even the fingerprint is matched. So, the alarm was turn ON after the fingerprinting.



```
15:12:58.301 -> Engine OFF, Turn ON Key
15:12:58.347 -> Engine OFF, Turn ON Key
15:12:58.395 -> Engine OFF, Turn ON Key
15:12:58.442 -> Engine OFF, Turn ON Key
15:12:58.489 -> Engine OFF, Turn ON Key
15:12:58.533 -> Engine OFF, Turn ON Key
15:12:58.533 -> Key is ON, Scan Fingerprint
15:12:59.703 -> No finger detected
15:12:59.797 -> Key is ON, Scan Fingerprint
15:13:00.871 -> No finger detected
15:13:00.963 -> Key is ON, Scan Fingerprint
15:13:02.087 -> Image taken
15:13:02.416 -> Image converted
15:13:02.512 -> Found a print match!
15:13:02.512 -> Found ID #1 with confidence of 126
15:13:03.071 -> Fingerprint matches, Vehicle is Started and Engine is ON
```

☑ Autoscroll ☑ Show timestamp — Newline — 19200 baud — Clear output

Figure 4.15 Output for serial monitor

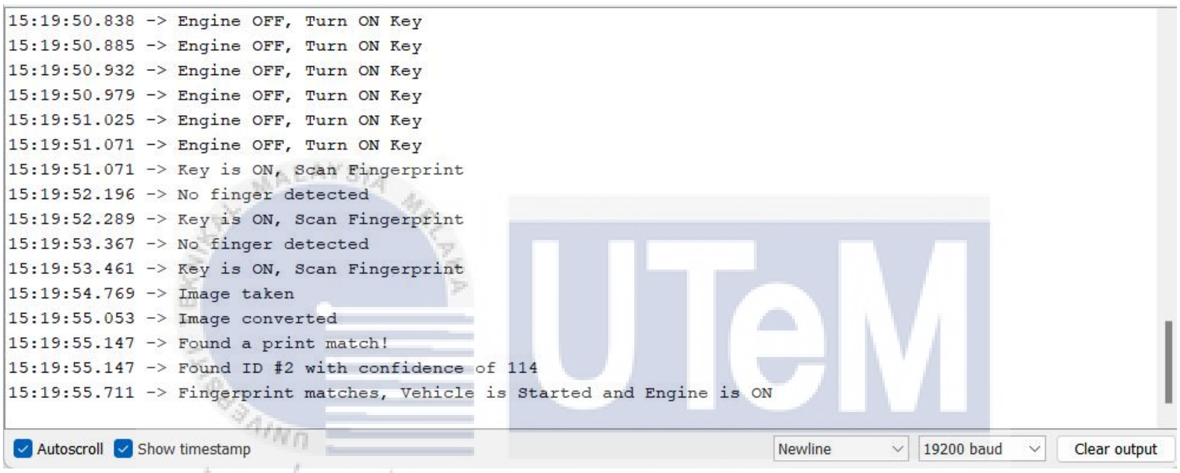


```
15:18:05.776 -> Engine OFF, Turn ON Key
15:18:05.823 -> Engine OFF, Turn ON Key
15:18:05.868 -> Engine OFF, Turn ON Key
15:18:05.868 -> Key is ON, Scan Fingerprint
15:18:06.946 -> No finger detected
15:18:07.040 -> Key is ON, Scan Fingerprint
15:18:08.117 -> No finger detected
15:18:08.211 -> Key is ON, Scan Fingerprint
15:18:09.571 -> Image taken
15:18:09.618 -> Could not find fingerprint features
15:18:09.711 -> Key is ON, Scan Fingerprint
15:18:10.834 -> Image taken
15:18:11.161 -> Image converted
15:18:11.254 -> Found a print match!
15:18:11.254 -> Found ID #1 with confidence of 137
15:18:11.816 -> Fingerprint matches, Vehicle is Started and Engine is ON
```

☑ Autoscroll ☑ Show timestamp

Newline 19200 baud

Figure 4.16 Output for serial monitor

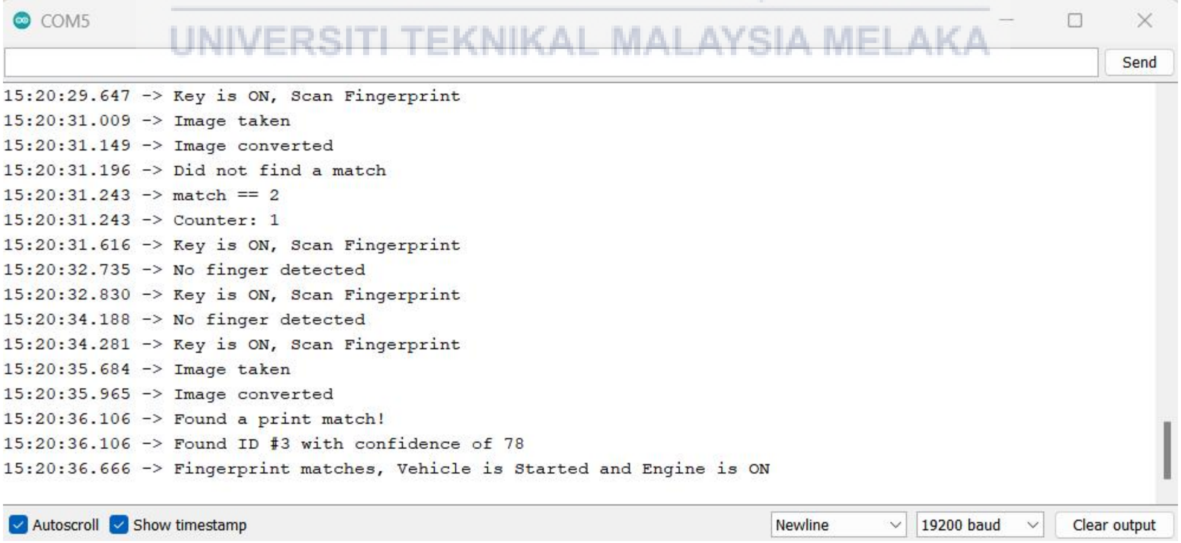


```
15:19:50.838 -> Engine OFF, Turn ON Key
15:19:50.885 -> Engine OFF, Turn ON Key
15:19:50.932 -> Engine OFF, Turn ON Key
15:19:50.979 -> Engine OFF, Turn ON Key
15:19:51.025 -> Engine OFF, Turn ON Key
15:19:51.071 -> Engine OFF, Turn ON Key
15:19:51.071 -> Key is ON, Scan Fingerprint
15:19:52.196 -> No finger detected
15:19:52.289 -> Key is ON, Scan Fingerprint
15:19:53.367 -> No finger detected
15:19:53.461 -> Key is ON, Scan Fingerprint
15:19:54.769 -> Image taken
15:19:55.053 -> Image converted
15:19:55.147 -> Found a print match!
15:19:55.147 -> Found ID #2 with confidence of 114
15:19:55.711 -> Fingerprint matches, Vehicle is Started and Engine is ON
```

☑ Autoscroll ☑ Show timestamp

Newline 19200 baud

Figure 4.17 Output for serial monitor



```
15:20:29.647 -> Key is ON, Scan Fingerprint
15:20:31.009 -> Image taken
15:20:31.149 -> Image converted
15:20:31.196 -> Did not find a match
15:20:31.243 -> match == 2
15:20:31.243 -> Counter: 1
15:20:31.616 -> Key is ON, Scan Fingerprint
15:20:32.735 -> No finger detected
15:20:32.830 -> Key is ON, Scan Fingerprint
15:20:34.188 -> No finger detected
15:20:34.281 -> Key is ON, Scan Fingerprint
15:20:35.684 -> Image taken
15:20:35.965 -> Image converted
15:20:36.106 -> Found a print match!
15:20:36.106 -> Found ID #3 with confidence of 78
15:20:36.666 -> Fingerprint matches, Vehicle is Started and Engine is ON
```

☑ Autoscroll ☑ Show timestamp

Newline 19200 baud

Figure 4.18 Output for serial monitor

Based from Figure 4.15, Figure 4.16 Figure 4.17, and Figure 4.18 shown the output for serial monitor of fingerprint. The results shown, the fingerprint is matched with the fingerprint ID two time of ID #1, ID #2 and ID #3. So, the vehicle is started and the engine is ON without any error. In conclusion, the successful stored fingerprint can turn ON the engine and the vehicle is started without turning the alarm ON.

	Time Taken for engine to start after place the matched fingerprint (s)									
Number of Trial	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Number of ID										
#1	1.001	0.962	1.041	1.003	1.040	0.842	1.042	1.041	1.164	1.041
#2	1.041	0.922	0.882	0.922	1.040	1.082	1.041	0.922	1.123	1.039

Table 4.2 Time taken for engine to start after place the matched fingerprint

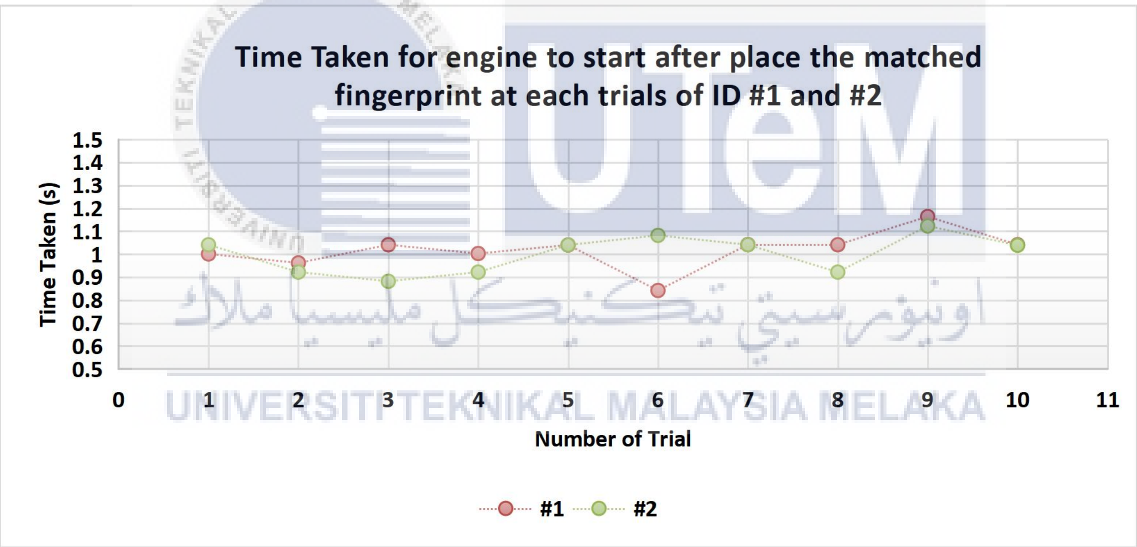


Figure 4.19 Graph of time taken for engine to start after place the matched fingerprint

From the graph, the trends shows that the time taken for the engine to start after placed the matched fingerprint at each trials of ID #1 and #2 is almost constant. The time taken is between range of 0.8s-1.2s, which is meant that the fingerprint only need less than 1.5s to start the vehicle engine. So, the duration of time it takes for an engine to start after matching a fingerprint might vary depending on a number of factors. A fingerprint-based engine start system's hardware, software, and implementation specifics can all affect how quickly the system responds overall.

4.5 Summary

In summary, Based on the objective of the project to design of Fingerprint-Based Vehicle Starting System. The main goal was to use fingerprint identification technology to create a safe and practical vehicle starting system. This required designing how fingerprint sensors would be integrated with the car's ignition mechanism.

Second objective is construction of an operating prototype, The project's goal was to turn the design into a workable prototype. In this stage, the required hardware and software components for the fingerprint-based car starting system were physically implemented.

Last objective is performance analysis of the prototype. The project attempted to analyse the efficacy and efficiency of the developed prototype. This entailed evaluating the precision, responsiveness, and general dependability of the fingerprint recognition technology in initiating the car.

However, based on this projects the fingerprint Recognition is success. The system's fingerprint recognition component was put into place without any problems. The prototype showcased the effectiveness of the biometric technology integration by demonstrating the capacity to store fingerprints safely and match them to start the car.

But unfortunately, there is difficulties in Integrating GSM and GPS. In spite of the fingerprint recognition system's success, there were difficulties in integrating GSM and GPS features. These features failed because of coding flaws that were identified by the project. The expected features, such communication capability (GSM) and remote tracking (GPS), were thus not completely realized.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

The project's main goal was effectively accomplished by creating and deploying a workable fingerprint-based car starting system. The system's dependable fingerprint detection improved the ignition process's security and user friendliness.

Unfortunately, difficulties encountered throughout the coding process interfered with the GSM and GPS features' integration, causing them to malfunction. Solving these code problems is essential to releasing the system's full potential and enabling capabilities like remote tracking and communication.

To sum up, the research establishes a strong basis for a cutting-edge and safe car ignition system. Future work will focus on fixing code problems, streamlining the GPS and GSM connection, and improving the system further for real-world use. The successful integration of these improvements will aid in the development of an all-encompassing and effective smart ignition vehicle system.

5.2 Potential for Commercialization

There is a lot of commercial potential in the development of a fingerprint-based smart ignition car system. There is a few salient features underscore its potential for commercialization.

First, Enhancing Security. By adding a biometric layer for authentication, fingerprint recognition technology improves vehicle security. For customers searching for cutting-edge car security solutions, this feature is really alluring.

Second, Practicality and User Experience. The fingerprint-based ignition system provides an easy-to-use and simple interface. With just a touch, users may start their cars, doing away with the need for conventional keys or key fobs. This ease of use is probably going to draw in a wide audience.

Third, The need in the market for smart car systems. The need for linked and intelligent car systems is rising in the automotive sector. Your project fits in with this trend by providing a distinctive and cutting-edge solution that will appeal to tech-savvy customers.

Next, Distinctiveness in the Marketplace. The integration of fingerprint technology sets your project unique from typical car ignition systems. This distinction has the potential to be a powerful selling factor, particularly in a market where buyers are looking for cutting-edge and distinctive features in their cars.

Last but not least, Possibility of Integration with Current Automobiles. The design of your project could be able to work with not simply new models but also older ones. Its versatility expands its market potential by making it appealing to a wider variety of car buyers.

5.3 Future Works

The following might be done to improve the accuracy of the project estimating findings in the future:

i. Fixing Coding Mistakes. Resolve and fix the coding problems affecting the GPS and GSM functions. This step is essential for releasing your system's full potential and turning on features like communication and remote tracking.

ii. Optimizing the Integration of GPS and GSM. After resolving code concerns, concentrate on refining the GSM and GPS integration. Assure smooth connection between these technologies and the vehicle system to improve connectivity, anti-theft measures, and remote tracking.

iii. Improved Security Elements. To improve the vehicle's overall security even more, consider adding more security measures. Multi-factor authentication, encrypted communication methods, and interaction with car security systems are a few examples of this.

iv. Enhancements to the User Interface and Usability. To improve usability and user experience, think about improving the user interface. Positive system contact is enhanced with a user-friendly interface that provides clear instructions and feedback.

REFERENCES

[1] Patents: Automobile-theft preventer US 1300150 A. [online] Accessed 12/10/15. Available from: http://www.google.com/patents/US1300150.

[2] Security of car keys. Andrej Simko. 04/25/14. Accessed 12/10/15. Available from: http://www.slideshare.net/Andrejimko/security-of-car-keys.

[3] Richardson, A. (n.d.). *SECURITY OF VEHICLE KEY FOBS AND IMMOBILIZERS*.https://www.cs.tufts.edu/comp/116/archive/fall2015/arichardson.pdf

[4] Samuel, Henry. Three Quarters of Cars Stolen in France 'Electronically Hacked.' The Telegraph. 10/29/15. Accessed 12/10/15. Available from: http://www.telegraph.co.uk/news/worldnews/europe/france/11964140/Three-quarters-of-cars-stolen-in-France-electronically-hacked.html

[5] Exploiting RFIDs Car Immobilizers and the ExxonMobil Speed pass. Bono, Stephen and Green, Mathew and Stubblefield, Adam and Rubin, Avi. Accessed 12/10/15.Available from: https://securityevaluators.com/knowledge/case_studies/rfid/

[6] LOZIER, Herbert. 90 Firsts in American Automotive History. Popular Science. New York: Time4 Media, 1964, pp. 81-83. Accessed 12/10/15. Available from: https://encrypted.google.com/books?id=_iwDAAAAMBAJ&pg=PA80&lpg=PA80&dq=automotive+firsts&source=bl&ots=HmsMDHdRn&sig=7e_6YR85hodRWm50gtphsei23s&hl=en&ei=G1NwTLPRG8Tflgf68OTODg&sa=X&oi=book_result&ct=result&resnum=8&ve d=0CDcQ6AEwBw#v=onepage&q&f=false

[7] Open Garage. Car Hacker's Handbook. Accessed 12/10/15. Available from: http://opengarages.org/handbook/2014_car_hackers_handbook_compresse d.pdf

[8] Requirements of Remote Keyless Entry Systems. Maxim Integrated. Accessed 12/10/15. Available from: https://www.maximintegrated.com/en/app-notes/index.mvp/id/3395

[9] National Forensic Science Technology Center. 2009. *A Simplified Guide to FingerprintAnalysis* <https://www.forensicsciencesimplified.org/prints/Fingerprints.pdf>

[10] Jamil Abedalrahim Jamil Alsayaydeh, Win Adiyansyah Indra, Adam Wong Yoon Khang, Vadym Shkarupylo, & Dhanigaletchmi A. P. P. Jkatisan. (2019). *DEVELOPMENT of VEHICLE IGNITION USING FINGERPRINT*, *14*(23), 4045–4053.

[11] 'Advantages and Disadvantages of Microcontroller'. 2020. *GeeksforGeeks* (GeeksforGeeks) <https://www.geeksforgeeks.org/advantages-and-disadvantages-of-microcontroller/> [accessed 22 June 2023]

[12] ARUN BATI, SRUSHTI V C, PRIYANKA G, AKASH R, & Prof. GAGANAMBHA. (2022). *FINGER PRINT VEHICLE STARTER and THEFT CONTROL USING GPS*, *10*(6), C822–C827.

[13] Yusuf, S. D., Dalhatu, A., Umar, I., & Loko, A. Z. (2022). Simulation and Construction of Fingerprint Vehicle Starter System Using Microcontroller. *International Journal of Research and Scientific Innovation*, *09*(09), 135–144. https://doi.org/10.51244/ijrsi.2022.9912

[14] Karthikeyan.A&Sowndharya.J, "Fingerprint Based Ignition System", International Journal Of Computational Engineering Research, Vol. 2, Issue No.2, pp. 236-243. 2012.

[15] Dr.V.Nandagopal, Dr.V.Maheswari, & C.Kannan. (2018). *VEHICLE STARTING SYSTEM USING FINGER PRINT*, *119*(18), 1753–1760.

[16] Abhilash Mudpe, Gauravi Narote, Mahendra Mahale, Priyanka Arshanapelli, & Ms. Priyanka Kothoke. (2020). *Finger Print Based Vehicle Start Using Arduino*, *9*(4), 2293–2299.

[17] Gopu Priyanka, Bala Bhadruni Pranavi, Krishnamsetty Manaswini, A. S. R. Sai Srinivas, & A. V. Rajan. (2020). *Fingerprint Based Vehicle Starter and Vehicle Tracking System*, *3*(5), 1195–1197.

[18] Pratiksha Jadhav, Nadira Mulla, Namrata Sutar, Sakshi Mane, & S. H. Mali. (2022). *Fingerprint Starter Vehicle Using Arduino* [Review of *Fingerprint Starter Vehicle Using Arduino*]. *8*(3), 3585–3589.

[19] Rohan Chandra Pradhan, Saumya Rai, Smit M Shah, & Brinda. (2018). *Biometric Security System Using Arduino for Vehicles*, *6*(10), 24–27.

[20] AVNISH TIWARI, PAYAL VERMA, & JYOTIKA MISHRA. (n.d.). *Fingerprint Vehicle Starter Project*, 1–8.

[21] Rajat Verma, Raj Kumar, Kanishk Rawat, Ankit Pawar, & Hitesh Maheshwari. (2019). *SECURING VEHICLES with FINGERPRINT BASED SECURITY*, *6*(8), 5392–5394.

APPENDICES

A) Coding 1

```
#include <Adafruit_Fingerprint.h>


#if (defined(__AVR__) || defined(ESP8266))
&& !defined(__AVR_ATmega2560__)
// For UNO and others without hardware serial, we must use
software serial...
// pin #2 is IN from sensor (GREEN wire)
// pin #3 is OUT from arduino  (WHITE wire)
// Set up the serial port to use softwareserial..
SoftwareSerial mySerial(4, 5);

#else
// On Leonardo/M0/etc, others with hardware serial, use
hardware serial!
// #0 is green wire, #1 is white
#define mySerial Serial1

#endif


Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);

uint8_t id;

void setup()
{
  Serial.begin(9600);
  while (!Serial);  // For Yun/Leo/Micro/Zero/...
  delay(100);
  Serial.println("\n\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); }
  }

  Serial.println(F("Reading sensor parameters"));
  finger.getParameters();
  Serial.print(F("Status:                               0x"));
Serial.println(finger.status_reg, HEX);
  Serial.print(F("Sys                ID:                0x"));
Serial.println(finger.system_id, HEX);
  Serial.print(F("Capacity:                               "));
Serial.println(finger.capacity);
  Serial.print(F("Security             level:             "));
Serial.println(finger.security_level);
  Serial.print(F("Device             address:             "));
Serial.println(finger.device_addr, HEX);
  Serial.print(F("Packet               len:               "));
Serial.println(finger.packet_len);
  Serial.print(F("Baud                rate:               "));
Serial.println(finger.baud_rate);
}

uint8_t readnumber(void) {
  uint8_t num = 0;

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

void loop()                     // run over and over again
{
  Serial.println("Ready to enroll a fingerprint!");
  Serial.println("Please type in the ID # (from 1 to 127) you
want to save this finger as...");
  id = readnumber();
  if (id == 0) {// ID #0 not allowed, try again!
     return;
  }
  Serial.print("Enrolling ID #");
  Serial.println(id);

  while (!  getFingerprintEnroll() );
}

uint8_t getFingerprintEnroll() {

  int p = -1;
  Serial.print("Waiting  for  valid  finger  to  enroll  as  #");
Serial.println(id);
  while (p != FINGERPRINT_OK) {
    p = finger.getImage();
    switch (p) {
```

```
  case FINGERPRINT_OK:
    Serial.println("Image taken");
    break;
  case FINGERPRINT_NOFINGER:
    Serial.println(".");
    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 #");  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;
  }

  return true;
}
```

B) Coding 2

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <SoftwareSerial.h>
#include <Adafruit_Fingerprint.h>

#define buzzerPin 6
#define keyPin  9
#define enginePin  13

int keyState;
int engineStopState;

int match = 0;
bool isStolen = false;
int counter = 0;
int engineStop = 2;
int stolenFlagPin = 3;

SoftwareSerial mySerial(4, 5);
LiquidCrystal_I2C lcd = LiquidCrystal_I2C(0x27, 16, 2);
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);

void setup() {

  Serial.begin(19200);
  pinMode(buzzerPin, OUTPUT);
  pinMode(enginePin, OUTPUT);
  pinMode(stolenFlagPin, OUTPUT);
  pinMode(engineStop, INPUT);
  pinMode(keyPin, INPUT);

  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("Scan");
  lcd.setCursor(0, 1);
  lcd.print("Fingerprint");
  Serial.println("Place Fingerprint");
  finger.begin(57600);
  delay(5);
  if (finger.verifyPassword()) {
    Serial.println("Found fingerprint sensor!");
  } else {
    Serial.println("Did not find fingerprint sensor :(");
    while (1) {
      delay(1);
    }
  }

  Serial.println(F("Reading sensor parameters"));
  finger.getParameters();
  finger.getTemplateCount();

  if (finger.templateCount == 0) {
    Serial.print("Sensor doesn't contain any fingerprint data.
Please run the 'enroll' example.");
  }
  else {
    Serial.println("Waiting for valid finger...");
    Serial.print("Sensor              contains              ");
Serial.print(finger.templateCount);           Serial.println("
templates");
  }

  digitalWrite(stolenFlagPin, LOW);
  digitalWrite(buzzerPin, HIGH);
  delay(100);
  digitalWrite(buzzerPin, LOW);
  delay(100);
  digitalWrite(buzzerPin, HIGH);
  delay(100);
  digitalWrite(buzzerPin, LOW);
  delay(100);
  digitalWrite(buzzerPin, HIGH);
  delay(100);
  digitalWrite(buzzerPin, LOW);
}


void loop() {
  Serial.println("Engine OFF, Turn ON Key");
  keyState = digitalRead(keyPin);
  while (keyState == HIGH) {
    Serial.println("Key is ON, Scan Fingerprint");
    keyState = digitalRead(keyPin);
    getFingerprintID();
    delay(50);
    getMatch();
    delay(50);
  }
  digitalWrite(enginePin, LOW);
  delay(50);
}

uint8_t getFingerprintID() {
  finger.begin(57600);
```

```
uint8_t p = finger.getImage();
switch (p) {
  case FINGERPRINT_OK:
    Serial.println("Image taken");
    break;
  case FINGERPRINT_NOFINGER:
    Serial.println("No finger detected");
    return p;
  case FINGERPRINT_PACKETRECIEVEERR:
    Serial.println("Communication error");
    return p;
  case FINGERPRINT_IMAGEFAIL:
    Serial.println("Imaging error");
    return p;
  default:
    Serial.println("Unknown error");
    return p;
}

// OK success!

p = finger.image2Tz();
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!
p = finger.fingerSearch();
if (p == FINGERPRINT_OK) {
  Serial.println("Found a print match!");
} else if (p == FINGERPRINT_PACKETRECIEVEERR) {
    Serial.println("Communication error");
    return p;
  } else if (p == FINGERPRINT_NOTFOUND) {
    Serial.println("Did not find a match");
    match = 2;
    return p;
  } else {
    Serial.println("Unknown error");
    return p;
  }

  // found a match!
  Serial.print("Found ID #"); Serial.print(finger.fingerID);
  Serial.print("       with       confidence       of       ");
Serial.println(finger.confidence);
  match = 1;
  return finger.fingerID;
}

// returns -1 if failed, otherwise returns ID #
int getFingerprintIDez() {
  uint8_t p = finger.getImage();
  if (p != FINGERPRINT_OK)  return -1;

  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);
  match = 1;
  return finger.fingerID;

}

void getMatch() {
  if (match == 2) {
    Serial.println("match == 2");
    counter++;
    Serial.println("Counter: " + String(counter));
    match = 0;
    digitalWrite(buzzerPin, HIGH);
    delay(100);
    digitalWrite(buzzerPin, LOW);
    delay(100);
    digitalWrite(buzzerPin, HIGH);
    delay(100);
    digitalWrite(buzzerPin, LOW);

    if (counter > 2) {
      // 3 times doesnt match - Vehicle is stolen //
      Serial.println("3 times fingerprint does not match,
vehicle is stolen");
      digitalWrite(stolenFlagPin, HIGH); //send SMS arduino to
send SMS
      isStolen = true;
      delay(5000);
      while (isStolen) {
        Serial.println("Vehicle is stolen!");
        digitalWrite(enginePin, LOW);
        digitalWrite(buzzerPin, HIGH);
        delay(300);
        digitalWrite(buzzerPin, LOW);
        delay(300);
        engineStopState = digitalRead(engineStop);
        if (engineStopState == LOW) {
          isStolen = false;
          digitalWrite(stolenFlagPin, LOW);
          counter = 0;
          digitalWrite(buzzerPin, LOW);
        }
      }
    }
  }

  if (match == 1) {
    digitalWrite(buzzerPin, HIGH);
    delay(500);
    digitalWrite(buzzerPin, LOW);
    Serial.println("Fingerprint matches, Vehicle is Started
and Engine is ON");
    while (match == 1) {
      digitalWrite(enginePin, HIGH);
      engineStopState = digitalRead(engineStop);
      if (engineStopState == HIGH) {
        isStolen = true;
      }
      while (isStolen) {
        Serial.println("Vehicle is stolen!");
        digitalWrite(enginePin, LOW);
        digitalWrite(buzzerPin, HIGH);
        delay(300);
        digitalWrite(buzzerPin, LOW);
        delay(300);
        match = 0;
        engineStopState = digitalRead(engineStop);
        if (engineStopState == LOW) {
          isStolen = false;
          digitalWrite(stolenFlagPin, LOW);
          digitalWrite(buzzerPin, LOW);
        }
      }
      keyState = digitalRead(keyPin);
      if (keyState == LOW) {
        digitalWrite(enginePin, LOW);
        digitalWrite(buzzerPin, LOW);
        match = 0;
      }
      delay(500);
    }
  }
}
```

C) Coding 3

```
#include <NMEAGPS.h>
#include <SoftwareSerial.h>

// Configure software serial port

// Variable to store text message
String textMessage, getGPS, lati, logi, link;

String phone = "+601126502500";
NMEAGPS gps;  //Declared GPS Object
NMEAGPS fix;  // all GPS fields like location, etc.
uint8_t fixCount;
int engineStop = 2;
int stolenFlagPin = 3;
int stolenFlagPinState;
int stolenFlagPinStateCounter = 0;
SoftwareSerial serialgps(5, 6);
SoftwareSerial SIM900(7, 8);

void setup()
{
  Serial.begin(19200);
  SIM900.begin(19200);
  Serial.println( F("Start") );

  // AT command to set SIM900 to SMS mode
  SIM900.print("AT+CMGF=1\r");
  delay(100);
  // Set module to send SMS data to serial out upon receipt
  SIM900.print("AT+CNMI=2,2,0,0,0\r");
  delay(100);
  pinMode(engineStop, OUTPUT);
  pinMode(stolenFlagPin, INPUT);
  digitalWrite(engineStop, LOW);
  //delay(10000);
}

void loop()
{
  NMEAGPS fix;  // Declare fix here

  stolenFlagPinState = digitalRead(stolenFlagPin);
  if (stolenFlagPinState == HIGH && stolenFlagPinStateCounter
== 0) {
    digitalWrite(engineStop, HIGH);
    link = "3 times fingerprint does not match, vehicle is
stolen";
    sms();
    stolenFlagPinStateCounter++;
  }

  else if (stolenFlagPinState == LOW) {
    stolenFlagPinStateCounter = 0;
  }



  // Check for GPS characters and parse them
  if (SIM900.available() > 0) {
    textMessage = SIM900.readString();
    Serial.print(textMessage);
    delay(10);
  }

  if (textMessage.indexOf("ENGINEOFF") >= 0) {
    getGPS = "TRUE";
    serialgps.begin(9600);
    textMessage = "";
    digitalWrite(engineStop, HIGH);
    Serial.println("Alarm Set ON and Engine Set OFF");
  }

  if (textMessage.indexOf("ALARMOFF") >= 0) {
    textMessage = "";
    Serial.println("Alarm Set OFF");
    //digitalwrite to flag On engine
    digitalWrite(engineStop, LOW);
  }

  if (getGPS == "ALARM") {
    SIM900.begin(19200);
    Serial.println( F("Alarm State") );
    // AT command to set SIM900 to SMS mode
    SIM900.print("AT+CMGF=1\r");
    delay(100);
    // Set module to send SMS data to serial out upon receipt
    SIM900.print("AT+CNMI=2,2,0,0,0\r");
    delay(100);
    getGPS = "FALSE";
  }

  if (getGPS == "TRUE") {
    Serial.println("Shut Down Engine");
    if (gps.available())
    {
      // Once per second, a complete GPS fix structure is
ready.  Get it.
      fix = gps.fix();

      // Count elapsed seconds
      fixCount++;

      if (fixCount >= 5)
      {
        fixCount = 0; // reset counter
        printGPSdata();
      }
    }
    //textMessage = "";
  }

}


void printGPSdata()
{
  Serial.print( F("Latitud/Longitud: ") );
  if (fix.valid.location) {
    lati = String(fix.latitude(), 5);
    logi =  String(fix.longitude(), 5 );
    Serial.print( lati);
    Serial.print( F(", ") );
    Serial.print( logi);
  }
  Serial.println();
  link = "Location (latitude,logitude): " + lati + ", " + logi
+ " .Engine is Immobilized" ;
  delay(3000);
  getGPS = "ALARM";
  sms();
}


void sms()
{
  SIM900.begin(19200);
  SIM900.println("AT+CMGF=1");
  delay(200);
  SIM900.println("AT+CMGS=\"" + phone + "\"\r");
  delay(200);
  SIM900.print(link);
  SIM900.println (char(26)); // ctrl-z
  delay(1000);
  Serial.println("SMS Sent");
  delay(10000);
}
```