DEVELOPMENT OF INTELLIGENT HOME SECURITY SYSTEM

A Thesis Submitted to the Department of Electrical and Electronic Engineering of BRAC University

By

Sayedul Islam ID-10221007
Zakir Hossain ID-11321003
Sourav Dutta ID-10321006
Shafiul Hossain ID-09210034

In partial fulfillment of the requirements for the degree of Bachelor of Science in Electrical and Electronic Engineering
SPRING 2017
BRAC University, Dhaka

Supervised by

Professor. Dr Md Adnan Kiber
Department of Electrical and Electronic Engineering, Dhaka University

CERTIFICATE

This is to certify that this thesis project work entitled” **DEVELOPMENT OF INTELLIGENT HOME SECURITY SYSTEM WITH RASPBERRY PI**” submitted by Sayedul Islam, Sourav Dutta, Zakir Hossain and Shafiul Hossain legitimate thesis work carried out under my supervision and guidance and fulfilling the nature and standard required for the partial fulfillment of the degree of Bachelor of Science in Electrical and Electrical Electronic Engineering .The work includes in this thesis has not been submitted elsewhere for the degree.

Date:

Signature of Supervisor

Signature of students

....

Professor. Dr Md Adnan Kiber

Sayedul Islam

....

Zakir Hossain

…………………………………………….

Sourav Dutta

....

Shafiul Hossain

……………………………………….

ACKNOWLEDGEMENT

First of all, we are extremely thankful to our Creator who has made all these come into being and allowed us to accomplish this work.

Secondly, we are greatly thankful to our supervisor Professor.Md Adnan Kiber without his guidance and support this project wouldn't have been possible.

Last but not the least we have to thank Brac University for having faithful in us and allow us to conduct our project.

ABSTRACT

This research work deals with the design and implementation of smart surveillance monitoring hardware system using Raspberry pi and PIR sensor.It increases the security of home using image processing and digital signal processing.This causes a major problem in our society as theft and burglars enter our home without noticeable.Therefore our aim is to construct a smart home security system to overcome these type of problem.This device consists of a raspberry pi which operates and control motion detector and stream live video and records it for future playback. Whenever a motion is detected a pi camera capture the picture and send it to owners smartphone through SMS and later saved on the server.

Table of Contents

Acknowledgement ………………………………………………………………………………………….. 3

Abstract ………………………………………………………………………………………………… 3

Chapter 1 : Introduction

1.1 Introduction ……………………………………………………………………………………… 8

1.2 Purpose ……………………………………………………………………………………………8

1.3 The Impact of security and safety system: …………………………………………………..9

1.4 Motivation ……………………………………………………………………………………….9

1.5 Overview of the content ………………………………………………………………………..10

Chapter 2: Classification of the Security System

2.1 Classification of the Security System ……………………………………………………..11

2.1.2 GE personal security Alarm kit: ………………………………………………………13

2.1.3 Monitored security system: ……………………………………………………………….14

2.1.4 Unmonitored security system: ………………………………………………………….15

2.1.5 Electronic security system:………………………………………………………………..16

2.1.6 Fingerprint Authorization: ……………………………………………………………….17

2.1.7 Optical scanners: ………………………………………………………………………………..17

2.1.8 Ultrasonic scanners: ……………………………………………………………………………17

2.1.9 Capacitive scanners: ……………………………………………………………………………..18

2.2.0 Password Authorization: ………………………………………………………………………18

2.2.1 RFID Authorization: ……………………………………………………………………………… 19

2.2.2 Face Recognization: ……………………………………………………………………………… 20

2.2.3 Face detection in pictures: ………………………………………………………………………20

2.2.4 Real-time face detection: …………………………………………………………………………21

2.2.5 Fire-Detection Systems: …………………………………………………………………………..21

2.2.6 Summary: …………………………………………………………………………………………………..22

Chapter 3: Hardware and component

3.1 Introduction: ……………………………………………………………………………………………..23

3.2 Component used in the hardware: ……………………………………………………………………23

3.3.1 Arduino Mega 2560 ……………………………………………………………………………………24

3.3.2 Fingerprint Scanner TTL (GT-511C3) …………………………………………………………26

3.3.3 125Khz RFID Module …………………………………………………………………………………..27

3.3.4 RFID Tag ……………………………………………………………………………………………………..28

3.3.5 16*2 lCD display …………………………………………………………………………………………29

3.3.6 4*4 Key Capacitive Keypad ………………………………………………………………………30

3.3.7 Simcom Sim900a GSM+GPRS module ………………………………………………………31

3.3.8 Electric Solenoid Mini Door Lock DC 12V ………………………………………………32

3.3.9 Buzzer ……………………………………………………………………………………………33

3.3.10 Raspberry Pi 3 Model B ……………………………………………………………………33

3.3.11 Arduino UNO ……………………………………………………………………………………34

3.3.12 HC-SR501 PIR SENSOR MODULE …………………………………………………………35

3.3.13 Adaptor …………………………………………………………………………………………36

3.3.14 Vero board ……………………………………………………………………………………36

3.3.15 Micro servo ……………………………………………………………………………………37

3.3.16 Raspberry pi camera …………………………………………………………………………37

3.3.17 Flame Sensor …………………………………………………………………………………38

3.3.18 LM2596 dc-dc adjustable step-down module ………………………………………39

3.3.19 12V Door lock driver …………………………………………………………………………40

3.3.20 LED Lights-5mm ………………………………………………………………………………41

3.3.21 HDMI PORT ……………………………………………………………………………………… 41

3.3.22 Smartphone ……………………………………………………………………………………42

Chapter 4 Structural System Design Flowchart and Circuit

4.0 Introduction: ………………………………………………………………………………………43

4.2 Overview and Block Diagram of the System: ………………………………………………… 43

4.1 Pin Configuration and Symmetric Diagram: …………………………………………………45

4.2 Digital Door Lock System: ……………………………………………………………………… 48

4.3 Face Detection System: ……………………………………………………………………………52

4.4 Fire Detection and Alarm System: ………………………………………………………………52

Chapter 5: System Software Design and Configuration an Implementation

5.0 Introduction ……………………………………………………………………………………………… 54

5.1 Software algorithm: …………………………………………………………………………………………. 54

5.2 Arduino IDE: **…………………………………………………………………………………………………. 56**

5.3 Codes and Libraries: ………………………………………………………………………………………. 57

5.4 Results and Discussion…………………………………………………………………………………..82

Chapter 6: Conclusion and future implementation

6.1 Conclusion ……………………………………………………………………………………..……………….83

6.1.1 Limitation …………………………………………………………………………………………………..83

6.2 future implementation ………………………………………………………………………………… 83

References ………………………………………………………………………………………………… 85

Chapter 1

Introduction

1.1 Introduction

During the last few decades in Bangladesh, the rise in crime rates has increased in great numbers. In Bangladesh small crimes like burglary, cutting window and entering are mostly likely ignored where as high profile crimes are highly prioritized by the law enforcement agencies.Although it's quite true that not every citizen of Bangladesh cannot afford the trained private security guard for their homes and small business.

Furthermore foreigner travel in Bangladesh for their business purpose and other issues with a poor security and sometimes they are the prey of criminals and suffer in many ways. Nowadays in diplomatic areas more police forces are deployed by the Bangladesh government for the security of the foreigners.A highly technically advanced security system in the residential and business area can be secured more efficiently and aid Bangladesh government to secure the foreigner effectively. However to ensure the security of the civilian of Bangladesh we have the plan to design and develop advance smart security system by the support of technical knowledge.

1.2 Purpose

The purpose of this paper is to provide some solutions to the deficit in our household's, offices' and banks' security through the use of technology. Although the security risks cannot be fully eliminated but the safety of homes, offices, banks and industries can be dramatically improved by the introduction of simple but smart advanced system.

The Impact of security and safety system:

In Bangladesh crimes like burglaries, robberies and residential break-ins occurs over five to six times a day.The previous record of Bangladesh crime reports shows that most of the crimes occurs during the day and secondly due to backdated security system in the whole country.Most of the time the person who is in charge or responsible for the safe keeping are involved with the crime so it’s hard decision for anyone to judge which is more reliable and technological . Recent data of Bangladesh crime report stated that 74.70% is the level of crime. In past 3 years, crime increases 72.94% and home was broken and things stolen is 69.05%. This is really regrettable that safety walking along during night is 25.69% and the only moderate is safety walking along the road during daylight is 47.32%. Homes without well-protected security systems are 2.7 times targeted by burglars. Recently Bangladesh government install surveillance camera several diplomatic areas and important sector of the Dhaka city which may help significantly reduce of crime rate compared to past few year.

There are approximately more than 300,000 private security guards and private security agencies whose operation mostly in Dhaka and the other big cities. These sector is providing around Tk 300 crore to the national GDP which is high and also provides employment of many people. We are providing a technological home security system which decreases the crime rate significantly and also offer a new horizon of employment for more people by establishing a new career of the production area in this field as well as increase the GDP of our country significantly.

1.3 Motivation

The Holey Artisan bakery café attack in Dhaka that took place on the night of 1st July 2016, around 21:20 local on the other hand Kishoreganj Sonali Bank branch robbery which again motivates us to help and cut down these crime numbers.In order to do so, we have constructed a smart home security system which minimizes some percentage of relevant crimes from our society.With great motivated from the above statement, we research and able to find the solution with the advanced security and safety system.To live a peaceful life a well-protected home is necessary.Homes without well-protected security systems are mostly targeted by burglars repeatedly so protect home and family from intruder it is very important to have a decent security system.In future record and from video footage there could be a chance to identify the burglar during a burglary.

1.4 Overview of the content

We come up with a smart home security and safety system using image processing and digital signal processing for the sake of security and safety issues.We think about a three layer of the security system where we include fingerprint sensor,password reader,RFID examining,fire alarm and video streaming are highly highlighted in this project. The overview of our project is when the system is turn ON and the person put his/her finger to the fingerprint scanner it generates an image of the ridges and valleys of the finger and detects whether the person is authorized or unauthorized. If ridges and valleys match then the system permit the user to access next process. In the next level, the system will ask for the password, so the user need to type his/her password and the system will check whether the password given is correct or wrong and go with the previous way. Correctly accessed by the user will face RFID scanner and the successful user or admin get the permission to access the door for ten seconds. Every wrong process of this three layers security check user cannot access the next process and admin will get a message by GSM module. Beside these,a live video will stream inside the home and also detect the human face. If any face detected by the system it will capture and save the image and also notify admin by sending a message. Fire detected also added in this security system to detect any flame of fire.

Chapter 2

Classification of the Security System

2.1 Most of the security system are designed in such a way so that it can prevent unauthorized entry from the intruders, burglars and detect invasion minimized of valuable utilities and property damage as well as personal protection in the residential, commercial and industrial zone. On the other hand Security system also increase the economic growth of the society.

There are numbers of security systems in this world. Some are as follows

1. Danalock Bluetooth Z-wave smart lock
2. Dropcam pro Wifi Wireless video Monitoring camera
3. Everspring Z-Wave door/Window sensor
4. GE personal security Alarm kit
5. Monitored security system
6. Unmonitored security system
7. Electronic security system

2.1.1 Everspring Z-Wave door/Window sensor:

This Z-wave powered unit will notify an alert whenever there is crack or break in the sensor, besides this you would also know who's is coming or going out of your home. It has the feature to Compatible with any contact switch devices.It is mainly battery powered.

2.1.2 GE personal security Alarm kit:

This home monitoring system is a truly economical solution for basic home security. The 120-decibel alarm alerts you to any potential intruders. This device has three window alarms and one deluxe door alarm. Its battery consumption is up to 1 years use and the best part is that there is no wire necessary to become active.

Figure 2.1.4 GE personal security alarm kit

2.1.3 Monitored security system:

Monitored system alarm is the most commonly used and has some pros and cons like this system alarm a call center when the user gets triggered. Then the call center calls the police. But the problem is that this system goes through the outdoor phone line. If the burglar smart enough he can locate the line to cut them before breaking in and the call center would never be notified about that. So to minimize the problem you can use a cellular phone or radio as an alternative.

Another problem is that the burglar has quite a bit of time to get in and get a few valuables by the time when the call center and the police get notified. This system is more expensive than any other system.

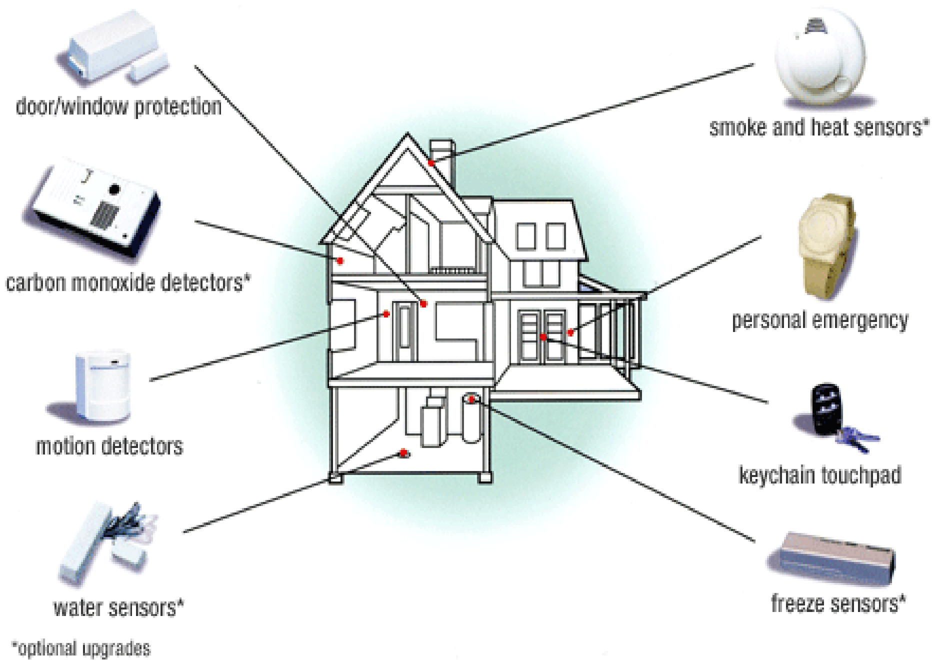


Figure 2.1.5 monitored security system

2.1.4 Unmonitored security system:

The Unmonitored security system is a system where sets off a loud distress signal inside and outside the house when an alarm is tripped. This system relies on your neighbors nearby to call the police if you are not home. A major benefit to this system is you will not have to spend money monitoring fees, making it much more economical. The system can be installed with flashing lights so that people can know from where the alarm is being sounded.

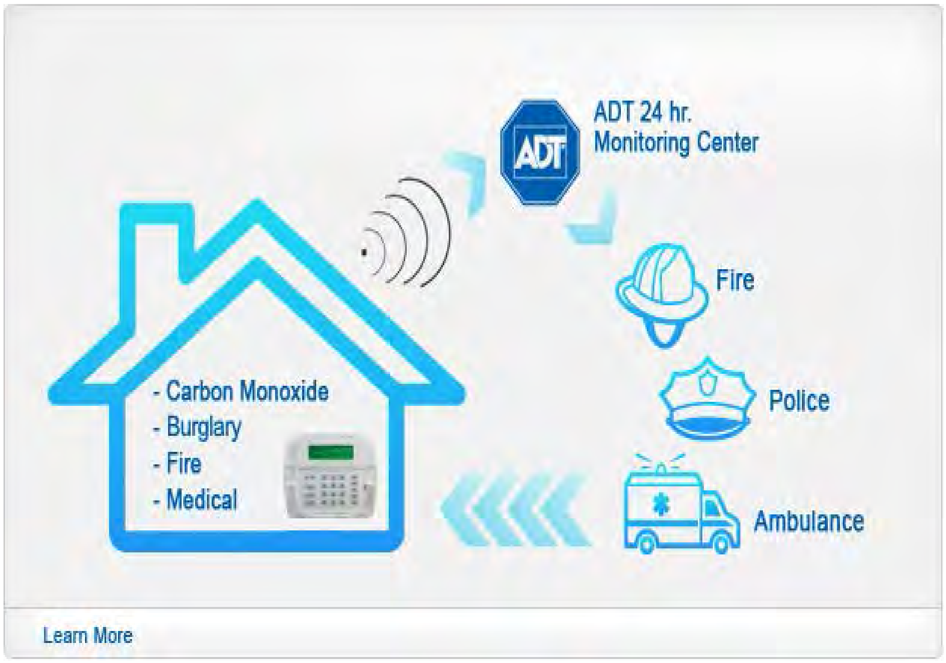


Figure 2.1.6 unmonitored security system

2.1.5 Electronic security system:

Electronic security systems developed with one or more danger sensing units which placed at the front of the system and generate some kind of electrical output when danger is sensed. The output of the sensor unit is nourished from a data link to a decision making signal processing unit, and this unit's output is nourished via another data link, to a 'danger' response unit such as an alarm or an electromechanical trigger or shutdown device. A simple electronic doorbell or shop-entry alarm system is an example of electronic security system

2.1.6 Fingerprint Authorization:

There are three types of scanners used to authorize fingerprint. They are 1. Optical scanner, 2. Ultrasonic scanner, and 3. capacitive scanner. In our system, we use a capacitive scanner to authorize fingerprint. To make up a fingerprint both optical scanners and capacitive scanners generate an image of the ridges and valleys. But the difference between optical and capacitive scanners is that optical scanners use light whereas capacitive scanners use electrical current to sense the print.

2.1.7 Optical scanners:

This method is the oldest method to capture and compare fingerprint which works by capturing an optical image and using algorithms to detect the ridges and valleys and analyzing the darkest and lightest area of the image. Security level increase depending on the sensor resolution so the higher the resolution of the sensor, the better details of the image. It's very dark when your finger is placed on the scanner, that’s why optical scanners also incorporate arrays of LEDs as a flash to light up the picture come scan time.

2.1.8 Ultrasonic scanners:

The latest fingerprint scanning technology to enter the smartphone space is an ultrasonic sensor which actually captures the details of a fingerprint. The hardware consists of both an ultrasonic transmitter and a receiver. An ultrasonic pulse is transmitted against the finger that is placed on the scanner. Some of this pulse is absorbed and some of it is bounced back to the sensor, depending upon the ridges, pores and other details that are unique to each fingerprint

2.1.9 Password Authorization:

Password-based security door lock system is an access control system that allows only authorized persons to access a restricted area. This system is best for home security system and also for corporate offices and ATMs. The system contains a small electronic unit with a numeric keypad which is fixed outside the entry door. When an authorized person enters a user ID and password with the keypad the door will open and after a few times later the door will lock again. If the code entered incorrectly three times in a row, the code lock will switch to block mode and the function prevents any attempts by the hackers try a large number of codes in a sequence. If the users forget his/her password the code lock can be accessed by a unique 10 digits of the administrator password. The secret code can be changed anytime by the master code. A buzzer can be providing on the system for audio acknowledgment of the key impression. A short beep sound can make an impression the system acknowledgment when a key pressed on the numeric keypad.

2.2.0 RFID Authorization:

A Radio-Frequency Identification system has three parts which are a scanning antenna, a transceiver with a decoder to interpret the data and a transponder - the RFID tag - that has been programmed with information.The scanning antenna puts out radio-frequency signals in a relatively short range. The RF radiation does two things those are, it provides a means of communicating with the transponder and it delivers the RFID tag with the energy to communicate.This is an absolutely key part of the technology; RFID tags do not need to contain batteries, and can, therefore remain usable for very long periods of time.

The scanning antennas can be permanently affixed to a surface; handheld antennas are also available. They can take whatever shape you need; for example, you could build them into a door frame to accept data from persons or objects passing through.

RFID tags can be read in a wide variety of circumstances, where barcodes or other optically read technologies are useless. The tag need not be on the surface of the object and is therefore not subject to wear.

2.2.1Face Recognization:

Face detection technique is used to recognize the face which usually works by the capacity to dependably find a face and its landmarks. This is basically a segmentation issue and vast majority of the exertion goes to solving this problem. In fact, the genuine acknowledgment in light of components separated from these facial landmarks is just a minor last step. There are two sorts of face identification issues:

1. Face detection in pictures and
2. Real-time face detection

2.2.2 Face detection in pictures:

Face detection systems attempt to remove a small amount of the entire face, so eliminating the majority of the background and different areas of an individual's head like hair that is a bit much for the face detection. This is frequently done by running a "window" over the picture and the face location framework judges if a face is available inside the window with static pictures but static images have large space of possible locations of a face in an image so it can be situated anyplace from the upper left to the lower right of the picture and 21 extensive or little. Most face detection systems utilize an example-based learning approach to deal with choose whether or not a face is available in the window at that given moment. A neural network enables it to group a picture as a "face" or 'non-face' by pictures for effective training. There is another procedure for figuring out if there is a face inside the face location framework's window - utilizing Template Matching. The contrast between a fixed target pattern and the window is computed. If the window contains a pattern which is near the objective pattern then the window is judged as containing a face and uses an entire bank of fixed sized templates to identify facial components in a picture. By utilizing a few formats of various sizes, appearances of changed scales are distinguished. There is another implementation of layout coordinating is utilizing a deformable format. Rather than utilizing a few settled size formats, we utilize a deformable layout and thereby change the span of the format hoping to identify a face in a picture.

2.2.3 Real-time face detection:

This process is really a far less complex than detecting a face in a static image. The reason behind this is surrounding our environment, people are continually moving around, blink, fidget, wave our hands about, etc. Since in real-time face detection, the framework is given a progression of edges in which to distinguish a face, by utilizing spatiotemporal filtering the region of the casing that has changed can be recognized and the individual identified. Besides, correct face areas can be effortlessly distinguished by utilizing a couple of straightforward standards, for example, 1) the head is the little blob over a bigger blob - the body 2) head movement must be sensibly moderate and bordering - heads won't bounce around erratically.

Real-time face detection has in this way turn into a moderately basic issue and is conceivable even in unstructured and uncontrolled situations utilizing these simple picture handling systems and reasoning rules

2.2.4 Fire-Detection Systems:

Before deciding to replace an old fashioned fire alarm we should be aware of the different types of fire alarm systems that are on the market. The two main types of fire alarm systems are conventional and addressable.

2.2.5 Summary:

There are different kind of security systems which we talked throughout the whole chapter in details and we discussed their structures, working principles, both advantages and disadvantages shortly. We tried to clarify the overview of our proposed security system and the subsystems of this project. For example, we talked about fingerprint scanner, their types and working principle. Furthermore, we also discussed another subsystem like password authorization, RFID authorization, face detection and their types as well as fire detection and their types in briefly.

Chapter 3

Hardware and component

3.1 Introduction:

The main theme of our project is to build a smart home security system by using the most efficient and reliable components in order to achieve a full hundred percent output of the system .Secondly we tried to minimize the overall cost of the equipment so that at a very low cost we can get better efficiency.The major component used in this project are Arduino mega

2560, as a microcontroller, RFID module, GSM+GPRS module, Raspberry Pi , Raspberry Pi camera module, and some other additional component.

3.2 Component used in the hardware:

Components we used in this project are as follows:

1. Arduino Mega 2560

2. Fingerprint Scanner
3. 125Khz RFID Module
4. RFID Tag
5. 16*2 lCD display
6. 4*4 Key Capacitive Keypad
7. Simcom Sim900a GSM+GPRS module
8. Electric Solenoid Mini Door Lock DC 12V
9. Buzzer
10. Raspberry Pi 3 Model B
11. PIR sensor
12. Adaptor
13. Vero board
14. Micro servo
15. Raspberry Pi camera module
16. Arduino UNO
17. Flame Sensor
18. Lm2596 dc-dc adjustable step-down module
19. 12V Door lock driver
20. LED Lights-5mm
21. HDMI cable
22. Smart phone

3.3.1 Arduino Mega 2560

This system uses the Arduino Mega 2560, which is a high-end microcontroller unit in comparison to most other similar boards and has been chosen so that handling large amounts of data is not an issue, as it has a fairly large RAM. The device driver programs are solely responsible for controlling the hardware devices and executing low-level hardware-specific routines. These custom device drivers running on the Arduino Mega 2560 microcontroller are designed specifically to suit the needs of our proposed security system.

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560. There are 54 input/output pins.

→ 15 PWM (Pulse width modulation) outputs

→ 16 analog inputs

→ 4 UARTs (hardware serial ports)

→ A 16 MHz crystal oscillator

→ USB connection

→ A power jack

→ An ICSP header

→ A reset button

USB connection or external power supply is the main power source of Arduino Mega which is selected automatically and the external power can come through an AC to DC adapter or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack and a battery can be inserted in the Gnd and Vin pin headers of the power connection. The board can operate on an external supply of 6 to 20 volts and the recommended range is 7 to 12 volts. Also, there are some restrictions on the board's power supply. If the power supply is less than 7 volts then the 5V pin may supply less than five volts and the board is unstable but if the supply is more than 12 volts then the voltage regulator may overheat and damage the board.

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter it has

256KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the EEPROM library).

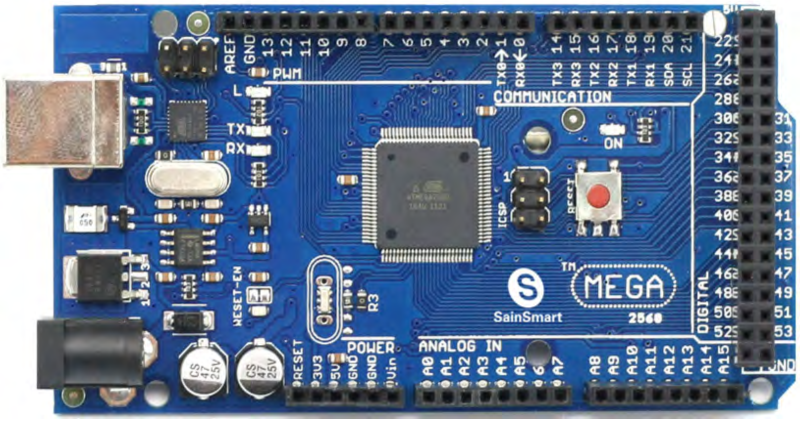


Figure 3.3.1 Arduino mega 2560

The Arduino Mega can be programmed with the Arduino software. The ATmega2560 on the Arduino Mega comes pre-burned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files). You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega2560 via a 100 Nanofarad capacitor. When this line is asserted, the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. The Arduino Mega2560 has a resettable poly fuse that protects your computer's USB ports from shorts and overcurrent. The Mega2560 is designed to be compatible with most shields designed for the Uno, Diecimila or Duemilanove. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

3.3.2 Fingerprint Scanner TTL (GT-511C3)

The fingerprint scanner used for the system is TTL-(GT-511C3). This model has been chosen for reliability and for cost considerations, and it's capability of running verification algorithm within itself, eradicating the need of exhausting fingerprint matching algorithm to be implemented and run on the scarce memory of the Android device. This scanner will be attached to the Arduino board using a JST-SH Jumper 4 Wire Assembly. They are Vcc, GND, Tx, Rx. The easiest way to demo this would be to connect Pin 1and Pin 2 to Arduino D3 and D4 respectively and run FPS Blink. If the FPS blinks blue that mentions the device is working.

This module is used for both reading and identifying the fingerprints with an optical sensor and 32-bit CPU. The fingerprint scanner can store different fingerprints and the database of prints can even be downloaded from the unit and distributed to other modules.AS well as the fingerprint "pattern", the analyzed version of the print the module take the image of the print and raw image pull by the optical sensor.

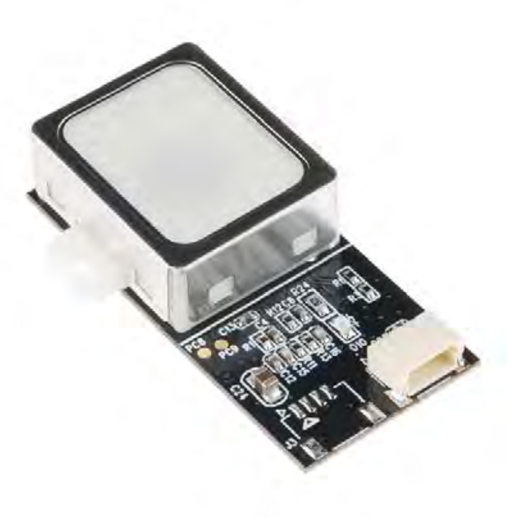


figure 3.3.2 fingerprint scanner TTL

This is the updated version of the GT-511 which has an increased memory capacity. The module can store up to 200 different fingerprints and is now capable of 360° recognition.

3.3.3 125Khz RFID Module

Radio frequency identification (RFID) is a wireless device that is basically used in electromagnetic fields to transfer data, for the purpose of automatically identifying and tracking tags attached to objects. There is a signal which is transmitted through the antenna for activate the tags. The signal itself is a form of energy that can be used to power the tag. The radio frequency converts into usable power by the transponder which is the part of RFID tag, Also send and receive messages. RFID 125 KHz card mini-module is designed for reading code from 125KHz card compatible read-only tags and read/write a card. It can be applied in office/home/industry security, personal identification, access control and production control systems etc.

Finally, RFID is the method for automatic identification and Data Capture.

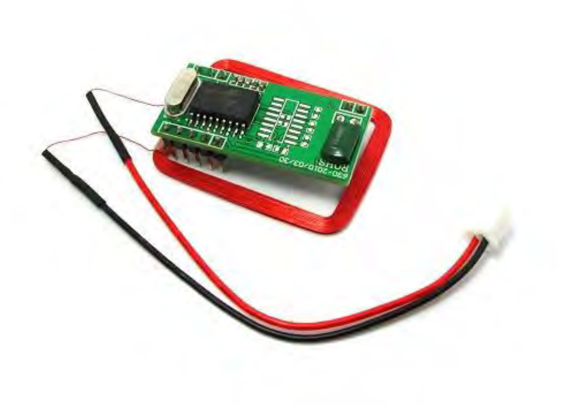


Figure 3.3.3 125khz Rfid module

The Radio Frequency Identification (RFID) Reader Module provides a low-cost solution for reading RFID transponder tags from up to 4 inches away. The module can be used in a wide variety of commercial applications, including access control, user identification, robotics navigation, inventory tracking, payment systems, car immobilization, and manufacturing automation.

3.3.4 RFID Tag

The RFID tags contained electronically stored information. There are two types of tags:

1) Active Tags
2) Passive Tags

The active tags have a local power source such as a battery and many operate at hundreds of meters from the RFID reader and the passive tags collect energy from nearby RFID readers interrogating radio waves.

The RFID tags are widely used in industries. Library systems, real time location system, toll tracking, access control are some of the industrial application of RFID tags. Since RFID tags can be attached to cash, clothing, and possessions, or implanted in animals and people, the possibility of reading personally linked information without consent has raised serious privacy concerns.

Since RFID tags can be attached to cash, clothing, and possessions, or implanted in animals and people, the possibility of reading personally-linked information without consent has raised serious privacy concerns.

Figure 3.3.4 RFID tag

3.3.5 16*2 lCD display

LCD (Liquid Crystal Display) screen is an electronic display module and finds a wide range of applications. A 16x2 LCD display is a very basic module and is very commonly used in various devices and circuits. These modules are preferred over seven segments and other multi-segment LEDs. The reasons being: LCDs are economical; easily programmable; have no limitation of displaying special & even custom characters (unlike in seven segments), animations and so on.

Figure 3.3.5 16*2 lCD display

A 16x2 LCD means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in the 5x7 pixel matrix. This LCD has two registers, namely, Command and Data.

The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc. The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD. Click to learn more about the internal structure of an LCD.

3.3.6 4*4 Key Capacitive Keypad

The TTP229 Touch IC is capacitive sensing design specifically for touchpad controls. The device built in regulator for the touch sensor. Stable sensing method can cover diversity conditions. Human interfaces control panel links through the non-conductive dielectric material. The main application is focused on replacing the mechanical switch or button. This Module can replace conventional 4×4 keypad with a touch based keypad. The module has two working modes. A 8 key mode that provides an independent 8 channel output or a 16 key mode that can be used with the I2C interface of the module, thereby saving, even more pins and connections on the application Arduino or Microcontroller board.

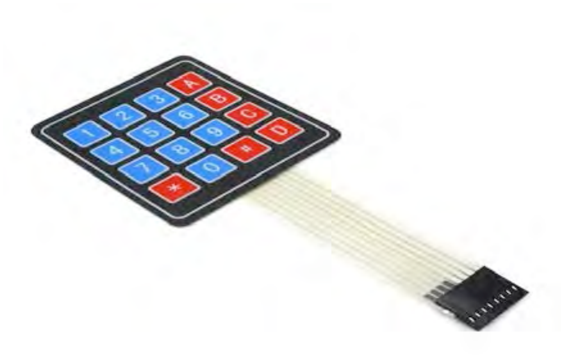


Figure 3.3.6 4*4 key capacitive keypad

Operating voltage for this keypad is 2.4V~5.5V and has a built-in regulator. The standby current at 3V for 16 input keys is typically 2.5uA and also has a 2-wires serial output interface, both can use for 8 and 16 direct input keys mode. It has 8 separate outputs can select output driving types by option and provides two kinds of sampling rate that slow sampling rate 8 Hz and fast sampling rate 64Hz at sleep mode. Furthermore, it offers multi-key or single-key feature by option.

3.3.7 Simcom Sim900a GSM+GPRS module

SIM900A modules were used for all IO port pin leads which is easy to use and has basic features. It has the board RS232 serial port which supports hardware flow control, convenient and user-friendly with and PC / IPC and other devices. For easy voice communication development, onboard 3.5mm headphone and microphone are given to the module. It leads all the IO ports, and communications section IO port compatibility design made for easy connection 3.3V/5V SCM system. It has efficient synchronous buck circuit board which has conversion efficiency up to 90% and support for wide voltage range (5 ~ 24V) so that it is ideal for industrial applications. The board has onboard power anti-reverse protection, TVS power protection, SIM card ESD protection and the protection function, in addition, it has board RTC backup battery (XH414H-IV01E), there is no worry about the power-down problem. Its onboard antenna can effectively improve the signal reception and also can adopt international A-level PCB material by using immersion gold processing technology which is, stable and reliable, processed using the new components, copper plated pin and durable. The module is designed so perfectly that each interface has a screen annotation by using a glance and also connector location and reasonable arrangements designed to facilitate smoothly. PCB size is 80mm * 58mm and with mounting holes, small and exquisite. ATK-SIM900A module supports RS232 serial port and with hardware flow control it supports for 5V ~ 24V that facilitate the wide scope of work, so that the module can be very convenient to connect with the product and giving the product, including voice, SMS, GPRS data transmission and other functions with more efficiently.

3.3.8 Electric Solenoid Mini Door Lock DC 12V

The supply voltage of this electric device is 12 V DC which has to lock the telescopic length of 10mm and power form is interrupted. The unlock time for the door is 1 second. It’s red wire connects to the positive supply and black wire connects to the negative supply. Once supply current is available, the electric lock will retract and unlock the door.

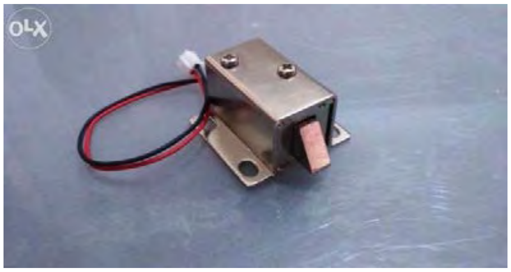


Figure 3.3.8 electric solenoid mini door lock dc 12V

3.3.9 Buzzer

Early devices were based on an electromechanical system identical to an electric bell without the metal gong. Similarly, a relay may be connected to interrupt its own actuating current, causing the contacts to buzz. Often these units were anchored to a wall or ceiling to use it as a sounding board. The word "buzzer" comes from the rasping noise that electromechanical buzzers made.

In the present day, the buzzer is usually used for novelty uses, judging Panels, educational purposes, electronic metronomes, microwave ovens and other household appliances, electrical alarms and many other electronic devices.

Figure 3.3.9 buzzer

3.3.10 Raspberry Pi 3 Model B

The raspberry pi 3 is the third generation raspberry pi.It has a 1.2GHz 64-bit quad-core ARMv8 CPU and 802.11n Wireless Lan.Bluetooth version 4.1 is very fast in transmitting the data from one device to another.It has a 4 USB port and 1gb ram, full HDMI port is also available.raspberry pi has compatibility with raspberry pi 1 and pi 2.

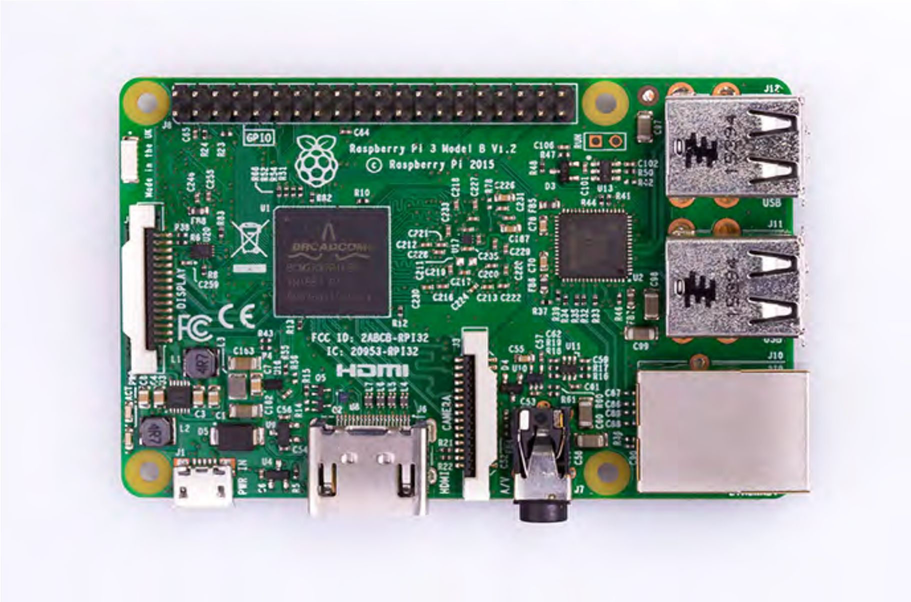


Figure 3.3.10 Raspberry pi 3 model b

3.3.11 Arduino UNO

Arduino is a software company, project, and user community that designs and manufactures computer open-source hardware, open-source software, and microcontroller-based kits for building digital devices and interactive objects that can sense and control physical devices. The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins, 6 analog inputs, a 16MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button.Analog inputs, a 16MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button.

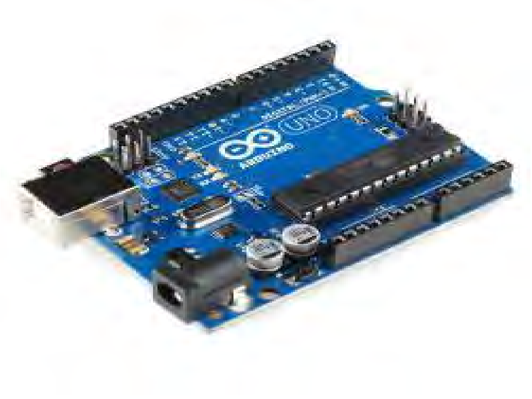


Figure 3.3.11 Arduino Uno

3.3.12 HC-SR501 PIR SENSOR MODULE

PIR sensors used to sense the motion, almost always used to detect whether a human has moved in or out of the sensors range. These are small, low-power, easy to use, inexpensive, and don't wear out. So that reason they are mostly found in the appliances and gadgets used in homes or businesses.

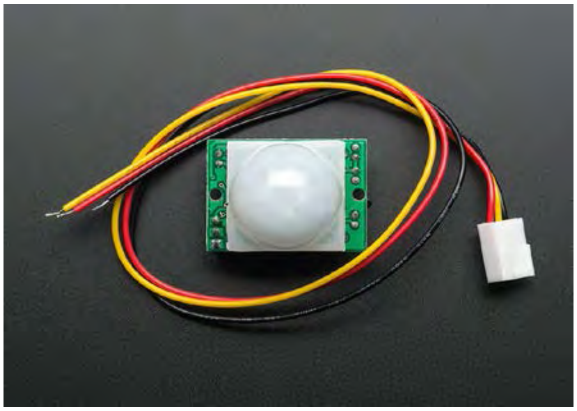


Figure 3.3.12 hc-sr501 PIR sensor module

3.3.13 Adaptor

Adaptor for powering up the arduino

Figure 3.3.13 adaptor

3.3.14Vero board

A Vero board is a design in such way that the strip of copper clad features a grid pattern.The distance between the two holes is 2.54mm apart.With the help of vero board you can get more reliable and permanent circuit.

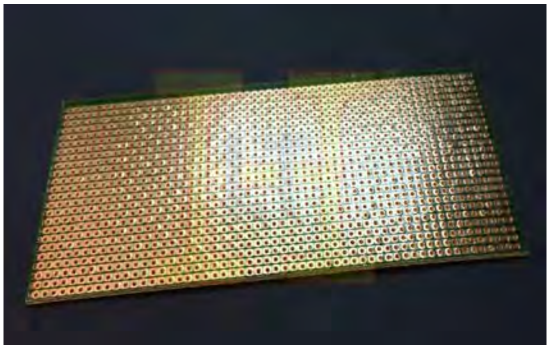


Figure 3.3.14 Vero board

3.3.15 Micro servo

This is a servo motor where it can do a rotary actuator or linear actuator that allows any object to take control of angular or linear position, velocity, and acceleration.

Figure 3.3.15 micro servo

3.3.16 Raspberry pi camera

The Raspberry Pi camera module can be used to take high-definition video, as well as stills photographs. This module has a five-megapixel fixed-focus camera that supports 1080p30, 720p60 and VGA90 video modes, as well as stills capture. It attaches via a 15 cm ribbon cable to the CSI port on the Raspberry Pi.

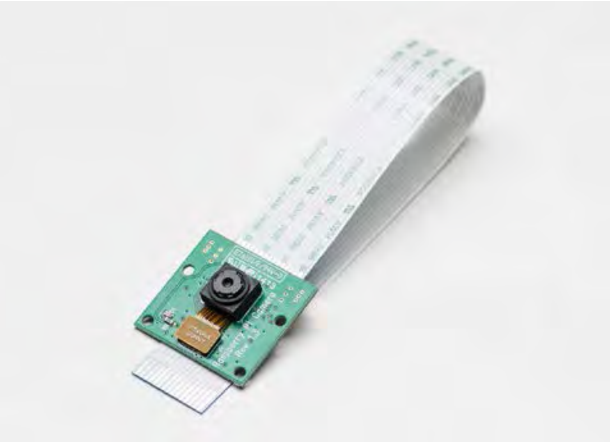


Figure 3.3.16 raspberry pi camera

3.3.17 Flame Sensor

A flame detector is a sensor designed to detect and respond to the presence of a flame or fire. Responses to a detected flame depend on the installation but can include sounding an alarm, deactivating a fuel line, and activating a fire suppression system. A flame detector can often respond faster and more accurately than a smoke or heat detector due to the mechanisms it uses to detect the flame. Near-infrared (IR) array flame detectors, also known as visual flame detectors, employ flame recognition technology to confirm fire by analyzing near IR radiation using a charge-coupled device (CCD). Infrared (IR) flame detectors monitor the infrared spectral band for specific patterns given off by hot gasses.

These are sensed using a specialized fire-fighting thermal imaging camera (TIC), a type of thermographic camera. False alarms can be caused by other hot surfaces and background thermal radiation in the area. Water on the detector's lens will greatly reduce the accuracy of the detector, as will exposure to direct sunlight. A single-frequency IR flame detector is typically sensitive to wavelengths around 4.4-micrometers, which is a spectral characteristic peak of hot carbon dioxide as is produced in a fire. The usual response time of an IR detector is 3–5 seconds. Dual IR (IR/IR) flame detectors compare the threshold signal in two infrared ranges. Often one sensor looks at the 4.4 micrometer carbon dioxide (CO2) emission, while the other sensor looks at a reference frequency. Sensing the CO2 emission is appropriate for hydrocarbon fuels; for non-carbon based fuels, e.g., hydrogen, the broadband water bands are sensed.

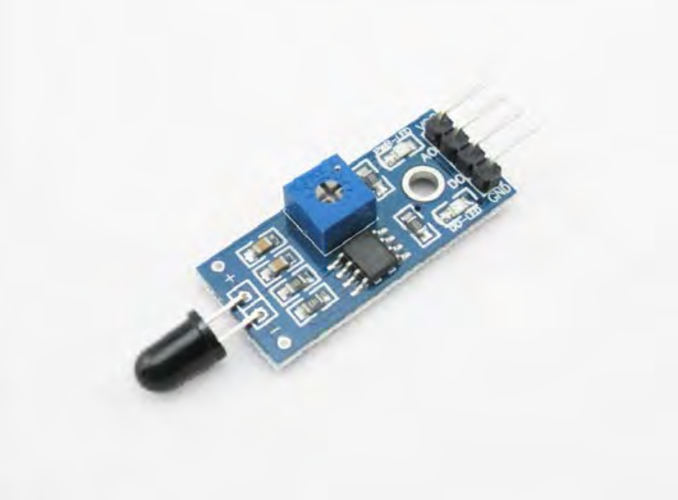


Figure 3.3.17 flame sensor

3.3.18 LM2596 dc-dc adjustable step-down module

There are number a number of applications when a single DC voltage is not sufficient. For examples like SD Card or XBee modules which both require a 3.3v. This can easily be obtained from a microcontroller platform such as the Arduino or by wiring your own voltage regulator. However, if you want to have access to a variable voltage, the circuitry becomes slightly more complicated. An LM2596 DC-DC Adjustable Step-Down Module will allow you to quickly step down a voltage source without having to rely on a voltage regulator circuit which dissipates power as heat. It will also give you the flexibility of adjusting the output voltage on the fly with the aid of a potentiometer

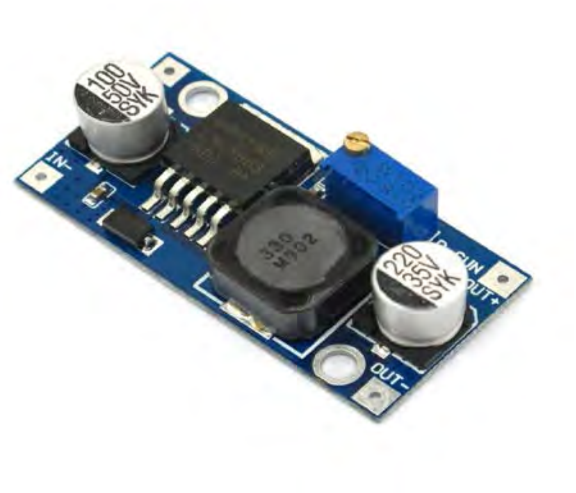


Figure 3.3.18 lm2596 dc-dc adjustable step-down module

3.3.19 12V Door lock driver

The door lock operates at 12V dc for example the case of microcontroller device can tolerate more than 5V dc. This door lock driver hold the capacity to move a servo motor.

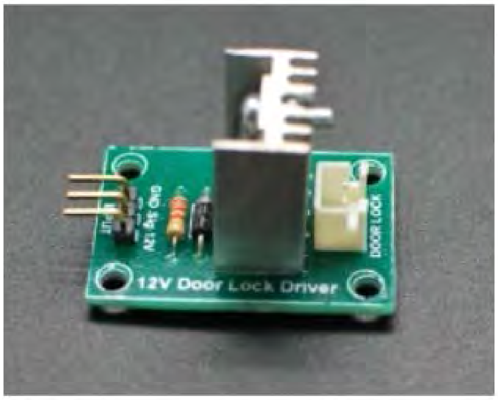


Figure 3.3.19 12V door lock driver

3.3.20 LED Lights-5mm

A light-emitting diode (LED) is a two-lead semiconductor light source. LEDs - those blinky things. A must have for power indication, pin status, optoelectronic sensors, and fun binky displays.

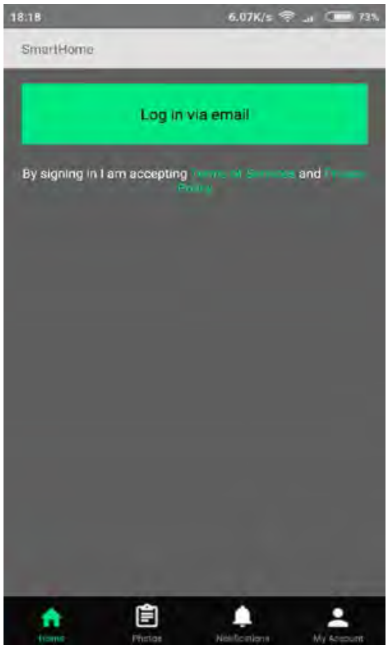
HDMI PORT

HDMI is a both forward and backward process that creates a communication from sending to receiving the device. As a result, the role the port plays on the input side of the signal is critically important. These ports are connected to the electronics that decode the incoming signal from the source component, verifying the HDCP handshake, and sending display data to the source. In addition, these ports must be able to pass along the HDCP coding, if there are other devices upstream.

HDMI ports send audio, video and control information to switchers, televisions, projectors and audio/video receivers. These devices also send the high-definition content protection (HDCP) key, which must be acknowledged at the other end of the cable.

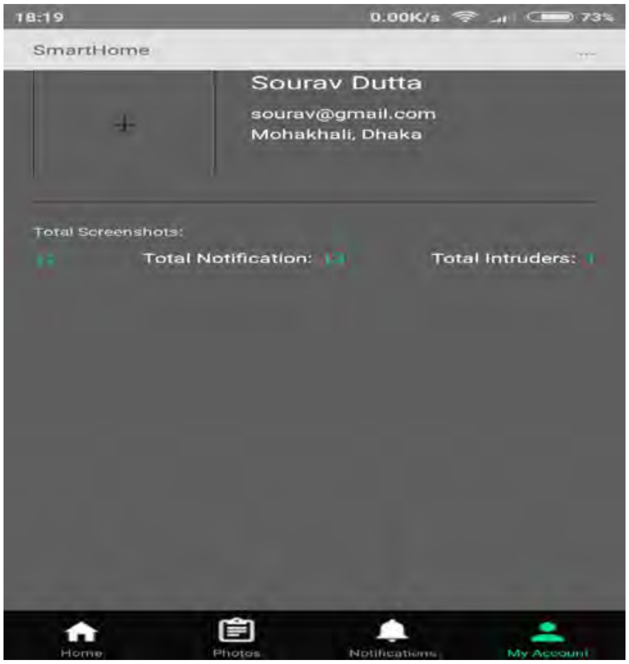
3.3.21 Smartphone Application

Nowadays smart phone is so popular and even the cost is so cheap that anyone can afford the phone. It is helpful to create a communication to long distances.We develop a prototype applicatuion with android studio in reactnative.In this section the admin can create his profile by email address.When someone tries to get access with a wrong password three times, the raspberry pi camera wiil take a picture and send it to a ftp server.The mobile app will hit the ftp server when a new data is formed.The admin will be notified with a push notification.There is an archive in the app where all the images will be stored from the installation date.



By signing in I am accepting Terms of Services and Privacy Policy

Figure3.3.21 step 1



Sourav Dutta

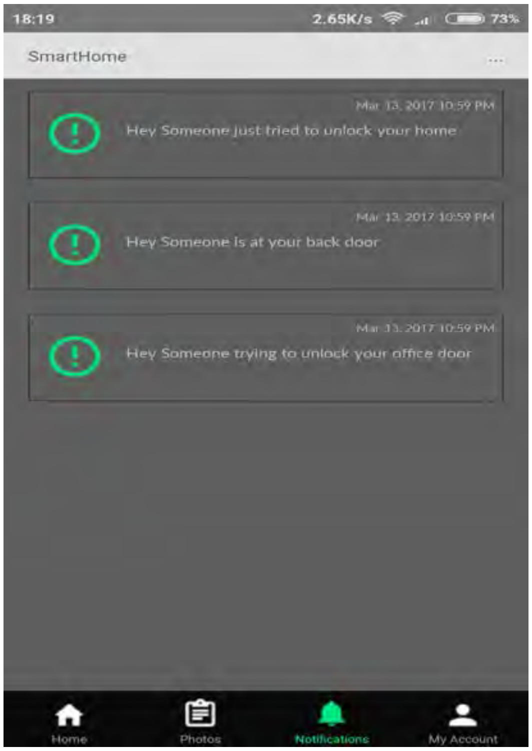
sourav@gmail.com
Mohakhali, Dhaka

Total Screenshots: [illegible]

Total Notification: 13

Total Intruders: 1

Figure 3.3.22 step 2



Mar 13, 2017 10:59 PM
Hey Someone just tried to unlock your home

Mar 13, 2017 10:59 PM
Hey Someone is at your back door

Mar 13, 2017 10:59 PM
Hey Someone trying to unlock your office door

Figure3.3.23 step 3

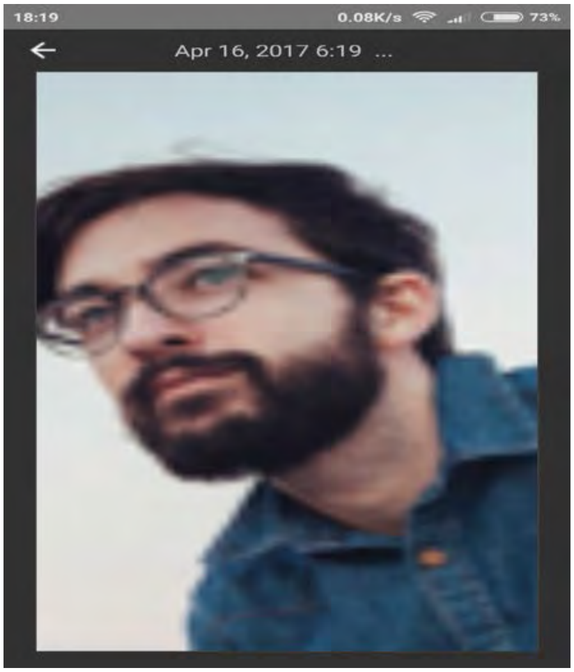


Figure3.3.24 step 4

Chapter 4

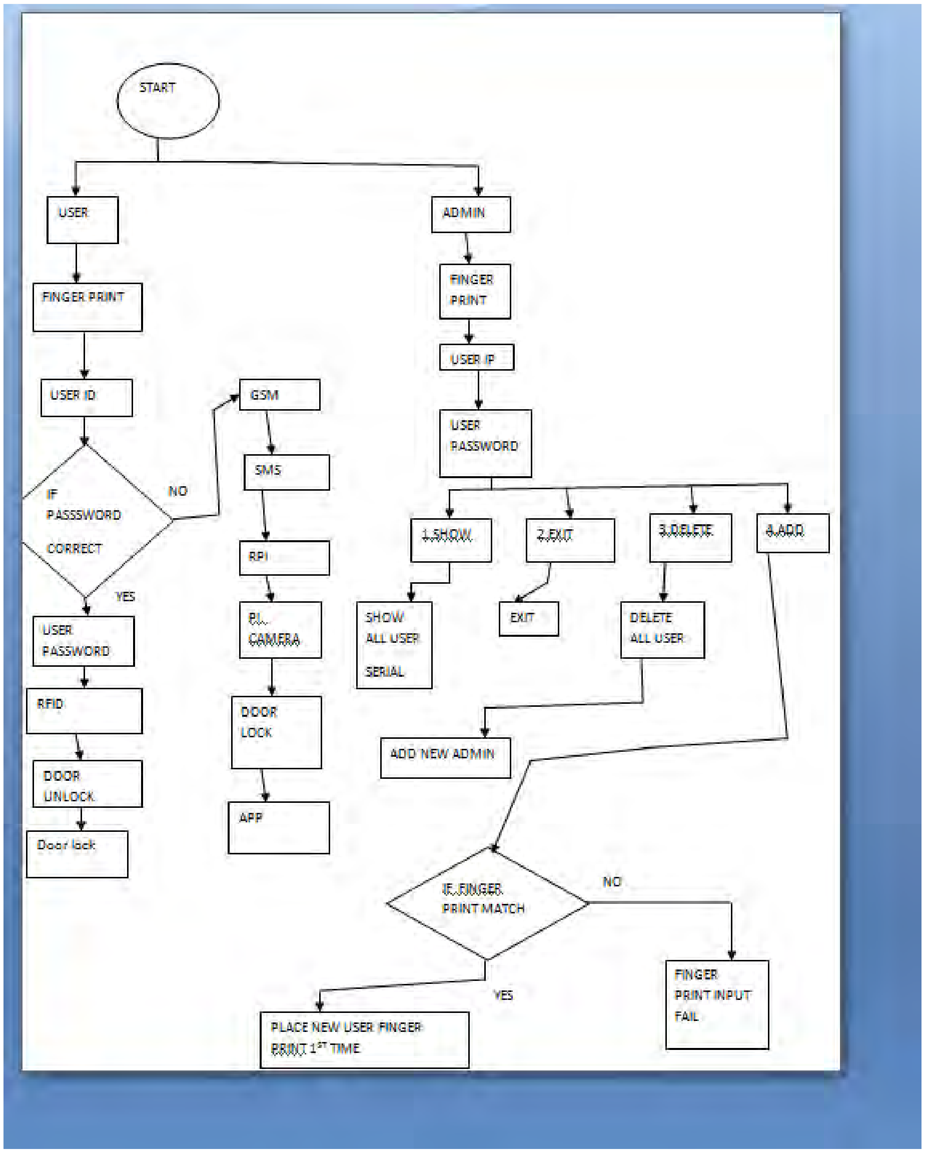
Structural Design Flowchart and Circuit

4.0 Introduction:

The prime concern of this project is to hinder any perpetrators, trying to illegitimately break into an establishment, and alert the admin instantaneously. The authentication devices (i.e. Fingerprint Scanner, RFID, Keypad, and display) is positioned on the outside of the door and the security system can be only accessed from outside. However, an unlock switch is placed inside the door in order to open it from the inside at any time. This is because, naturally the person who would be able to access the door from the inside, would not be a perpetrator. Therefore it is futile to place another set of authentication devices on the inside. At real implementation of the system, it is thought that the core devices such as the microcontroller unit, Raspberry Pi, GSM module etc. would be at a secretive place such as inside the wall or on a false ceiling so that they are out of reach of people.

4.1 Overview and Block Diagram of the System:

When the system is activated and user put his/her finger to the fingerprint scanner it generates an image of the ridges and valleys of the finger and detects whether the person is authorized or unauthorized. If ridges and valleys match then the system permit the user to access next process. In the next level the system will ask for the user's password, so the user need to type his/her password and the system will check whether the password given is correct or wrong and go with the previous way. Correctly accessed the user will face RFID scanner and the successful user or admin get the permission to access the door for ten seconds. Every wrong process of this three layers security check user cannot access the next process and admin will get a message by GSM module. Beside these, a live video streaming will stream inside the home and also detect a human face and store the information in the app or server. If any face detected by the system it will capture and save the image, led in ON and also notify admin by sending message. Fire detected also added in this security system to detect any flame of fire. If system sense any fire then LED and buzzer will ON and admin will get a notify message from the system.


START
USER
ADMIN
FINGER PRINT
FINGER PRINT
USER IP
USER ID
GSM
USER PASSWORD
SMS
IF PASSSWORD CORRECT
NO
YES
RPI
1.SHOW
2.EXIT
3.DELETE
4.ADD
USER PASSWORD
PI CAMERA
SHOW ALL USER SERIAL
EXIT
DELETE ALL USER
RFID
DOOR LOCK
DOOR UNLOCK
ADD NEW ADMIN
APP
Door lock
IF FINGER PRINT MATCH
NO
YES
FINGER PRINT INPUT FAIL
PLACE NEW USER FINGER PRINT 1ST TIME

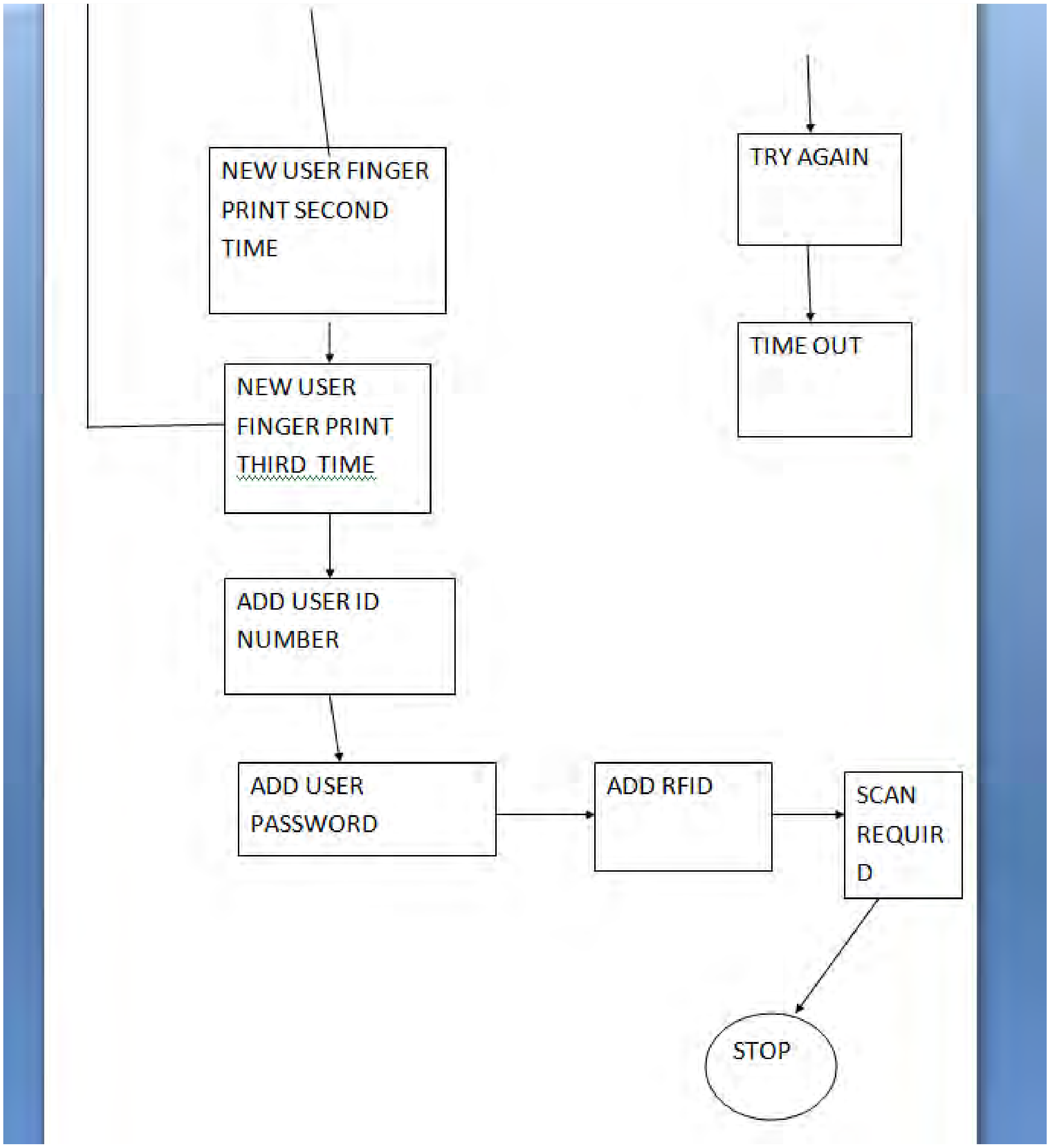


Figure 4.2 overview of the flow chart

4.1 Pin Configuration and Symmetric Diagram:

The keypad are connected to the digital pins 2, 3, 4, 5, 6, 7, 8 and 9 of the Arduino Mega, among which, pins 2, 3, 4, and 5 are connected to the column of the keypad and pins 6, 7, 8 and 9 are connected to the row of the keypad. Pins 50 and 51 of the Arduino board are connected to the RX and TX pins of the fingerprint scanner module, respectively. The module powered by the 5V output pin from the Arduino board. The 125kHz RFID module connected to the Arduino Board on pin no 52 and 53 to the RX and TX pins respectively. The module powered by the 5V output pin from the Arduino board. The GSM module to the Arduino is sent or received through the RX and TX pins which are connected to pins 10 and 11 respectively. The mode of communication used by the GSM module is serial communication. A separate power adapter of 12V, 3A is used to provide the required amount of power to the GSM module to be functional. The electronic solenoid door lock is the main output of the whole system which is linked with the Arduino board by the use of an electronic solenoid door lock driver. The signal pin from the door lock driver is connected to pin number 49 of Arduino. The driver module is powered using a 12V adapter. A 16 X 2 LCD display has 6 data pins which are connected to the pins 33, 35, 37, 39, 41 and 43 of the Arduino board. The power input of the display is 5V which is fed directly from the Arduino board. A 10K analog potentiometer is used adjust the brightness of the LCD. The LED module contains two pins where one is connected to the pin 12 and one of them is the ground. An 89dB buzzer module contains three pins, +5V, ground and a signal pin. The signal pin is connected to the digital pin 13 of Arduino. The Raspberry Pi communicates with the primary control unit of the system using its GPIO pins. Raspberry Pi’s GPIO pin number 26 is connected to the Arduino’s digital pin 49 in order to send a signal regarding the face detection. The Raspberry Pi camera adapter. A 16 X 2 LCD display has 6 data pins which are connected to the pins 33, 35, 37, 39, 41 and 43 of the Arduino board. The power input of the display is 5V which is fed directly from the Arduino board. A 10K analog potentiometer is used adjust the brightness of the LCD. The LED module contains two pins where one is connected to the pin 12 and one of them is the ground. An 89dB buzzer module contains three pins, +5V, ground and a signal pin. The signal pin is connected to the digital pin 13 of Arduino. The Raspberry Pi communicates with the primary control unit of the system using its GPIO pins. Raspberry Pi’s GPIO pin number 26 is connected to the Arduino’s digital pin 49 in order to send a signal regarding the face detection. The Raspberry Pi camera module is connected to the dedicated camera port of the

Raspberry Pi development board. The Flame Sensor is connected to Arduino A0 pin which is one of the analog input pins of the microcontroller unit and provides a real-time output voltage signal on the thermal resistance, the VCC was connected to a 5V power source and GND.

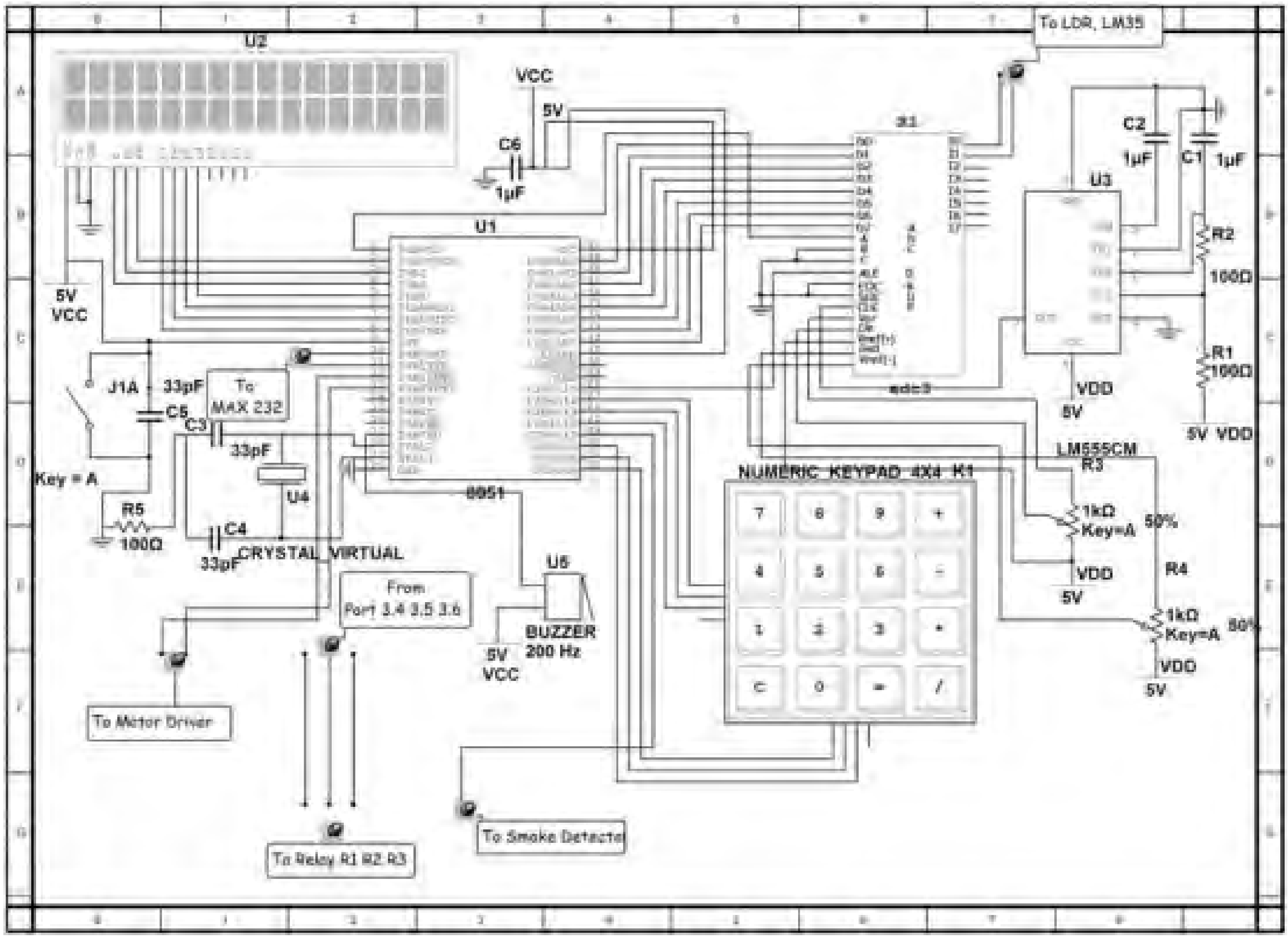


Figure 4.1 Schematic diagram

4.2 Digital Door Lock System:

The system is designed to have four different modes once activated. The modes are shown below in a tabular format.

State	Mode	Door Lock Status	LED/ Buzzer Status	GSM Status	Fingerprint Status	RFID Status	LCD Status	Comments
1	Setup	Inactive	Inactive	Inactive	Active	Active	Active – Instructions for enrollment is displayed sequentially.	Code for enrollment is burned into the microcontroller. Admin information is obtained and stored in the main code.
2	Admin Access	Unlocked	Green; Off	Inactive	Active	Active	Active – Admin Menu and subsequent messages/ instructions are displayed.	Admin Menu is displayed after the door is unlocked.
3	User Access	Unlocked	Blue; Off	Active – Send	Active	Active	Active – Displays	Admin Menu will not be accessible by the

				SMS			“Door Unlocked”	users
4	Wrong Input	Locked	Red; On	Active – Send SMS	Inactive	Inactive	Active – Displays “Wrong ID/Password” / “Fingerprint not Found” / “RFID Not Found” / “Suspicious Activity Detected” / “Text Sent to Admin”	
5	Sleep	Locked	Yellow; Off	Inactive	Inactive	Inactive	Active – Displays “Digital Door Lock”	

Figure 4.2 Operating mode of the system

The system is run at setup mode where a different code is burned into the microcontroller before the main code is uploaded. This is done to enroll the fingerprint of the admin and store necessary information which is required for first boot of the security system. Admin fingerprint enrollment process starts upon requesting to place the finger of the admin three times on the scanner. If the fingerprint is retrieved successfully by the scanner, it is shown on the LCD otherwise, the admin is asked again to place finger until the print is successfully read by the scanner. The RFID card number and a five digit passcode are entered into the main code which will only be usable by the admin. The fingerprint of the admin is stored at rank 0 by default, therefore the admin's user rank is always 0. Once the RFID, passcode and the fingerprint of the admin is stored inside the database, the final code of the security system is ready to be uploaded and run.

The security system is initiated upon the press of "*" button from the keypad which acts as an input signal to start the system and promptly asks for the fingerprint of the user. When a finger is placed on the fingerprint scanner, it cross-matches with the finger print that was previously stored in its database. If the fingerprint matches with the one in the database, the system proceeds to the next step of verification, otherwise it goes to State 4 where it indicates that the fingerprint scanned by the scanner is not recognized, by displaying the message on the LCD while flickering the buzzer and red LED once at the same time before it goes back to sleep mode. If the fingerprint matches with the admin it will be displayed on the LCD.

The next step of verification is a five digit passcode which is entered using the 16X 2 keypads. Right after the fingerprint authentication, the user is prompted to input a five-digit user identification number followed by a corresponding five digit user password. Entering incorrect identification number or the password or even if any of the data in this step mismatches the fingerprint provided in the first step, the system displays a message on the LCD saying the password or user identification number is invalid, concurrently flickering the buzzer and the red LED once. Subsequently, if the password or user identification number is invalid for three consecutive times, the system understands that a suspicious activity is taking place and goes to state 4 where it sends a text message via the GSM module to the admin of the system. This also activates an alarm with a combined flickering of the buzzer and red LED fifteen times to grab the attention of any neighboring individual or passerby. The system automatically goes back to sleep mode right after the alarm goes off.

On another scenario, if the user identification number and the password is correctly entered, the system proceeds to the final step which consists of the RFID authentication. The LCD displays to place the RFID card near the designated area to grant the RFID access. If the RFID card number matches with the one which has been pre-stored in the database and also matches with the information provided in the previous steps, only then the access is granted and the solenoid door lock is unlocked. This information is displayed on the LCD and is indicated by glowing the blue LED as long as the door is kept opened. A text message is also sent to the admin informing which user has accessed the system. The system stays in state 3 for five minutes after which the door is locked and the system goes into sleep mode automatically.

In another situation, if the RFID card number is detected invalid i.e. if there is a conflict between the RFID card number with the information previously provided or if the RFID card is not registered, the LCD displays a message saying that the RFID card is invalid again flickering the buzzer and the red LED once. The system goes back to sleep after the buzzer and LED goes off.

On an alternative circumstance, if the person accessing the system is the admin; the system will run in Admin Access Mode which is State 2. Upon carrying out all the steps of authentication correctly; the admin will be able to access a menu, named ‘Admin Menu’, which will pop up automatically on the LCD with the unlocking of the solenoid door lock. At this stage, the system will change its state to unlocked state unless obliged to perform otherwise. The ‘Admin Menu’ will have four different options which will only be available when the system is accessed by the admin. The options are:

1. Delete – In this process all the information that had been stored inside the system i.e. fingerprints, user identification number, and the RFID numbers will be erased and immediately the admin will be prompted to enroll his/ her fingerprint, identification number and password and RFID number to access the system once again. This process is triggered by the input of the “2” key on the keypad. The system goes back to sleep when the process is complete.

2. Add – This is the process to authorize another user to the system and is initiated when "3" is pressed on the keypad. The user is prompted to enroll his/ her fingerprint then to enter user identification number and password and RFID respectively following the enrollment process which had been explained earlier. The information of the new user is then stored in the database and the user is then authorized to access the security system in future. User ranks are assigned chronologically by the system automatically. The system goes back to sleep mode once the procedure is complete.
3. Show – This process is commenced when the input from the keypad is "1". This shows the list of all the users who are currently authorized and enrolled to the system. In fact, the user identification number and their corresponding password is displayed on the LCD one after another with certain time delays. The menu screen is displayed again on the LCD when the process is complete. At this stage, the state of the system remains unchanged.
4. Exit – Exits the menu and resets the system as it goes back to sleep mode.

The system automatically goes to state 5 which is the sleep mode when the door is unlocked and the system is inactive for 5 seconds.

4.3 Face Detection System:

The face detection system works only works when the Raspberry Pi system is switched on by the admin. In a situation when no human is supposed to be present inside the room where the system is present, the Raspberry Pi face detection system starts rolling a live video in search for a human face. If any human face is found during those restricted hours, the Raspberry Pi sends a signal through one of its GPIO pins. That signal is fed through to the Arduino Mega as it is the primary controller for the entire system. The received signal triggers the GSM module which sends a text message to the admin of the system indicating that an intruder is present in the room. This also activates the emergency alarm of the system and with the blinking of a red LED. The Raspberry Pi immediately captures the picture of the person and saves it in its microSD card for later use for the admin. For a better user experience, one picture of one type of face is saved despite reading the same type of face multiple times.

4.4 Fire Detection and Alarm System:

The fire detection and alarm system is using the flame sensor to detect any fire inside the room or nearby. However, unfortunately, due to a limited budget a low-priced flame sensor with a range of maximum 50cm was used in this project. Upon detection of fire, the sensor sends a signal directly to the Arduino Mega which triggers the alarm and activates the blinking of red LED. A text message is also sent to the admins mobile phone regarding the fire.

Chapter 5

System Software Design configuration and Implementation

5.1 Introduction

In our project, we chose Arduino Mega as our microcontroller which is based on C or C++ language.For effective communications and fast processing between hardware components to microcontroller, we used this programming language. We also used Raspberry pi and Raspberry pi camera for face detection and flame sensor are for fire detection.

5.1 Software algorithm Design:

Step 1: Start

Step 2: User // who will try to enter

Step 3: Finger Print //user will put the finger on finger print scanner

Step 4: Password // user will press 4 digit of secret code

Step 5: RFID // on RFID scanner

Step 6: Door unlocked // If all the secret code attempted successfully

Step 7: Door locked // door will locked after few seconds of successful attempt

If any of the step of 3, 4, 5 has been attempted for three unsuccessful times then

Step 8: Go back to step 5

Step 9: GSM will be activated

Step 10: Mobile SMS // a message will be forwarded to admin

Step 11: Raspberry Pi // RPi will active to send command to Pi camera

Step 12: Pi Camera // Pi camera will take an image of user

Step 13: Database // image will be stored in a database

Step 14: Android Application // image will be sent to android app of the admin with push notification

There is another step for Admin

Step 1: Start

Step 2: Admin

Step 3: Finger print

Step 4: ID

Step 5: Password

Step 6: Three option will arrive i) Show; ii) Delete; iii) Add; iv) Exit

Step 7: press no i // it will show all the admins

Step 8: Go back to step 6

Step 9: press no ii // it will delete admin

Step 10: Go back to step 6

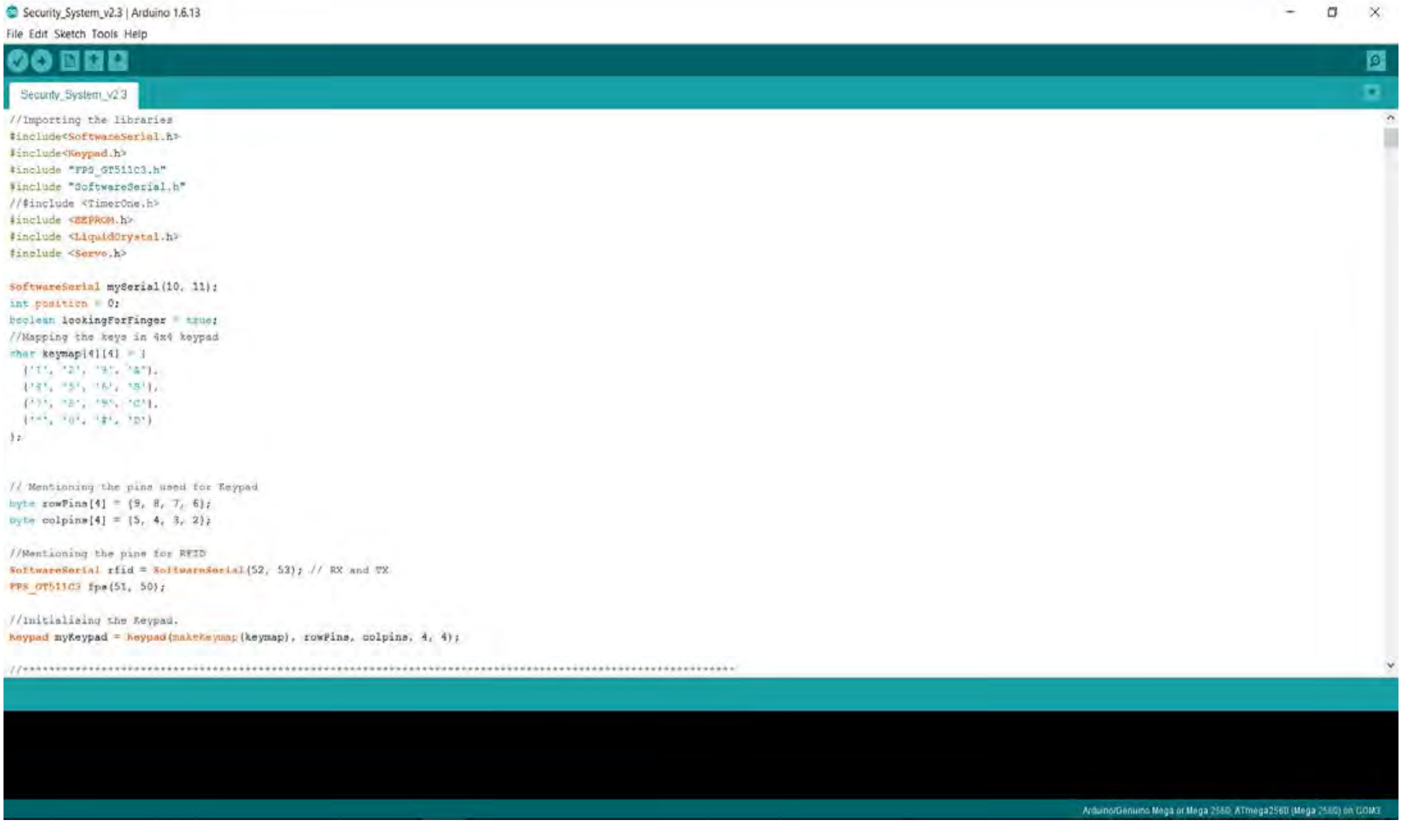
Step 11: press no iii // it will add admin

Step 12: press no iv // exit from the step 6

Step 13: Go to step 11

5.2 Arduino IDE:

One of the major components of this security and safety system is the software coding in order to make the system fully functional. The core of the system is the Arduino Mega development board which is programmed using Arduino IDE version 1.6.13. The programming language supported by the board is based on C and C++ languages and is coded using the Arduino development environment. The usage of programming in this project facilitates effective communications and fast processing between hardware components that are connected to the microcontroller.


Security_System_v2.3 | Arduino 1.6.13
File Edit Sketch Tools Help
Security_System_v2.3
Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) on COM3

```
//Importing the libraries
#include<SoftwareSerial.h>
#include<Keypad.h>
#include "FPS_GT511C3.h"
#include "SoftwareSerial.h"
//#include <TimerOne.h>
#include <EEPROM.h>
#include <LiquidCrystal.h>
#include <Servo.h>

SoftwareSerial mySerial(10, 11);
int position = 0;
boolean lookingForFinger = true;
//Mapping the keys in 4x4 keypad
char keymap[4][4] = {
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'}
};


// Mentioning the pins used for Keypad
byte rowPins[4] = {9, 8, 7, 6};
byte colpins[4] = {5, 4, 3, 2};

//Mentioning the pins for RFID
SoftwareSerial rfid = SoftwareSerial(52, 53); // RX and TX
FPS_GT511C3 fps(51, 50);

//Initialising the Keypad.
Keypad myKeypad = Keypad(makeKeymap(keymap), rowPins, colpins, 4, 4);

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

Figure 5.2 Arduino IDE Screenshot

Arduino IDE is written in Java and is a cross-platform application. It is designed to help programmers to configure the Arduino microcontrollers to their preference. Arduino programs are written in C or C++ but require two functions, setup() and loop(), to make a runnable cyclic executive program. The setup() function initializes all the settings and runs once at the start of the program. The loop() function executes the microcontrollers main job and is called repeatedly until the board powers off.

5.3 Codes and Libraries:

Programming also enables the microcontroller to learn about different devices that are attached to it. For each of the different modules used for the architecture of the system, different software libraries were used.

A software library is a suite of data and programming code that is used to develop software programs and applications. It is designed to assist both the programmer and the programming language compiler in building and executing software.

A software library generally consists of pre-written code, classes, procedures, scripts, configuration data and more. Typically, a developer might manually add a software library to a program to achieve more functionality or to automate a process without writing code for it. For example, when developing a mathematical program or application, a developer may add a mathematics software library to the program to eliminate the need for writing complex functions. All of the available functions within a software library can just be called/used within the program body without defining them explicitly. Similarly, a compiler might automatically add a related software library to a program on run time.

In other words, libraries are a collection of code that makes it easy to connect to a variety of modules or devices such as a sensor, display etc. For example, the built-in LiquidCrystal library makes it easy to talk to character LCD displays.

Admin register code:

```
#include <EEPROM.h>


/** the current address in the EEPROM (i.e. which byte we're going to write to next) **/



void setup() {
  /** Empty setup. **/
  for(int i=0; i<10; i++){
    EEPROM.write(i, 49); //49 is the ascii value for '1'. So, index 0-4 = 11111 (PIN) and
index 5-9=11111 (Password) which is the admin PIN and Password
  }
}


void loop() {
}
```

Face detection sms code:

```
#include <SoftwareSerial.h>
SoftwareSerial mySerial(10, 11);

int GPIOpin = 7;     // the number of the pushbutton pin
const int ledPin =  9;      // the number of the LED pin

// variables will change:
int GPIOstate = 0;         // variable for reading the pushbutton status

void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(GPIOpin, INPUT);
  mySerial.begin(9600);   // Setting the baud rate of GSM Module
  delay(1000);
}

void loop() {
  // read the state of the pushbutton value:
  GPIOstate = digitalRead(GPIOpin);
  // check if the pushbutton is pressed.
  // if it is, the GPIOstate is HIGH:
  if (GPIOstate == HIGH) {
    // turn LED on:
    digitalWrite(ledPin, HIGH);
    delay(5000);
    digitalWrite(ledPin, LOW);
    SendMessage();
  }
  if (mySerial.available() > 0)
    Serial.write(mySerial.read()); //sudo nano/etc/profile

}
void SendMessage()
{
  mySerial.println("AT+CMGF=1");    //Sets the GSM Module in Text Mode
  delay(1000);  // Delay of 1000 milli seconds or 1 second
  Serial.println(Serial.read());
  mySerial.println("AT+CMGS=\"+8801670669566\""); // Replace x with mobile number
  delay(1000);

  mySerial.println("SECURITY THREAT: INTRUDER DETECTED");// The SMS text you want to send

  delay(1000);

  mySerial.println((char)26);// ASCII code of CTRL+Z

  delay(1000);

}
```

Fingerprint blink test code

```
#include "FPS_GT511C3.h"
#include "SoftwareSerial.h"

// Hardware setup - FPS connected to:
//    digital pin 4(arduino rx, fps tx)
//    digital pin 5(arduino tx - 560ohm resistor fps tx - 1000ohm resistor - ground)
//    this brings the 5v tx line down to about 3.2v so we dont fry our fps

FPS_GT511C3 fps(11,10);

void setup()
{
  Serial.begin(9600);
  fps.UseSerialDebug = true; // so you can see the messages in the serial debug screen
  fps.Open();
}


void loop()
{
  // FPS Blink LED Test
  fps.SetLED(true); // turn on the LED inside the fps
  delay(1000);
  fps.SetLED(false);// turn off the LED inside the fps
  delay(1000);
}
```

Fingerprint check verified id code

```
#include "FPS_GT511C3.h"
#include "SoftwareSerial.h"
#include <LiquidCrystal.h>

// Hardware setup - FPS connected to:
//	digital pin 4(arduino rx, fps tx)
//	digital pin 5(arduino tx - 560ohm resistor fps tx - 1000ohm resistor - ground)
//		this brings the 5v tx line down to about 3.2v so we dont fry our fps
```

```
FPS_GT511C3 fps(51, 50);
LiquidCrystal lcd(33, 35, 37, 39, 41, 43);
void setup()
{
  Serial.begin(9600);
  delay(100);
  fps.Open();
  lcd.begin(16, 2);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("FPS Check ID");
  delay(2000);
  fps.SetLED(true);
}

void loop()
{

  // Identify fingerprint test
  if (fps.IsPressFinger())
  {
    fps.CaptureFinger(false);
    int id = fps.Identify1_N();
    if (id < 200)
    {

      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print("Verified ID");
      lcd.print(id);
    }
    else
    {
      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print("Finger not found");
    }
  }
  else
  {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Please press finger");
  }
  delay(100);
}
```

Fingerprint enroll code

```
#include "FPS_GT511C3.h"

#include "SoftwareSerial.h"

#include <LiquidCrystal.h>




//FPS connected to pin 4 and 5 - see previous schemas

FPS_GT511C3 fps(51, 50);

LiquidCrystal lcd(33, 35, 37, 39, 41, 43);

void setup()

{

  Serial.begin(9600);

  //display messages on the classical serial teminal - DEBUG

  fps.UseSerialDebug = true;

  fps.Open();

  //call Enroll to add fingerprint

  enroll();

}




void enroll()

{

  // get the first available id for new finger print
```

```
int enrollid = 0;
bool usedid = true;
while (usedid == true)
{
  usedid = fps.CheckEnrolled(enrollid);
  if (usedid == true) enrollid++;
}

//enrollment start here with the first free id
fps.EnrollStart(enrollid);

lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Press finger #");
lcd.print(enrollid);

// ***** FIRST MEASURE *****
// wait for finger press
while (fps.IsPressFinger() == false) delay(100);
bool bret = fps.CaptureFinger(true);
int iret = 0;
if (bret != false)
{
```

```
//has a finger print captured
lcd.clear();
lcd.setCursor(0, 0);
lcd.println("Remove finger");
// Enroll step 1
fps.Enroll1();
//wait for finger removed
while (fps.IsPressFinger() == true) delay(100);

// ***** SECOND MEASURE *****
// Now we need to check the finger print
// wait for finger press
lcd.clear();
lcd.setCursor(0, 0);
lcd.println("Press same finger again");
while (fps.IsPressFinger() == false) delay(100);
bret = fps.CaptureFinger(true);
if (bret != false)
{
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.println("Remove finger");
  //enroll step 2
```

```
fps.Enroll2();
//wait for finger removed
while (fps.IsPressFinger() == true) delay(100);
// ***** THIRD MEASURE *****
//Check Again the finger print
lcd.clear();
lcd.setCursor(0, 0);
lcd.println("Press same finger yet again");
while (fps.IsPressFinger() == false) delay(100);
bret = fps.CaptureFinger(true);
if (bret != false)
{
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.println("Remove finger");
  iret = fps.Enroll3();
  if (iret == 0)
  {
    //*** SUCCESS third measure are the same -> NOW finger print is stored
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.println("Enrolling Successfull");
  }
  else
  {
    //*** FAIL For some reason -> NOTHING STORED
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Enrolling Failed with error code:");
    lcd.println(iret);
  }
}

else
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.println("Failed to capture third finger");
}

else
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.println("Failed to capture second finger");
}

else
```

```
    lcd.clear();

    lcd.setCursor(0, 0);

    lcd.println("Failed to capture first finger");

  }


  //loop is useless here

  void loop()

  {

    delay(100000);

  }
```

Fingerprint reset code

```
#include <FPS_GT511C3.h>


#include <LiquidCrystal.h>


#include <EEPROM.h>

FPS_GT511C3 fps(50,51);

LiquidCrystal lcd (25, 27, 29, 31, 33, 35);

void setup() {

  // initialize the LED pin as an output.

 // pinMode(13, OUTPUT);

fps.DeleteAll();

fps.SetLED(true);
```

```
lcd.begin(20,4);
lcd.clear();
  lcd.setCursor(0,0);
 lcd.print("Deleting Prints...");

  /***
    Iterate through each byte of the EEPROM storage.

    Larger AVR processors have larger EEPROM sizes, E.g:
    - Arduno Duemilanove: 512b EEPROM storage.
    - Arduino Uno:        1kb EEPROM storage.
    - Arduino Mega:       4kb EEPROM storage.

    Rather than hard-coding the length, you should use the pre-provided length function.
    This will make your code portable to all AVR processors.
  ***/

  for (int i = 0 ; i < 10 ; i++) {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print((char)EEPROM.read(i));
    delay(2000);
  }
  // turn the LED on when we're done
  lcd.clear();
  lcd.setCursor(0,0);
 lcd.print("Finished");
 fps.SetLED(false);
}

void loop() {
```

flame detector code

```
#include <SoftwareSerial.h>



SoftwareSerial mySerial(10, 11);
int sensorPin = A0; // select the input pin for the LDR
int sensorValue = 0; // variable to store the value coming from the sensor
int led = 22; // Output pin for LED

void setup() {

  pinMode(led, OUTPUT);
  mySerial.begin(9600);   // Setting the baud rate of GSM Module
  delay(1000);
  Serial.begin(9600);
}
void SendMessage()
{
  mySerial.println("AT+CMGF=1");    //Sets the GSM Module in Text Mode
  delay(1000);  // Delay of 1000 milli seconds or 1 second
  Serial.println(Serial.read());
  mySerial.println("AT+CMGS=\"+8801672332231\""); // Replace x with mobile number
  delay(1000);
  mySerial.println("Fire Detected");// The SMS text you want to send
  delay(1000);
  mySerial.println((char)26);// ASCII code of CTRL+Z
  delay(1000);
}

void loop() {
  sensorValue = analogRead(sensorPin);
Serial.println(sensorValue);
  if (sensorValue < 100) {
    digitalWrite(led, HIGH);
    SendMessage();
    delay(1000);
    digitalWrite(led, LOW);
  }
```

GSM send sms code

```
#include <SoftwareSerial.h>


SoftwareSerial mySerial(10, 11);


void setup()
{
  mySerial.begin(9600);   // Setting the baud rate of GSM Module
  Serial.begin(9600);    // Setting the baud rate of Serial Monitor (Arduino)
  delay(1000);
}



void loop()
{
  if (Serial.available() > 0)
    Serial.println(Serial.available());
  switch (Serial.read())
  {
    case 's':
      SendMessage();
      break;
    case 'r':
      RecieveMessage();
      break;
  }

  if (mySerial.available() > 0)
    Serial.write(mySerial.read());
}


void SendMessage()
{
  mySerial.println("AT+CMGF=1");    //Sets the GSM Module in Text Mode
  delay(1000);  // Delay of 1000 milli seconds or 1 second
  Serial.println(Serial.read());
  mySerial.println("AT+CMGS=\"+8801670669566\""); // Replace x with mobile number
  delay(1000);
  mySerial.println("0011 -CTG1001- Approaching Uttara Crossing");// The SMS text you want
to send
  delay(1000);
  mySerial.println((char)26);// ASCII code of CTRL+Z
  delay(1000);
```

```
}



void RecieveMessage()
{
  mySerial.println("AT+CNMI=2,2,0,0,0"); // AT Command to recieve a live SMS
  Serial.println(mySerial.println("AT+CNMI=2,2,0,0,0"));
  delay(1000);
}
```

Flame detector code

```
#include <SoftwareSerial.h>



SoftwareSerial mySerial(10, 11);
int sensorPin = A0; // select the input pin for the LDR
int sensorValue = 0; // variable to store the value coming from the sensor
int led = 22; // Output pin for LED

void setup() {
  pinMode(led, OUTPUT);

  mySerial.begin(9600);   // Setting the baud rate of GSM Module

  delay(1000);

  Serial.begin(9600);

}

void SendMessage()

{

  mySerial.println("AT+CMGF=1");    //Sets the GSM Module in Text Mode

  delay(1000);  // Delay of 1000 milli seconds or 1 second

  Serial.println(Serial.read());

  mySerial.println("AT+CMGS=\"+8801672332231\""); // Replace x with mobile number

  delay(1000);

  mySerial.println("Fire Detected");// The SMS text you want to send

  delay(1000);

  mySerial.println((char)26);// ASCII code of CTRL+Z

  delay(1000);

}


void loop() {

  sensorValue = analogRead(sensorPin);
```

```
Serial.println(sensorValue);
  if (sensorValue < 100) {
    digitalWrite(led, HIGH);
    SendMessage();
    delay(1000);
    digitalWrite(led, LOW);
  }
}
```

RFID Attendance

```
// Libraries
#include <SPI.h>
#include <RFID.h>
#include "pitches.h"
#include <Ethernet.h>
#include <LiquidCrystal.h>

byte mac[] = { 0x90, 0xA2, 0xDA, 0x0D, 0x96, 0xB4 };
byte ip[] = { 192,168,1,110 }; // Direction ip local
char server[]={"example.com"};

EthernetClient client;

// RFID definition
RFID rfid(10,5);
byte USER[5] = {124,11,145,50,212}; // Define serial here
//Put here the another allowed cards
```

```
//LCD address and type declaration
LiquidCrystal lcd(9, 8, 7, 4, 3, 2);
byte serNum[5];
byte data[5];

void setup(){
  Serial.begin(9600);
  Ethernet.begin(mac, ip); //Initiation ethernet shield
  delay(1000); // Waiting 1 second after initializing
  lcd.begin(16, 2);
  lcd.print(Ethernet.localIP());
  Serial.print("My IP address: ");
  Serial.println(Ethernet.localIP());

  Serial.begin(9600); // Serial communication initialization
  SPI.begin(); // SPI communication initialization
  rfid.init(); // RFID module initialization
  lcd.setCursor(0,1);
  lcd.print("Waiting for Card");
  Serial.println("Waiting for Card");
}

void loop(){
  boolean USER_card = true;  // Here you can create a variable for each user
  //Put here the another variable for user

  if (rfid.isCard()){ // valid card found
    if (rfid.readCardSerial()){ // reads the card
      data[0] = rfid.serNum[0]; // stores the serial number
      data[1] = rfid.serNum[1];
      data[2] = rfid.serNum[2];
      data[3] = rfid.serNum[3];
      data[4] = rfid.serNum[4];
    }

  for(int i=0; i<5; i++){
    if(data[i] != USER[i]) USER_card = false; // if it is not a valid card, put false to this user
    // Here you can check the another allowed users, just put lines like above with the user name
  }
    if (client.connect(server, 80)) {

    if (USER_card){
    Serial.println("Connection Successfull");
    client.print("GET http://example.com/arduino.php?v=");

client.print(data[0]);client.print(data[1]);client.print(data[2]);client.print(data[3]);client.print(data
[4]);
    client.println(" HTTP/1.0"); client.println("User-Agent: Arduino 1.0");
    client.println(); client.stop(); lcd.clear();
    lcd.print("USER!");
    Serial.println("Hello USER!"); // print a message
    delay(1000);
  }

  /*
  // another cards analysis put many blocks like this as many user you have
  else if (USER2_card){
    Serial.println("Connection Successfull");
    client.print("GET http://example.com/arduino.php?v=");
client.print(data[0]);client.print(data[1]);client.print(data[2]);client.print(data[3]);client.print(data
[4]);
    client.println(" HTTP/1.0"); client.println("User-Agent: Arduino 1.0");
    client.println(); client.stop(); lcd.clear();
    lcd.print("USER2!");
    Serial.println("Hello USER2!"); // print a message
    delay(1000);
  }
  */

  else{ // if a card is not recognized
    lcd.clear();
    lcd.print("Card not");
    lcd.setCursor(0,1);
    lcd.print("recognized!");
    Serial.println("Card not recognized!"); // print a message
     delay(1000);
  }
  if (USER_card){// add another user using an logical or condition ||
    // Welcome messages and access permission
   lcd.setCursor(0,1);
   lcd.print("Access Granted!");
    Serial.println("Access Granted!... Welcome!"); // print a message
    delay(1000);
  }
  Serial.println();
  lcd.clear();
  lcd.print("    Welcome!");
  lcd.setCursor(0,1);
```

```
 Serial.println("Waiting for Card");
 lcd.print("Waiting for Card");
 client.stop();
 delay(1000);
 }
 else{
  Serial.println("Connection Unuccessfull");
  lcd.clear();
  lcd.print("Connection");
  lcd.setCursor(0,1);
  lcd.print("Unuccessful");
  delay(1000);
  lcd.clear();
  lcd.print("Try Again!");
  client.stop();
  delay(1000);
   }
 }
}
```

APPLICATION CODE

```
'use strict';

import React, { PropTypes, Component } from 'react';

import {
    AsyncStorage,
    Linking,
    Platform,
    Alert
} from 'react-native';

var _ = require('lodash');
var config = __DEV__ ? require('./config/config-dev.json') :
require('./config/config.json');
```

```
import TopBar from "./components/topbar";
import UserService from "./services/user";
import Storage from "./services/storage";

import PushNotification from 'react-native-push-notification';
import ReactNativeUA from 'react-native-ua';
var DeviceInfo = require('react-native-device-info');

var appChannelId = '';
const PUSH_STORAGE_KEY = "PushSettings";

export default class SmartHomeCommon extends Component {
    state = {
        homePresses: 0,
        missionPresses: 0,
        notificationPresses: 0,
        accountPresses: 0,
        selectedTab: 'home',
        notificationCount: 0,
        userToken: '',
        topBarWithBackButton: false,
        topBarWithBackButtonCenter: '',
        topBarWithBackButtonRight: [],
        backBtnAction: {},
        settingsVisible: false,
        showMissionDetails: false,
        showMissionId: '',
        showThreadDetails: false,
        threadId: '',
        threadMissionId: '',
        showResetPassModal: false,
        resetUserToken: ''
    };

    constructor (props) {
        super(props);

        var storedPushSettings = Storage.getItem(PUSH_STORAGE_KEY);
        var pushOnOff = true;

        if (storedPushSettings != null) {
            //storedPushSettings.soundSetting;
            //storedPushSettings.vibrationSetting;
            pushOnOff = storedPushSettings.pushNotificationSetting;
        }

        // Check if user enabled notifications
        ReactNativeUA.are_notifications_enabled().then(enabled => {
            if (!enabled && pushOnOff) {
                ReactNativeUA.enable_notification();
            }
        });
        ReactNativeUA.enable_action_url();
        ReactNativeUA.handle_background_notification();
        ReactNativeUA.set_quiet_time_enabled(false);
        ReactNativeUA.add_tag(Platform.OS);

    }

    componentWillMount() {
        this.initialize();

        ReactNativeUA.on_notification((notification) => {
            if (Platform.OS == 'ios') {
                PushNotification.localNotification({
                    title: notification.alert.title,
                    autoCancel: false,
                    bigText: notification.alert.body,
                    vibrate: true,
                    message: notification.alert.body
                });
            }

            this.manageParsedNotification(notification.data);
        });
    }

    componentDidMount() {
        Linking.addEventListener('url', this._handleOpenURL.bind(this));
    }

    componentWillUnMount () {
        Linking.removeEventListener('url', this._handleOpenURL.bind(this));
    }

    async setNotificationCount () {
        try {
            if (UserService.isLoggedIn()) {
                var response = await
NotificationService.fetchUnreadNotifications();
                var unreads  = response.content.filter((notification) =>
notification.status != 'READ');

                this.setState({
                    notificationCount: unreads.length
                }, () => {

PushNotification.setApplicationIconBadgeNumber(unreads.length);
                });
            } else {
                this.setState({
                    notificationCount: 0
                }, () => {
                    PushNotification.setApplicationIconBadgeNumber(0);
                });
            }
        } catch (error) {
            console.log(error);
            this.setState({
                notificationCount: 0
            }, () => {
                PushNotification.setApplicationIconBadgeNumber(0);
            });
        }
    }

    _handleOpenURL(event) {
        /** open designated page from deep link -- smarthome:// */

    }

    async initialize(tabItem) {
        this.setState({loggingIn: true});
        try {
            await UserService.initialize();
        } catch (error) {
            console.log("Unable to initialize UserService. However, this
should not be too much of a problem.", error);
        }
        finally {
            this.setState({loggingIn: false});
        }

        if (UserService.isLoggedIn()) {
            /*try {
                var appVersionInfo = await UserService.getAppVersionInfo();
                var latest = true;

                if (Platform.OS == 'android') {
                    latest = (appVersionInfo.version.android <=
DeviceInfo.getVersion());
                } else if (Platform.OS == 'ios') {
                    latest = (appVersionInfo.version.ios <=
DeviceInfo.getBuildNumber());
                }

                if (!latest) {
                    Alert.alert(
                        'New Version of SmartHome',
                        'A new version of SmartHome is available. Please
update for getting better and problem-free experience!',
                        [
                            {
                                text: 'Later',
                                style: 'cancel'
                            },
                            {
                                text: 'Upgrade',
                                onPress: async () => {
                                    if (Platform.OS == 'ios') {
                                        Linking.openURL("itms-
apps://itunes.apple.com/no/app/smarthome/id90909090?mt=8").catch(err =>
console.error('An error occurred opening store', err));
                                    } else {

Linking.openURL("market://details?id=com.smarthome.apps.android").catch(err
=> console.error('An error occurred opening store', err));
                                    }
                                }
                            }
                        ]
                    );
                }

            } catch (error) {
                console.log('version check error', error);
            }*/

            var token = UserService.profile.token;
            console.log('common js: ', token);
            if (tabItem) {
                this.setState(
                    {selectedTab: tabItem},
                    () => {
                        this.setState({userToken: token}, () =>
{this.configurePushNotification(UserService.profile.id);} );
                    });
            } else {
                this.setState({userToken: token}, () =>
{this.configurePushNotification(UserService.profile.id);} );
            }

            this.setNotificationCount();
        }
    }

    manipulateTopBar (backBtn, title, rightActionContent, backBtnAction) {
        this.setState(
            {
                topBarWithBackButton: backBtn,
                topBarWithBackButtonCenter: backBtn ? title : '',
                topBarWithBackButtonRight: backBtn ? rightActionContent : [],
                backBtnAction: backBtn ? backBtnAction : {},
                homePresses: 0,
                missionPresses: 0,
                notificationPresses: 0,
                accountPresses: 0,
            });
    }

    openSettings () {
        this.setState({settingsVisible: true});
    }
    closeSettings () {
        this.setState({settingsVisible: false});
    }

    configurePushNotification (userId) {
        //ReactNativeUA.set_named_user_id(userId);
    }

    async manageParsedNotification (notificationData) {
        var parsedNotification = await
NotificationService.parseNotification(notificationData);

        switch (parsedNotification.modalType) {
            case 'mission':
                this.openMissionModal(parsedNotification.payload.missionId);
                break;

            case 'thread':
                var threadId = parsedNotification.payload.threadId;
                var threadMissionId = parsedNotification.payload.missionId;

                this.openThreadModal(threadMissionId, threadId);
                break;

            case 'notification':
                  this.setState({
                      selectedTab: 'notification'
                  });
                  break;
        }
    }

    openMissionModal (missionId){
        this.setState({
            showMissionDetails: true,
            showMissionId: missionId
        });
    }

    closeMissionDetails (){
        this.setState({
            showMissionDetails: false,
            showMissionId: ''
        });
    }

    openThreadModal (missionId, threadId) {
       this.setState({
          showThreadDetails: true,
          threadId: threadId,
          threadMissionId: missionId
       });
    }
    closeThreadModal () {
       this.setState({
          showThreadDetails: false,
          threadId: '',
          threadMissionId: ''
       });
    }

    openResetPassModal (userId, token) {
        this.setState({
           showResetPassModal: true,
           resetUserToken: token
        });
    }

    closeResetPassModal () {
        this.setState({
           showResetPassModal: false,
           resetUserToken: ''
        });
    }
}
```

5.4 Results and Discussion

We faced many problems in running the python code also in the face detection steps as the device could not recognize who is the main admin. On the other hand in the very beginning raspberry pi could not take all the command at once, as time elapse it started to show results. The message sent through GSM was failing several times and giving errors but after trying in the end we accomplish. Moreover we do not have any ftp server so we could not able to merge the app with the server at this stage. In future we would try to connect the app with ftp server as the client demands.

Chapter 6
Conclusion and Future Works

6.1 Conclusion

We build a smart security system by using the most reliable and efficient components as well as minimize the overall cost of the equipment in order to achieve better output of the system. The overall cost of the designed smart security system is also relatively low compared to available smart system in the market. Beside that we faced many problems to do this project which we tried to overcome. We also have some future plan to add more features in this project

6.1.1 Limitation

1. By using Solenoid door lock someone can only unlock the door but have to open the door manually.
2. The total system is powered by electricity. When a country wide power failure happens, the system will work on the backup batteries dedicated only to the system, but in case the backup power fails it will restrict entry to the building as door lock system and other components of the system will not work.
3. The face detection system implemented using the raspberry pi saves several images of the same person each time a face is detected even of the admin himself or herself as the system is unable to recognize a specific face
4. Raspberry Pi face detection library available is not that much effective in detecting faces properly

6.2 Future implementation

In current system, the door is to be manually pushed open and close after the solenoid magnetic lock is unlocked by authentication. In future, the door will be opened and closed automatically by use of mechanical servos after an authentication is made.

In the face detection system, the video can be improved by using a camera of higher resolution and the detection system can be improved by using a more upgraded face library. By using the upgraded library, a specific face can be detected and stored in the system which will enable us to avoid the recognition of a known face over and over again, and only store image and alert for unknown or unexpected faces. The library will also enable the system to detect the side faces which will be more efficient in terms of security.

The flame sensing system can be implemented using better sensors with a wider range and precision so that the sensing is more effective and precise and also ensure less number of sensor to be used while implementing it on large area or room.

In future, the face recognition system can be used to restrict the door open switch to be used by any unregistered person and the password will be automatically reset and the admin will be notified immediately.

A mobile application can also be developed for the system. The system will be connected to the internet. This way, the admin will receive notifications through the mobile application.

References

https://circuitdigest.com/microcontroller-projects/raspberry-pi-iot-intruder-alert-system

https://people.kth.se/~maguire/DEGREE-PROJECT-REPORTS/131201-Rafid_Karim_and_Haidara_Al-Fakhri-with-cover.pdf

https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0ahUKEwjTr42K54nTAhVFsY8KHW_KBRAQFggcMAA&url=http%3A%2F%2Fwww.rndrepository.com%2Fsmartgriduwi%2Ffiles%2FPaper%2520in%2520IJCA.pdf&usg=AFQjCNEgOB-3eHDvAjr2pzxR2L5UZV0-uw&sig2=05SepzzrzFwGhpsvchB-xw&bvm=bv.151426398,d.c2I

https://www.safety.com/blog/50-best-smart-home-security-products/

https://www.sparkfun.com/products/11792

https://www.lelong.com.my/vero-board-board-pcb-strip-3x6-hwaleetrading-170086735-2017-11-Sale-P.htm

https://www.arduino.cc/en/Main/arduinoBoardMega2560 [43]

https://www.sparkfun.com/products/11792 [65]

https://en.wikipedia.org/wiki/Radio-frequency_identification [47]

http://www.rhydolabz.com/documents/gps_gsm/sim900_rs232_gsm_modem_opn.pdf [8]

https://www.mpja.com/download/31227sc.pdf [10]

https://www.raspberrypi.org/products/camera-module/ [21]

https://en.wikipedia.org/wiki/Flame_detector [40]