# Visvesvaraya Technological University, Belagavi.

PROJECT REPORT

on

# "BIOMETRIC BASED CAR UNLOCKING SYSTEM"

**Project report submitted in partial fulfillment of the requirement for the award of the degree of**

**Bachelor of Engineering**

**in**

**Electronics and Communication Engineering**

For the academic year 2019-20

Submitted by

| USN | Name |
|---|---|
| 1CR15EC095 | Madhu R |
| 1CR15EC168 | S Venkat Parrdhu |
| 1CR15EC066 | B E Harsha Vardhan |

Under the guidance of

Internal
Guide 1- Prof.Richa Tengshe
Designation-Professor
Department of ECE
CMRIT, Bangalore

Department of Electronics and Communication Engineering

**CMR Institute of Technology, Bengaluru – 560 037**

**For college use only**

| Reviewer Names | Sign | Accepted | Correction Needed | Rejected |
|---|---|---|---|---|
| 1. | | | | |
| 2. | | | | |

# CERTIFICATE

This is to certify the Project work entitled “**Biometric based car unlocking system**”, carried by the following bonafide students of **CMR Institute of Technology, Bengaluru** in partial fulfillment of the requirements for the award of **Bachelor of Engineering in Telecommunication Engineering** of the V**isvesvaraya Technological University, Belagavi-590018** during the academic year 2019-20. This is certified that all the corrections and suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The Project report has been approved as it satisfies the academic requirements prescribed for the said degree.

| USN | Name |
|---|---|
| 1CR15EC095 | Madhu R |
| 1CR15EC168 | S Venkat Parrdhu |
| 1CR15EC066 | B E Harsha Vardhan |

| | | |
|---|---|---|
| ---------------------- | ----------------------- | ----------------------- |
| **Signature of Guide** | **Signature of HoD** | **Signature of Principal** |
| **Mrs. Richa Tengshe** | **Dr. R Elumalai** | **Dr. Sanjay Jain** |
| **Professor** | **HOD and Professor** | **CMRIT** |
| **Dept. of ECE, CMRIT** | **Dept. of ECE, CMRIT** | |

## External Viva

**Name of the Examiner** **Signature with date**

**1.**

# ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without mentioning the people whose proper guidance and encouragement has served as a beacon and crowned my efforts with success. We take an opportunity to thank all the distinguished personalities for their enormous and precious support and encouragement throughout the duration of this seminar.

We take this opportunity to express our sincere gratitude and respect to **CMR Institute of Technology, Bangalore** for providing us the opportunity to carry out our project work.

We have a great pleasure in expressing our deep sense of gratitude to **Dr. Sanjay Jain,** Principal, CMRIT, Bangalore, for his constant encouragement.

With a profound sense of gratitude, we acknowledge the guidance and support extended by **Dr.R Elumalai ,** HoD and **Prof.Richa Tengshe** ,Professor , Department of Electronics and Communication Engineering, CMRIT, Bangalore. Their incessant encouragement and invaluable technical support have been of immense help in realizing this project work. Their guidance gave us the environment to enhance our knowledge, skills and to reach the pinnacle with sheer determination, dedication and hard work.

We also extend our thanks to the faculties of Electronics and Communication Department who directly or indirectly encouraged us throughout the course of project work.

We also thank our parents and friends for all their moral support they have given us during the completion of this work.

# Table of Contents

Chapter 1

# INTRODUCTION

## 1.1 Problem statement

Car is a mode of transport, pride, and necessity. In common, out of 10 families 3 are affording a car for transport, 3 are purchasing for showcase, 2 for necessity, especially in India. Usual count of a family in India varies from 3-5 members. For traveling, irrespective of distance, people use car for transportation. Cost of car, depends on model they purchase, varies from lakhs to crores. Knowing about this fact, even a random person tries cheap ways of affording a car(includes illegal methods).Typically in India, car lock comprises of key, button system. Breaking into these lock systems requires some techniques and simple gears.

1. The notch that is present in the door side can be easily altered by simply tapping on it using a flat hard metallic scale
2. Electronic button unlocking system can be hacked using a device which absorbs the bandwidth of radio waves Emitted by the button unlocking system and use the same signal to unlock the car.

Our purpose is to prevent losing of car by upgrading the unlocking system using biometric, specifically fingerprint.

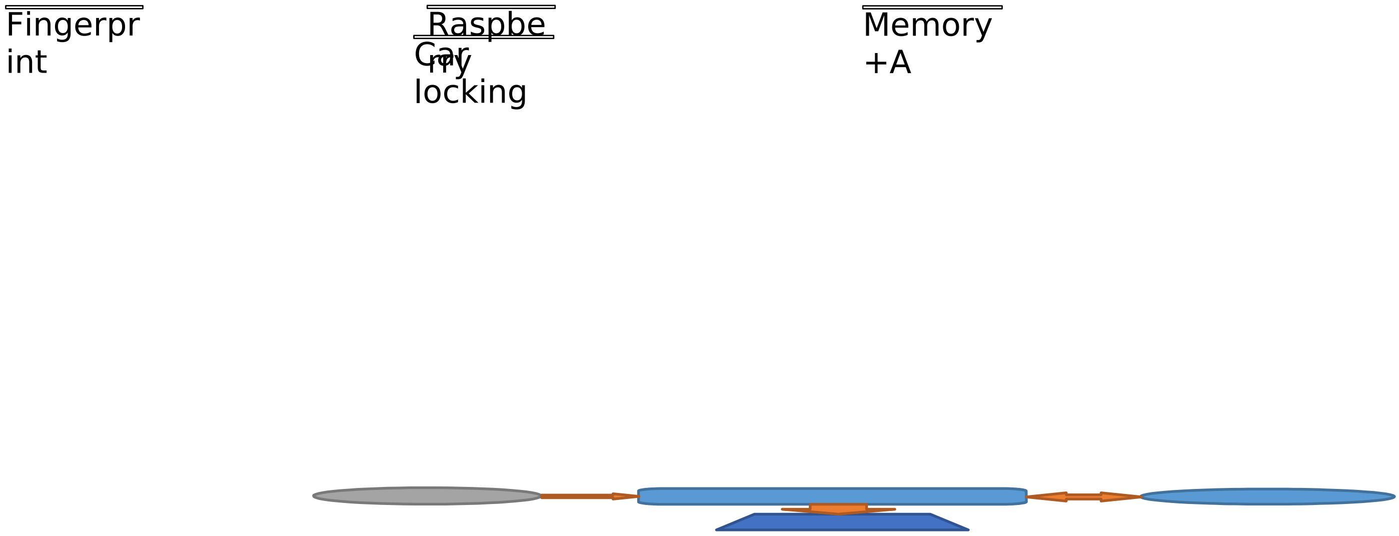
## 1.2 Solution to problem

Procedure for fingerprint verification technique is as follows

1. First each authorized person will register their fingerprint by scanning through sensor
2. Store the fingerprints in memory
3. If the person has to access the car, they should verify their fingerprint on scanner present at the handle.
4. When it is verified, it will unlock the car

## 1.3 Block diagram

The block diagram is described below



## 1.4 Explanation of block diagram

### Fingerprint sensor:

Finger-scan technology is the most widely deployed biometric technology, with a number of different vendors offering a wide range of solutions. Among the most remarkable strengths of fingerprint recognition, we can mention the following:

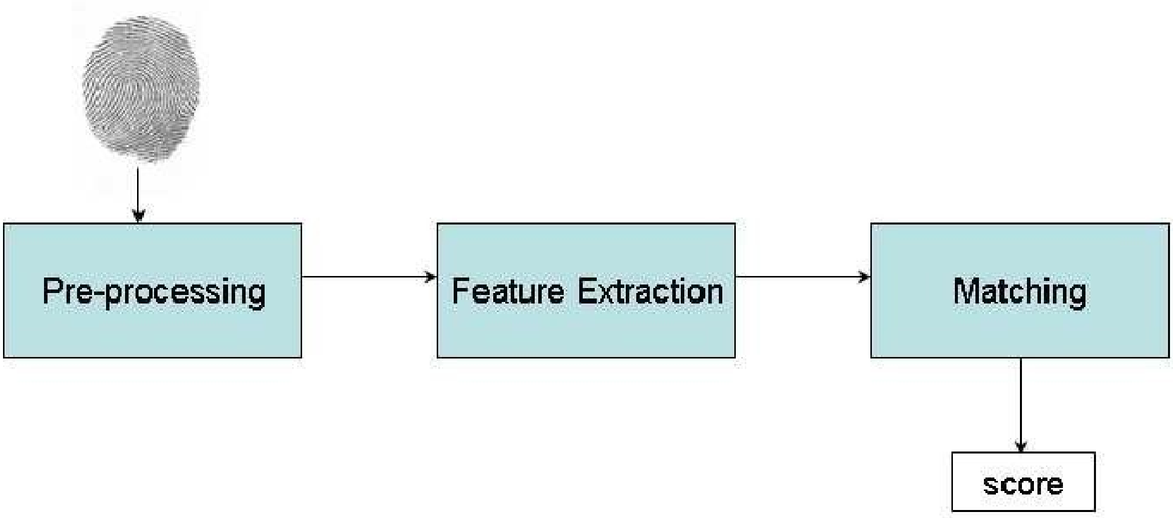
- Its maturity, providing a high level of recognition accuracy.
- The growing market of low-cost small-size acquisition devices, allowing its use in a broad range of applications, e.g., electronic commerce, physical access, PC logon, etc.
- The use of easy-to-use, ergonomic devices, not requiring complex user-system interaction.

On the other hand, a number of weaknesses may influence the effectiveness of fingerprint recognition in certain cases:

- Its association with forensic or criminal applications.
- Factors such as finger injuries or manual working can result in certain users being unable to use a fingerprint-based recognition system, either temporarily or permanently.
- Small-area sensors embedded in portable devices may result in less information available from a fingerprint and/or little overlap between different acquisitions.

Basic modules of Fingerprint Recognition:

This section provides a basic introduction to fingerprint recognition systems and their main parts, including a brief description of the most widely used techniques and algorithms.



The main modules of a fingerprint verification system are:

*a) fingerprint sensing*, in which the fingerprint of an individual is acquired by a fingerprint scanner to produce a raw digital representation;

*b) pre-processing*, in which the input fingerprint is enhanced and adapted to simplify the task of feature extraction;

*c) feature extraction*, in which the fingerprint is further processed to generate discriminative properties, also called feature vectors; and *d) matching,* in which the feature vector of the input fingerprint is compared against one or more existing templates.

The templates of approved users of the biometric system, also called clients, are usually stored in a database. Clients can claim an identity and their fingerprints can be checked against stored fingerprints.

CPU(Central processing unit):

- CPU is the heart and brain
- It interprets and executes machine level instructions
- Controls data transfer from/to Main Memory (MM) and CPU
- Detects any errors
- It boots up the OS and handles h/w of

computer Car locking system:

Function:

The locking system in a vehicle must grant access only to authorized persons. It is the means via which the vehicle doors and boot lid are locked and unlocked and the engine is started. The locking system is operated with a key or remote control.

Different mechanical locking systems:

In years gone by, purely mechanical locking systems were the norm. Each door or lid has an independent mechanism which could be operated from the outside with a key or from the inside with a knob. Central locking systems, for which pneumatic drives were used originally, brought about significant improvements in comfort and convenience. These systems feature a built-in vacuum reservoir which triggers the locks on all doors when the key is turned in a lock.

Electric locking systems are common in today's vehicles. Most of these combine a key with infrared or wireless remote control. This means that they can be triggered remotely, i.e. without contact between key and vehicle. Today, most vehicle manufacturers only fit a lock which can be operated with a key in one door, so the car can be unlocked in an emergency. The very latest systems enable entirely keyless vehicle access. Drivers only need to have the transmitter in their pockets, for example. The doors are then unlocked when the driver touches a door handle which has a built-in contact point.

Components of central locking systems

The locking system comprises the following components: Door handle/Handle strip:

The door handle is the traditional means by which a vehicle is opened and closed from inside or outside. The external door strip usually houses the door lock. Door strips are increasingly used as design elements in modern cars. They can be chrome-plated or paint-finished in the same color as the vehicle.

Door lock/Actuator:

The latching mechanism in a vehicle is installed directly in its doors. It contains both a latch and an electric motor (actuator) which controls the central locking. The latch opens or closes the doors, whereas the door locks or unlocks the vehicle. Today, all door latches are powered by electric drives.

Fuel filler cap:

The fuel filler cap must securely seal the fuel tank. Some fuel filler caps have locks, others do not. Fuel filler caps with locks are usually found on vehicles which have either a fuel filler flap which does not lock or no fuel filler flap at all. Fuel filler caps without locks are found on vehicles whose fuel filler flap is locked automatically via the central locking system.

Transponder:

The transponder is usually integrated inside the key bow. It is the means by which the electronic immobilizer identifies that the correct key is being used. The transponder's code is read out as the key nears the ignition lock. If the code is correct, the electronic immobilizer sends the start enable to the engine.

Remote control:

Remote controls are being used with increasing frequency in small cars, replacing the functions of a conventional key to all intents and purposes. A signal transmitter sends a signal or a coded order instruction to a receiver inside the vehicle, which usually controls a number of functions. Infrared remote controls have a range of up to 15 m. They rely on direct "visual" contact between transmitter and receiver. Today, infrared remote controls are only used rarely as they have been overtaken by other technologies. Wireless remote controls transmit on radio frequencies and have a range of up to approximately 100 m.

Keys:

The basic function of keys and remote controls is the locking and unlocking of doors, luggage compartments, fuel filler caps, etc. they are also used to control the interior lighting electronic immobilizer alarm system and the window lifters.

The keys comprise two units: the milled, toothed key blade and the key bow. The latter is home to an increasing number of electronic functions such as the remote control for the central locking system or the boot lid.

Start/stop system:

Traditionally, a vehicle key was needed to unlock the steering lock and to start the engine. Subsequently, the vehicle key was enhanced with the addition of a transponder-based release mechanism for the electronic immobilizer. Today, keyless systems are increasingly being used to start engines. In a keyless system, a transmitter – which usually also houses the controller for the central locking – is inserted into a reader in the vehicle and the engine is then started by pressing a button. A more recent development has seen the use of systems that work without any contact at all. Here, it is sufficient to simply "take along" the transmitter (carrying it in a trouser pocket, for example) the pedals before starting the engine by pressing a button.

Steering lock:

Steering locks have been a mandatory requirement set by insurance companies since 1969. They provide protection against theft. They are the means by which the steering column is unlocked and the engine is started – either electrically or in by conventional mechanical means.

# Chapter 2

# LITERATURE SURVEY

## 2.1 Present existing products in market and their flaws

### Key locking system:

For several decades, car keys have been used to physically secure vehicles. Initially, simple mechanical keys were introduced to open the doors, unlock the steering, and operate the ignition lock to start the engine. Given physical access to a mechanical key, or at hand of a detailed photograph, it is possible

1. These authors contributed the research on Hitag2.

2. These authors contributed the research on VW Group. to create a duplicate. In addition, mechanical tumbler locks and disc locks are known to be vulnerable to techniques such as lock-picking and bumping that allow to operate a lock without the respective key. Finally, for most types of car locks, locksmith tools exist that allow to decode the lock and create a matching key.

### Electronics in a Car Key

With electronic accessories becoming available, additional features were integrated into the locking and starting systems of cars: some of them to improve the comfort, others to increase security. On the side of the car key, this implies some electronic circuitry integrated in its plastic shell

### 1.1.1 Immobilizer Transponders

One of the most notable events in the history of car security was the introduction of the immobilizer, which significantly reduced the number of stolen cars and so-called joyrides conducted by teenagers.

An electronic immobilizer improves the security of the car key with respect to starting the engine. Technically, most immobilizers rely on Radio Frequency IDentification (RFID) technology: An RFID transponder is embedded in the plastic shell of the car key and contains a secret that is required to switch on the ignition and start the engine. An antenna coil around the ignition lock establishes a bidirectional communication link and provides the energy for the transponder in order to verify its authenticity with a range of a few centimeters. All modern immobilizers use cryptography for authentication between transponder and vehicle, typically based on a challenge-response protocol.

For many years, only weak, proprietary cryptography was implemented in immobilizer transponders worldwide. This may have been caused by the limited energy available on RFID-powered devices, technological limitations, and cost considerations. The first type of immobilizer transponder to be broken was the widespread DST40 cipher used in Texas Instrument’s Digital Signature Transponder(DST), which was reverse-engineered and broken at Usenix Security 2005: The

40-bit secret key of the cipher can be revealed in a short time by means of exhaustive search. This paper was at the same time one of the first published attacks on a commercial device in the literature. A few years later, at Usenix Security 2012, researchers published several cryptanalytic attacks on NXP's Hitag2 transponders, the most widely used car immobilizer at that time. The authors showed that an attacker can obtain the 48-bit secret key required to bypass the electronic protection in less than 360 seconds. One year later, in a paper submitted to Usenix Security 2013 (and finally published in 2015), the security mechanism of the Megamos Crypto transponder were found to be vulnerable to cryptanalytic attacks. The 96-bit secret key of the cipher is mapped into a 57-bit state of a stream cipher that can be rolled back. A flawed key generation (multiple bits of the secret key are set to zero) additionally found in various transponders decreases the attack time from the order of days to a few seconds using a Time-Memory Tradeoff (TMTO).

As a result, the majority of RFID immobilizers used in today's vehicles can be cloned: the secret of the transponder can be obtained by an adversary to circumvent the added security provided by the immobilizer.

The cryptography of these immobilizers has to be considered broken as their added protection to prevent criminals from starting the engine of a car is very weak.

Button locking system:

Today, certain modern cars (especially made by luxury brands) are equipped with PKES systems that rely on a bidirectional challenge-response scheme, with a small operating range of about one meter:

When in proximity of the vehicle, the car key generates a cryptographic response to a challenge transmitted by the car. A valid response unlocks the doors, deactivates the alarm system, and enables the engine to start. As a consequence, the only remaining mechanical part in some cars is a door lock for emergencies (usually found behind a plastic cover on the driver’s side), to be used when the battery is depleted.

Unfortunately, PKES does not require user interaction(such as a button press) on the side of the car key to initiate the cryptographic computations and signal transmission. The lack of user interaction makes PKES systems prone to relay attacks, in which the challenge and response signals are relayed via a separate wireless channel: The car key (e.g., in the pocket of the victim) and vehicle (e.g., parked hundreds of meters away) will assume their mutual proximity and successfully authenticate. Since the initial publication of these relay attacks in 2011, tools that automatically perform relay attacks on PKES systems are available on the black market and are potentially used by criminals to open, start, and steal vehicles.

RKE systems rely on a unidirectional data transmission from the remote control, which is embedded in the car key, to the vehicle. Upon pressing a button, an active Radio Frequency (RF) transmitter in the remote control usually generates signals in a freely usable frequency band. These include the 315MHz band in North America and the 433MHz or 868MHz band in Europe, with a typical range of several tens to hundreds of meters. Note that a few old cars have been using infrared technology instead of RF. RKE systems enable the user to comfortably lock and unlock the vehicle from a distance, and can be used to switch on and off the anti-theft alarm, when present.

The first remote controls for cars used no cryptography at all: The car was unlocked after the successful reception of a constant “fix code” signal. Replay attacks on these systems are straightforward. We encountered a Mercedes Benz vehicle manufactured around 2000 that still relies on such fix code RKE systems.

The next generation of RKE systems are so-called rolling code systems, which employ cryptography and a counter value that is increased on each button press. The counter value (and other inputs) form the plaintext for generating a new, encrypted (or otherwise authenticated) rolling code signal. After decryption/verification on the side of the vehicle, the counter value is checked by comparing it to the last stored counter value that was recognized as valid:

An increased counter value is considered new and thus accepted. A rolling code with an old counter value is rejected. This mechanism constitutes an effective protection against replay attacks, since a rolling code is invalidated once it has been received by the vehicle.

In principle, such unidirectional rolling code schemes can provide a suitable security level for access control. However, as researchers have shown in the case of Keeloq in 2008, the security guarantees are invalidated if they rely on flawed cryptographic schemes: Keeloq was broken both by cryptanalysis and, in a more realistic setting, by side channel attacks on the key derivation scheme executed by the receiver unit. Although it is frequently mentioned that Keeloq is widely used for vehicle RKE systems, our research indicates that this system is prevalently employed for garage door openers.

Another attack, targeting an outdated automotive RKE scheme of an unspecified vehicle (built between 2000 and 2005), was demonstrated by Cesare in 2014: An adversary has to eavesdrop three subsequent rolling codes. Then, using phase-space analysis, the next rolling code can be predicted with a high probability. However, apart from this attack the cryptographic security of automotive RKE systems has not been investigated to our knowledge.

In particular, a large-scale survey and security analysis of very wide-spread rolling code systems has not been carried out.

A different, simple but effective method used by criminals to break into cars is to jam the RF communication when the victim presses the remote control to lock the car. The victim may not notice the attack and thus leave the car open. A variant of the attack is “selective jamming”, i.e., a combined eavesdropping-and-jamming approach: The transmitted rolling code signal is monitored and at the same time jammed, with the effect that the car is not locked and the attacker possesses a temporarily valid (one-time) rolling code. Consequently, a car could be found appropriately locked after a burglary.

Note that one successful transmission of a new rolling code from the original remote to the car usually invalidates all previously eavesdropped rolling codes, i.e., the time window for the attack is relatively small.

Furthermore, it is usually not possible to change the signal contents, for example, convert a “lock” command into an “unlock”. This limitation is often overlooked and severely limits the practical threat posed by this type of attack.

## 2.2 History of fingerprint sensor

This section narrates work done by other researchers to fingerprint enhancement through classification using Level 1 or Level 2 features or sometimes both. In this section, the contributions of all researchers to the field of fingerprint image enhancement using Level1 features are summarized.

Sherlock et al. (1994) proposed image enhancement algorithm based on nonstationary directional Fourier domain filtering. The directional filter used first and foremost to smooth the input image whose orientation in all fields matched to the local ridge orientation. The output of this stage is reduced noise image or high-quality image compared to the input image.

Fourier Domain filtering mainly uses local ridge patterns and local ridge parameters, directional band pass filters, and local ridge spacing. To implement the filtering techniques in the digital computer the image must be spatially sampled and the continuous function used in Fourier Transform is replaced by discrete functions. All images were sampled at a low resolution of 512 by 512 pixels and edge effects of discrete Fourier Transform were reduced to 10% using separable split-cosine window. The result of the enhancement used for various classifications of the input images.

Chikkerur et al. (2005) proposed fingerprint enhancement using Short Term Fourier Transforms (STFT), which is based on not stationary signals. In this paper, researchers extended the properties of STFT to two dimensional (2D) fingerprint images. They proposed a new algorithm for image enhancement process based on contextual filtering in Fourier domain. The new algorithm simultaneously yields local ridge orientation and local ridge frequency level 1 feature. The intrinsic features of the fingerprint image can be computed using single unified approach rather than multiple algorithms. Hsieh, C. T. et al. (2003) proposed an effective and efficient algorithm for fingerprint image enhancement, which not only improves the quality of the clarity of the image but also improves the continuity of the ridge structure based on global texture and local orientation.

The global texture is exposed using multi resolution analysis and local orientation through wavelet transforms. In wavelet based fingerprint analysis first input image is converted into normalized image.

Paul & Lourde (2006) proposed the new method for image enhancement using the applications of wavelet transforms. Before the inventions of these techniques, popular other techniques were Gabor filtering and Fourier filtering. The new method outperformed compared to this method in terms of efficiency and execution time.

Ye et al., 2007 additionally utilized a 2D discrete wavelet transform to digitally compress fingerprint and to reconstruct the original image, whenever necessary using some reconstructing attributes. Few quantitative measurements are used to evaluate the quality of wavelet transform, which helps in image enhancement process. In this paper researcher also used a different measure to evaluate the performance of wavelet transform and obtained higher efficiency.

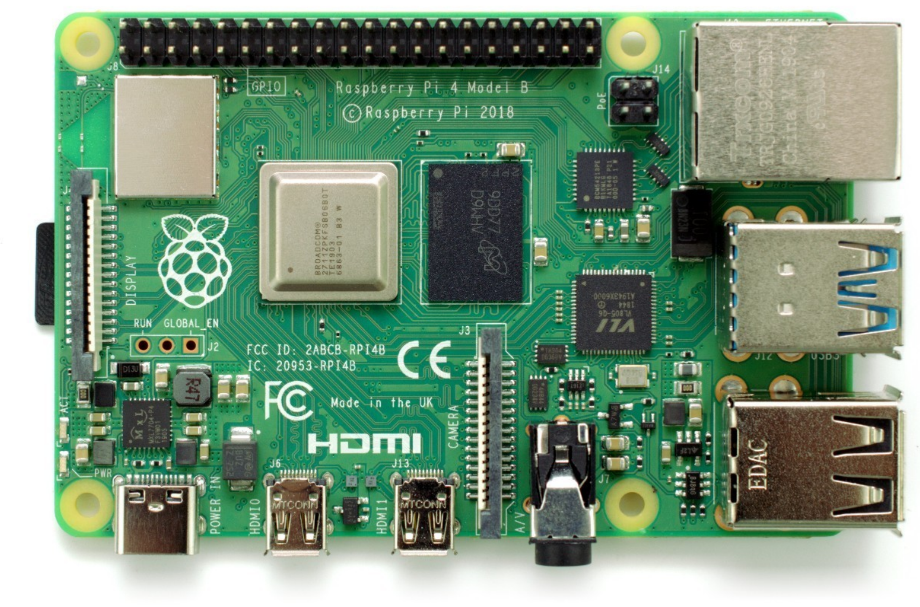
Farina et al., (1999) worked on a binary image, the input is either already taken as a binary image or converted into binary from the greyscale image and also the image is skeletonised. Due to differences in a number of minutiae occur in real, there is a necessity of post pre-processing, in order to maintain the consistency of image and to reduce the computational cost.

Babatund (2012) modified some of the existing sub-models mathematical algorithms for fingerprint image enhancement and obtained new version. The different sub models of the new version are segmentation, normalization, ridge orientation estimation, ridge frequency estimation, Gabor filtering, and Binarization and Thinning. In order to test this new version, the author used windows vista home basic operating system and Matrix Laboratory (Matlab) as Frontend engine.

# Chapter 3

# HARDWARE

## 3.1 Raspberry Pie 4 Model B



This is the latest product in the popular Raspberry Pi range of computers. It offers ground-breaking increases in processor speed, multimedia performance, memory, and connectivity compared to the prior-generation Raspberry Pi 3 Model B+, while retaining backwards compatibility and similar power consumption. For the end user, Raspberry Pi 4 Model B provides desktop performance comparable to entry-level x86 PC systems.

This product's key features include a high-performance 64-bit quad-core processor, dual-display support at resolutions up to 4K via a pair of micro-HDMI ports, hardware video decode at up to 4Kp60, up to 4GB of RAM, dual-band 2.4/5.0 GHz wireless LAN, Bluetooth 5.0, Gigabit Ethernet, USB 3.0, and PoE capabilities.

The dual-band wireless LAN and Bluetooth have modular compliance certification, allowing the board to be designed into end products with significantly reduced compliance testing, improving both cost and time to market.

Specification

Processor: Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz

Memory: 1GB, 2GB or 4GB LPDDR4 (depending on model)

Connectivity: 2.4 GHz and 5.0 GHz IEEE 802.11b/g/n/ac wireless LAN, Bluetooth 5.0, BLE Gigabit Ethernet, 2 × USB 3.0 ports, 2 × USB 2.0 ports.

GPIO: Standard 40-pin GPIO header(fully backwards-compatible with previous boards)

Video & sound: 2 × micro HDMI ports (up to 4Kp60 supported), 2-lane MIPI DSI display port, 2-lane MIPI CSI camera port, 4-pole stereo audio and composite video port

Multimedia: H.265 (4Kp60 decode), H.264 (1080p60 decode, 1080p30 encode); OpenGL ES, 3.0 graphics

SD card support: Micro SD card slot for loading operating system and data storage

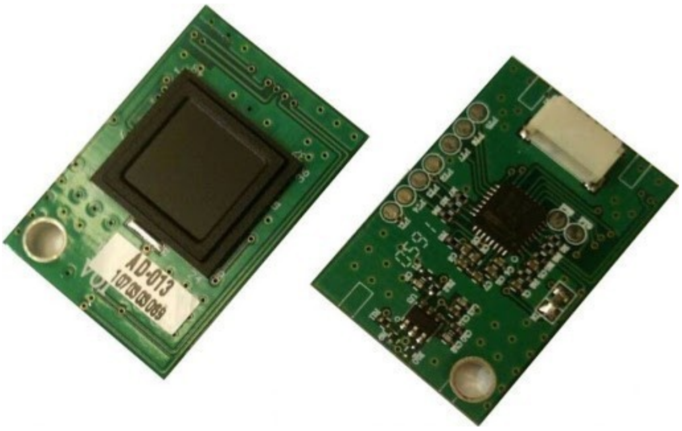
Input power: 5V DC via USB-C connector (minimum 3A 1 ), 5V DC via GPIO header (minimum 3A), Power over Ethernet (PoE)–enabled (requires separate PoE HAT)

Environment: Operating temperature 0–50oC

Compliance: For a full list of local and regional product approvals, please visit https://www.raspberrypi.org/documentation/ hardware raspberrypi/conformity.md

Production lifetime: The Raspberry Pi 4 Model B will remain in production until at least January 2026.

## 3.2 Fingerprint module (AD-013)



This fingerprint sensor module is consist from a fingerprint sensor & a MCU. The module provide one UART interface output port for connecting to customer side main board.

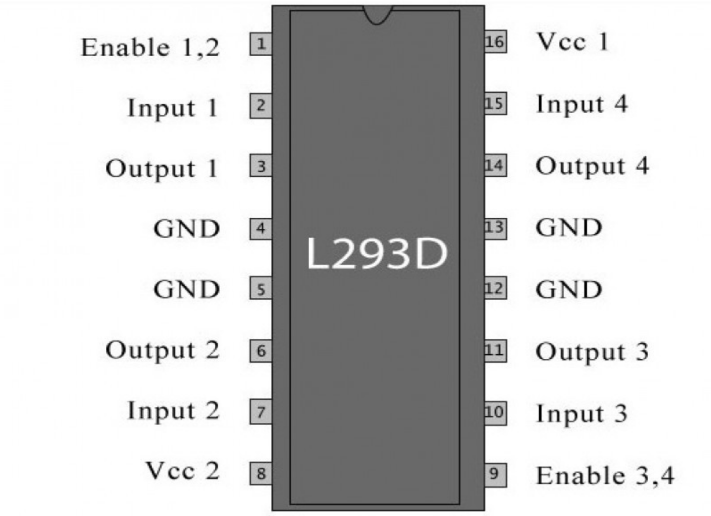
The operation of AD-013 is as following, a fingerprint image captured by pixel array, delivery fingerprint ridge or valley signals to A/D converter and digital processor, then to the serial peripheral interface for data-reading. Due to MCU already installed fingerprint algorithm, the user can through

UART interface use MCU SOC command to operate module like fingerprint enroll, match operations etc.

### 3.3 tty-USB converter

This converter acts as intermediate between Raspberry Pie and fingerprint module. The UART data is converted into USB data and delivered to host. The connection in tty comprise of 3.3V, 5V, Tx, Rx, and Gnd. Since our sensor works in 3.3V logic, we use 3.3V source, Tx, Rx, and Gnd using jumper wires.

### 3.4 L293 IC series



A motor driver IC is an integrated circuit chip which is usually used to control motors in autonomous robots. Motor driver ICs act as an interface between microprocessors in robots and the motors in the robot. The most commonly used motor driver IC’s are from the L293 series such as L293D, L293NE, etc. These ICs are designed to control 2 DC motors simultaneously. L293D consist of two H- bridge. H-bridge is the simplest circuit for controlling a low current rated motor.

The L293D IC receives signals from the microprocessor and transmits the relative signal to the motors. It has two voltage pins, one of which is used to draw current for the working of the L293D and the other is used to apply voltage to the motors. The L293D switches it output signal according to the input received from the microprocessor. The L293D is a 16 pin IC, with eight pins, on each side, dedicated to the controlling of a motor. There are 2 INPUT pins, 2 OUTPUT pins and 1 ENABLE pin for each motor. L293D consist of two H-bridge. H-bridge is the simplest circuit for controlling a low current rated motor.

# Chapter 4

# SOFTWARE

## 4.1 AD-013 SoC details

### Buffer and fingerprint Database:

There are a 72K-byte image buffer ImageBuffer and two 7-byte feature file buffers CharBuffer1 and CharBuffer2 in the chip. Users can read/write any of the above buffers by instructions. CharBuffer1 or CharBuffer2 can be used to store general feature file as well as template feature file.

The capacity of fingerprint database changes with FLASH memory capacity, which is identified by the system automatically. Fingerprint templates are stored sequentially according to the SN, while the definition of SN is: 0~(N-1) (N=fingerprint database capacity).

Note: Only through SN index will users access the fingerprint database.

### Features and Templates:

Fingerprint feature file occupies 425 bytes, including general information as well as minutiae information. Template file occupies 2129 bytes, sum of five features files of the same fingerprint.

### Feature file structure:

The minutiae number of a feature file is no more than 99. Of the total 2129 bytes (size of feature file is 425 bytes),the first

56 bytes is the file header used for general information; The latter 369 bytes are to store minutiae information, 4 bytes for each minutiae.

Instruction Form specification:

MCU SOC can form complete fingerprint identification module with several necessary periphery circuit(sensor, flash, power supply, etc.). The module is in Slave mode all the time. The host can issue different instructions to the module, for various functions. The host instruction, modules ACK and data exchanges are all work according to given format data packet. The host should packet instructions and data which need transmitting as well as analyze received data packets based-on the following format.

Data packet Form:

Instruction /data packet altogether be classified into three categories:

Packet flag=01 Command packet

Packet flag=02 Data packet, and with continue packet

Packet flag=08 The last data packet,i.e. end packet

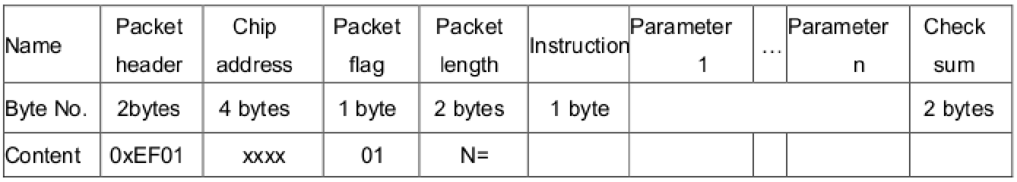
All data packets should be with packet header:0xEF01

Packet length= The total byte number from packet length to Sum (instruction, parameter or data), including Sum, but not the byte number of packet length itself;

Sum is the total bytes from packet flag to Sum, the carry will be ignored if it exceed 2 bytes;

The default chip address is 0xFFFFFFFF before its issue. Once the host issues chip address by instruction, all data packets should receive and transmit according to the address. Chip will reject packets with wrong address

**01 Command packet format:**

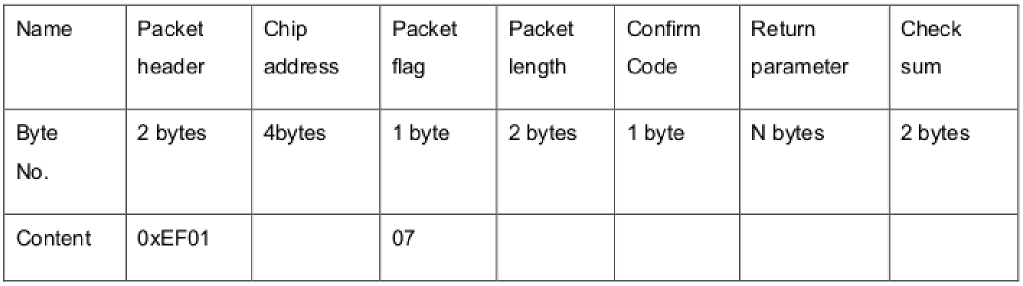


| Name | Packet header | Chip address | Packet flag | Packet length | Instruction | Parameter 1 | … | Parameter n | Check sum |
|---|---|---|---|---|---|---|---|---|---|
| Byte No. | 2bytes | 4 bytes | 1 byte | 2 bytes | 1 byte | | | | 2 bytes |
| Content | 0xEF01 | xxxx | 01 | N= | | | | | |

Instruction ACK:

ACK is to report relevant command running condition and result to the host, the ACK packet contains parameter and can be with continue data packet. Only when the host received the ACK packet of SOC can it confirm the condition of SOC packet receiving and instruction implementing.

ACK packet format:



| Name | Packet header | Chip address | Packet flag | Packet length | Confirm Code | Return parameter | Check sum |
|---|---|---|---|---|---|---|---|
| Byte No. | 2 bytes | 4bytes | 1 byte | 2 bytes | 1 byte | N bytes | 2 bytes |
| Content | 0xEF01 | | 07 | | | | |

4.2 Python ‘pyserial’ library

‘pyserial’ is library available in python repository (python.org) used for serial communication, like UART, SPI, I2C, USB etc., through serial ports.

Following are 3 important functions in pyserial library:

serial.Serial() takes port name, baud rate, size of byte, parity and stop bits as parameters. This is used for initializing the serial port used for communication.

serial.write(), takes byte sequence as parameter, for writing data to serial peripheral device

serial.read(), takes number of bytes as parameter, for reading data from serial peripheral device

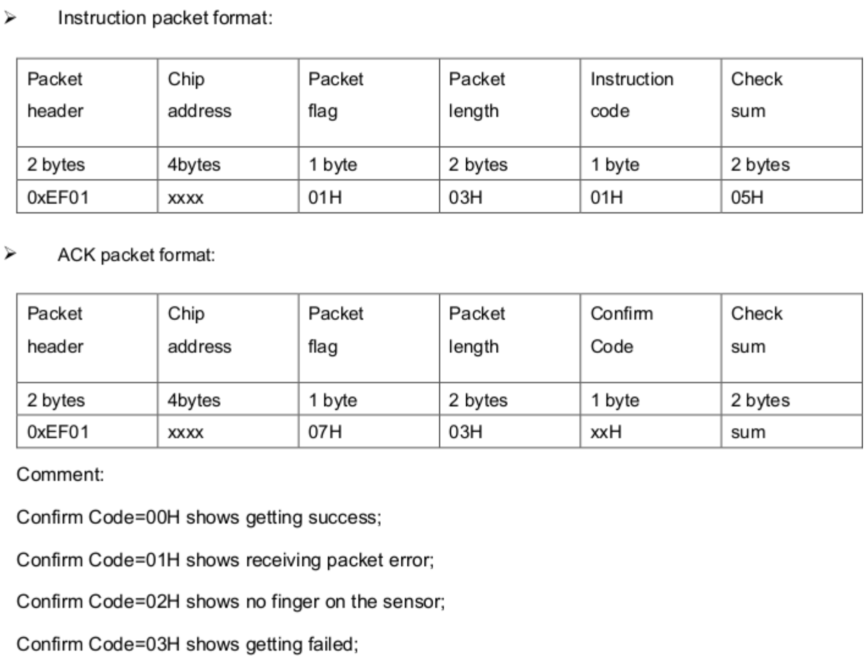
In python, bytes is inbuilt object represented as b'<sequence>'. The byte value follow ASCII protocol, Ex : [34,24,56] in bytes form is b'"\x188'

" is ASCII value of 34, \x18 is hexadecimal number for 24, 8 present in 56th place of ASCII table

## 4.3 AD-013 commands

In our project, we use 7 commands for our registering and verifying the fingerprint. Following defined 7 functions, are written in resource.py python file, for sending each command and receive acknowledgment for the corresponding

1)GetImage(arg) function: this function detects finger on sensor, then get the fingerprint image and store it in ImageBuffer. Returning to confirm code to show: getting success, no finger, etc. arg is initialized serial port.



- Instruction packet format:

| Packet header | Chip address | Packet flag | Packet length | Instruction code | Check sum |
|---|---|---|---|---|---|
| 2 bytes | 4bytes | 1 byte | 2 bytes | 1 byte | 2 bytes |
| 0xEF01 | xxxx | 01H | 03H | 01H | 05H |

- ACK packet format:

| Packet header | Chip address | Packet flag | Packet length | Confirm Code | Check sum |
|---|---|---|---|---|---|
| 2 bytes | 4bytes | 1 byte | 2 bytes | 1 byte | 2 bytes |
| 0xEF01 | xxxx | 07H | 03H | xxH | sum |

Comment:

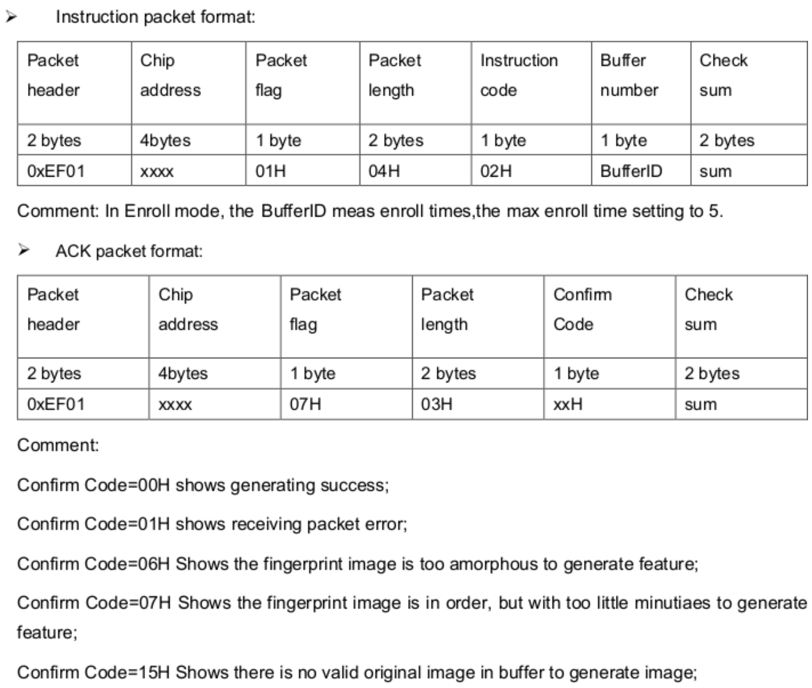
Confirm Code=00H shows getting success;

Confirm Code=01H shows receiving packet error;

Confirm Code=02H shows no finger on the sensor;

Confirm Code=03H shows getting failed;

2)GenChar(arg, N) function: Generating the original image in ImageBuffer to fingerprint feature file and store it in CharBuffer. arg is the initialized serial port.



- Instruction packet format:

| Packet header | Chip address | Packet flag | Packet length | Instruction code | Buffer number | Check sum |
|---|---|---|---|---|---|---|
| 2 bytes | 4bytes | 1 byte | 2 bytes | 1 byte | 1 byte | 2 bytes |
| 0xEF01 | xxxx | 01H | 04H | 02H | BufferID | sum |

Comment: In Enroll mode, the BufferID meas enroll times,the max enroll time setting to 5.

- ACK packet format:

| Packet header | Chip address | Packet flag | Packet length | Confirm Code | Check sum |
|---|---|---|---|---|---|
| 2 bytes | 4bytes | 1 byte | 2 bytes | 1 byte | 2 bytes |
| 0xEF01 | xxxx | 07H | 03H | xxH | sum |

Comment:

Confirm Code=00H shows generating success;

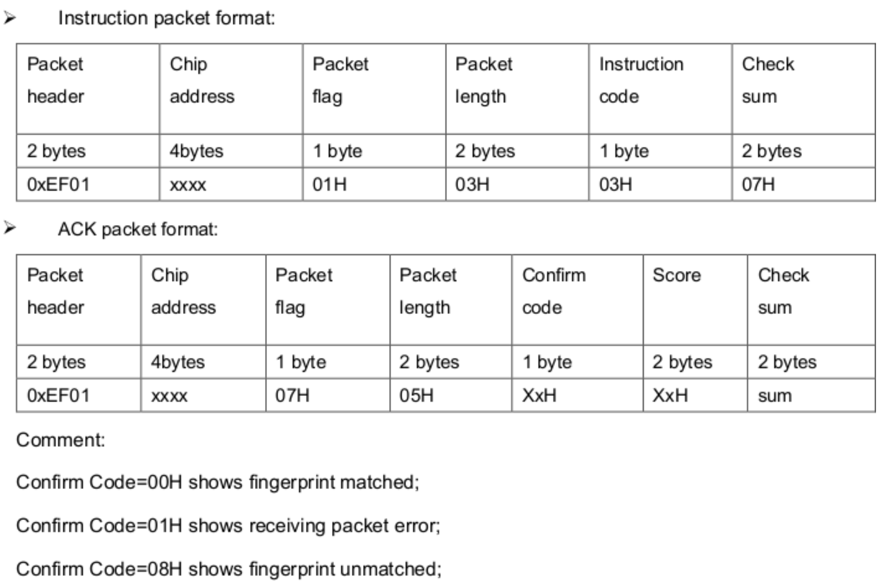
Confirm Code=01H shows receiving packet error;

Confirm Code=06H Shows the fingerprint image is too amorphous to generate feature;

Confirm Code=07H Shows the fingerprint image is in order, but with too little minutiaes to generate feature;

Confirm Code=15H Shows there is no valid original image in buffer to generate image;

3)Match(arg) function:Pattern-matching the feature files in CharBuffer1 and CharBuffer2, arg is initialized serial port.



➢ Instruction packet format:

| Packet header | Chip address | Packet flag | Packet length | Instruction code | Check sum |
|---|---|---|---|---|---|
| 2 bytes | 4bytes | 1 byte | 2 bytes | 1 byte | 2 bytes |
| 0xEF01 | xxxx | 01H | 03H | 03H | 07H |

➢ ACK packet format:

| Packet header | Chip address | Packet flag | Packet length | Confirm code | Score | Check sum |
|---|---|---|---|---|---|---|
| 2 bytes | 4bytes | 1 byte | 2 bytes | 1 byte | 2 bytes | 2 bytes |
| 0xEF01 | xxxx | 07H | 05H | XxH | XxH | sum |

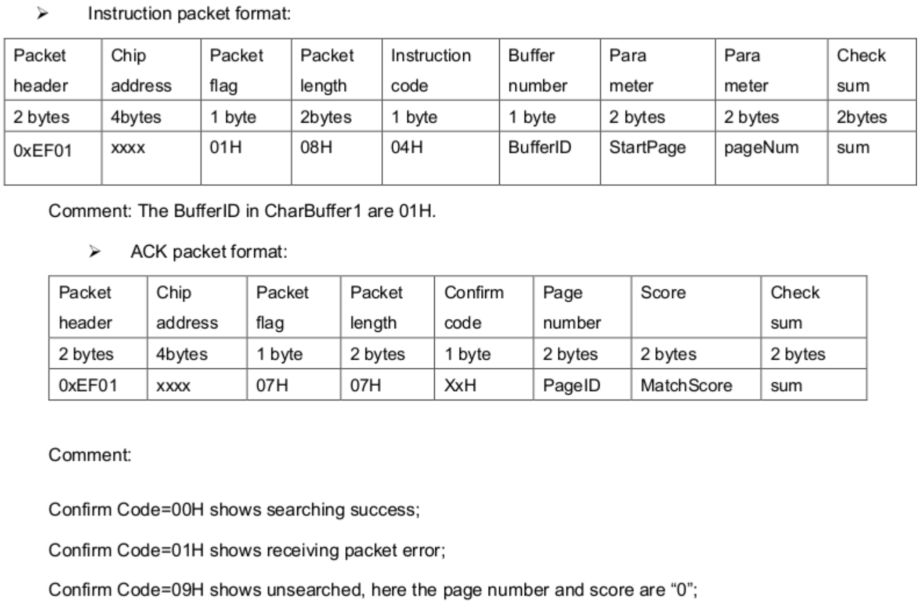
Comment:

Confirm Code=00H shows fingerprint matched;

Confirm Code=01H shows receiving packet error;

Confirm Code=08H shows fingerprint unmatched;

4)Search(arg, arg1, arg2): To search the whole or part of fingerprint database with feature files in CharBuffer1 or CharBuffer2. If get, jump to the original page. Arg is initialized serial port, arg1 is start ID,arg2 is stop ID of database



➢ Instruction packet format:

| Packet header | Chip address | Packet flag | Packet length | Instruction code | Buffer number | Para meter | Para meter | Check sum |
|---|---|---|---|---|---|---|---|---|
| 2 bytes | 4bytes | 1 byte | 2bytes | 1 byte | 1 byte | 2 bytes | 2 bytes | 2bytes |
| 0xEF01 | xxxx | 01H | 08H | 04H | BufferID | StartPage | pageNum | sum |

Comment: The BufferID in CharBuffer1 are 01H.

➢ ACK packet format:

| Packet header | Chip address | Packet flag | Packet length | Confirm code | Page number | Score | Check sum |
|---|---|---|---|---|---|---|---|
| 2 bytes | 4bytes | 1 byte | 2 bytes | 1 byte | 2 bytes | 2 bytes | 2 bytes |
| 0xEF01 | xxxx | 07H | 07H | XxH | PageID | MatchScore | sum |

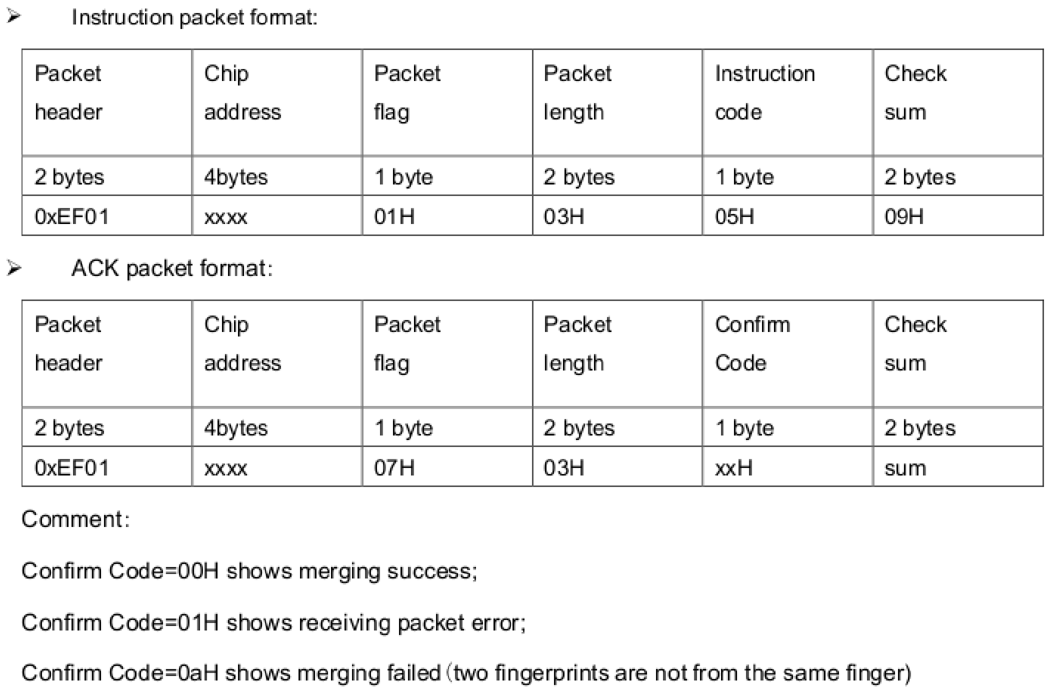
Comment:

Confirm Code=00H shows searching success;

Confirm Code=01H shows receiving packet error;

Confirm Code=09H shows unsearched, here the page number and score are "0";

5)RegModel(arg) function: Merging feature files in CharBuffer1 and CharBuffer2 to generate templates, store the result in CharBuffer1 and CharBuffer2. arg is preinitialized serial port.



- Instruction packet format:

| Packet header | Chip address | Packet flag | Packet length | Instruction code | Check sum |
|---|---|---|---|---|---|
| 2 bytes | 4bytes | 1 byte | 2 bytes | 1 byte | 2 bytes |
| 0xEF01 | xxxx | 01H | 03H | 05H | 09H |

- ACK packet format:

| Packet header | Chip address | Packet flag | Packet length | Confirm Code | Check sum |
|---|---|---|---|---|---|
| 2 bytes | 4bytes | 1 byte | 2 bytes | 1 byte | 2 bytes |
| 0xEF01 | xxxx | 07H | 03H | xxH | sum |

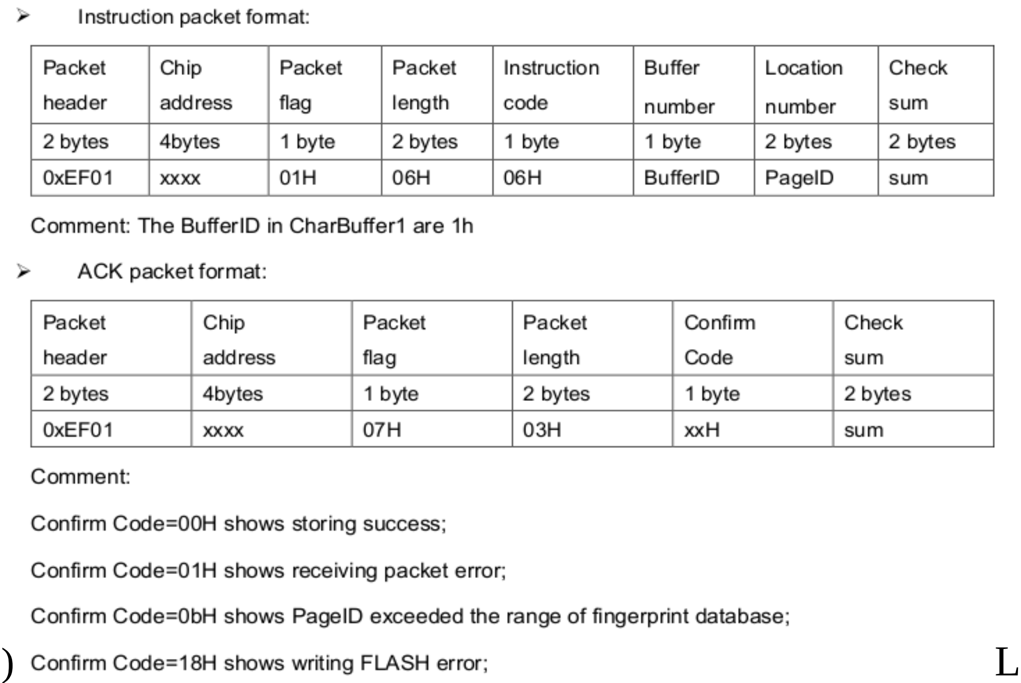
Comment：

Confirm Code=00H shows merging success;

Confirm Code=01H shows receiving packet error;

Confirm Code=0aH shows merging failed（two fingerprints are not from the same finger)

6) StoreChar(arg, arg1): Storing the template files in CharBuffer1 to the location of PageIDNum flash database. arg is initialized serial port, arg1 is ID assigned for stored fingerprint.



- Instruction packet format:

| Packet header | Chip address | Packet flag | Packet length | Instruction code | Buffer number | Location number | Check sum |
|---|---|---|---|---|---|---|---|
| 2 bytes | 4bytes | 1 byte | 2 bytes | 1 byte | 1 byte | 2 bytes | 2 bytes |
| 0xEF01 | xxxx | 01H | 06H | 06H | BufferID | PageID | sum |

Comment: The BufferID in CharBuffer1 are 1h

- ACK packet format:

| Packet header | Chip address | Packet flag | Packet length | Confirm Code | Check sum |
|---|---|---|---|---|---|
| 2 bytes | 4bytes | 1 byte | 2 bytes | 1 byte | 2 bytes |
| 0xEF01 | xxxx | 07H | 03H | xxH | sum |

Comment:

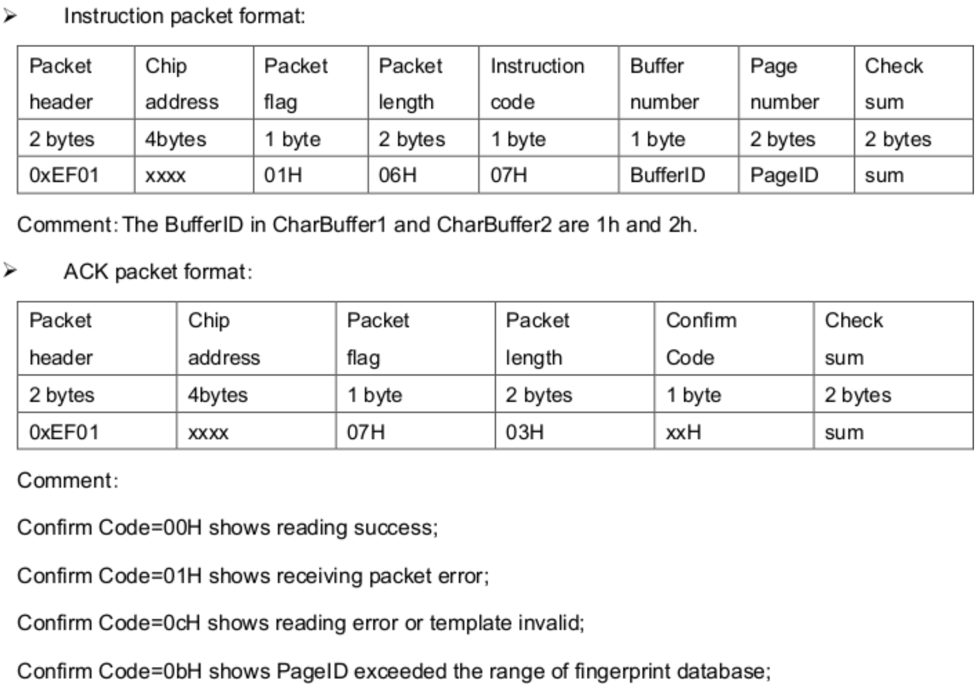
Confirm Code=00H shows storing success;

Confirm Code=01H shows receiving packet error;

Confirm Code=0bH shows PageID exceeded the range of fingerprint database;

Confirm Code=18H shows writing FLASH error;

7) LoadChar(arg, arg1) function: Readin the fingerprint templates which appointed IDNum in flash database to template buffer CharBuffer1 or CharBuffer2. arg is the initialized serial port, arg1 is ID of fingerprint to be loaded to one of feature buffer



- Instruction packet format:

| Packet header | Chip address | Packet flag | Packet length | Instruction code | Buffer number | Page number | Check sum |
|---|---|---|---|---|---|---|---|
| 2 bytes | 4bytes | 1 byte | 2 bytes | 1 byte | 1 byte | 2 bytes | 2 bytes |
| 0xEF01 | xxxx | 01H | 06H | 07H | BufferID | PageID | sum |

Comment: The BufferID in CharBuffer1 and CharBuffer2 are 1h and 2h.

- ACK packet format:

| Packet header | Chip address | Packet flag | Packet length | Confirm Code | Check sum |
|---|---|---|---|---|---|
| 2 bytes | 4bytes | 1 byte | 2 bytes | 1 byte | 2 bytes |
| 0xEF01 | xxxx | 07H | 03H | xxH | sum |

Comment:

Confirm Code=00H shows reading success;

Confirm Code=01H shows receiving packet error;

Confirm Code=0cH shows reading error or template invalid;

Confirm Code=0bH shows PageID exceeded the range of fingerprint database;

## 4.4 2 python files for enrolling and verifying

We are creating 2 python modules for enrolling and verifying the fingerprint named as enroll.py and verify.py

In enroll.py, the functions from resource.py are called for capturing fingerprint image, and store them as template in database along with assigning an ID.

In verify.py, we generate a feature file from fingerprint image and check it in database for its presence and print the appropriate results.

## 4.5 Interfacing our model with locking system of car

The actuator in car, which is main part of locking system, comprise of system of gears rotated to lock/unlock it, This state of latch depends on rotational direction of dc motor.

Using L298N motor driver IC, we control the dc motor of latch for locking and unlocking of car door. We vary 2 GPIO pin states of raspberry CPU for controlling rotation direction of dc motor. We insert this logic in verify.py module for opening of car door only for authorized persons.

# Chapter 5
# RESULTS AND FUTURE WORK

## 5.1 Details

The enroll.py module enrolls the fingerprint template in database. We save multiple fingerprint in database and when we cross-check it using verify.py, results match with expected. So declare the interfacing of fingerprint sensor with raspberry pie as working. Then we update verify.py module by adding logic to operate the actuator, test it by scanned fingerprint, then door locks/unlocks.

Since we did not fully utilize the raspberry CPU, a application specific device can be designed for our purpose and reduce the cost of device, if produced in bulk quantities.

Chapter 6

# REFERENCES

[1]A. Aditya Shankar, “Finger Print Based Door Locking System”, International Journal Of Engineering And Computer Science, ISSN:2319-7242 Volume 4 Issue 3 March 2015.

[2]R.M.Vithlani, “Biometric Automobile Ignition Locking System”, International Journal of Electronics and Communication Engineering and Technology (IJECET), Volume 7, Issue 5, September-October 2016.

[3]Ch.Surendra Kumar, “ Biometric Authenication Based Vehicular Safety System Using Arm Processor”, International Journal of Engineering Science & Advanced Technology Volume-4, Issue-5.

[4]Roopam Arora, “ Start-Up The Engine Using Fingerprinting”, International Journal of Computer Engineering and Applications, Volume IX, Issue X, Oct. 15 ISSN 2321-3469.

[5] Mohamed Basheer. K. P and Dr. T. Abdul Razak, “Enhanced Biometric Based Authentication for Network Security using IRIS”. International Journal of Computer Engineering and Technology (IJCET), 4(6), 2014, pp. 412–422.

[6] Priya Darshini.V “Multilevel Security System for Auto motives using RFID and Biometric Techniques in LabVIEW”, International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering Vol. 2, Issue 4, April 2013.

[7] Anil Jain, Arun Ross and Salil Prabhakar, “Fingerprint Matching Using Minutiae And Texture Features,” Fingerprint Matching Using Minutiae And Texture Features”, in Proc. of Int’l Conference on

Image Processing (ICIP), pp.282-285, Thessaloniki, Greece, Oct 7 - 10, 2001.

[8] Mukesh Kumar Thakur, Ravi Shankar Kumar, Mohit Kumar, Raju Kumar "Wireless Fingerprint Based Security System using Zigbee", International Journal of Inventive Engineering and Sciences(IJIES) ISSN: 2319–9598, Volume-1, Issue-5, April 2013.

[9] Mary Lourde R and Dushyant Khosla,"Fingerprint Identification in Biometric Security Systems", International Journal of Computer and Electrical Engineering, Vol. 2, No. 5, October,2010.

[10] Karthikeyan.a "Fingerprint Based Ignition System" International Journal Of Computational Engineering Research / ISSN: 2250–3005

[11]R. Tani, J. -S. Yoon, S. -I. Yun, W. -J. Nam, S. Takasugi, J. -M. Kim, J. -K. Park, S. -Y. Kwon, P. -Y. Kim, C. -H. Oh, B. -C. Ahn, "Panel and Circuit Designs for the World's First 65-inch UHD OLED TV", SID 2015 Digest, pp. 950 – 953, 2015.

[12]S. M. Jung, J. M. Nam, D. H. Yang, and M. K. Lee, "A CMOS integrated capacitive fingerprint sensor with 32-bit RISC microcontroller," IEEE Journal of Solid-State Circuits, vol. 40, pp. 1745-1750, 2005.

[13]S. Tomita, T. Okada, and H. Takahashi, "An in-cell capacitive touch sensor integrated in an LTPS WSVGA TFT-LCD," J. of the Soc. for Information Display, vol 20, pp. 441–449, 2012.

[14]P. Koundinya, X. Zhaoy, T. Fengy, S. Theril, and W. Shiy, "Touch-Fingerprint Display for Supporting Identity Sensing,", SID 2014 Digest, pp. 1610-16