Anti-theft Security System Using Face Recognition

BY

CHONG GUANG YU

A REPORT

SUBMITTED TO

University Tunku Abdul Rahman

in partial fulfilment of the requirements

for the degree of

BACHELOR OF INFORMATION TECHNOLOGY (HONS)

COMPUTER ENGINEERING

Faculty of Information and Communication Technology

(Perak Campus)

MAY 2018

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: ANTI-THEFT SECURITY SYSTEM USING FACE RECOGNITION

Academic Session: 2018 MAY

I CHONG GUANG YU

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in

Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes

Verified by,

_____ _____

(Author's signature) (Supervisor's signature)

Address:

491-A, JLN CHONG HAI, _____

BUKIT SIPUT, 85020 SEGAMAT Supervisor's name

JOHOR, MALAYSIA

Date: 18 AUGUST 2018 **Date**: _____

Anti-theft Security System Using Face Recognition

BY

CHONG GUANG YU

A REPORT

SUBMITTED TO

University Tunku Abdul Rahman

in partial fulfilment of the requirements

for the degree of

BACHELOR OF INFORMATION TECHNOLOGY (HONS)

COMPUTER ENGINEERING

Faculty of Information and Communication Technology

(Perak Campus)

MAY 2018

DECLARATION OF ORIGINALITY

I declare that this report entitled "**ANTI-THEFT SECURITY SYSTEM USING FACE RECOGNITION**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : _____

Name : CHONG GUANG YU

Date : 18 AUGUST 2018_____

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisors, Mr Leong Chun Farn who has given me this bright opportunity to engage in an Ambient Image Processing project. It is my first step to establish a study in this field. A million thanks to you.

To my family, I would like to say thanks for their spiritual and material support. Thank you for always standing by my side since I started my degree programme. Last but not least, thanks to my friends and schoolmate for helping and accompanying each other in my university life.

ABSTRACT

This project is about an anti-theft security system using face recognition based on Raspberry pi. Anti-theft security system has been launched for many years but most of them are just a CCTV, IP camera or door sensor alert system. It could be more efficient with uses of face recognition. The design of anti-theft security system is based on human face recognition and remote monitoring technology. It will verify the person identity that goes near to the camera within certain distance. Only the people who match the identity with database and key in the correct password have the right to entry. A mobile application telegram and image processing technique LBP based have been involved in this system. Electric door lock solenoid, keypad and LCD display will combined operate the accessibility of door. An ultrasonic sensor is used to detect the person distance between the door. Face recognition will only start to function when someone goes near the door within certain distance. An alert message and photo taken will send to owner mobile phone via telegram with WIFI when the face recognition algorithms detect a stranger comes near the door and also a buzzer will be trigger to act as an alarm system. Owner can control the door locker and check the camera with the telegram application. The security system is designed for actual use in home condition with the uses of a microcontroller ARM Quad Cortex-A53.

TABLE OF CONTENTS

Table of Contents

Title		i
Declaration of originality		ii
Acknowledgements		iii
Abstract		iv
Table of Contents		v
List of Tables		viii
List of Figures		viii
List of Abbreviations		xi
Chapter 1: Introduction		
1.1	Problem statement and motivation	1
1.2	Project Scope	1
1.3	Project Objective	2
1.4	Impact, Significance and Contribution	2
1.5	Background information	3
1.6	Achievement	4
1.7	Report Organization	5
Chapter 2: Literature Review		
2.1	Discussion on other research	6
2.2	Critical Remarks of previous works	9
Chapter 3: System Design		
3.1	Description of Project	11
3.2	Process Flow of System	15
3.2.1	Raspberry Pi	15

3.2.2 Telegram 15
3.3 System Setup Procedures 15
3.3.1 Raspbian OS 15
3.3.2 Telegram 16
3.3.2 VNC Viewer 17
3.4 Software Development 18
3.4.1 Process of Face Recognition 18
3.4.2 Process of Training Dataset 20
3.4.3 Process of Prepare Face Dataset 22

Chapter 4: Design Specification

4.1 Methodologies 24
4.1.1 System Development Methodology 24
4.1.2 Project Flow 25
4.2 Tools 26
4.2.1 System Hardware 27
4.2.1.1 Raspberry pi 3 model B 27
4.2.1.2 Raspberry pi camera module 29
4.2.1.3 I2C LCD1602 display 30
4.2.1.4 Buzzer 31
4.2.1.5 Solenoid locker 31
4.2.1.6 HC-SR04 Ultrasonic Sensor 32
4.2.2 System Software 33
4.2.2.1 Raspbian OS 33
4.2.2.2 Python IDE 33
4.2.2.3 Blynk 34
4.2.2.4 Telegram 34

Chapter 5: Implementation and Testing

5.1 Testing Face Recognition 35
5.2 Testing System 36

5.2.1 Security System 36
5.2.2 Surveillance System 37
5.2.3 Telegram Control 38
5.2.4 LCD Display 40
5.2.5 Blynk 41
5.3 Implementation 42

Chapter 6: Conclusion 43
6.1 Project Review, Discussion and Conclusions 43
6.2 Future work 44

Bibliography 45

Appendices 46

List of Tables

Table Number	Title	Page
Table 4.2.1	Hardware components required	26
Table 4.2.1.1.1	Specification details of Raspberry pi 3 model B	28
Table 4.2.1.2.1	Specification details of Raspberry pi camera module V1.3	29
Table 4.2.1.3.1	Connection pin of LCD module and Raspberry pi GPIO pinout	30
Table 4.2.1.6.1	Specification details of HC-SR04 Ultrasonic sensor	32
Table 4.2.1.6.2	Connection pin of HC-SR04 Ultrasonic sensor and Raspberry pi GPIO pinout	32

List of Figures

Figure Number	Title	Page
Figure 2.1.1	Overview of the proposed system configuration layout	7
Figure 2.1.2	Overview face detection and feature extraction process	8
Figure 2.1.3	Block diagram of Door locking System	9
Figure 2.2.1	Keypad password and RFID door lock	9
Figure 2.2.2	Fingerprint door lock	10
Figure 3.1.1	System Block Diagram	11
Figure 3.1.2	System Flowchart	12
Figure 3.1.3	Breadboard Diagram	13
Figure 3.1.4	Schematic Diagram	14
Figure 3.3.1.1	Raspbian OS Screenshot	16
Figure 3.3.2.1	Telegram Screenshot	17
Figure 3.3.3.1	VNC Viewer Screenshot	18
Figure 3.4.1	Flow of Face recognition	18
Figure 3.4.1.1	Coding for Face Recognition initialization	19

Figure 3.4.1.2	Coding for face recognition	20
Figure 3.4.2.1	Coding for Training dataset initialization	20
Figure 3.4.2.2	Coding for listing the Training dataset	21
Figure 3.4.2.3	Coding for perform training	22
Figure 3.4.3.1	Coding for prepare face dataset initialization	22
Figure 3.4.3.2	Coding collect face image dataset	23
Figure 4.1.1.1	Prototyping model	24
Figure 4.1.2.1	Flow of Project	25
Figure 4.2.1.1.1	Raspberry pi 3 model B GPIO pinout diagram	27
Figure 4.2.1.2.1	Raspberry pi camera module V1.3 with CSI ribbon cable	29
Figure 4.2.1.3.1	LCD1602 display with I2C LCD controller	30
Figure 4.2.1.4.1	Mini buzzer	31
Figure 4.2.1.5.1	Solenoid locker	31
Figure 4.2.1.6.1	HC-SR04 Ultrasonic sensor	32
Figure 4.2.2.1.1	Raspbian OS	33
Figure 4.2.2.1.2	Python IDE	33
Figure 4.2.2.3.1	Overview of Blynk	34
Figure 4.2.2.4.1	Telegram	34
Figure 5.1.1	Sample dataset used for testing	35
Figure 5.1.2	Result obtained with the use of sample dataset	35
Figure 5.2.1.1	Screenshot notification when stranger detected	36
Figure 5.2.1.2	Screenshot notification when stranger key in wrong password	36
Figure 5.2.1.3	Screenshot notification when correct password but face not in database	36
Figure 5.2.1.4	Screenshot notification when wrong password but face verified	37

Figure 5.2.1.5	Screenshot notification send to other when reverse password entered	37
Figure 5.2.2.1	Screenshot commend to activate surveillance cam	37
Figure 5.2.2.2	Screenshot of surveillance cam	38
Figure 5.2.2.3	Screenshot command to end surveillance cam	38
Figure 5.2.3.1	Screenshot command to capture photo	38
Figure 5.2.3.2	Screenshot command to remote access control the locker	39
Figure 5.2.3.3	Screenshot command to set time restrictions	39
Figure 5.2.3.4	Screenshot command to check door access records	40
Figure 5.2.4.1	LCD display when system is ready	40
Figure 5.2.4.2	LCD display when password entered	40
Figure 5.2.4.3	LCD display when wrong password	40
Figure 5.2.4.4	LCD display when locker unlocked	41
Figure 5.2.4.5	LCD display when meet time restriction period	41
Figure 5.2.5.1	Blynk user interface	41
Figure 5.3.1	Prototype 1	42
Figure 5.3.2	Final product	42

List of Abbreviations

CCTV	*Closed-circuit television*
IP camera	*Internet protocol camera*
PCA	*Principal component analysis*
LBPH	*Local binary patterns histograms*
LCD	*Liquid-crystal display*
LED	*Light Emitting Diode*
PIR	*passive infrared sensor*
ARM	*Acorn RISC Machines*
SDLC	*Systems Development Life Cycle*
GPIO	*General Purpose Input Output*
Wi-Fi	*Wireless Fidelity*
IoT	*Internet of Things*
IT	*Information Technology*
HDMI	*High-Definition Multimedia Interface*
I2C	*Inter-Integrated Circuit*
RAM	*Random Access Memory*
USB	*Universal Serial Bus*
RFID	*Radio-frequency identification*
V	*Voltage*
OS	*Operating System*

CHAPTER 1: INTRODUCTION

1.1 Problem statement and motivation

This project is for provide a precise solution with a new design with improvement embedded system to protect the household with giving higher security level and notification on time. Anyway there are a few problems that are still exist on security product on market. The issues like the face detection only will start to function when someone press the doorbell. Surveillance system will destroy by theft and without notify owner when stranger come in. Lastly, face recognition can be by pass with owner face photo.

Nowadays, the home security with only door locker is not enough to protect your house and family. However the smart home security system on market still not popular for household uses because of the high price tag. Besides that, the smart home security system on market still can be improved to higher security level and build with low cost material to fulfil the need of market. Regarding to this issue, anti-theft security system using face recognition project have been started for solve this issue. It can help to solve the issues like it can trigger the alarm and capture an image send to owner when a stranger is detected in front of door. The theft will get alert so he got no time to start breaking the door and destroy the security system. For overcome the weakness of face recognition that can be by pass with owner face image so the door will only unlock with the correct password input with a keypad.

1.2 Project Scope

The scope of this project is developing an embedded system with the implement of face recognition algorithm. This project involves a new design with low cost material but higher security level. Hence, it was design to against the weakness of security system on market. The boundary or coverage of this project was limited to household uses. The design architecture, the structure of the embedded system and the programming skill are included in this project.

1.3 Project Objective

Despite of there are many similar products are available in the market, however there are still a lot of improvement can be done. The objectives of this project are:

- To design and develop an anti-theft security system which is portable, high efficiency, low cost and easy to operate.
- To detect the present of person when he or she reach in front the door and the the face recognizer will recognize owner face.
- To notify owner and start surveillance system in real time when stranger detected.
- To record the door access about time, user and locker status.
- To have door access time restrictions

1.4 Impact, Significance and Contribution

The main impact and significance of this project is to design and develop an anti-theft security system which is portable, high efficiency, low cost and easy to operate. One of the features that very useful and convenient to user is remote access control. User can control the door access with application when he not at home or even during vacation at foreign country.

Furthermore, it also has the door access audit trail so user can check the record detail for during which time anyone going in your house. House owner can use this feature to track their family member and children when they come back and go in the house and their friend who enter the house also will be capture and recorded along with time log. House owner can found out the record if any incident happened and easier for track.

Moreover, users no need to worry about stolen or lost key because the system is keyless. It makes our life easier and convenient users only need to input user face image data and remember the password for door entry. It also increases the difficulty for theft to enter the house. Users no need to prepare a bunch of spare key for visitors who might have stay for a short period of time. No need to hide the spare key under carpet or somewhere else anymore.

Besides that, the door access can set the day or time restrictions. User can restrict the access of babysitter, repairman and even a stranger. Children can be limited to only have the right to enter the house before 12am midnight. Good behavior of child can be train with the access control. Industrials would like this features since they want their worker always have punctuality on start of work and end of work. On the same time, when the worker work for overtime their working hours can be easier with the log file.

Indeed, this project is to ensure that the problems stated as above can be solved and benefit everyone from having a better, well performed and more functional anti-theft security system for their home to increase security level.

1.5 Background information

The development of Information Technology (IT) has led the rapid change in human lifestyle. Modern advance of electronic and communication technology have made the application of computer, networking and mobile devices to be implemented in our daily life. These changes have catalyzed the development of Internet of Things (IoT) which includes the smart home technologies.

Smart home is the integration of Information Technology and various services through computing, networking and electronic components to bring a better living quality. According to Oxford Dictionaries, smart home is defined as "A home equipped with lighting, heating, and electronic devices that can be controlled remotely by smartphone or computer". Smart homes are built to improve the home comfort, security and efficiency of energy management. In addition, it can be used to assist the elder people and those with disabilities to provide a comfortable and secure living environment. So it's important that a smart home should include the three major elements which are internal network, intelligent control and home automation.

According to the Mobile Statistic Report (2015-2019) by The Radicati Group, the number of worldwide mobile devices (phones & tablets) is 9,568 million in the year of 2015. Meanwhile, the statistic published by Population Reference Bureau (PRB, 2015) shows that there is only 7,336 million of populations globally in 2015. The Radicati Group predicts the total number of mobile devices in the world will

exceeds 14,000 million by the year of 2019. The increasing popularity of mobile devices and rapid expansion of the Internet have made the smart home to be connected with mobile gadgets and, accessibility and administrative of the system has become one of the key performance of a smart home system.

Increasing urbanization and expanding Internet technologies, connected things are expected to more than 2.6 billion in urbanized area by the year of 2017, and there is around 1 billion of connected things are installed in smart homes (Gartner, 2015). The connected things are including smart LED lighting, smart locks and various sensors. Thus, this project will able to give a clear picture to system developer and programmer in designing a single platform smart home system. The smart home system also plays a role to balance the challenge of resource constraints against environmental sustainability. Smart home systems are able to control energy consumption efficiently when increasing IoTs are connected to our home.

Besides, most of the ready smart home products or systems are expensive. As discussed in the Problem Statement, a full set of smart home solution may costs more than RM 10,000. It's definitely can't afford by most of the consumers. In addition, most of the smart home solutions are in close system. It is not compatible with other manufacturers' device and refuse additional appliances. Thus, from this project, it is able to encourage the existing smart home manufacturer to produce a more affordable and extensible smart home system.

Last but not least, this project also can be used in any academic study and make it as a reference. Any future study which related can also be done base on this project.

1.6 Achievement

1. Face detect and face recognition successful implement on door lock system.
2. A door lock system which unlock and lock the door by using phone, keypad or remote has been implemented.
3. Surveillance cam has been implemented on this system.
4. A low cost and efficient security system successful build.
5. A door access audit trail system successful build.
6. A time restrictions system has been integrated on this security system.

1.7 Report Organization

Chapter 1 is about the introduction of project, problem statement and motivation, project scope, project objective, background information, impact, significance and contribution. Chapter 2 discuss about literature review, comparison of previous work and review. Chapter 3 talks about the system design of project, describe the flow of system, explain the process and the system setup. It included block diagram, breadboard diagram, schematic diagram and system flow chart. Chapter 4 is about design specification, requirement and methodology of system design. Besides that, chapter 5 discuss about the implementation and testing which included figure to explain and show how the system work. Chapter 6 is about the conclusion and the improvement can be done for this project in future.

CHAPTER 2: LITERATURE REVIEW

2.1 Discussion on other research

Based on the research found, there are some project works related to the face recognition on security system. Through a research paper "Web-based online embedded door access control and home security system based on face recognition" written by Sahani, M., Nanda, C., Sahu, A.K. and Pattnaik, B.. The strength and weakness on their product can be identified after analyse.

The strength of their product is they used wireless network technique ZigBee based. The ZigBee module combine with electromagnetic door lock module to operate the door accessibility. The proposed system is designed with wireless access control so the lock module can be added easily if have the need. Email and SMS are used to notify the house owner when detected a stranger face. It helps to reduce the need of server so the user can directly login and control the embedded system in real time. User can control the system with SMS, email and website.

However, it still can be improve from it weaknesses. The face recognition can be by pass with a photo of owner face. The system can be improve with add on password authentication, sound recognition or fingerprint authentication. The product cost can be lower with reduce the SMS module and use the WIFI module as replacement. Since our phone always connected to internet and the latency should be lower if compare with GSM network.

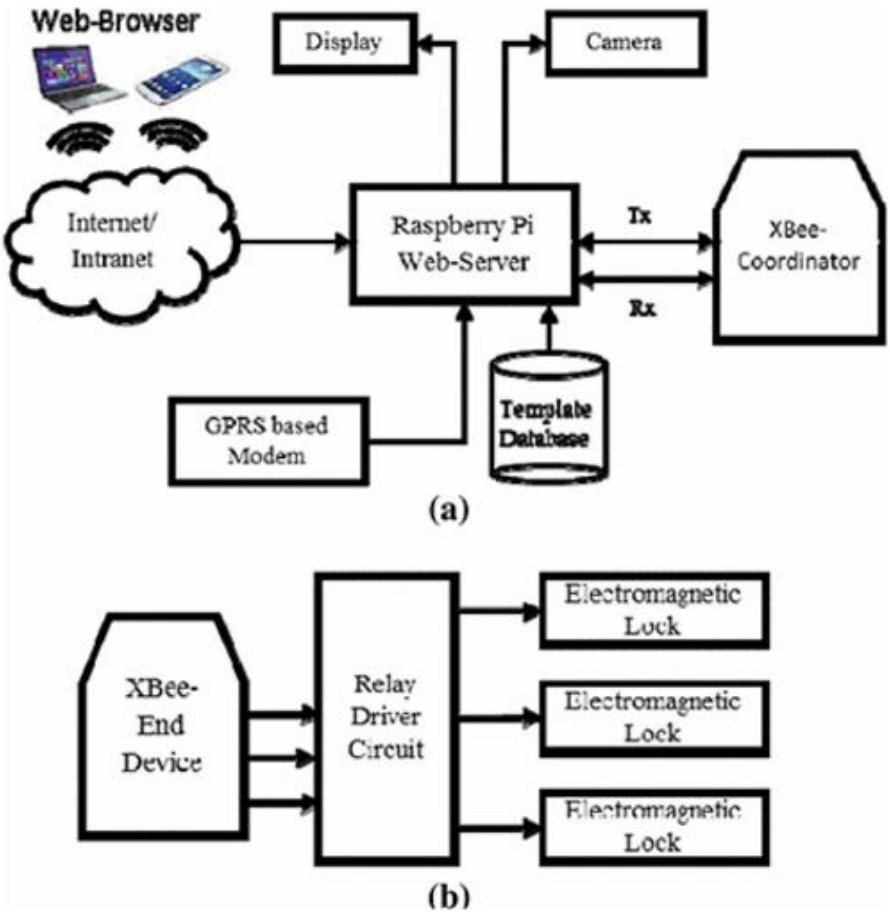


Figure 2.1.1 Overview of the proposed system configuration layout; (a) wireless information unit, (b) wireless control unit.

Furthermore, another research was done by Thabet, A. and Amor, N. with the title "Enhanced smart doorbell system based on face recognition". There have a few factors that marked their system strength and weaknesses.

First and foremost, the strength of the system is optimizing on the size of data in database. The face images are stored with 2D array with 8 bit intensity values in grey scale. When detected a face image it will subtract a result vector from eigen face vector to get the actual data for matching the data image with database in the shortest time. PCA had been used to create a linear combination of vector to minimize memory usage (Pissarenko, D. ,2002). A better algorithm was proposed on this research with higher recognition rates. For improve the accuracy of result, face alignment and scaling filters had been utilized on image process.

Nevertheless, weakness of the embedded system can be found. The camera for face detection will only start to capture when the doorbell button was pressed. It could be improve with install an infrared PIR motion sensor to detect the present of visitor so it will automate the action of pressing the doorbell. It would be more effective and can notify the house owner on time when any stranger present in the house. The security can be improve due to this modification.

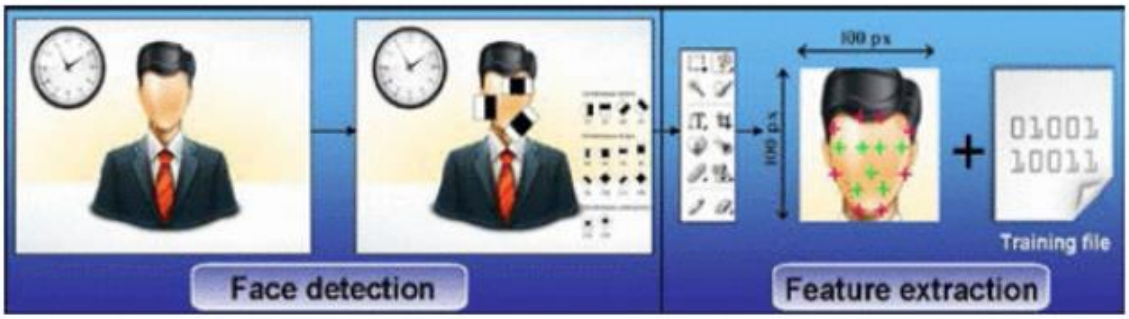


Figure 2.1.2 Overview face detection and feature extraction process

A project name "Automatic Door Locking System" developed by (Majgaonkar, Et Al., 2016). There are some strength and weakness on his product which can be identified.

The project is developed with the uses of Bluetooth device and Microcontroller technology which is a low budget product. User can lock and unlock the door with password and the password can be reset. It has two level of security, the first level is key in password for Bluetooth connection with phone and the second level is unlock the door with self-develop Android mobile application.

Bluetooth module use as a data receiver to get the password and transfer to microcontroller. Microcontroller act as a data processing center to verify the entered password and then unlock the locker. A battery use as a power supply and a LED use as a indicator.

The door locking system will process to verify the password data send by mobile application and unlock the locker. Bluetooth module require the same baud rate with Microcontroller for communication if not it will not able to interpret data sent out by

mobile phone. The data send out by mobile phone will received by Bluetooth module and transfer to Microcontroller. The microcontroller will process the data and verify the password for unlock the door.

The weakness of this product is it cannot remote access control the system because it use the technology of Bluetooth but not Wi-Fi connection. It can only be access when you near by the device. There is also no way for you to key in password except with the use of mobile application. The door cannot be unlock if the mobile application crash or when you lost your mobile phone.

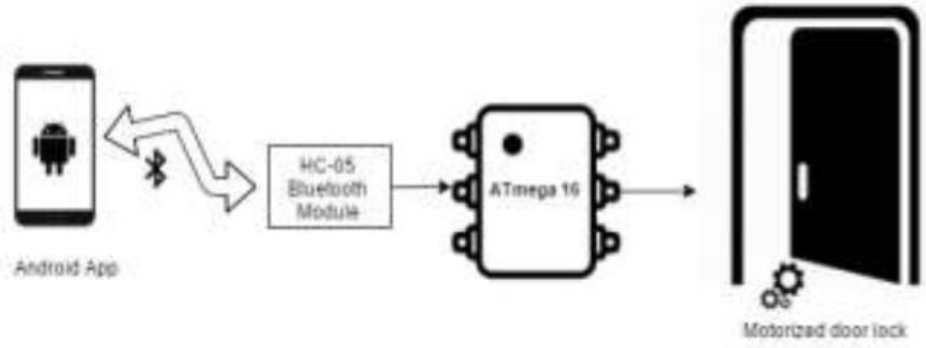


Figure 2.1.3 Block diagram of Door locking System

2.2 Critical Remarks of previous works

Figure 2.2.1 Keypad password and RFID door lock

The older security door lock system is using keypad password to control the mechanism knob. It require user to remember the password. The cons of this system is for the older age and younger user it is not easy for them to memorize the combination of password and also it is easy for thief to crack it with the technology nowadays. Another product also had been produce to identify user identity that is RFID card. It just like a key but in card form and also it can unlock the door within second. However it is also easy for other to make a copy of your RFID card with the technology today.

Figure 2.2.2 Fingerprint door lock

Fingerprint is also widely used as a way to identify a person identity. That why it is also had been used for security door lock. Everyone have their own unique fingerprint except some disabled person and very small group of people in this world they do not have fingerprint. Nevertheless there are still a bigger group of person could not be identify by fingerprint sensor. This is because they have a poor quality of fingerprint cause by their age, occupation and the other factors. The main reason why biometrics fingerprint door lock not widely used at household is because the high price tag.

CHAPTER 3: SYSTEM DESIGN

3.1 Description of Project

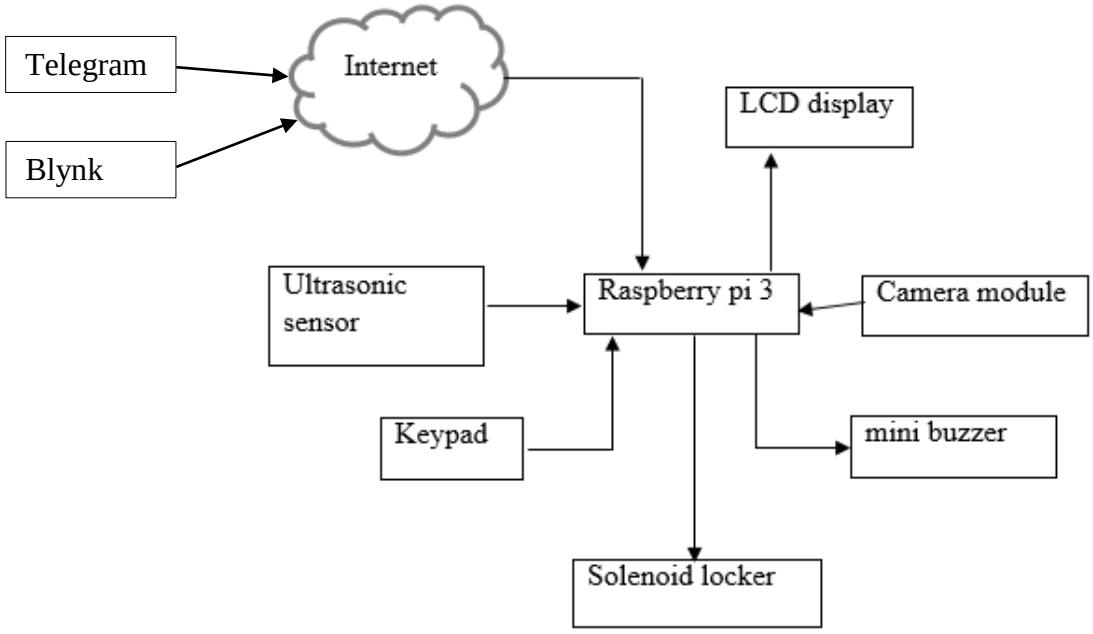


Figure 3.1.1 System Block Diagram

Anti-theft Security System need the use of software part and hardware part combine together to perform its functionality.

Firstly, the software part will be used is Telegram and Blynk. Both of them are mobile applications. System will start surveillance function and capture photo send to owner to notify him with Telegram when stranger try to enter. Owner can control the locker on and off wirelessly with the mobile application Blynk.

OpenCV will be used in this project to code the image processing algorithm. It is an open source computer vision library compatible to run in Raspberry pi.

For the hardware part, a Raspberry pi 3 model B, Solenoid electric door locker, keypad, 1602 LCD display, ultrasonic sensor module, camera module and mini buzzer. Raspberry pi will take responsible to process all the workload. When the ultrasonic sensor detect a person come near the door it will trigger the camera to start capture the person face and compare with the database if a stranger detected it will notify house owner with telegram via Wi-Fi and start giving alert with the buzzer. If the person was match with database then he or she will need to key in the password with keypad to unlock the solenoid locker.

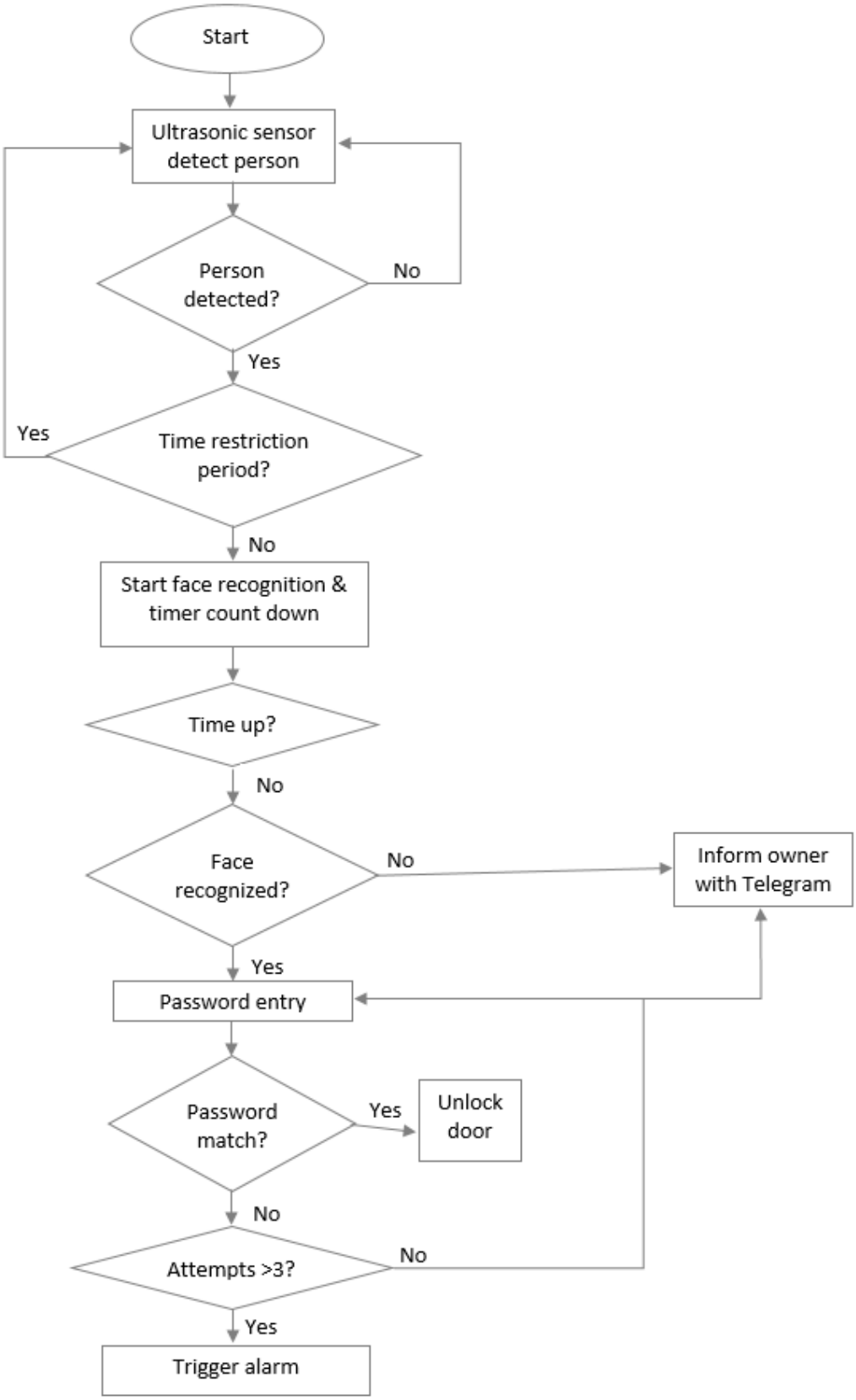


Figure 3.1.2 System Flowchart

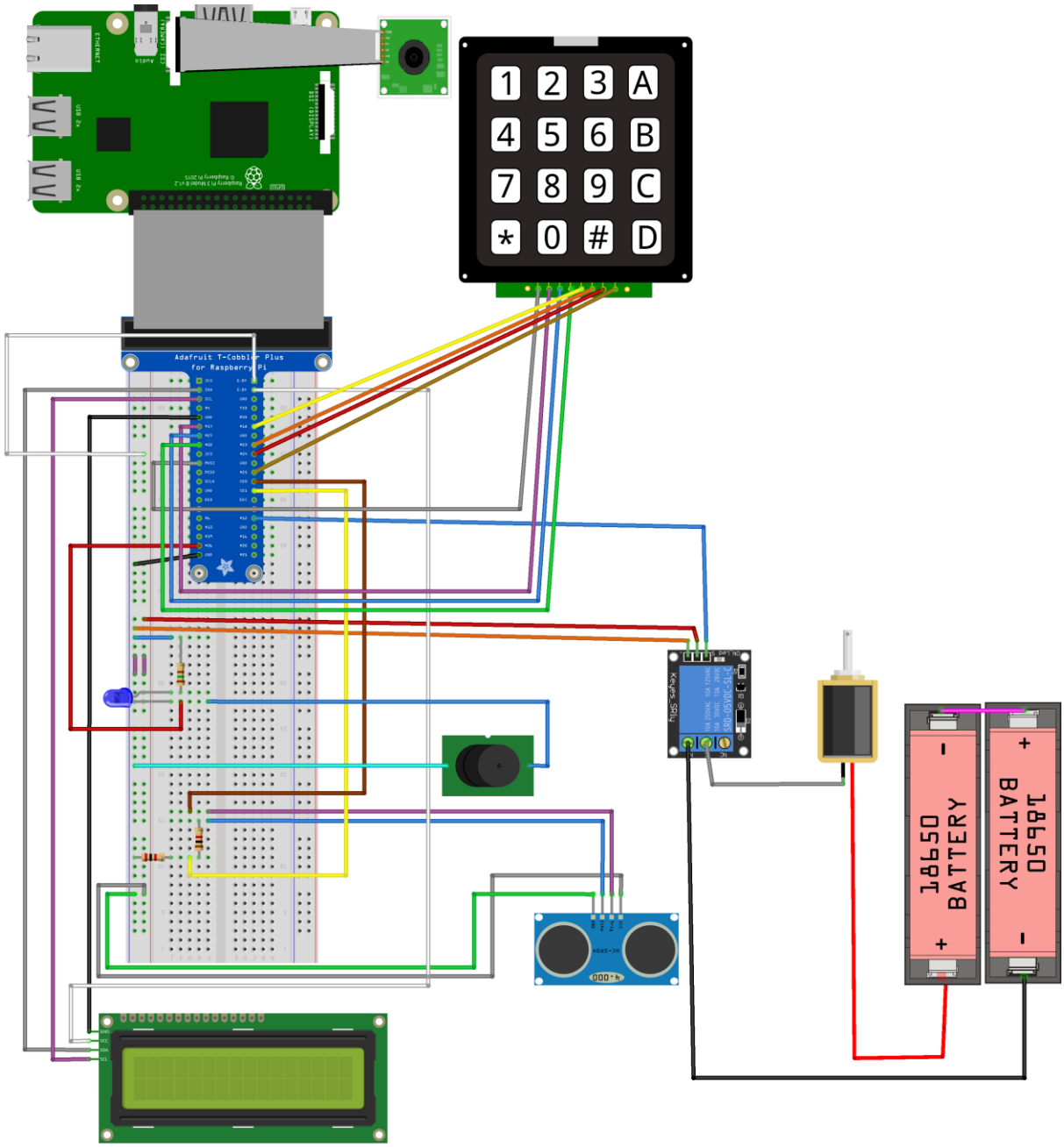


Figure 3.1.3 Breadboard Diagram

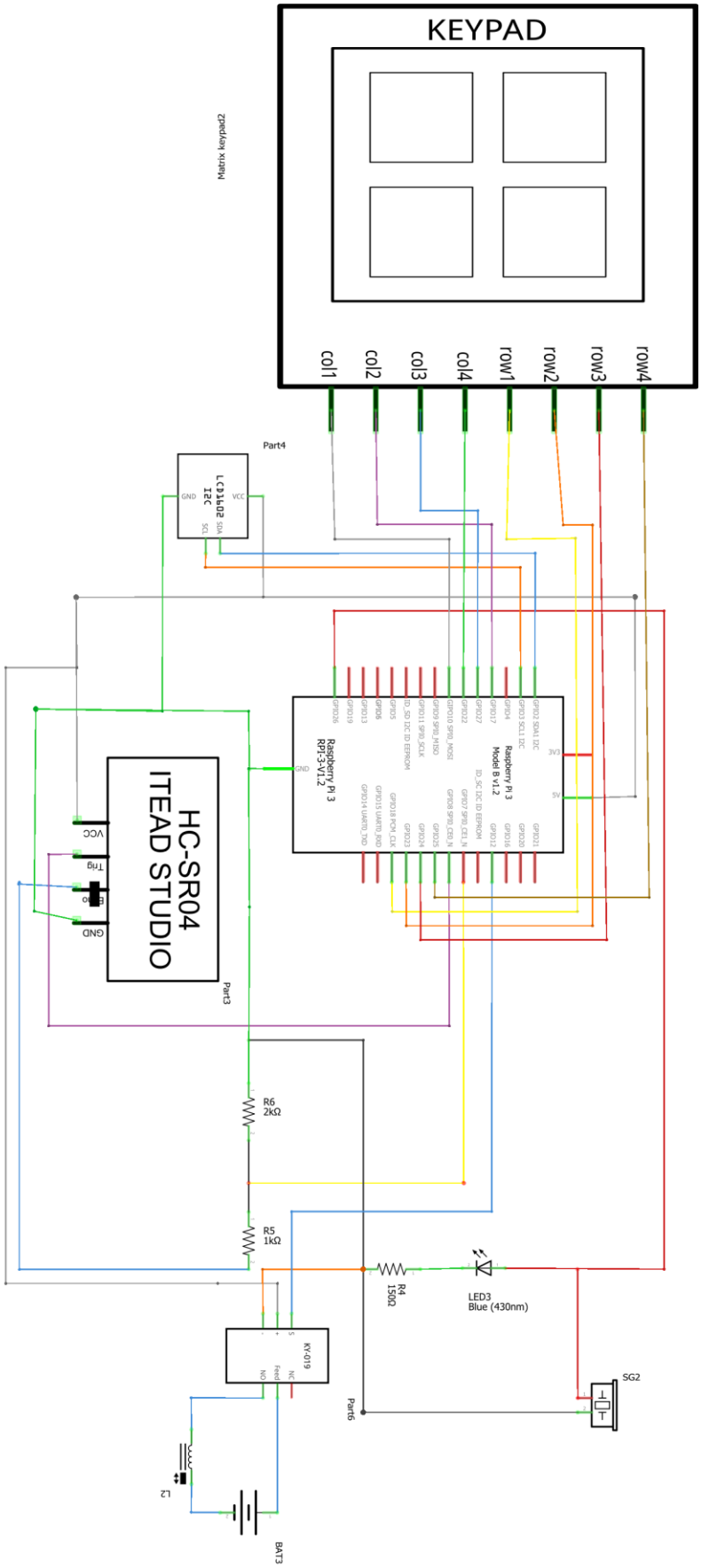


Figure 3.1.4 Schematic Diagram

3.2 Process Flow of System

3.2.1 Raspberry Pi

When the Raspberry Pi start up the code will be run are main.py, led_vkey.js, lockstcheck.py and telegrambot.py. The main.py is function to control the whole system, led_vkey.js is to allow wireless network control by the mobile application Blynk to control the locker, telegrambot.py is to allow the communication between telegram and Raspberry Pi. The lockstcheck.py is use for door access records when the door locked and when the door unlocked.

3.2.2 Telegram

Telegram will send notify to owner about the surveillance status and capture photo. Telegram can be used to control the locker by sending command “On” or “Off” to control the locker and also capture photo by the command “C”. Command “S” can use to start a live cam and end it with the command “Stop”. Besides that, command “Shutdown” and “Rbt” can used to turn off the system and reboot the system. For checking the door access records can use command “Cr”. Time restrictions start time and finish time can be set with command “Rts d’DAY’ h’HOUR’ m’MINUTE’ ” and “Rtf d’DAY’ h’HOUR’ m’MINUTE’ ”.

3.3 System Setup Procedures

3.3.1 Raspbian OS

The first thing to do with the Raspberry Pi is to install the operating system, Raspbian. The image of Raspbian is downloaded from Raspberry Pi official website. After that, the image file is written into a MicroSD card by image writing tool.

After installing the operating system, Raspberry Pi is booted and have a system configuration, including changing log in password, update the Locale setting, set hostname and creating a new user. The initial set up is followed by connecting the Raspberry Pi to the Internet wirelessly. With the Internet access, the operating system is updated to the latest version. In order to avoid the changing of Raspberry Pi’s IP address

while rebooting Router or Raspberry Pi, it's a necessity that we use a fixed IP address. The IP address has been set to a static address within the network range.

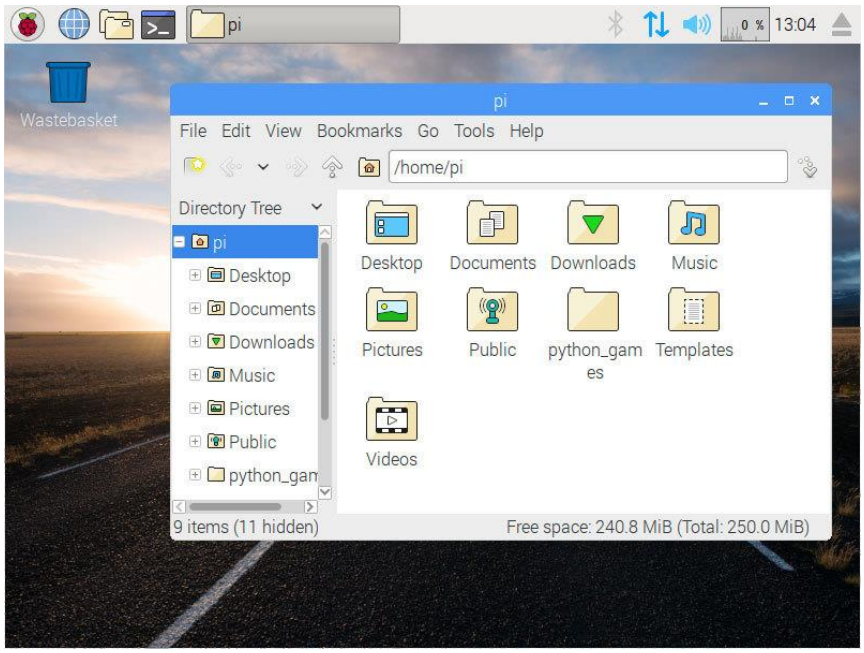


Figure 3.3.1.1 Raspbian OS Screenshot

3.3.2 Telegram

. Telegram app in Playstore is searched and downloaded. It is logged in with the user's phone number. Go to the website https://web.telegram.org/ to access Web Telegram through a PC. Go to the website https://botsfortelegram.com/project/the-bot-father/ to add BotFather through Web Telegram by clicking "Add BotFather To Telegram". After added, type '/start' at the chat bar of BotFather and press send. BotFather will reply a long message contain a lot of commands, then type "/newbot". Then, BotFather will reply a message "Alright, a new bot. How are we going to call it? Please choose a name for your bot.", "Pi3" is typed as the name of the bot. Next, BotFather will reply another message "Good. Now let's choose a username for your bot. It must end in `bot`. Like this, for example: TetrisBot or tetris_bot.", "Pi3bot" is typed as the username. Lastly, BotFather will reply a message contained the token. This token is used to access the HTTP API which is to communicate between this bot and the other software. This token is later added in the Python code which is going to be run in Raspberry Pi.

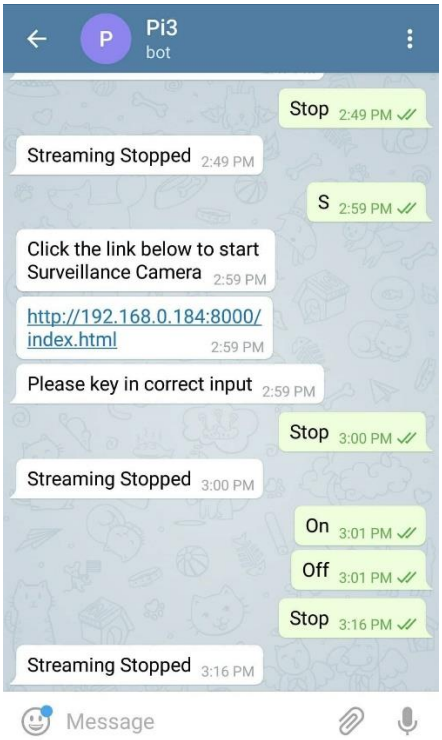


Figure 3.3.2.1 Telegram Screenshot

3.3.3 VNC Viewer

Enable VNC Server with select Menu > Preferences > Raspberry Pi Configurations > Interfaces, and select “Enabled” for VNC. Now open VNC Server, click on the “Menu” on top-right corner and select “Licensing…”. Select “Sign in to your RealVNC account”. After signing in, go to VNC Viewer, click on “File” on top-left corner, and select “New Connection…”. Type the IP address in the VNC server, and type a name for this connection, and click “OK”. Connect to the Raspberry Pi by double clicking the connection which has just been created. A pop up message will be prompted to enter its username and password. VNC set up is done.

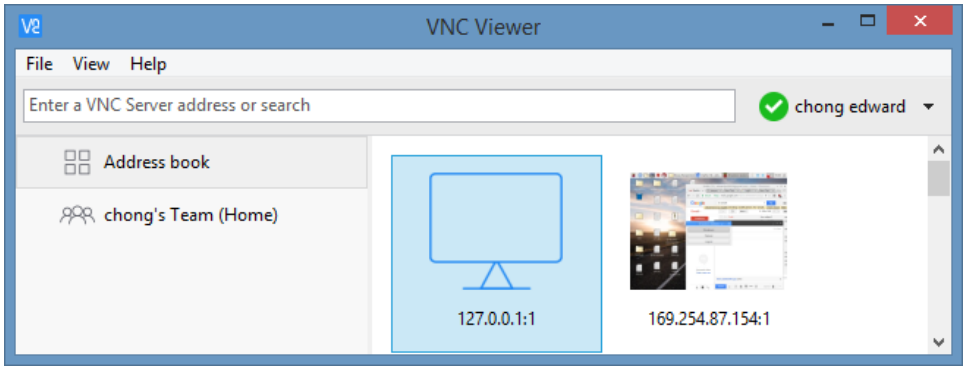


Figure 3.3.3.1 VNC Viewer Screenshot

3.4 Software Development

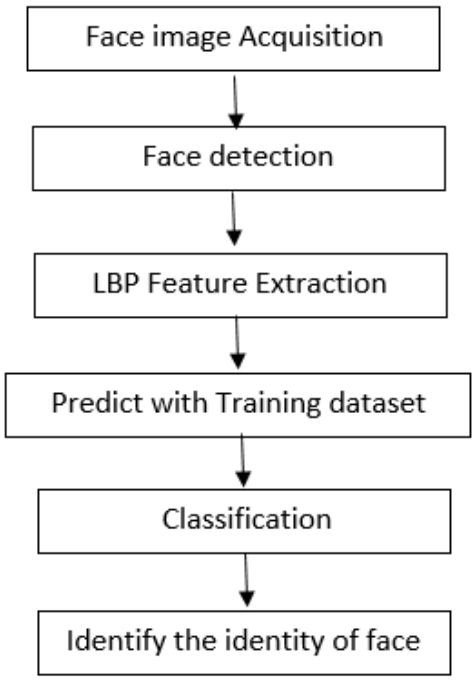


Figure 3.4.1 Flow of Face recognition

3.4.1 Process of Face Recognition

With the OpenCV library which developed by Intel it can be easily connect to Pi Cam module and start image capturing and video capturing. Firstly, enable the camera module with the code, os.system("sudo modprobe bcm2835-v4l2") . Create a Local Binary Patterns Histograms for face recognition, cv2.face.createLBPHFaceRecognizer(). Load the training set prepared earlier, recognizer.load('Face_Recognition/trainer/trainer.yml').

Haar Cascade face detection module by OpenCV is loaded, cascadePath = "Face_Recognition/haarcascade_frontalface_default.xml". Create classifier from prebuild model, faceCascade = cv2.CascadeClassifier(cascadePath). Start the video frame capturing, cam = cv2.VideoCapture(0).

```
def facerecog():
    os.system("sudo modprobe bcm2835-v412")

#Open a text file of facerecog for record
    f = open("facerecog.txt", "w")
    f.write("0")


# Create Local Binary Patterns Histograms for face recognization
    recognizer = cv2.face.createLBPHFaceRecognizer()

# Load the trained mode
    recognizer.load('Face_Recognition/trainer/trainer.yml')

# Load prebuilt model for Frontal Face
    cascadePath = "Face_Recognition/haarcascade_frontalface_default.xml"

# Create classifier from prebuilt model
    faceCascade = cv2.CascadeClassifier(cascadePath);

# Set the font style
    font = cv2.FONT_HERSHEY_SIMPLEX

# Initialize and start the video frame capture
    cam = cv2.VideoCapture(0)
    K_count = 0
    U_count = 0
```

Figure 3.4.1.1 Coding for Face Recognition initialization

Start reading from video frame with the code, cam.read(). Convert the captured frame into grayscale, cv2.cvtColor(im,cv2.COLOR_BGR2GRAY). Detect the faces from captured image frame, faceCascade.detectMultiScale(gray, 1.2, 6). Recognize the captured face image with previous training dataset and get the confidence value, Id , conf = recognizer.predict(gray[y:y+h,x:x+w]). The higher the confidence value, the higher error rate on recognition. Zero confidence value mean prefect recognition normally it will only happen when u comparing exactly the same captured image manually. From the code, the confidence value is tuned at more than or equal to 70.0.

If the confidence value obtained more than or equal to that value it will be classified as a stranger because it did not match with any trained dataset.

```
while True:
# Read the video frame
    ret,im =cam.read()

# Convert the captured frame into grayscale
    gray = cv2.cvtColor(im,cv2.COLOR_BGR2GRAY)

# Get all face from the video frame
#faces = faceCascade.detectMultiScale(gray, 1.2,5)
    faces = faceCascade.detectMultiScale(gray, 1.2, 6)

# For each face in faces
    for(x,y,w,h) in faces:

    # Create rectangle around the face
        cv2.rectangle(im, (x-20,y-20), (x+w+20,y+h+20), (0,255,0), 4)

    # Recognize the face belongs to which ID
        Id , conf = recognizer.predict(gray[y:y+h,x:x+w])

    # Check the ID if exist
    # Confidence level more than 70 then the ID is Unknown
        print("cam_conf: " +str(conf))
        if (conf <= 70):
```

Figure 3.4.1.2 Coding for face recognition

3.4.2 Process of Training Dataset

```
def getImagesAndLabels(path):

    # Get all file path
    imagePaths = [os.path.join(path,f) for f in os.listdir(path)]

    # Initialize empty face sample
    faceSamples=[]

    # Initialize empty id
    ids = []
```

Figure 3.4.2.1 Coding for Training dataset initialization

A function getImagesAndLabels() had been create to take the path to captured owner image dataset as a input and return two list which is faceSamples and ids.

faceSamples contain the detected face and the ids contain the id correspond to the face from dataset. The list imagePaths append all the path names from dataset images. Then initialize two lists which is faceSamples and ids.

```
# Loop all the file path
for imagePath in imagePaths:

    # Get the image and convert it to grayscale
    PIL_img = Image.open(imagePath).convert('L')

    # PIL image to numpy array
    img_numpy = np.array(PIL_img,'uint8')

    # Get the image id
    id = int(os.path.split(imagePath)[-1].split(".")[1])
    print(id)

    # Get the face from the training images
    faces = detector.detectMultiScale(img_numpy)

    # Loop for each face, append to their respective ID
    for (x,y,w,h) in faces:

        # Add the image to face samples
        faceSamples.append(img_numpy[y:y+h,x:x+w])

        # Add the ID to IDs
        ids.append(id)

# Pass the face array and IDs array
return faceSamples,ids
```

Figure 3.4.2.2 Coding for listing the Training dataset

The for loop will loop every face image to detect the face and update the two lists. Dataset images created with jpg format so PIL image module is used to read the face image in dataset in greyscale format and it had been converted to numpy arrays to make it compatible with OpenCV. Image “id” will be extracted from image name from dataset and will be used to label on recognized face. The function detector.detectMultiScale( ) will return a list of face from dataset. Each face returns in the rectangle format, top left x pixel, top left y pixel, width and height.

Run a for loop on region of interest and append it to faceSamples list and append it to ids list. After done the for loop it will return faceSamples list and ids list.

```
# Get the faces and IDs
faces,ids = getImagesAndLabels('dataset')

# Train the model using the faces and IDs
recognizer.train(faces, np.array(ids))

# Save the model into trainer.yml
recognizer.save('trainer/trainer.yml')
```

Figure 3.4.2.3 Coding for perform training

Start the training with recognizer.train(faces, np.array(ids)), it requires faces and ids belong to each face image id. Lastly, the trained data will be saved at a file name "trainer.yml".

3.4.3 Process of Prepare Face Dataset

```
# Start capturing video
vid_cam = cv2.VideoCapture(0)

# Detect object in video stream using Haarcascade Frontal Face
face_detector = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

# For each person, one face id
face_id = 3

# Initialize sample face image
count = 0
```

Figure 3.4.3.1 Coding for prepare face dataset initialization

First, assign the face_id with number like first family member "1", second family member "2". Do not repeat the number except you want to make replacement.

```
while(True):

    # Capture video frame
    _, image_frame = vid_cam.read()

    # Convert frame to grayscale
    gray = cv2.cvtColor(image_frame, cv2.COLOR_BGR2GRAY)

    # Detect frames of different sizes, list of faces rectangles
    faces = face_detector.detectMultiScale(gray, 1.2, 5)

# Loops for each faces
for (x,y,w,h) in faces:

    # Crop the image frame into rectangle
    cv2.rectangle(image_frame, (x,y), (x+w,y+h), (255,0,0), 2)

    # Increment sample face image
    count += 1

    # Save the captured image into the datasets folder
    cv2.imwrite("dataset/User." + str(face_id) + '.' + str(count) + ".jpg", gray[y:y+h,x:x+w])

    # Display the video frame, with bounded rectangle on the person's face
    cv2.imshow('frame', image_frame)

# To stop taking video, press 'q' for at least 100ms
if cv2.waitKey(100) & 0xFF == ord('q'):
    break

# If image taken reach 50, stop taking video
elif count>=50:
    break
```

Figure 3.4.3.2 Coding collect face image dataset

Start capture the video frame with vid_cam.read(). Then covert the image frame to greyscale, cv2.cvtColor(image_frame, cv2.COLOR_BGR2GRAY). Detect the face with face_detector.detectMultiScale(gray, 1.2, 5). Run for loop for each captured face. Crop the captured face image into rectangle with cv2.rectangle( ). Increment of captured face image with count += 1. Save the captured face image into dataset folder with the function cv2.imwrite( ). Image file will be named as User.1.50.jpg (User.<id>.<number of image>.jpg). Show the captured image with the function cv2.imshow( ). When the count of captured image reach 50 break the for loop.

CHAPTER 4: DESIGN SPECIFICATION

4.1 Methodologies

4.1.1 System Development Methodology

System development life cycle (SDLC) describe the stages of development project. The life cycle defines a methodology for improving the quality of overall development process. By using SDLC model, project can be implemented efficiently and smoothly. Various SDLC model have been developed to help the processes of developing system, for example, waterfall model, spiral model, prototyping model and so on. In this project Prototyping model is used.

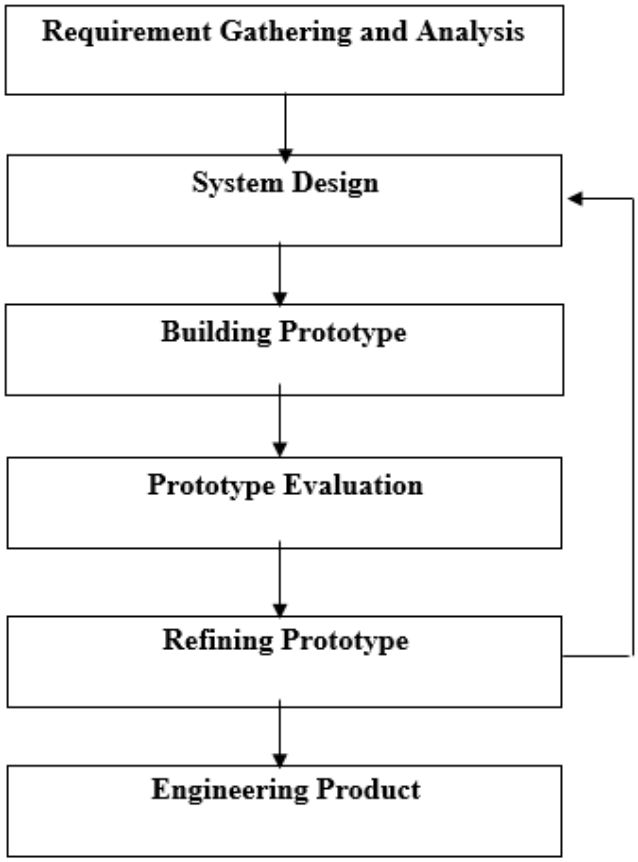


Figure 4.1.1.1 Prototyping model

4.1.2 Project Flow

In order to make the project development running smoothly and complete in scheduled period, the project is divided into five phases which started with project planning, hardware development and software development. After that it's followed by simulation and lastly, build the prototype. The process flow of the project is shown as follow.

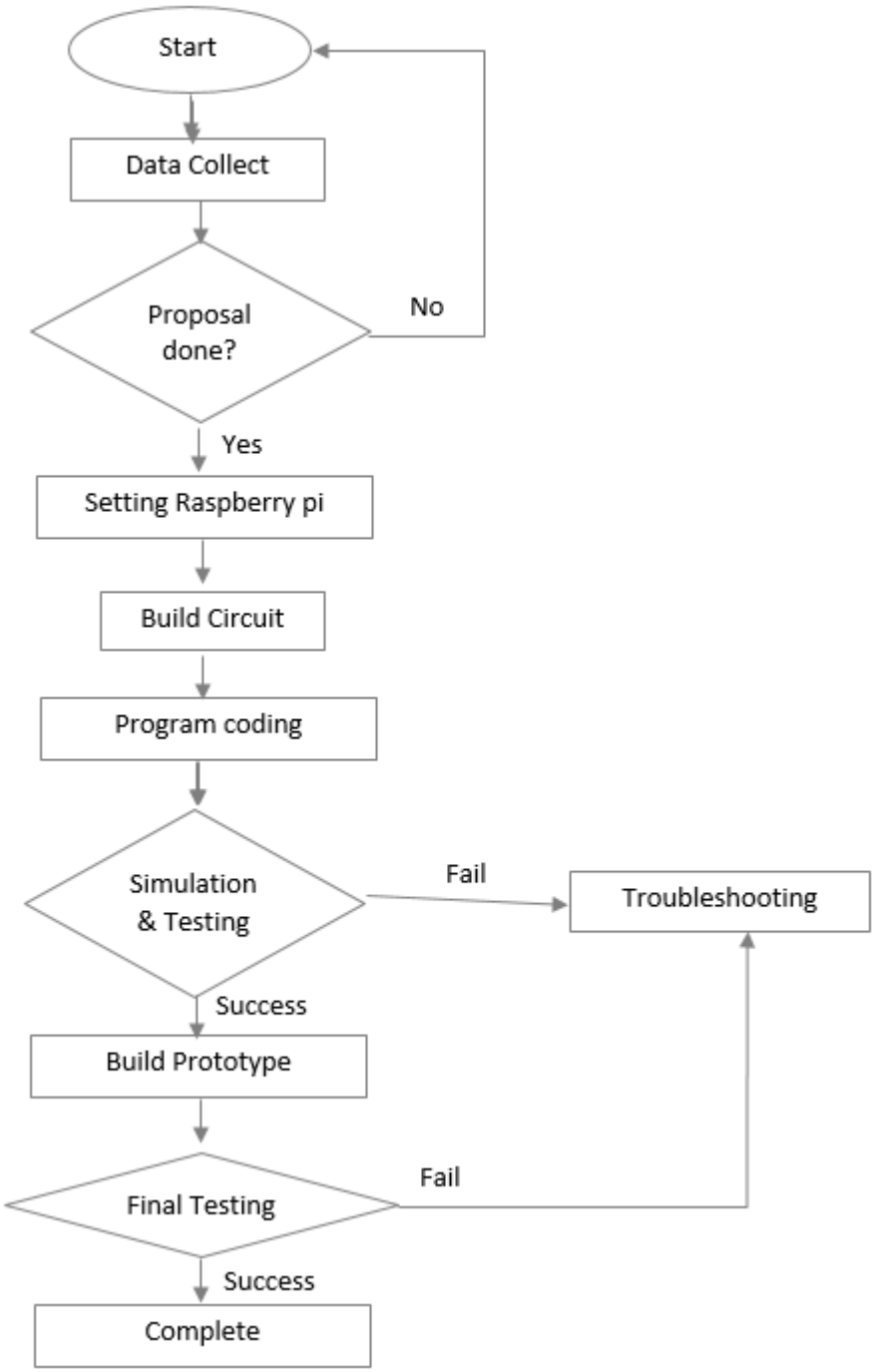


Figure 4.1.2.1 Flow of Project

4.2 Tools

This Anti-Theft Security System is design with some hardware combine with software to work. The software need to handle math calculation on image processing to verify the identity of user. Hardware used also need to have the ability to process the whole system flow smoothly.

Several hardware components are prepared for the system development.

Item	Quantity	Remarks
Raspberry pi 3 model B	1	Single board computer to process the whole system flow.
Raspberry pi camera module	1	Capture image
I2C LCD1602 display	1	Display password input
Buzzer	1	Alert stranger and indicate status
Solenoid locker	1	Lock and unlock the door
LED light bulb	2	Indicate status
4x4 matrix keypad	1	For password input
5V relay module	1	Control larger current solenoid
Resistor	1	Protect the component with lower down voltage
Breadboard	1	Provide a construction base for developing circuit
Ultrasonic sensor	1	Estimate the present of people in front door
18650 battery	2	Provide sufficient current to control solenoid

Table 4.2.1 Hardware components required

On the other hand, this project also need program coding to control all the hardware. For the image processing part, OpenCV is chosen because it is open source distribution. So the development cost for whole project would be over budget. Beside that Python programming language is used for this project to control the system. Due to it have a lot of library can make the development easier to control the hardware components.

4.2.1 System Hardware

4.2.1.1 Raspberry pi 3 model B

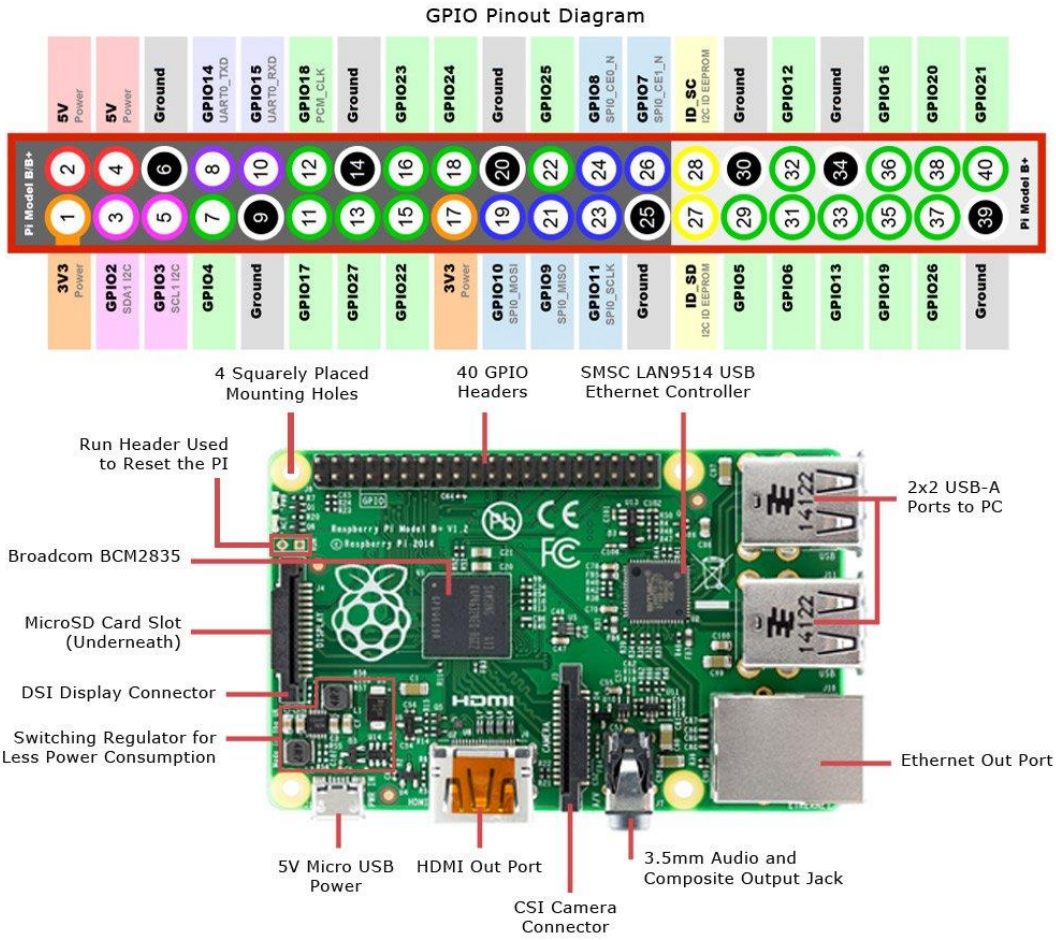


Figure 4.2.1.1.1 Raspberry pi 3 model B GPIO pinout diagram

The Raspberry Pi 3 Model B was designed with open source Linux-based and non-Linux-based operating system. This development board is third generation of small single board computers developed by Raspberry Pi Foundation. Third generation has been chosen for this project is because older generation of Raspberry pi need to install Wi-Fi adapter for internet connection. Raspberry pi 3 model B already have built in Wi-Fi chipset, more RAM and faster processor compare to older generation. Below is the specification details of Raspberry pi 3 model B.

Device	Raspberry Pi 3 Model B
SoC	BCM2837
CPU	Quad Cortex A53 @ 1.2GHz
Instruction set	ARMv8-A
GPU	400MHz VideoCore IV
RAM	1GB SDRAM
Storage	Micro-SD
Ethernet (Mbps)	10 / 100
Wireless	802.11n / Bluetooth 4.0
Video Output	HDMI
Audio Output	HDMI / 3.5mm audio jack
USB port	4
GPIO	40
Interfaces port	CSI Camera port and DSI Display port

Table 4.2.1.1.1 Specification details of Raspberry pi 3 model B

4.2.1.2 Raspberry pi camera module

Figure 4.2.1.2.1 Raspberry pi camera module V1.3 with CSI ribbon cable

Raspberry pi camera module V1.3 has been chosen for this project is because it is cheaper compare with newer version of pi camera module and also it is more efficiency and use lesser power compare with USB webcam. The resolution of this pi camera is good enough to fulfill the requirement of this project so newer version with higher price tag is not necessary. Pi camera modules can connected directly to Raspberry pi's GPU so it can encode video at 1080p 30 frames per second. There is also use lesser CPU resources because it attached to GPU. In contrast USB webcam require more power input and consume more CPU resources. Below is the specification details of Raspberry pi camera module V1.3.

Still resolution	5 Mega pixels
Video modes	1080p30, 720p60, 640 × 480p60/90
Sensor	OmniVision OV5647
Photo resolution	2592 × 1944 pixels
Pixel size	1.4 µm × 1.4 µm
Fixed focus	1 m to infinity

Table 4.2.1.2.1 Specification details of Raspberry pi camera module V1.3

4.2.1.3 I2C LCD1602 display

Figure 4.2.1.3.1 LCD1602 display with I2C LCD controller

The LCD 16x2 display is use to display information for user and give user an interface to input entry password. The chosen display come with an I2C LCD controller it make the connection simpler because it require lesser raspberry pi GPIO pins for it to work. It total have 4 input pins which is VCC, GND, SDA, and SCL. The data for display only used 2 pins and it will connected to raspberry pi GPIO pins and another pin VCC will connect to 5V power source, GND will connect to ground pin.

I2C LCD1602 display input pin	**Raspberry pi GPIO pinout (number pin)**
VCC	5V (04)
GND	GPIO 17 (11)
SDA	GPIO 2 (03)
SCL	GPIO 3 (05)

Table 4.2.1.3.1 Connection pin of LCD module and Raspberry pi GPIO pinout

4.2.1.4 Buzzer

Figure 4.2.1.4.1 Mini buzzer

The buzzer is used to notify user for every single press on the keypad it will produce a “beep” sound. It also indicate the status of lock and unlock of the door. After unlock the door or lock the door it will notify user. When user fail on third attempt correct password input it will produce a long “beep” sound to alert the user.

4.2.1.5 Solenoid locker

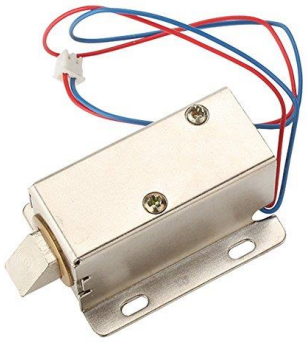


Figure 4.2.1.5.1 Solenoid locker

This solenoid locker require large current to drive it up. It require a minimum of 400mA and 7.5V. A 9V battery only have 200mA so it is not sufficient to trigger it. For provide enough power and lessen the frequent of changing the battery. For ideal experience two pcs of 18650 rechargeable battery had been used to drive this solenoid. The solenoid locker is used to push and pull the locker to unlock and lock the door.

4.2.1.6 HC-SR04 Ultrasonic Sensor

Figure 4.2.1.6.1 HC-SR04 Ultrasonic sensor

HC-SR04 Ultrasonic sensor has been chosen for this project is because it is cheap and can fulfill the need of project. It is use to detect the present of person. It is the main component to make this project energy saving and consume lesser power. The system will only activated when detected person in certain distance.

Working Voltage	**DC 5 V**
Working Current	**15mA**
Working Frequency	**40Hz**
Max Range	**4m**
Min Range	**2cm**
MeasuringAngle	**15 degree**
Trigger Input Signal	**10uS TTL pulse**
Echo Output Signal	**Input TTL lever signal and the range in proportion**
Dimension	**45*20*15mm**

Table 4.2.1.6.1 Specification details of HC-SR04 Ultrasonic sensor

HC-SR04 Ultrasonic sensor input pin	**Raspberry pi GPIO pinout (number pin)**
VCC	5V (+)
TRIG	GPIO 8 (24)
ECHO	GPIO 7 (26)
GND	5V(-)

Table 4.2.1.6.2 Connection pin of HC-SR04 Ultrasonic sensor and Raspberry pi GPIO pinout

4.2.2 System Software

4.2.2.1 Raspbian OS

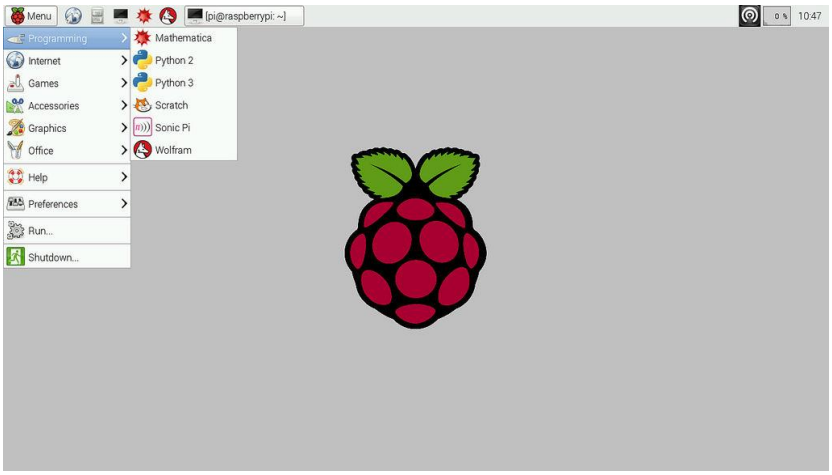


Figure 4.2.2.1.1 Raspbian OS

Raspbian is the free and foundation's official supported operating system based on Debian optimized for the raspberry pi hardware. Raspbian provide more than pure OS if compare to the other operating system. It comes with over 35000 packages, pre-compiled software bundled in a nice format for easy installation on raspberry pi. Software like python IDE, Scratch and more are included in this OS.

4.2.2.2 Python IDE

Figure 4.2.2.1.2 Python IDE

Python IDLE is a free and open source programming software and also is an IDE stand for integrated development environment for Python. In Raspbian OS, python IDE is a built in software and installed with python2 and pyhton3. In this project python IDE will be used to code most of the program including the face recognition and face detection with the use of OpenCV library.

4.2.2.3 Blynk

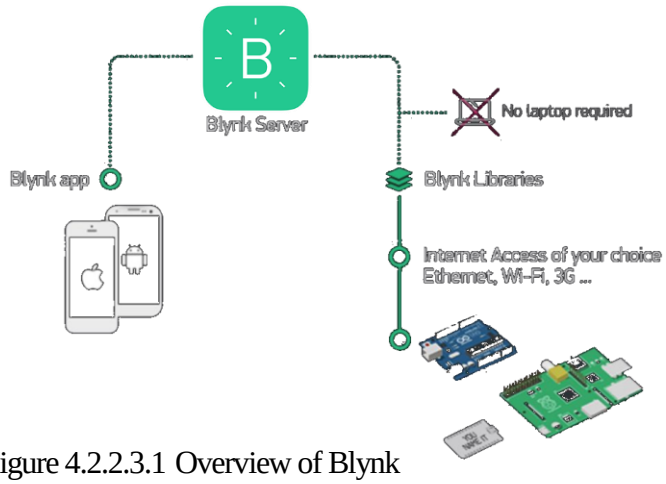


Figure 4.2.2.3.1 Overview of Blynk

Blynk is a mobile application which allow user to use it to monitor and control their self-develop IoT product. It can control Raspberry Pi over the internet. Graphical user interface can be easily build with just simply drag and drop widgets and with some simple programming on your device. . In this project Blynk will use to design an on off button to remote access control the locker.

4.2.2.4 Telegram

Figure 4.2.2.4.1 Telegram

Telegram is an online instant messaging application which can use at computer and mobile phone. However for this project I would like to use its extra feature which is create bot. Telegram allow third party developer to build their own bot. With the bot it can give respond accordingly to different command programmed by developer. In this project it will use to send notification automatically to user when the system have been trigger. User can also use it to control the system by sending command.

CHAPTER 5: IMPLEMENTATION AND TESTING

5.1 Testing Face Recognition

Figure 5.1.1 Sample dataset used for testing

```
9 is Correctly Recognized with confidence 49.93126589956086
7 is Correctly Recognized with confidence 40.524923649357895
8 is Correctly Recognized with confidence 35.926313049761966
3 is Correctly Recognized with confidence 31.43750110281589
4 is Correctly Recognized with confidence 39.8666770462547
5 is Correctly Recognized with confidence 53.44319669646407
6 is Correctly Recognized with confidence 43.5767624577832
2 is Correctly Recognized with confidence 45.88195803741032
1 is Incorrect Recognized as 2 conf 68.21861864102476
```

Figure 5.1.2 Result obtained with the use of sample dataset

Local binary patterns histograms face recognizer from OpenCV has been used for this project. The result show that tested with 9 different person's face have 8 correctly recognized but one incorrect recognized.

5.2 Testing System

5.2.1 Security System

Figure 5.2.1.1 Screenshot notification when stranger detected

Figure 5.2.1.2 Screenshot notification when stranger key in wrong password

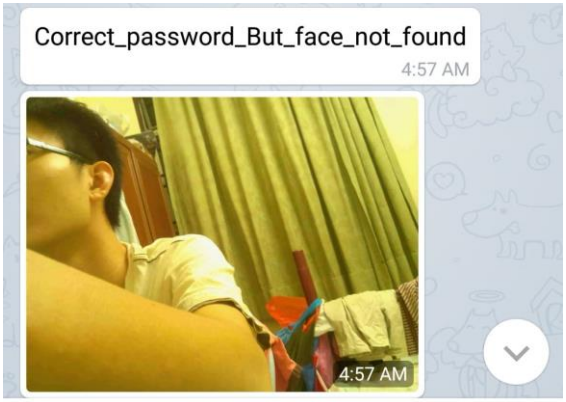


Figure 5.2.1.3 Screenshot notification when correct password but face not in database

Figure 5.2.1.4 Screenshot notification when wrong password but face verified

Figure 5.2.1.5 Screenshot notification send to other when reverse password entered

5.2.2 Surveillance System

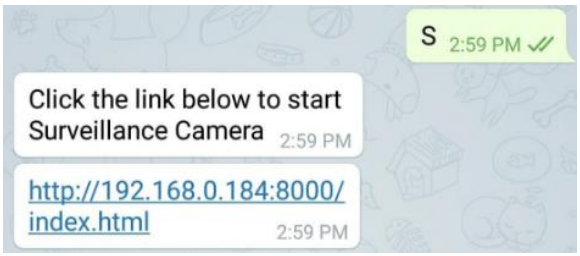


Figure 5.2.2.1 Screenshot commend to activate surveillance cam

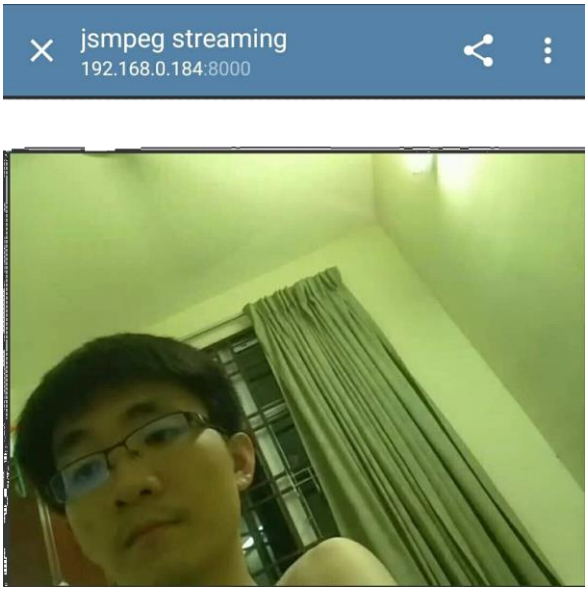


Figure 5.2.2.2 Screenshot of surveillance cam

Figure 5.2.2.3 Screenshot command to end surveillance cam

5.2.3 Telegram Control

Figure 5.2.3.1 Screenshot command to capture photo

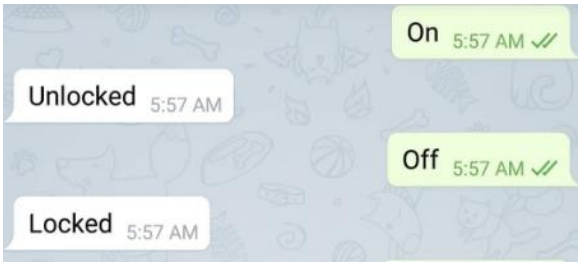


Figure 5.2.3.2 Screenshot command to remote access control the locker

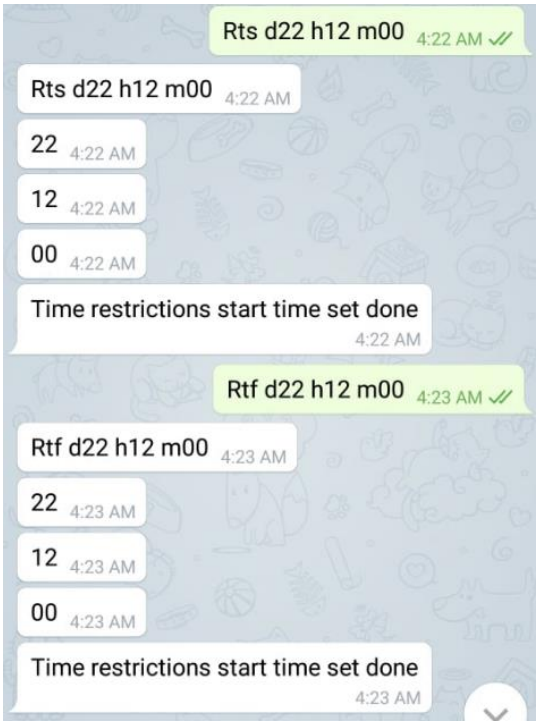


Figure 5.2.3.3 Screenshot command to set time restrictions

Figure 5.2.3.4 Screenshot command to check door access records

5.2.4 LCD Display

Figure 5.2.4.1 LCD display when system is ready

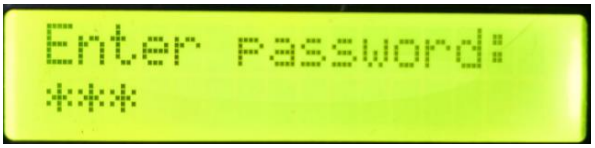


Figure 5.2.4.2 LCD display when password entered

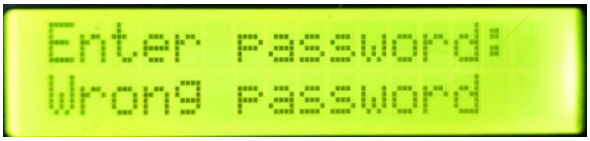


Figure 5.2.4.3 LCD display when wrong password

Figure 5.2.4.4 LCD display when locker unlocked

Figure 5.2.4.5 LCD display when meet time restriction period

5.2.5 Blynk

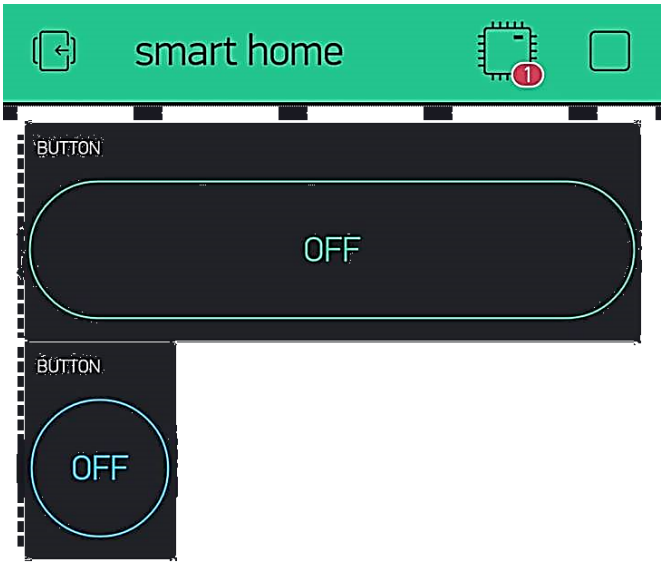


Figure 5.2.5.1 Blynk user interface

5.3 Implementation

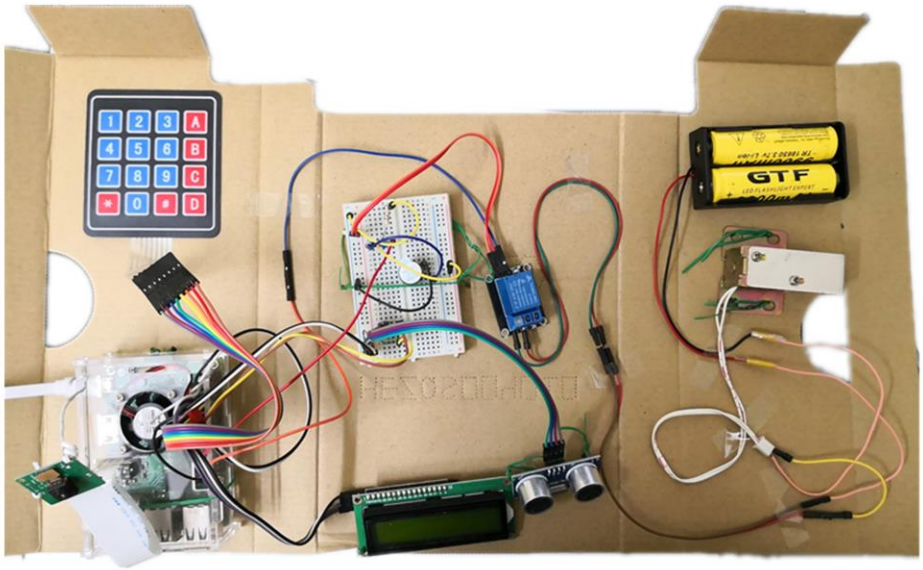


Figure 5.3.1 Prototype 1

Figure 5.3.2 Final product

CHAPTER 6: CONCLUSION

6.1 Project Review, Discussions and Conclusions

This project is used to implement the face recognition and the house security. Face recognition nowadays has been widely used on many areas especially on security. The house security can be improved with the implement of this product. It is design with low cost and efficient material. The improvement of technology had made the internet of thing no longer an expensive stuff and it can be modified and customized depend on our needs.

The system security on market with low budget have the problem like the face detection only will start to function when someone press the doorbell. Surveillance system will destroy by theft and without notify owner when stranger come in. Face recognition can be by pass with owner face photo. Unable to unlock the door without key.

The motivation is to develop a low cost security system to solve the issues exist on market product. A self-develop anti-theft security system using face recognition project have been started for solve the above issues. It can help to solve the issues like it can trigger the alarm and capture an image send to owner when a stranger is detected in front of door. The theft will get alert so he got no time to start breaking the door and destroy the security system. For overcome the weakness of face recognition that can be by pass with owner face image so the door will only unlock with the correct password input with a keypad. Besides, it also has features like remote access control, reverse password, door access records and time restriction control.

Appropriate hardware and software is used as they are the crucial component to make this project. The hardware used is Raspberry Pi 3 model B which is main hardware, connected to several sensors and also connected to the Telegram mobile application. This project uses Python, Shell script, JavaScript programming language. Some Linux command prompt is also used while installing the drivers or library for the Python coding.

Moreover, this project contents consists several features, draft ideas, researches information and methodology about the Anti-theft security system using face recognition.

There are many difficulties found such as understanding the project features, studying the limited case study related to this project and thinking about the basic structure and design of the system which make whole progress did not run smoothly. However, I still managed to solve it finally. During the process of doing the proposal, the hardware part and software part is categorized separately and then do research about the necessary component needed and its function involved in this project.

Finally, with the proper guidance and advice from lecturer, supervisor and also the researches on previous products that had been studied, this project has been successfully completed in more efficiency way.

6.2 Future Work

For the future work, there are some improvement can be done. The camera module can upgrade to infrared camera. It has better quality and not affected by light condition because it use the technology of infrared to capture photo. It had been widely used now on flagship mobile phone on market for higher security but it cost more compare with normal camera. The Raspberry Pi can upgrade to latest version on market which is more powerful and more features. So the system can process higher resolution image smoothly without stress on it. Face recognition algorithm still can be improve because the result is still have not achieve perfect recognize result.

BIBLIOGRAPHY

Erdem, H., Uner and A. (2009) ‘A multi-channel remote controller for home and office appliances’, *IEEE Xplore Digital Library*, vol. 55, no. 4, pp. 2184-2189.

Yuksekkaya, B., Kayalar, A.A., Tosun, M.B., Ozcan, M.K., Alkar and A.Z., (2006) ‘A GSM, internet and speech controlled wireless interactive home automation system’, *IEEE Xplore Digital Library,* vol. 52, no. 3, pp. 837-843.

Vernon, S., Joshi and S.S., (2011) ‘Brain—Muscle—Computer Interface: MobilePhone Prototype Development and Testing’, *IEEE Xplore Digital Library,* vol. 15, no. 4, pp. 531-538.

Faundez-Zanuy and M. (2005) ‘Privacy issues on biometric systems’ *, IEEE Xplore Digital Library,* vol. 20, no. 2, pp. 13-15.

P. B. Saurabh and D.S. Chaudhari (2012) ‘Principal Component Analysis for Face Recognition’, International Journal of Engineering and Advanced Technology, *, IEEE Xplore Digital Library,* vol. 1, pp. 91-94.

Sahani, M., Nanda, C., Sahu, A.K.,Pattnaik, B. (2015) 'Web-based online embedded door access control and home security system based on face recognition', IEEE Xplore Digital Library[online], 19-20 March 2015, pp.1-6. Available from: http://ieeexplore.ieee.org.libezp.utar.edu.my/xpls/icp.jsp?arnumber=7159473&tag=1[Accessed 1 August 2017]

Thabet, A. and Amor, N. (2015) ‘Enhanced smart doorbell system based on face recognition’, IEEE Xplore Digital Library[online], 21-23 Dec. 2015, pp.373-377. Available from: http://ieeexplore.ieee.org.libezp.utar.edu.my/document/7505106/[Accessed 1 August 2017]

Neelam Majgaonkar, Ruhina Hodekar & Priyanka Bandagale (2016) Automatic Door Locking System Vol.4 Issue-1, 2016 p.495 – 499.

APPENDICES: POSTER

UNIVERSITI TUNKU ABDUL RAHMAN

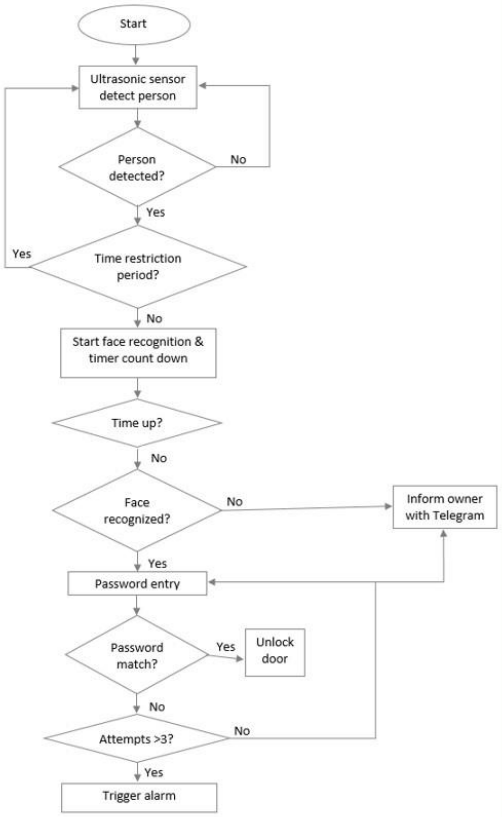
Anti-theft Security System Using Face Recognition

By Chong Guang Yu
Supervisor: Mr. Leong Chun Farn

INTRODUCTION

This project is for provide a better solution with a new design of embedded system and face recognition technology to protect the household with higher security level. The owner will get notification on time when stranger present in front of door. The stranger also will get warn when trying to enter the house.

METHODOLOGY



OBJECTIVE

The objectives of this project are:

- ✓ To design and develop an anti-theft security system which is portable, high efficiency, low cost and easy to operate.
- ✓ To detect the present of person when he or she reach in front the door and the face recognizer will recognize owner face.
- ✓ To notify owner and start surveillance system in real time when stranger detected.
- ✓ To record the door access about time, user and locker status.
- ✓ To have door access time restrictions

IMPLEMENTATION & TESTING

Notification

Remote access control UI

Surveillance System

Final Product

APPENDICES: Source Code

main.py

```
#!/usr/bin/env python

import RPi.GPIO as GPIO
import time
import os
import random
import string
import smbus
import subprocess
import sys, os
import range_sensor
import select
import cv2
import datetime

#set counter for password attempt
counter = 0

###################### LCD DISPLAY ################################
# Define some device parameters
I2C_ADDR  = 0x27 # I2C device address, if any error, change this address to 0x3f
LCD_WIDTH = 16   # Maximum characters per line

# Define some device constants
LCD_CHR = 1 # Mode - Sending data
LCD_CMD = 0 # Mode - Sending command

LCD_LINE_1 = 0x80 # LCD RAM address for the 1st line
LCD_LINE_2 = 0xC0 # LCD RAM address for the 2nd line
LCD_LINE_3 = 0x94 # LCD RAM address for the 3rd line
LCD_LINE_4 = 0xD4 # LCD RAM address for the 4th line

LCD_BACKLIGHT  = 0x08  # On
#LCD_BACKLIGHT = 0x00  # Off

ENABLE = 0b00000100 # Enable bit

# Timing constants
E_PULSE = 0.0005
E_DELAY = 0.0005

#Open I2C interface
#bus = smbus.SMBus(0)  # Rev 1 Pi uses 0
```

```
bus = smbus.SMBus(1) # Rev 2 Pi uses 1

def lcd_init():
  # Initialise display
  lcd_byte(0x33,LCD_CMD) # 110011 Initialise
  lcd_byte(0x32,LCD_CMD) # 110010 Initialise
  lcd_byte(0x06,LCD_CMD) # 000110 Cursor move direction
  lcd_byte(0x0C,LCD_CMD) # 001100 Display On,Cursor Off, Blink Off
  lcd_byte(0x28,LCD_CMD) # 101000 Data length, number of lines, font size
  lcd_byte(0x01,LCD_CMD) # 000001 Clear display
  time.sleep(E_DELAY)

def lcd_byte(bits, mode):
  # Send byte to data pins
  # bits = the data
  # mode = 1 for data
  #        0 for command

  bits_high = mode | (bits & 0xF0) | LCD_BACKLIGHT
  bits_low = mode | ((bits<<4) & 0xF0) | LCD_BACKLIGHT

  # High bits
  bus.write_byte(I2C_ADDR, bits_high)
  lcd_toggle_enable(bits_high)

  # Low bits
  bus.write_byte(I2C_ADDR, bits_low)
  lcd_toggle_enable(bits_low)

def lcd_toggle_enable(bits):
  # Toggle enable
  time.sleep(E_DELAY)
  bus.write_byte(I2C_ADDR, (bits | ENABLE))
  time.sleep(E_PULSE)
  bus.write_byte(I2C_ADDR,(bits & ~ENABLE))
  time.sleep(E_DELAY)

def lcd_string(message,line):
  # Send string to display

  message = message.ljust(LCD_WIDTH," ")

  lcd_byte(line, LCD_CMD)

  for i in range(LCD_WIDTH):
    lcd_byte(ord(message[i]),LCD_CHR)
```

```
################## Ultrasonic_sensor#######################

def initial():
   import range_sensor

   #os.system("python telegrambot.py &")
   os.system("pkill python3 ")

   while range_sensor.distance() >10:

     facerecog = open("facerecog.txt", "r")
     frc = facerecog.read()
     print (frc)
     facerecog.close()

     print (range_sensor.distance())

     time.sleep(1)

   else:

     #os.system("pkill ffmpeg ")
     #os.system("pkill python3 ")

     global start
     start = time.time()
     print("Start 1st : " +str(start))

     return startpass()

def clear_if_time_pass():

   k=0  #clear password
   lcd_string(" Locked Please ",LCD_LINE_1)
   lcd_string("enter password.... ",LCD_LINE_2)
   return initial()

def startpass():
   global counter

   ledbz = 37
   locker = 32

   GPIO.setmode(GPIO.BOARD)
   GPIO.setwarnings(False)
   GPIO.setup(locker, GPIO.OUT)
   GPIO.setup(ledbz, GPIO.OUT)
```

```
k=0
    lcd_init()

    MATRIX = [ [1,2,3,'A'],
               [4,5,6,'B'],
               [7,8,9,'C'],
               ['*',0,'#','D'] ]


    ROW = [19,11,13,15]
    COL = [12,16,18,22]

    lcd_string(" LOCKED Please ",LCD_LINE_1)
    lcd_string("enter password.... ",LCD_LINE_2)


    now = datetime.datetime.now()
    print (now)

    f = open("TimeRstrt/stD", "r")
    sdy= f.read()
    f.close()
    f = open("TimeRstrt/stH", "r")
    shr= f.read()
    f.close()
    f = open("TimeRstrt/stM", "r")
    smt= f.read()
    f.close()
    rstime = now.replace(day=int(sdy),hour=int(shr), minute=int(smt))
    print ("rstime")
    print (rstime)

    f = open("TimeRstrt/ftD", "r")
    fdy= f.read()
    f.close()
    f = open("TimeRstrt/ftH", "r")
    fhr= f.read()
    f.close()
    f = open("TimeRstrt/ftM", "r")
    fmt= f.read()
    f.close()
    rftime = now.replace(day=int(fdy),hour=int(fhr), minute=int(fmt))
    print ("rftime")
    print (rftime)
```

```
#time restrictions function
   if (rstime <=now)and(now <= rftime ):
      print ("lockkkkk")
      lcd_string(" Sorry, LOCKED ",LCD_LINE_1)
      lcd_string("Time Restriction",LCD_LINE_2)
      GPIO.output(ledbz,1)
      time.sleep(1)
      GPIO.output(ledbz,0)
      os.system("sh ./tg.sh redone Someone_come_at_time_restrictions! ")
      os.system("raspistill -q 5 -o img1.jpg -t 200 ")
      os.system("sh ./tg_photo.sh redone /home/pi/img1.jpg")
      os.system("python3 face_recognition.py &")
      time.sleep(20)
      return clear_if_time_pass()

   else:
      print ("unlockkkkk")
      os.system("python3 face_recognition.py &")



   file = open("pass.txt", "r")
   pw= file.read()
   #print("pw = " +str(pw))
   file.close()

   file = open("rvpass.txt", "w")
   file.write(pw[::-1])
   file.close()

   file = open("rvpass.txt", "r")
   rvpw = file.read()
   #print("rvpw = " +str(rvpw))
   file.close()

   facerecog = open("facerecog.txt", "r")
   frc = facerecog.read()
   print (frc)
   facerecog.close()
   print("checkpoint.....")

################### Keypad ##############################

   for j in range(4):
      GPIO.setup(COL[j], GPIO.OUT)
      GPIO.output(COL[j], 1)
```

```
for i in range(4):
    GPIO.setup(ROW[i], GPIO.IN, pull_up_down = GPIO.PUD_UP)

  try:
    while(True):
      for j in range(4):
        GPIO.output(COL[j],0)
        facerecog = open("facerecog.txt", "r")
        frc = facerecog.read()
        #print ("frc = ")
        #print (frc)
        facerecog.close()
        #global end
        end = time.time()
        #if (end == end + 10000000000):

        result = end - start
        #print("Result :: " +str(result))
        if (result > 60):
          print("JUMP testing.....")
          GPIO.output(locker,0)  #locker autolock when time up
          return clear_if_time_pass()
        #print(MATRIX[i][j])
        #print("MATRIX iiiii")
        for i in range(4):


          if GPIO.input(ROW[i]) == 0:
            keypress = (MATRIX[i][j])
            #k = k+1

            y = str(keypress)
            global start
            start = time.time()


            if y=="C": #lock the door
              GPIO.output(locker,0)
              GPIO.output(ledbz,1)
              time.sleep(0.4)
              GPIO.output(ledbz,0)
              lcd_string(" Locked Please ",LCD_LINE_1)
              lcd_string("enter password.... ",LCD_LINE_2)
              return initial()
```

```
                if y=="D":
                  print ("delete and Retry")
                  k=0   ##    enter password..
                  lcd_string("Clr, Pls Retry ",LCD_LINE_2)

                else:
                  k = k+1

                print(keypress)
                print (range_sensor.distance())
                print (y)
                print("COUNTER = " +str(counter))
                print ("k value "+str(k))
                lcd_string("Enter password: ",LCD_LINE_1)
                GPIO.output(ledbz,1)
                #lcd_string(y,LCD_LINE_2)
                time.sleep(0.2)
                GPIO.output(ledbz,0)

                if k==1:
                  z=y
                  lcd_string("*",LCD_LINE_2)
                if k==2:
                  k2="%s%s" % (z,y)
                  print (k2)
                  lcd_string("**",LCD_LINE_2)
                if k==3:
                  k3="%s%s" % (k2,y)
                  print (k3)
                  lcd_string("***",LCD_LINE_2)
                if k==4:
                  k4="%s%s" % (k3,y)
                  print (k4)
                  lcd_string("****",LCD_LINE_2)


                  if len(k4)==4:
                    os.system("pkill python3 ")
                    print("4keys entered")

                    if (k4 == pw)and(frc == "%^&*") :
                    #if k4 == "1234":
                      print('UNLOCKED')
                      print("frc = " +frc )
                      GPIO.output(locker,1)
                      GPIO.output(ledbz,1)
```

```
                    time.sleep(0.5)
                    GPIO.output(ledbz,0)
                    lcd_string("Door UNLOCKED >> ",LCD_LINE_1)
                        ##    enter password..
                    lcd_string("press C to Lock!",LCD_LINE_2)

                  elif (k4 == rvpw) :
                  #if k4 == "1234":
                    os.system("sh ./tg.sh redone Reverse_password_entered!!! ")
                    time.sleep(1)
                    os.system("raspistill -q 5 -o img1.jpg -t 200 ")
                    os.system("sh ./tg_photo.sh redone /home/pi/img1.jpg")
                    print('UNLOCKED')
                    GPIO.output(locker,1)
                    GPIO.output(ledbz,1)

                    time.sleep(0.5)
                    GPIO.output(ledbz,0)
                    lcd_string("Door UNLOCKED >> ",LCD_LINE_1)
                        ##    enter password..
                    lcd_string("press C to Lock!",LCD_LINE_2)

                  else:
                    counter +=1
                    print("send photo to phone>>>>>>>>>>.....")
                    print("frc = " +frc )
                    #print (frc)

                    #subprocess.call(['./notify.sh'])
                    if (k4 == pw):
                      os.system("sh ./tg.sh redone Correct_password_But_face_not_found ")

                    elif (frc == "%^&*"):
                      os.system("sh ./tg.sh redone Wrong_password_But_face_verification_pass
")

                    else:
                      os.system("sh ./tg.sh redone Warning_Stranger_Wrong_password!!! ")


                    #An error has occured
                    print('FAIL ATTEMPT')
                    time.sleep(1)
                    os.system("raspistill -q 5 -o img1.jpg -t 200 ")
                    os.system("sh ./tg_photo.sh redone /home/pi/img1.jpg")
                    lcd_string("Wrong password",LCD_LINE_2)
                    GPIO.output(ledbz,1)
```

```
                    time.sleep(1)
                    GPIO.output(ledbz,0)
                    time.sleep(2)
                    if (counter>=3):
                      print('Reached max attempts wait 10sec')
                      lcd_string("Reached max",LCD_LINE_1)
                      lcd_string("attempts! WAITing",LCD_LINE_2)

                      GPIO.output(ledbz,1)
                      time.sleep(5)
                      GPIO.output(ledbz,0)
                      counter = 0
                      lcd_string(" Locked Please ",LCD_LINE_1)
                      lcd_string("enter password.... ",LCD_LINE_2)
                    return initial()
                    #return start()


                while(GPIO.input(ROW[i]) == 0):
                  pass

            GPIO.output(COL[j],1)
    except KeyboardInterupt:
        print("stopped by User")
        GPIO.cleanup()



for i in range(1):
    initial()

    #ultrasonic_sensor()
    #start()
```

lockstcheck.py

```
import RPi.GPIO as GPIO
import datetime
import time

GPIO.setmode(GPIO.BOARD)
locker = 32
GPIO.setup(locker,GPIO.OUT)
Lst=0
Unst=0

if __name__ == '__main__':
    try:
        print ("In Progress")
        while True:
            time.sleep(1)
            date = datetime.datetime.now()
            dt=str(date)
            #print(dt +" locked")
            if (GPIO.input(locker) == False):
                #print("lock")
                Lst=Lst+1
                Unst=0
                #print(Lst)
                if(Lst==1):
                    print("record lock")
                    f = open("DaccessRecord", "a")
                    f.write(dt +"  locked" +"\n")
                    f.close()

            elif (GPIO.input(locker) == True):
                #print("unlock")
                Lst=0
                Unst=Unst+1
                #print(Unst)
                if(Unst==1):
                    print("record unlock")
                    f = open("DaccessRecord", "a")
                    f.write(dt +" UNLOCKED" +"\n")
                    f.close()


    except KeyboardInterrupt:
        print("stopped by User")
        GPIO.cleanup()
```

telegrambot.py

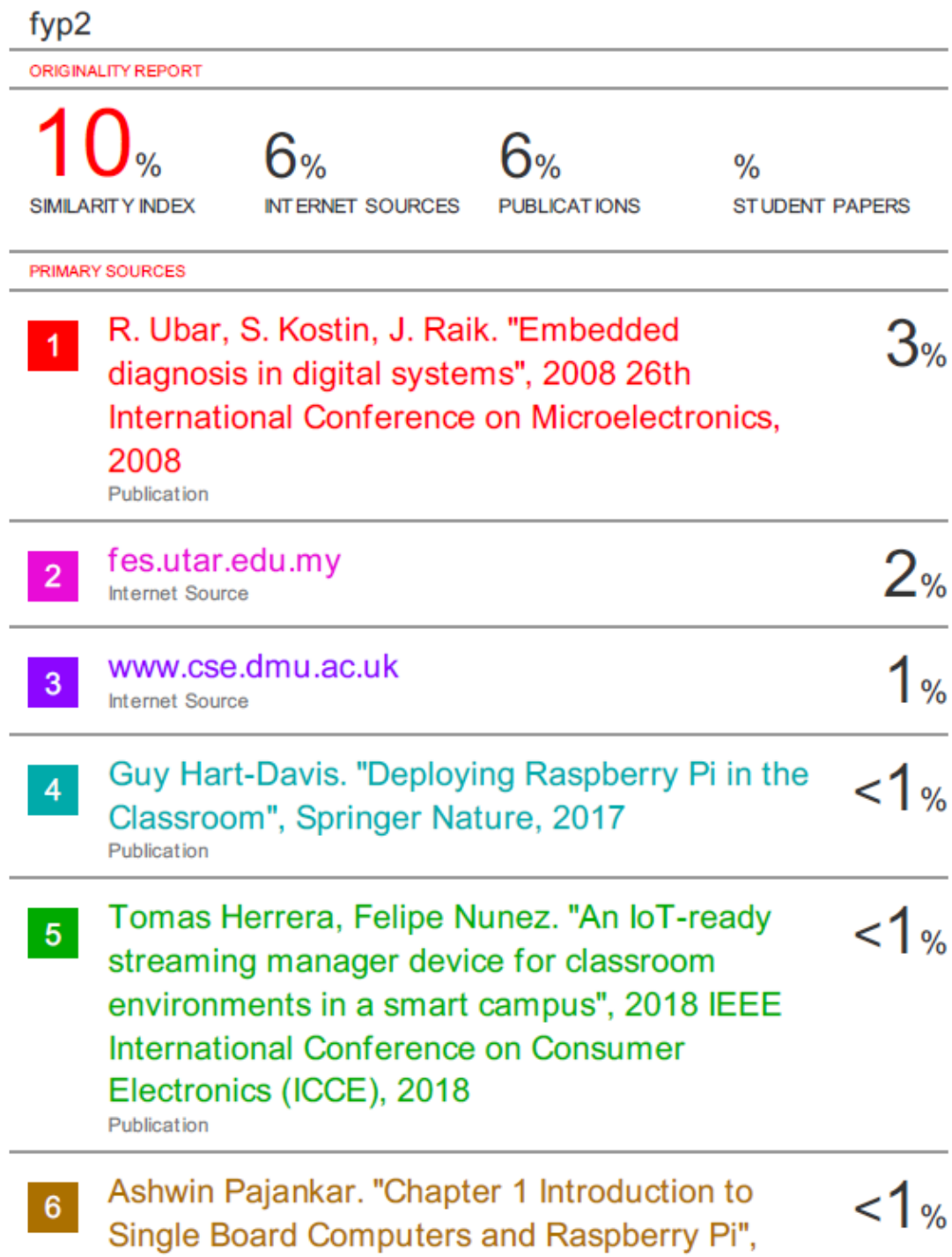
```
 import sys
import time
import telepot
import os
import RPi.GPIO as GPIO
import datetime
import re


#Control Locker
def on(pin):
        GPIO.output(pin,GPIO.HIGH)
        return
def off(pin):
        GPIO.output(pin,GPIO.LOW)
        return
# to use Raspberry Pi board pin numbers
GPIO.setmode(GPIO.BOARD)
# set up GPIO output channel
GPIO.setup(32, GPIO.OUT)

def replace_line(file_name, line_num, text):
    lines = open(file_name, 'r').readlines()
    lines[line_num] = text
    out = open(file_name, 'w')
    out.writelines(lines)
    out.close()

def handle(msg):
    chat_id = msg['chat']['id']
    command = msg['text']

    print('Got command: %s' % command)

    if command == 'On':
      bot.sendMessage(chat_id, text = "Unlocked")
      bot.sendMessage(chat_id, on(32))

    elif command =='Off':
      bot.sendMessage(chat_id, text = "Locked")
      bot.sendMessage(chat_id, off(32))


    elif command =='C':
      date = datetime.datetime.now()
```

```
 dti = date.isoformat()
      print( "%s" %date)
      print( "%s" %dti)

      dt = str(dti)
      os.system("raspistill -q 5 -o " +dt +".jpg -t 200")
      os.system("sh ./tg_photo.sh Pi3 /home/pi/"+dt +".jpg")

      bot.sendMessage(chat_id, text = "photo captured "+dt )

# start surveillance system

   elif command =='S':
      #os.system("cd && python server.py ")
      bot.sendMessage(chat_id, text = "Click the link below to start Surveillance Camera " )
      #bot.sendMessage(chat_id, text = "http://192.168.43.183:8000/index.html")
      bot.sendMessage(chat_id, text = "http://192.168.0.184:8000/index.html")

      os.system("pkill python3 ")
      os.system("pkill ffmpeg ")
      os.system("cd && cd pistreaming && python3 server.py &")

# stop surveillance system
   elif command =='Stop':
      bot.sendMessage(chat_id, text = "Streaming Stopped")
      #os.system("cd && python range_sensor.py ")
      os.system("pkill ffmpeg ")
      os.system("pkill python3 ")

   elif command =='Shutdown': #shutdown
      bot.sendMessage(chat_id, text = "System Turn Off " )
      os.system("shutdown -h now  ")
      bot.sendMessage(chat_id, text = "Loading " )


   elif command =='Rbt': #restart
      bot.sendMessage(chat_id, text = "Loading " )
      bot.sendMessage(chat_id, text = "System Rebooting " )
      os.system("reboot  ")
      bot.sendMessage(chat_id, text = "Loading " )

#check door access record
   elif command == 'Cr':
      bot.sendMessage(chat_id, text = "Below is your door access records " )
      file = open("DaccessRecord", "r")
      Crd = file.read()
      print("Crd = " +str(Crd))
```

```
        file.close()
        bot.sendMessage(chat_id, text = Crd )

#set Time restrictions start time
    elif command.startswith('Rts '):
        bot.sendMessage(chat_id, text = command )
        dy = re.search('(?<=d)\w+', command)
        bot.sendMessage(chat_id, text = dy.group(0) )
        f = open("TimeRstrt/stD", "w")
        f.write(dy.group(0))
        f.close()

        hr = re.search('(?<=h)\w+', command)
        bot.sendMessage(chat_id, text = hr.group(0) )
        f = open("TimeRstrt/stH", "w")
        f.write(hr.group(0))
        f.close()

        mt = re.search('(?<=m)\w+', command)
        bot.sendMessage(chat_id, text = mt.group(0) )
        f = open("TimeRstrt/stM", "w")
        f.write(mt.group(0))
        f.close()
        bot.sendMessage(chat_id, text = "Time restrictions start time set done" )

#set Time restrictions finsih time
    elif command.startswith('Rtf '):
        bot.sendMessage(chat_id, text = command )

        dy = re.search('(?<=d)\w+', command)
        bot.sendMessage(chat_id, text = dy.group(0) )
        f = open("TimeRstrt/ftD", "w")
        f.write(dy.group(0))
        f.close()

        hr = re.search('(?<=h)\w+', command)
        bot.sendMessage(chat_id, text = hr.group(0) )
        f = open("TimeRstrt/ftH", "w")
        f.write(hr.group(0))
        f.close()

        mt = re.search('(?<=m)\w+', command)
        bot.sendMessage(chat_id, text = mt.group(0) )
        f = open("TimeRstrt/ftM", "w")
        f.write(mt.group(0))
        f.close()
```

```
      bot.sendMessage(chat_id, text = "Time restrictions start time set done" )

   else:
      bot.sendMessage(chat_id, text = "Please key in correct input cc" )

bot = telepot.Bot('613252887:AAFk2OvH9sE05po1Pz7QWFKUgH22dU4aQJ0')
bot.message_loop(handle)
print('I am listening...')

while 1:
   try:
      time.sleep(10)

   except KeyboardInterrupt:
      print('\n Program interrupted')
      GPIO.cleanup()
      exit()

   except:
      print('Other error or exception occured!')
      GPIO.cleanup()
```

Plagiarism Check Result



fyp2

ORIGINALITY REPORT

10%	6%	6%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	R. Ubar, S. Kostin, J. Raik. "Embedded diagnosis in digital systems", 2008 26th International Conference on Microelectronics, 2008 Publication	3%
2	fes.utar.edu.my Internet Source	2%
3	www.cse.dmu.ac.uk Internet Source	1%
4	Guy Hart-Davis. "Deploying Raspberry Pi in the Classroom", Springer Nature, 2017 Publication	<1%
5	Tomas Herrera, Felipe Nunez. "An IoT-ready streaming manager device for classroom environments in a smart campus", 2018 IEEE International Conference on Consumer Electronics (ICCE), 2018 Publication	<1%
6	Ashwin Pajankar. "Chapter 1 Introduction to Single Board Computers and Raspberry Pi",	<1%

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	CHONG GUANG YU
ID Number(s)	1305524
Programme / Course	Computer Engineering (CT)
Title of Final Year Project	Anti-theft Security System Using Face Recognition

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index:_____ % **Similarity by source** Internet Sources: _____% Publications: _____ % Student Papers: _____ %	
Number of individual sources listed of more than 3% similarity: _____	
Parameters of originality required and limits approved by UTAR are as Follows: **(i) Overall similarity index is 20% and below, and** **(ii) Matching of individual sources listed must be less than 3% each, and** **(iii) Matching texts in continuous block must not exceed 8 words** ***Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.***	

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

_____ Signature of Supervisor

Name: _____

Date: _____

_____ Signature of Co-Supervisor

Name: _____

Date: _____

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (PERAK CAMPUS)

CHECKLIST FOR FYP1 THESIS SUBMISSION

Student Id	1305524
Student Name	CHONG GUANG YU
Supervisor Name	Mr Leong Chun Farn

TICK (√)	DOCUMENT ITEMS Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
	Title Page
	Signed form of the Declaration of Originality
	Abstract
	Table of Contents
	List of Figures (if applicable)
	List of Tables (if applicable)
	List of Symbols (if applicable)
	List of Abbreviations (if applicable)
	Chapters / Content
	Bibliography (or References)
	All references in bibliography are cited in the thesis, especially in the chapter of literature review
	Appendices (if applicable)
	Poster
	Signed Turnitin Report (Plagiarism Check Result – Form Number: FM-IAD-005)

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report. _____ (Signature of Student) Date:	Supervisor verification. Report with incorrect format can get 5 mark (1 grade) reduction. _____ (Signature of Supervisor) Date: