

KARNATAK LAW SOCIETY’S

GOGTE INSTITUTE OF TECHNOLOGY

UDYAMBAG, BELAGAVI-590008

(An Autonomous Institution under Visvesvaraya Technological University, Belagavi)

### (APPROVED BY AICTE, NEW DELHI)

Department of Information Science

# A Project Report on

**“SMART DOOR LOCK AND LIGHTING SYSTEM USING IOT”**

*Submitted in partial fulfillment of the requirement for the award of the degree of*

**BACHELOR OF ENGINEERING IN**

**INFORMATION SCIENCE ENGINEERING**

*Submitted by*

|  |  |
| --- | --- |
| **MOHAMMED MAAZ GHEEWALE** | **2GI18IS407** |

# Under the Guidance of

**Prof.**

# 2020 - 2021

KARNATAK LAW SOCIETY’S

## GOGTE INSTITUTE OF TECHNOLOGY

UDYAMBAG, BELAGAVI-590008

(An Autonomous Institution under Visvesvaraya Technological University, Belagavi)

**(APPROVED BY AICTE, NEW DELHI)**

## Department of Information Science Engineering



CERTIFICATE

Certified that the project entitled

### “SMART DOOR LOCK AND LIGHTING SYSTEM USING IOT”.

Carried out by **Ms. Mohammed Maaz Gheewale USN 2GI18IS407**

student of KLS Gogte Institute of Technology, Belagavi ,can be considered as a bonafide work for partial fulfillment for the award of **Bachelor of Engineering** in **Information Science Engineering** of the Visvesvaraya Technological University, Belagavi during the year **2020- 2021** It is certified that all corrections/suggestions indicated have been incorporated in the report. The project report has been approved as it satisfies the academic requirements prescribed for the said Degree.

|  |  |  |
| --- | --- | --- |
| **Guide** | **HOD** | **Principal** |
| (Prof. )  **Date:** | (Dr ) | (Dr. J.K Kittur) |
|  | **Final Viva-Voce** |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Name of the examiners** | **Date of Viva -voce** | **Signature** |
| **1.** |  |  |  |
| **2.** |  |  |  |

# DECLARATION BY THE STUDENT(S)

I, **Mr. Mohammed Maaz Gheewale** hereby declare that the project report entitled **“ Smart Door Lock and Lighting System using IOT ”** submitted by us to KLS Gogte Institute of Technology, Belagavi, in partial fulfillment of the Degree of **Bachelor of Engineering** in **Information Science** is a record of the project carried out at **KLS Gogte Institute of Technology, Belagavi** . This report is for the academic purpose.

We further declare that the report has not been submitted and will not be submitted, either in part or full, to any other institution and University for the award of any diploma or degree.

|  |  |  |
| --- | --- | --- |
| Name of the student | USN | Signature |
| Mohammed Maaz Gheewale | 2GI18IS407 |  |

Place: Belagavi , Karnataka Date:

# ACKNOWLEDGEMENT

It is a great privilege for us to express our profound gratitude to our guide **Prof………** , **Information Science** Engineering**, KLS Gogte Institute of Technology, Belagavi**, for her constant guidance, valuable suggestions, supervision and inspiration throughout the course work without which it would have been difficult to complete the work within scheduled time.

We would like to express our gratitude towards all the **faculty members** for his/her kind co- operation and encouragement which helped us in completion of this project. We are also indebted to the Head of the Department, **Dr……..**, Information Science Engineering, KLS Gogte Institute of Technology, for permitting us to pursue the project.

We would like to take this opportunity to thank all the respected teachers of this department for being a perennial source of inspiration and showing the right path at the time of necessity.

Mohammed Maaz Gheewale

(2GI18IS407)

# ABSTRACT

In the proposed approach, a smart door lock and lighting system using IOT for smart home is presented with the additional add on feature of facial recognition. A smart door lock system is a system which uses digital password for opening and closing the door. With the advancement of modern technologies areas related to robotics and computer vision, real time image processing has become a major technology under consideration. So here a try has been made for a novel approach for capturing images from the Pi Camera in real time environment and process them as we are required. Here in this project depicts a basic and simple equipment execution of face location framework utilizing Raspberry Pi, which itself is a minicomputer of a small estimate and is of a low cost. The framework is modified utilizing Python programming language. The destinations of the face recognition are to recognize appearances and its spatial area in any pictures or recordings

.The proposed framework distinguishes the faces present in a grey scale and colored image. Here in this project the idea of identification has been built up by composing distinguishable code for dataset generator, trainer and indicator. Effectiveness of the framework is examined by ascertaining the Face recognition rate for every one of the database. At last the data that will be shown alongside recognized photograph has been put away on database. This concept has a higher scope on security and surveillance projects and various operation.

# TABLE OF CONTENTS

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Content** | **Page No.** |
|  | i. | Declaration | i |
|  | ii. | Acknowledgement | ii |
|  | iii. | Abstract | iii |
|  | iv. | Table of contents | iv |
|  | v. | List of Tables | vi |
|  | vi. | List of Figures | vi |
|  | vii. | List of Abbreviations | vii |
|  |  |  |  |
| **1.** |  | **INTRODUCTION** | 1 |
|  | 1.1 | Introduction to Smart Door Lock And Lighting System Using Internet Of Things |  |
|  | 1.2 | Objectives |  |
|  | 1.3 | Methodology |  |
| **2.** |  | **LITERATURE REVIEW AND RESEARCH GAP** |  |
|  | 2.1 | Literature Review |  |
| **3.** |  | **PRIOR ART SEARCH** |  |
|  | 3.1 | Related Patents |  |
| **4.** |  | **PROJECT DETAILS & FLOWCHARTS** |  |
|  | 4.1 | Design Components |  |
|  |  | 4.1.1 Raspberry Pi 0 W |  |
|  |  | 4.1.2 Raspberry Camera Module |  |
|  |  | 4.1.3 Four channel Relay |  |
|  |  | 4.1.4 Solenoid Lock |  |
|  |  | 4.1.5 4x3 Matrix Keypad |  |
|  | 4.2 | Facial Recognition Feature |  |
|  |  | 4.2.1 Installation of Raspbian on the Raspberry Pi |  |
|  |  | 4.2.2 Face recognition |  |
|  | 4.3 | Flowchart |  |
|  | 4.4 | Block Diagram of the Smart Door Lock and Lighting System Using Internet of Things |  |
|  |  | 4.4.1 Circuit Diagram |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | 4.4.2 Overview Image of the Project |  |
| **5.** |  | **RESULTS AND DISCUSSION** |  |
|  | 5.1 | Result |  |
| **6.** |  | **CONCLUSION AND SCOPE FOR FUTURE WORK** |  |
|  | 6.1 | Future Scope |  |
|  | 6.2 | Conclusion |  |
|  |  | **References** |  |
| **Appendix - A** |  | **Program** |  |
| **Appendix - B** |  | **Bill of materials** |  |

### LIST OF TABLES

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Title** | **Page No.** |
| 4.1 | Technical Details of Solenoid Lock |  |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Title ( Caption)** | **Page No.** |
| 4.1 | Raspberry Pi (Zero W) |  |
| 4.2 | Pin Diagram of Raspberry Pi Zero W |  |
| 4.3 | Raspberry Pi Camera Module |  |
| 4.4 | Four Channel Relay |  |
| 4.5 | Four Channel Relay In-built Connection |  |
| 4.6 | Four Channel Relay Connection with Raspberry Pi |  |
| 4.7 | Solenoid Lock |  |
| 4.8 | Measurement circuit of Solenoid Lock |  |
| 4.9 | 4 x 3 Matrix Keypad |  |
| 4.10 | Installing Raspbian on Raspberry Pi |  |
| 5.1 | Turning ON the bulb which is connected to Raspberry Pi  Zero |  |
| 5.2 | Turning OFF the bulb which is connected to Raspberry Pi  Zero |  |
| 5.3 | Door Locking |  |
| 5.4 | Door Unlocking |  |

**LIST OF ABBREVIATIONS**

|  |  |
| --- | --- |
| **Abbreviation** | **Description** |
| OS | Operating system |
| LAN | Local area network |
| HDMI | High- Definition Multimedia Interface |
| USB | Universal Serial Bus |
| CSI | Camera Serial Interface |
| GPIO | General Purpose Input Output |
| SOC | System on Chip |
| ARM | Advanced RISC Machines |
| RISC | Reduced Instruction Set Computer |
| CPU | Central Processing Unit |
| RAM | Random Access Memory |
| LE | Low Energy |
| RCA | Radio Corporation of America |
| PWM | Pulse Width Modulation |
| SPI | Serial Peripheral Interface |
| MOSI | Master out Slave in |
| MISO | Master in Slave out |
| SCLK | Serial Clock |
| CE | Chip Enable |
| NO | Normally Opened |
| NC | Normally Closed |
| LED | Light Emitting Diode |
| VCC | Voltage Common Collector |
| GND | Ground |
| GB | Giga Byte |
| PC | Personal Computer |
| RPS | Regulated Power Supply |

# CHAPTER 1 INTRODUCTION

### Introduction To Smart Door Lock And Lighting System Using Internet Of Things :

In the proposed approach, a smart door lock and lighting system using IOT for smart home is presented with the additional add on feature of facial recognition. A smart door lock system is a system which uses digital password for opening and closing the door. The door lock is the foremost and endmost thing people come across in entering and leaving their homes respectively, the home automation function in digital door lock system allows users to comfortably control and monitor home environment and situation all at once. It also allows users to remotely overlook the situation inside the house through World wide web or any other public network. A smart lighting is proposed which can be remotely controlled using Internet.

Nowadays, technology is an integral part of everyone's lives. It influences several facts of everyday life and allows improved social synergy, easy transportation, the capability of indulging in entertainment and media and helps in the advancement in medicine. Internet has played a pioneering role in providing immediate solutions for various problems and has given the ability and has connected all the remote places which has contributed to significant reduction in cost and also energy consumption. Home automation or intelligent home is defined as initiation of technology inside the home surroundings to provide ease and safety to its inhabitants. The technology of the Internet of Things is used to examine and execute home automation. GPRS, GSM, Bluetooth, Wi-Fi and cellular networks support remote data transferring and are used to enter abundant levels of acumen within the home. Home automation has the ability to greatly assist and improve the quality of life of older people. IOT also greatly contributes to supply management and observance with ease of control. The user can remotely control the gate, home appliances, etc. comfortably and conveniently anywhere and anytime.

The Internet of things (IoT) is known as connecting objects like cell phones, personal computer and other devices to the world wide web, which introduces a new era in the area

of communication, where objects communicate with each other without human intervention. Most of the equipment and gadgets are controlled and monitored to help and assist humans. Moreover, various wireless technologies assist in communicating with remote places which play a great role in the intelligence of house surroundings. IOT is a sophisticated network of nodes with the unique ability of exchanging data and knowledge wirelessly which enables communication between two objects thereby making them smart and omitting the need of humans for machine to machine communication.

### Objectives:

The main objective of this project is to provide remote access to door lock and lighting system. The obvious motivation for providing such a kind of remote access to door lock is to make homes much more secure and enable us to remotely unlock or lock door for guests etc.

Smart lighting: The remote lighting system allows user to remotely control lighting i.e. switching it ON / OFF and also vary the intensity of light. This ensures that electricity is never wasted even if the user forgets to switch off the lighting as it can be remotely switched OFF.

Door Locking: This project also ensures that the user need not worry about whether the door is left unlocked or not and hence ensures peace of mind for the user.

Facial Recognition: This is an additional add-on feature which is implemented in this project. This feature is enabled for the door locking and unlocking system, which also imbibes a sense of security. It allows the user to add the images which are only restricted to the house owners.

### Methodology:

The smart digital door lock is a system which is used to monitor and control several devices in the home. With the locking system we have also included lighting system and the facial recognition feature.

The locking and lighting system is the most essential part of our day to day lives since keeping our houses locked gives us a sense of security and we can also save up on the electricity bill time to time. We have tried to implement this project which does both the above mentioned things with an additional feature of facial recognition. We have used the Raspberry pi 0 module which acts as a bridge between the software and the hardware parts of the projects. We have also taken the help of an application called Blynk App to connect our module over the internet. So, we have created a few buttons on the app which acts like a switch for turning ON/OFF the lights and locking/unlocking the door from anywhere around the world over the internet.

So basically, these buttons created are connected to the internet through the Blynk App and also in a way connected to the physical relays which are synced with the Raspberry pi 0 module. The programming for lighting/locking system as well as the facial recognition is written on the Raspberry Pi 0 module. It controls camera , keypad and communication between server and all the important processes are done by this module. The control module" Raspberry pi" is the centre of the door lock system where all the process cycle is done. This module is the server for verification of the user id. Camera is connected to Raspberry pi0 for surveillance purposes. All the operations are done by the control module which includes sending notification to the owner, to check that the guest is authorized to enter the house.

# CHAPTER 2

**LITERATURE REVIEW AND RESEARCH GAP**

### 2.1: Literature Review:

D. Surie, O. Laguionie, T. Pederson, [1] described the use of wireless sensor networking of everyday objects. By correlating the sensor output of such everyday objects, the wireless sensor network (WSN) as a whole can potentially provide functionality that an individual everyday object cannot.

Sirsath N. S, Dhole P. S, Mohire N. P, Naik S. C & Ratnaparkhi N.S [2] developed a Home Automation system that employs the integration of multi-touch mobile devices, cloud networking, wireless communication, and power-line communication to provide the user with remote control of various lights and appliances within their home.

Agarwal, Nikhil and Nayak, Subramanya G. [3] proposed the construction of a micro- controller based automated Home Security System. The door lock is password protected with an LED based resistive screen input panel which operates by detecting difference in light intensity captured by the photo diode which is emitted by surrounding red LEDs and reflected by the finger. IR Laser sensors are used to detect any obstacle while monitoring the windows and doors at night or when away.

[D. Pavithra](https://ieeexplore.ieee.org/author/37085746346); [Ranjith Balakrishnan](https://ieeexplore.ieee.org/author/37952912900) [4] proposed an efficient implementation for IoT (Internet of Things) used for monitoring and controlling the home appliances via World Wide Web. Home automation system uses the portable devices as a user interface. They can communicate with home automation network through an Internet gateway, by means of low power communication protocols like Zigbee, Wi-Fi etc. This project aims at controlling home appliances via Smartphone using Wi-Fi as communication protocol and raspberry pi as server system.

M. Ibrahim, A. Elgamri, S. Babiker and A. Mohamed .,[5] proposed an approach to build a cost-effective standardized environmental monitoring device using the Raspberry-Pi (R- Pi) single-board computer. The system was designed using Python Programming language and can be controlled and accessed remotely through an Internet of Things platform. It takes information about the surrounding environment through sensors and uploads it directly to the internet, where it can be accessed anytime and anywhere through internet. Experimental results demonstrated that the system is able to accurately measure:

temperature, humidity, light level and concentrations of the carbon monoxide harmful air pollutant. It's also designed to detect earthquakes through an assembled seismic sensor.

R. R. Harmon, E. G. Castro-Leon and S. Bhide, [6] proposed the smart city concept which represents a compelling platform for IT-enabled service innovation. It offers a view of the city where service providers use information technologies to engage with citizens to create more effective urban organizations and systems that can improve the quality of life. The emerging Internet of Things (IoT) model is foundational to the development of smart cities. Integrated cloud-oriented architecture of networks, software, sensors, human interfaces, and data analytics are essential for value creation. IoT smart-connected products and the services they provision will become essential for the future development of smart cities.

J. Ye, Q. Xie, Y. Xiahou and C. Wang, [7] The central controller uses the feedback information from the household appliances to find out the user's habits, and then adjust the system to adapt to the user. This ability makes the new system more convenient and friendly, and also overcome the poor adaptive capacity and portability defects of the traditional smart home system.

# CHAPTER 3 PRIOR ART SEARCH

### 3.1: Related Patents:

1. INTERNET OF THINGS (IOT) SYSTEM AND METHOD FOR SMART HOTEL:

Publication No: CN105759625A Publication date: 29 March 2016

The Internet of things system in a kind of wisdom hotel and Internet of Things method

Technical field

The present invention relates to the Internet of things system in a kind of wisdom hotel, further relate to the Internet of Things method of the Internet of things system in a kind of wisdom hotel, belong to hotel management technical field.

Background technology

Wisdom hotel refers to: with hotel for platform, utilize comprehensive wiring technology, the network communications technology, security precautions technology, automatic control technology, the intelligent facility relevant with hotel is integrated on intelligent processor, build facility, comfortable, efficiently, energy-conservation wisdom Hospitality management system, thus reaching to reduce energy consumption in hotels, reduce hotel's cost, promote the final goal of hotel occupancy rate.

Hotel traditional at present is directly registered to shop by user or website subscribes to the way to manage that shop is registered, and there is the technical problem that formality of checking in is loaded down with trivial details, move in efficiency of service low, hotel management cost of human resources is higher It addition, the equipment in guest room also cannot remotely be controlled by the registered user moved in, hotel's Intelligent Service degree is on the low side.

1. SMART-HOME AUTOMATION SYSTEM THAT SUGGESTS OR AUTMATICALLY IMPLEMENTS SELECTED HOUSEHOLD POLICIES BASED ON SENSED OBSERVATIONS:

Publication No: US20160259308A1 Publication date: 30 October 2018

This disclosure relates to smart-home environments. In particular, this disclosure relates to generation and/or distribution of device-implementable occupant policies for smart-device environments.

This section is intended to introduce the reader to various aspects of art that may be related to various aspects of the present disclosure, which are described and/or claimed below. This discussion is believed to be helpful in providing the reader with background information to facilitate a better understanding of the various aspects of the present disclosure. Accordingly, it should be understood that these statements are to be read in this light, and not as admissions of prior art.

People interact with a number of different electronic devices on a daily basis. In a home setting, for example, a person may interact with smart thermostats, lighting systems, alarm systems, entertainment systems, and a variety of other electronic devices. Unfortunately, the usefulness of these devices often times limited to basic and/or particular pre-determined tasks associated with the device.

As society advances, households within the society may become increasingly diverse, having varied household norms, procedures, and rules. Unfortunately, because so-called smart devices have traditionally been designed with pre-determined tasks and/or functionalities, comparatively fewer advances have been made regarding using these devices in diverse or evolving households or in the context of diverse or evolving household norms, procedures, and rules.

1. HOME AUTOMATION SYSTEM: Publication No: US5621662A Publication date: 30 October 2018

A home automation system comprises a number of sub-systems for controlling various aspects of a house, such as a security sub-system, an HVAC sub-system, a lighting control sub-system, and an entertainment sub-system. The network comprises a host computer connected through a host interface to a plurality of nodes. The network is in a free form topology and employ asynchronous communication. The host computer polls each node on the network to determine system configuration and to perform a diagnostic check on the system. The messages that are transmitted between the nodes are comprised of a source address, a destination address that uniquely identifies the location of each piece of hardware on the system, a message type field, and a data length segment. Each hardware device has a mirror image software object in the host computer to which messages are directed. The user interfaces for the various sub-systems share a common interfacing method whereby use of the system is greatly simplified.

This invention generally relates to a home automation system and, more particularly, to an interface between a host computer and a network, to a watchdog timer, to a method of polling nodes, to a software message scheme, to a common method of controlling sub-systems in the home automation system, and to a button keypad assembly.

Additionally, this invention relates to a temperature sensor for accurately measuring ambient temperature. More specifically, it relates to an apparatus for providing a temperature- indicating signal to a home automation system which maintains a desired temperature in a closed environment.

1. INTELLIGENT DOOR LOCK SYSTEM, AND INTELLIGENT DOOR LOCK AND INTELLIGENT ALARM DOOR:

Publication No: WO2016019590A1 Publication date: 11 Feb 2016

An intelligent door lock system, and an intelligent door lock and an intelligent alarm door based on the intelligent door lock system. The intelligent door lock system comprises a lock core module, a core control module, an information collection module and an alarm module, wherein the lock core module comprises a lock core and a lock core control device, and a pre-set crisis recognition signal is stored in the core control module; the information collection module collects information around a door lock, transfers the

information to the core control module, and conducts analysis and recognition on the information, so as to judge whether the crisis recognition signal appears in the information, and gives an alarm through the alarm module; and the lock core control device independently or the lock core control device and the core control module jointly controls the unlocking of the lock core. When the system unlocks the door lock, the door lock can intelligently recognize a crisis recognition signal secretly transmitted by a hijacked unlocker or an illegal opening/breaking recognition signal, and give an alarm.

The conventional lock system is composed of a key and a lock cylinder. The surface of the key body of the key forms different shapes of serrations. The inside of the lock cylinder is provided with corresponding card slots for the different shapes of the serrations. When the key body of the key is inserted into the interior of the lock cylinder, the spoon body The sawtooth is matched with the slot in the lock cylinder, and the key body is rotated by the key handle to cause the lock cylinder to rotate, and the door lock is opened. The conventional mechanical lock system has many inconveniences, such as poor safety performance, easy to be opened, and the like.

1. DOOR LOCK SYSTEM BASED ON FACE RECOGNITION: Publication No: CN104392526A

Publication date: 4 April 2015

Recognition of face, refers in particular to the computer technology utilizing com-parison and analysis. Recognition of face is a popular computer technology research field, and face tracking is detected, and automatically adjusts image zoom, and night infrared is detected, and automatically adjusts exposure intensity; It belongs to biometrics identification technology, is to distinguish biosome individuality to the biological characteristic of biosome (generally referring in particular to people) itself.

Existing door lock, simply by virtue of password or key, the maximum defect of coded lock be password easily stolen by others, guess and forget, its safe and secret degree is inadequate, along with the develop rapidly of science and technology and the appearance of large-scale integrated chip, existing door-locking system is also experiencing the conversion of upgrading. Door lock on the market adopts password identification method, although solve

the divisibility that mechanical door lock changes, easily stolen by others, guess and forget, key and the non-strong correlation of user, also exist great hidden danger in security.

A kind of door-locking system based on recognition of face, comprise single-chip microcomputer, it is characterized in that: also comprise man face image acquiring module, Face detection module, recognition of face pre-treatment module, pyroelectric sensor, watchdog circuit, clock circuit and lock-switch module, described man face image acquiring module, Face detection module, recognition of face pre-treatment module are connected successively with single-chip microcomputer, watchdog circuit is connected with single- chip microcomputer with clock circuit, and described pyroelectric sensor is connected with single-chip microcomputer by signal wire.

# CHAPTER 4

**PROJECT DETAILS AND FLOWCHARTS**

### : Design Components:

* + 1. Raspberry Pi Zero W

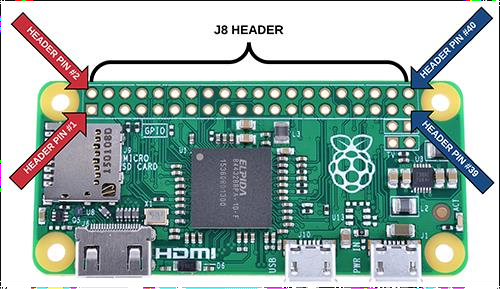
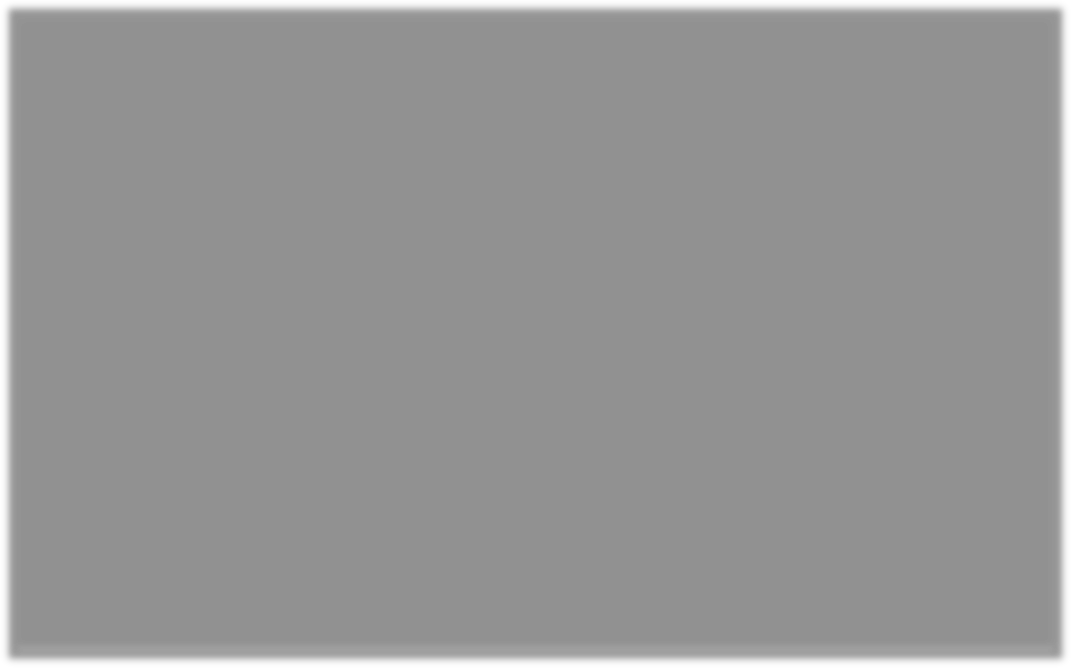


Figure 4.1: Raspberry Pi (Zero W)

The Raspberry Pi Zero W is the smallest and cheapest Raspberry Pi computer. It's an incredibly tiny piece of equipment, but still capable of running the same Raspbian OS (based on Linux) as every other Raspberry Pi computer. The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse.

With built in wireless LAN and Bluetooth, plus a mini-HDMI and micro-USB connector enables a range of input / output options. A built in CSI camera interface enables you to quickly add a camera module. It features a full 40-pin GPIO header, enabling you to connect and control devices. A separate version called Raspberry Pi Zero WH has pins pre-soldered to the header.

Raspberry Pi Zero W specifications :

* **Dimensions:** 65mm × 30mm × 5mm
* **SoC:** Broadcom BCM2835
* **CPU:** ARM11 running at 1GHz
* **RAM:** 512MB
* **Wireless:** 2.4GHz 802.11n wireless LAN
* **Bluetooth:** Bluetooth Classic 4.1 and Bluetooth LE
* **Power:** 5V, supplied via micro USB connector
* **Video & Audio:** 1080P HD video & stereo audio via mini-HDMI connector
* **Storage:** MicroSD card
* **Output:** Micro USB
* **GPIO:** 40-pin GPIO, unpopulated
* **Pins:** Run mode, unpopulated; RCA composite, unpopulated

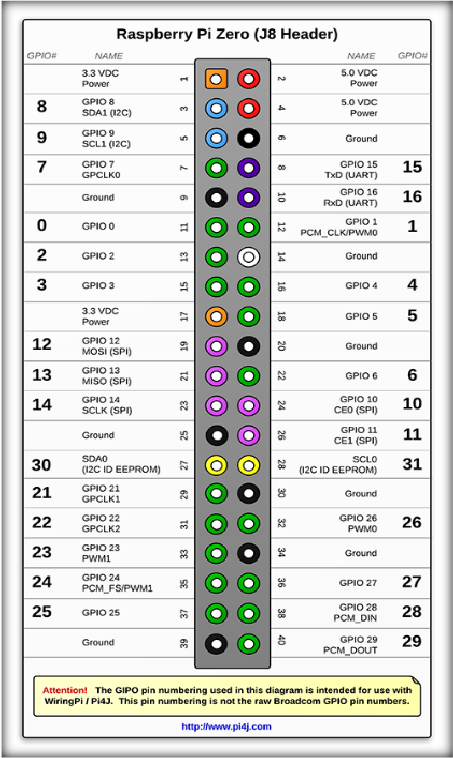


Figure 4.2: Pin Diagram of Raspberry Pi Zero W

A powerful feature of the Raspberry Pi is the row of GPIO (general-purpose input/output) pins along the top edge of the board. A 40-pin GPIO header is found on all current Raspberry Pi boards.

Voltages:

Two 5V pins and two 3V3 pins are present on the board, as well as a number of ground pins (0V), which are unconfigurable. The remaining pins are all general purpose 3V3 pins, meaning outputs are set to 3V3 and inputs are 3V3-tolerant.

Outputs:

A GPIO pin designated as an output pin can be set to high (3V3) or low (0V). Inputs:

A GPIO pin designated as an input pin can be read as high (3V3) or low (0V). This is made easier with the use of internal pull-up or pull-down resistors. Pins GPIO2 and GPIO3 have fixed pull-up resistors, but for other pins this can be configured in software. The GPIO pins can be used for other functions also. Some are available on all pins and others on some specific pins.

PWM (pulse-width modulation) :

-Software PWM available on all pins.

-Hardware PWM available on GPIO12, GPIO13, GPIO18, GPIO19.

SPI(Serial Peripheral Interface):

-SPI0: MOSI (GPIO10); MISO (GPIO9); SCLK (GPIO11); CE0 (GPIO8), CE1 (GPIO7).

-SPI1: MOSI (GPIO20); MISO (GPIO19); SCLK (GPIO21); CE0 (GPIO18); CE1 (GPIO17).

* + 1. Raspberry Pi Camera Module:

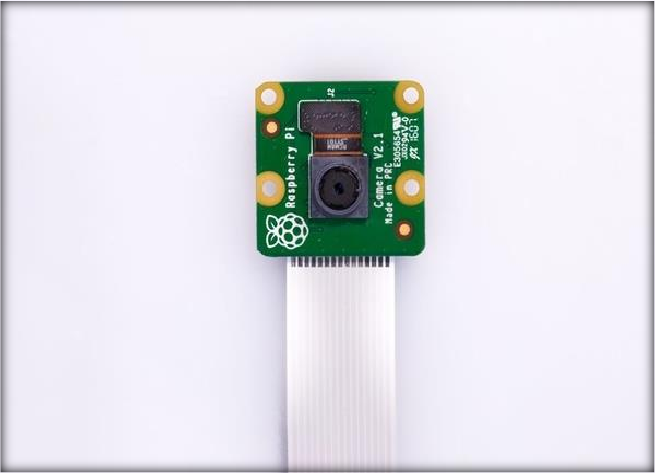


Figure 4.3: Raspberry Pi Camera Module

The Pi camera comes with a flex cable. The flex cable is inserted into the connector which is located between the Ethernet and HDMI port with the silver connectors facing the HDMI port. The flex cable connector is opened by pulling the tabs on the top of the connector upwards then towards the Ethernet port.

The flex cable then is inserted firmly into the connector. The top part of the connector then is pushed towards the HDMI connector and down, while the flex cable is held in place.

Here the Pi camera is being utilized for the face detection and the face recognition process where firstly are captured and stored it in the database using python and then again using the camera while the automation and the surveillance part.

* + 1. Four Channel Relay:

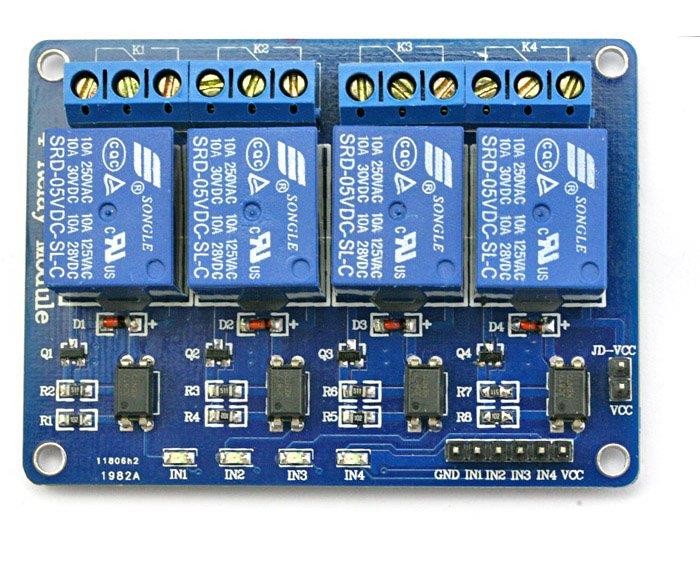
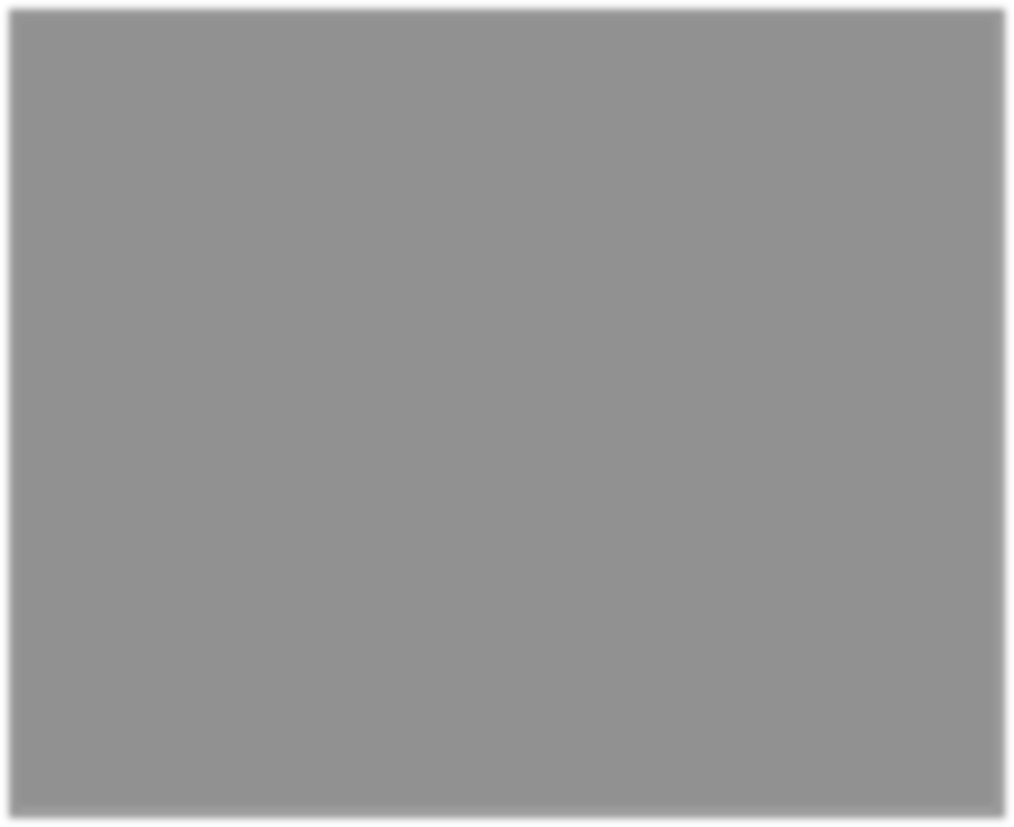


Figure 4.4 : Four Channel Relay

This relay module provides four relays that are rated for 7A at either 28VDC or 10A at 125VAC. Each relay has a Normally Open (NO) and a Normally Closed (NC) contact. This module could be used in various automation projects such as switching lights and motors. You can see that each relay is controlled by a corresponding input pin called In1 to IN4, there are also LEDs that show if the input is low or high. Another benefit is that each relay is optically isolated using an LTC-817.

Four-Channel Relay Module Specifications:

* Supply voltage – 3.75V to 6V
* Trigger current – 5mA
* Current when the relay is active - ~70mA (single), ~300mA (all four)
* Relay maximum contact voltage – 250VAC, 30VDC
* Relay maximum current – 10A

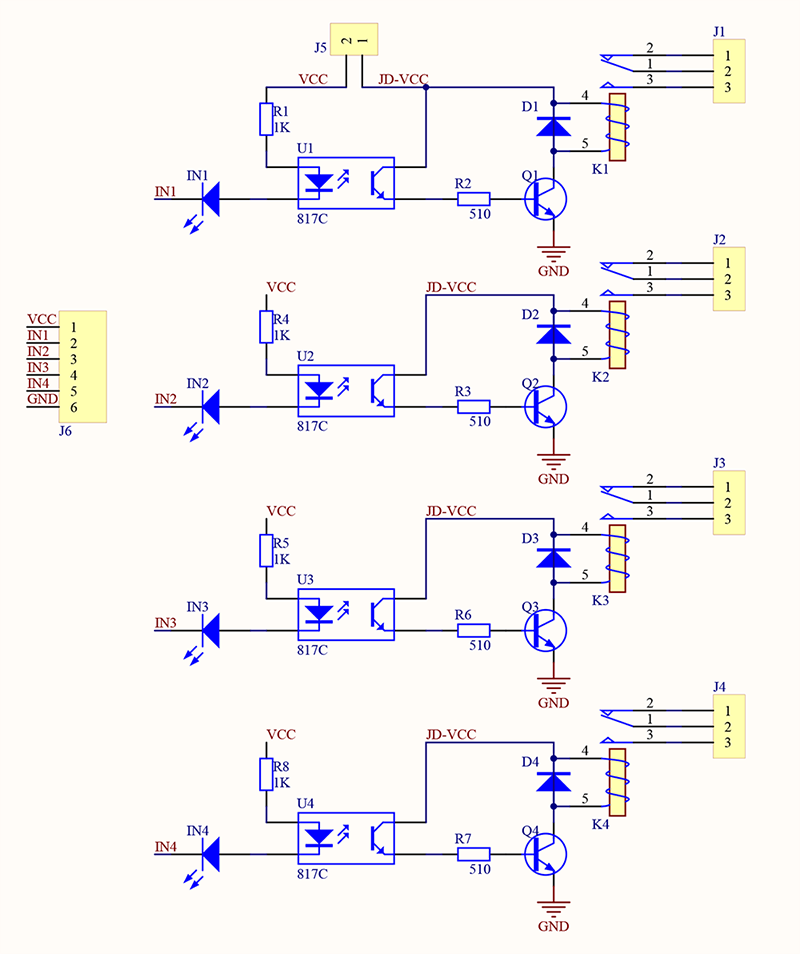
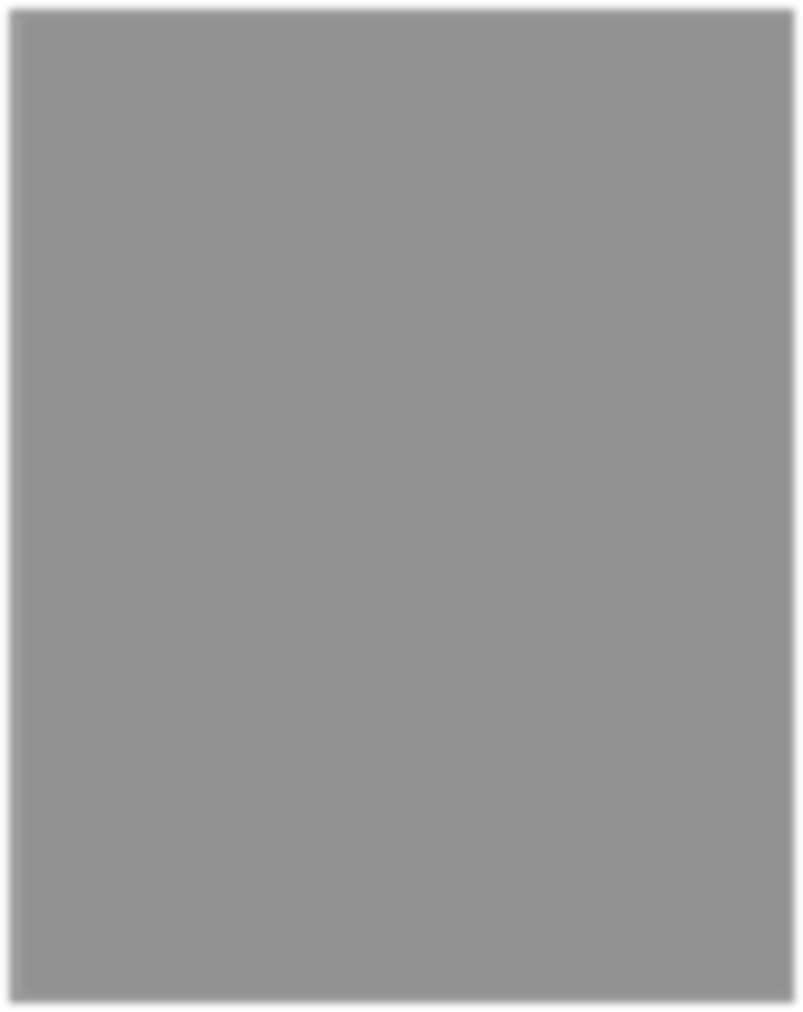


Figure 4.5 : Four Channel Relay In-built Connection

INPUT:

* VCC: Connected to positive supply voltage
* GND: Connected to negative supply voltage
* IN1: Signal triggering terminal 1 of relay module
* IN2: Signal triggering terminal 2 of relay module
* IN3: Signal triggering terminal 3 of relay module
* IN4: Signal triggering terminal 4 of relay module

OUTPUT:

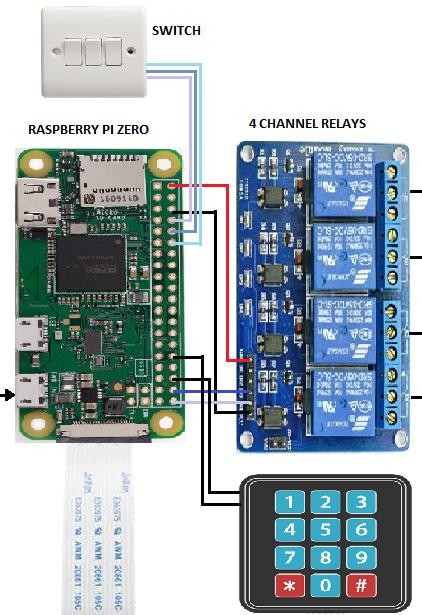
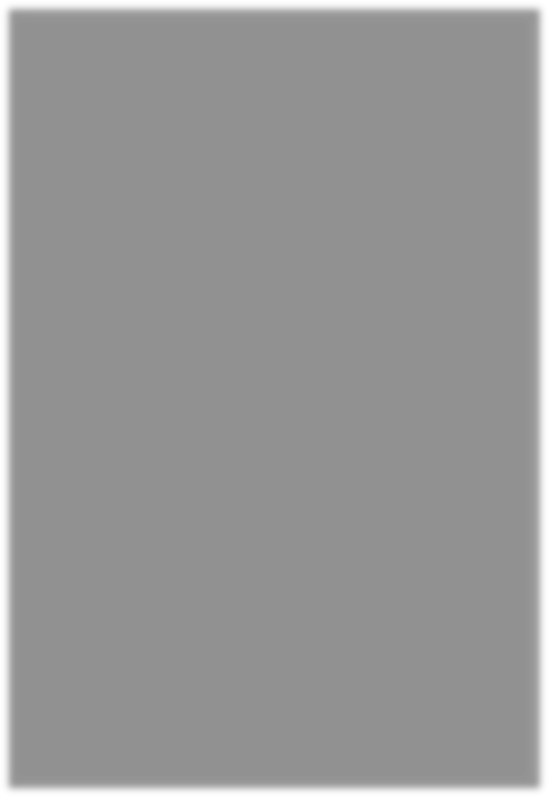


Figure 4.6 : Four Channel Relay Connection With Raspberry Pi

* Connect a load, DC 30V/10A, AC 250V/10A
* Connect the signal terminal IN3 ,IN4 of 4-channel relay to port 38, 40 of the Raspberry Pi,

4-channel relay Raspberry Pi IN3 38

IN4 40

Keypad Raspberry Pi K1 32

K2 36

Switch Raspberry Pi S1 8

S2 10

S3 12

* + 1. Solenoid Lock:

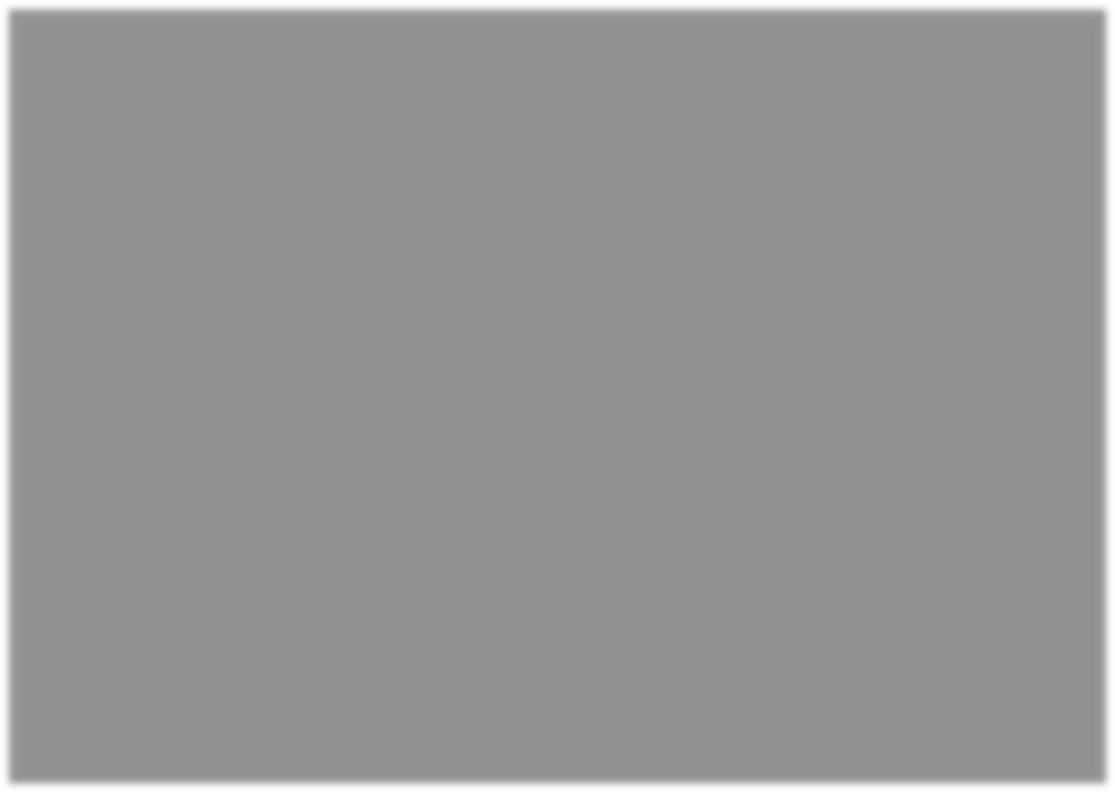


Figure 4.7: Solenoid Lock

Solenoids are basically electromagnets: they are made of a big coil of copper wire with an armature (a slug of metal) in the middle. When the coil is energized, the slug is pulled into the centre of the coil. This makes the solenoid able to pull from one end. This solenoid in particular is nice and strong, and has a slug with a slanted cut and a good mounting bracket. It's basically an electronic lock, designed for a basic cabinet or safe or door.

Normally the lock is active so you can't open the door because the solenoid slug is in the way. It does not use any power in this state. When 9-12VDC is applied, the slug pulls in so it doesn't stick out anymore and the door can be opened. The solenoids come with the slanted slug as shown above, but you can open it with the two Phillips-head screws and turn it around so its rotated 90, 180 or 270 degrees so that it matches the door you want to use it with.

|  |  |
| --- | --- |
| Operating voltage range | 9 – 12VDC. 12V typical |
| Current (energized) | 850-900mA (measured) |
| Current (de-energized) | 0 mA |
| Unlatch time | < 1 second |
| Maximum unlatched time | ~20 second |
| Case (L x W x H) | 53 x 40 x 28mm (2.1 x 1.6 x 1.1″) |
| Latch (L x W) | 10 x 10mm (0.4 x 0.4″) |
| Latch Throw | 10mm (0.4″) |
| Lead Length | 250mm (10″) |

Table 4.1: Technical Details of Solenoid Lock

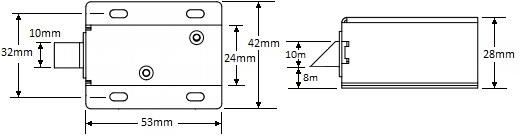


Figure 4.8: Measurement circuit of Solenoid Lock

* + 1. 4 **x** 3 Matrix Keypad **:**

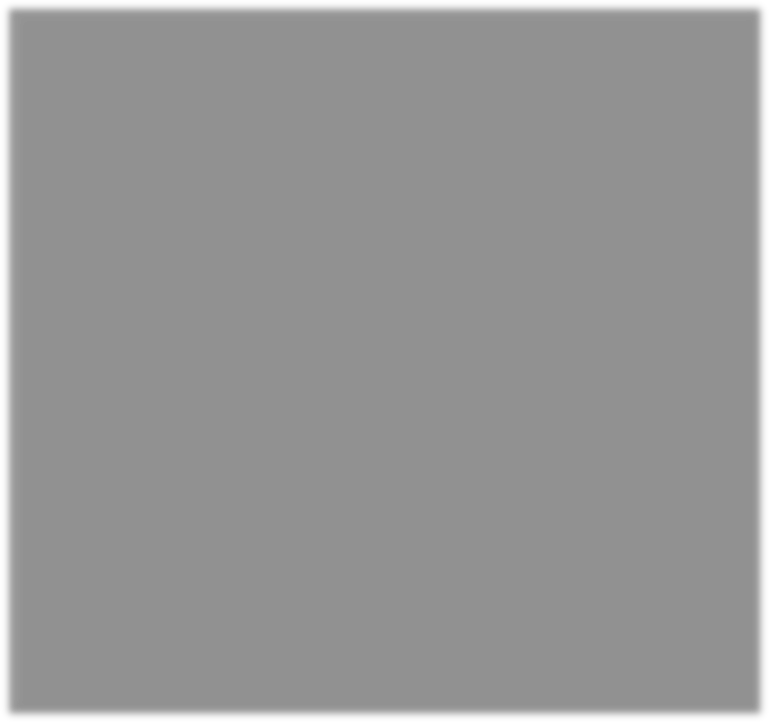


Figure 4.9: 4 **x** 3 Matrix Keypad

This Matrix Keypad 4 X 3 has 12 buttons, arranged in a telephone-line 3x4 grid. The keys are connected into a matrix, hence only 7 microcontroller pins (3-columns and 4-rows) are required to scan the pad.

Specifications of Matrix Keypad 4 X 3:

* + - * Flexible
      * Can be attached on any surface
      * Only 7 microcontroller pins are needed to access 12 buttons Dimensions:
      * Size: 70mm x 77mm x 1mm
      * Weight: 7.5gm Package includes:
      * 1 x Matrix Keypad 4 X 3

### Facial Recognition Feature:

* + 1. Installing Raspbian on the Raspberry Pi:

Installing Raspbian on the Raspberry Pi is really easy. Raspbian will be downloaded and writing the disc image to a micro SD card, at that point booting the Raspberry Pi to that microSD card. For this undertaking, one needs a microSD card (with no less than 8 GB), a PC with a space for it, and, obviously, a Raspberry Pi and fundamental peripherals (a mouse, console, screen, and power source). This isn't the main strategy for introducing Raspbian (more on that in a minute), yet it's a valuable method to learn on the grounds that it can likewise be utilized to introduce such a significant number of other working projects on the Raspberry Pi. When one realizes how to compose a circle picture to a microSD card, we open up a great deal of alternatives for Raspberry Pi projects.

Step1: Download the Raspbian

Turn on the PC and download the Raspbian disc image. One can locate the most recent variant of Raspbian on the Raspberry Pi Foundation's site here. It will take some time, particularly in the event when one intends to utilize the conventional download alternative as opposed to the other download sources. It can without much take a stretch of half hour or more to download.

Step2: Unzip the file

The Raspbian disc image is compressed, so it should be unzipped. The file uses the ZIP64 format, so depending on how current built-in utilities are, one needs to use certain programs to unzip it. Linux users will use the appropriately named Unzip.

Step3: Write the disc image to microSD card

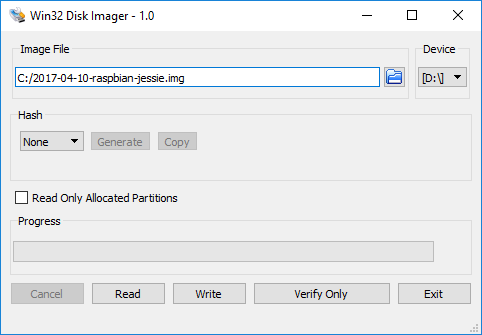


Figure 4.10: Installing Raspbian on Raspberry Pi

One has to pop the microSD card into our computer and write the disc image to it. The process of actually writing the image will be slightly different across these programs, but it’s pretty self- explanatory no matter what is being used. Each of these programs will have us select the destination and the disc image (the unzipped Raspbian file). Choose, double- check, and then button to write.

Step 4: Put the microSD card in your Pi and boot up

When the disc image has been kept in touch with the microSD card, it is prepared to go. Put that microSD into your Raspberry Pi, plug in the peripherals and power source. The present release to Raspbian will boot straightforwardly to the desktop. Our default credentials are username pi and password raspberry.

* + 1. : Face Recognition

1. Prepare your Raspberry Pi

For face recognition to work well, we’re going to need some horsepower, so we recommend a minimum of Raspberry Pi 3B+, ideally a Raspberry Pi 4. The extra memory will make all the difference. To keep as much resource as possible available for our project, we’ve gone for a Raspberry Pi OS Lite installation with no desktop.

Make sure you’re on the network, have set a new password, enabled SSH if you need to, and updated everything with sudo apt -y update && sudo apt -y full-upgrade. Finally, go into settings by running sudo raspi-config and enable the camera in ‘Interfacing Options’.

1. Attach the camera

This project will work well with the original Raspberry Pi Camera, but the new official HQ Camera will give you much better results. Be sure to connect the camera to your Raspberry Pi 4 with the power off. Connect the ribbon cable as instructed in [hsmag.cc/HQCameraGetStarted](http://hsmag.cc/HQCameraGetStarted). Once installed, boot up your Raspberry Pi 4 and test the camera is working. From the command line, run the following: raspivid -o test.h264 -t 10000.

This will record ten seconds of video to your microSD card. If you have an HDMI cable plugged in, you’ll see what the camera can see in real-time. Take some time to make sure the focus is correct before proceeding.

1. Install dependencies

The facial recognition library we are using is one that has been maintained for many years by [Adam Geitgey.](https://github.com/ageitgey/face_recognition) It contains many examples, including Python 3 bindings to make it really simple to build your own facial recognition applications. What is not so easy is the number of dependencies that need to be installed first. There are way too many to list here, and you probably won’t want to type them out, so head over to [hsmag.cc/FacialRec](http://hsmag.cc/FacialRec) so that you can cut and paste the commands. This step will take a while to complete on a Raspberry Pi 4, and significantly longer on a Model 3 or earlier.

1. Install the libraries

Now that we have everything in place, we can install Adam’s applications and Python bindings with a simple, single command: sudo pip3 install face\_recognition. Once installed, there are some examples we can download to try everything out. cd git clone --single-branch [https://github.com/ageitgey/face\_recognition.git.](https://github.com/ageitgey/face_recognition.git) In this repository is a range of examples showing the different ways the software can be used, including live video recognition.

For Example (image)

The examples come with a training image of Kaiyum

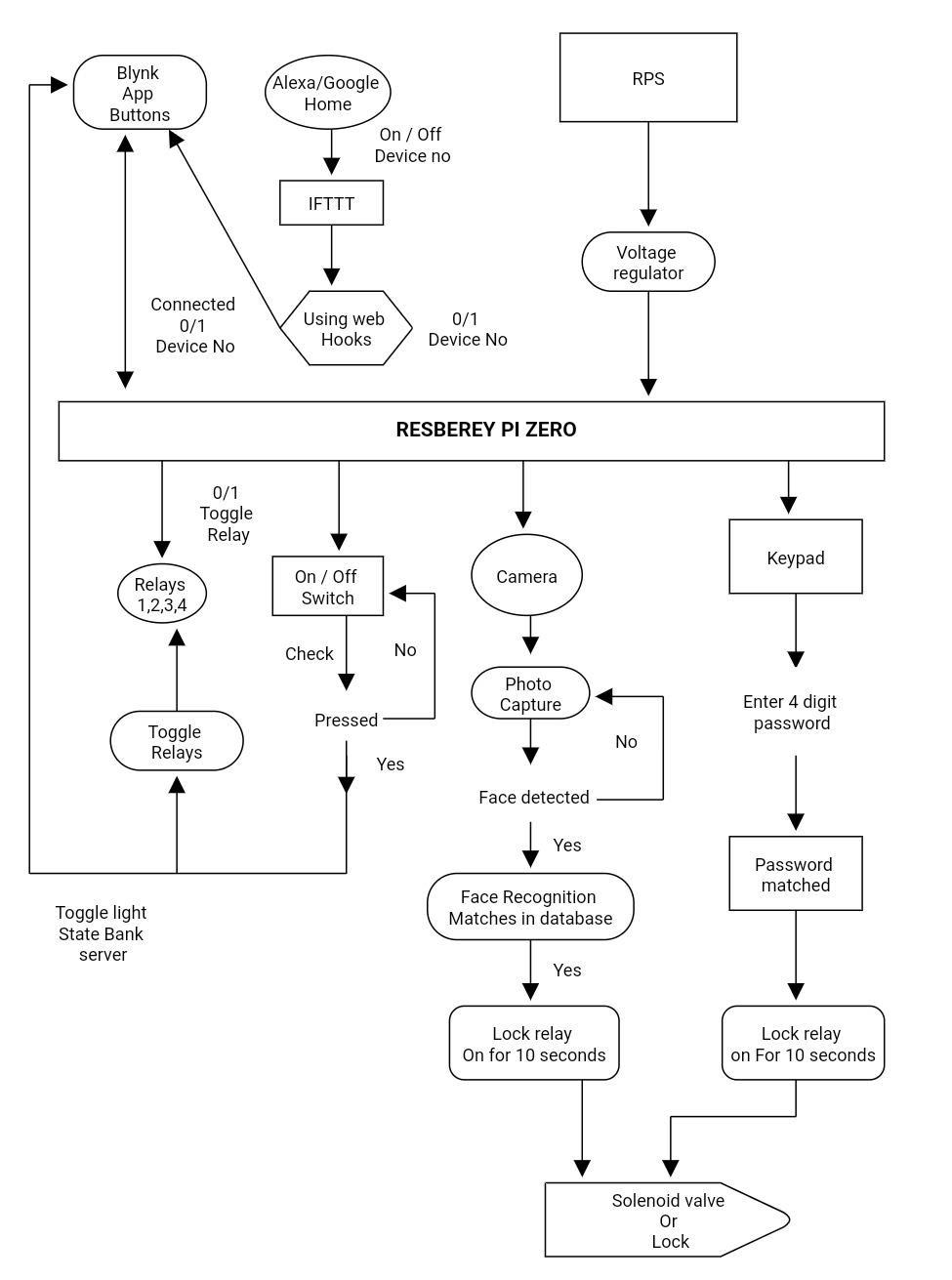
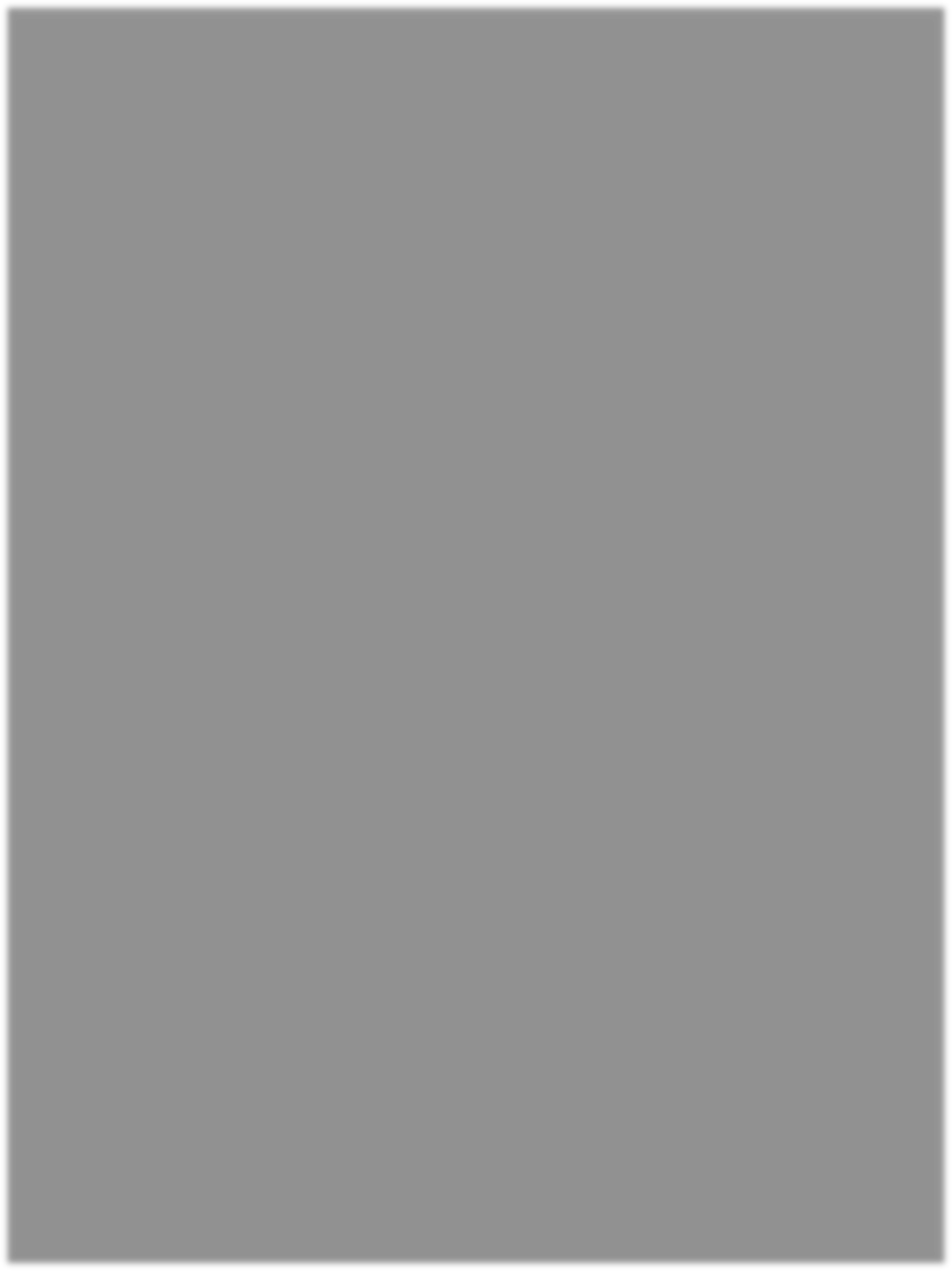
To run the example: cd ./face\_recognition/examplespython3 facerec\_on\_raspberry\_pi.py On your smartphone, find an image of Kaiyum using your favourite search engine and point it at the camera. Providing focus and light are good you will see: “I see someone named Kaiyum!” If you see a message saying it can’t recognise the face, then try a different image or try to improve the lighting if you can. Also, check the focus for the camera and make sure the distance between the image and camera is correct.

Training time:

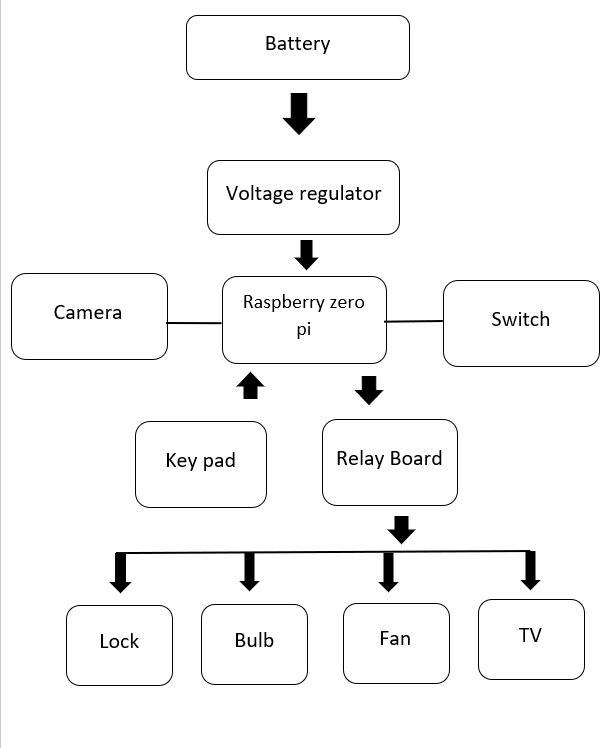
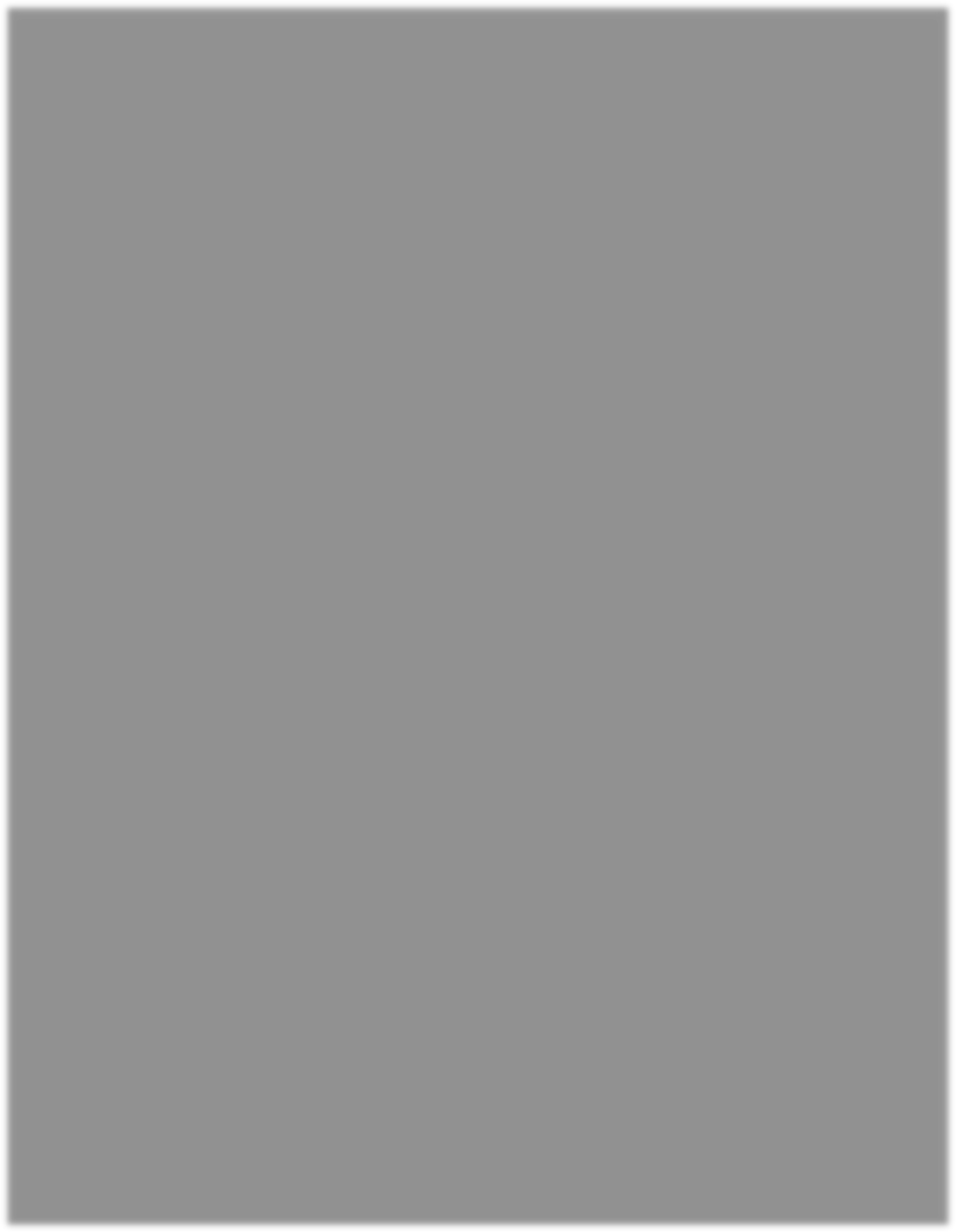
The final step is to start recognising your own faces. Create a directory and, in it, place some good-quality passport-style photos of yourself or those you want to recognise. You

can then edit the facerec\_on\_raspberry\_pi.py script to use those files instead. You’ve now got a robust prototype of face recognition. This is just the beginning. These libraries can also identify ‘generic’ faces, meaning it can detect whether a person is there or not, and identify features such as the eyes, nose, and mouth.

### : FLOWCHART



* 1. **BLOCK DIAGRAM OF THE SMART DOOR LOCK AND LIGHTING SYSTEM USING INTERNET OF THINGS:**



### CIRCUIT DIAGRAM



Raspberry pi

Relay

Solenoid lock

Pi camera

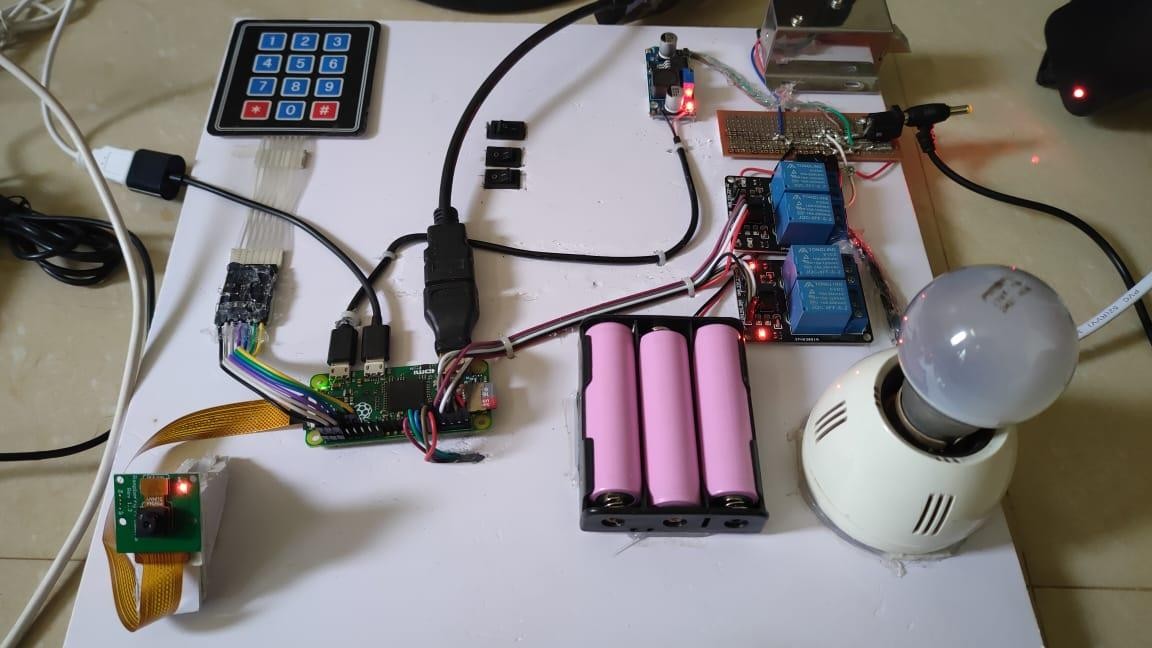
GPIO

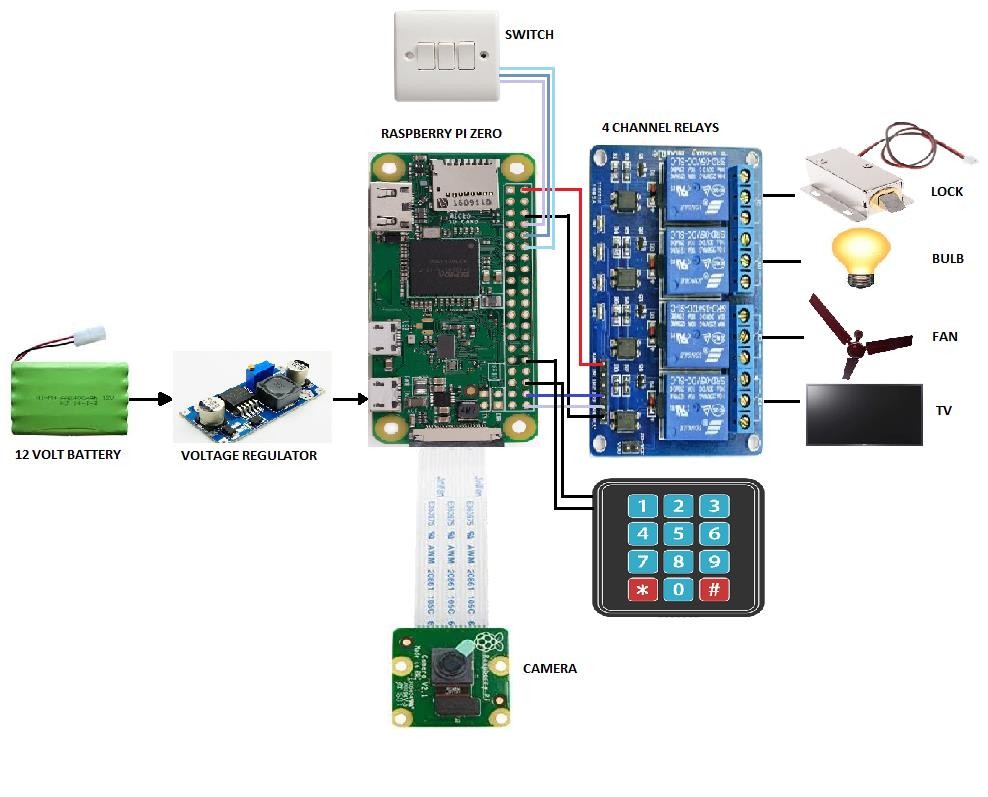
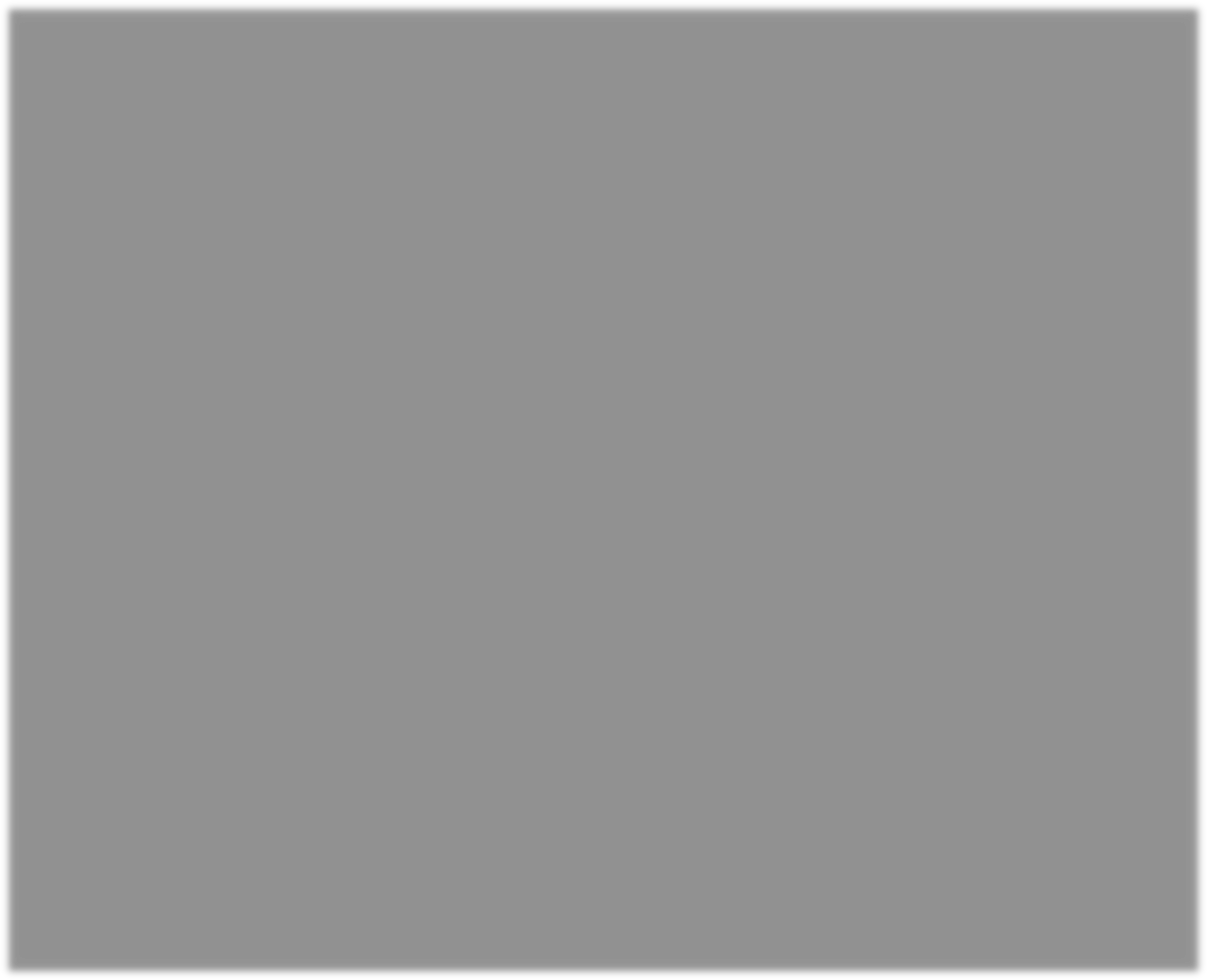
Camera slot

Lightning system

Here one can see that the Pi camera is attached to the camera slot and the Solenoid Lock and Relay is attached to the GPIO pins of the Raspberry Pi. As it is a two channel relay so it is connected to the Lightning System and the Solenoid lock. Here the camera module is being used for a door locking system for surveillance purposes. The door only opens after a successful recognition process, when the face is stored previously in the database and that image has got the access to open the door then it will work otherwise it won’t open and also another feature called Keypad system for door lock system where by entering 4 digit password as a secret key and lightning System works both manually by switches and also by Blynk app Server and by modern features called Alexa, Google home etc..

* + 1. **OVERVIEW IMAGE OF THE PROJECT**





# CHAPTER 5 RESULTS AND DISCUSSIONS

### Results

The main objective of the proposed work here is to create a system where it will be easy to operate home appliances and equipment very easily by making this system user friendly. Developing a smart home system was not easy at first. The most important part of this proposed work is human surveillance which is important due to the security issues of smart homes. For surveillance using face detection and face recognition is being used which are the most modern form of surveillance. For this purpose, Raspberry Pi is being used camera and which is a open source which is a part of Python language. Python here acts as the main platform where most of the work is going to be done.

Here mainly we have introduced two features :

* + 1. Remote access to switches of electrical devices
    2. Door lock system which uses face recognition

1. For example, we have shown connections for remote access to switches of electric bulbs

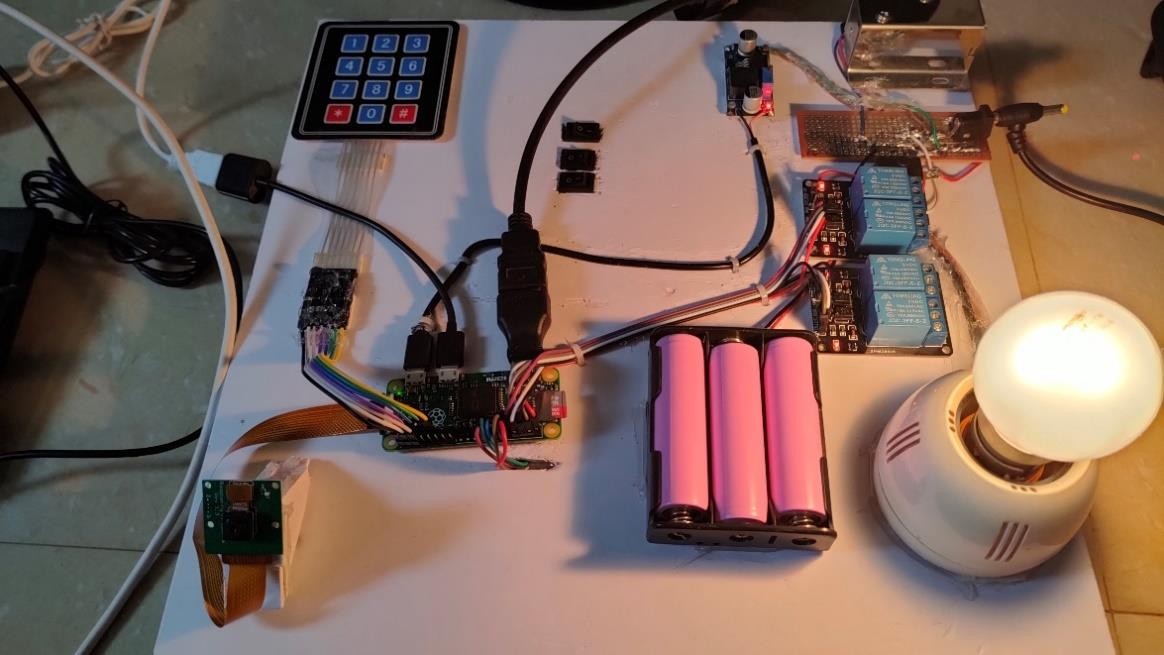


Figure 5.1: Turning ON the bulb which is connected to Raspberry Pi Zero

Raspberry Pi will be used for the *remote* switch. Raspberry Pi will host a local webpage (which contains the *remote* switch) and also connect to WIFI. So, User may use a phone (connected to WIFI) to open the local webpage and turn on/off the light.

For this project equal effort is needed both in software and hardware. Firstly, we connect electric devices to raspberry pi as shown in the figures, then we should program the raspberry pi using python language to give access to the web we are using for remote control.

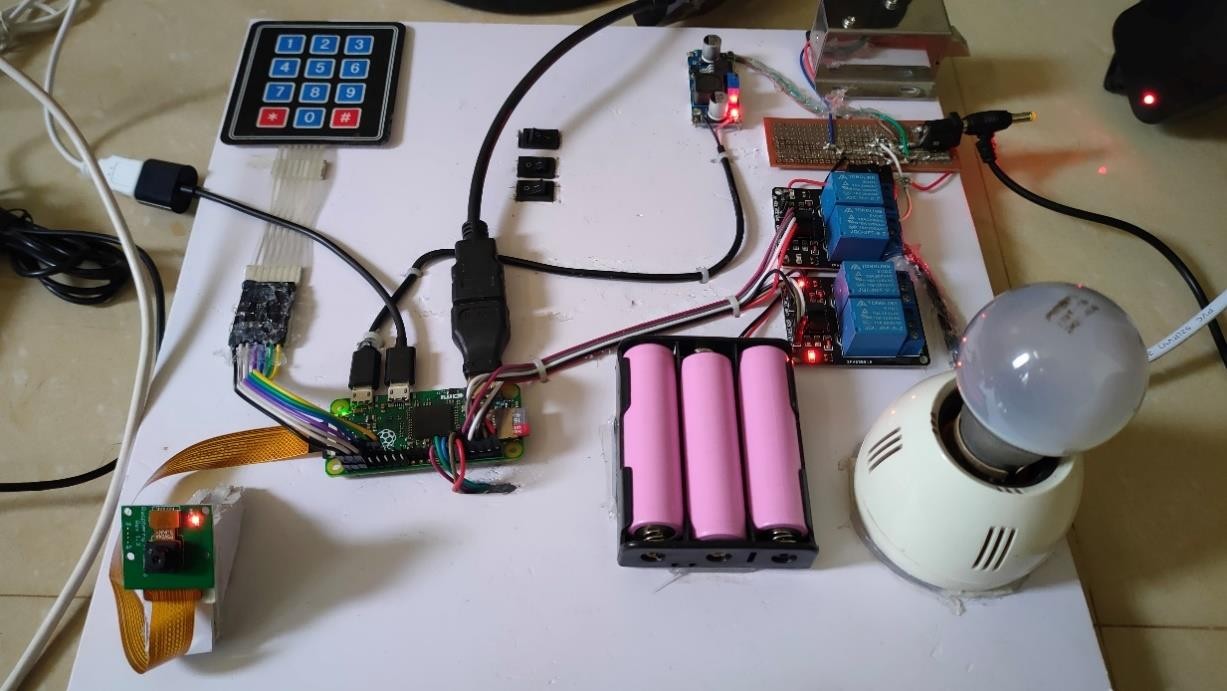


Figure 5.2:Turning OFF the bulb which is connected to Raspberry Pi Zero

1. Door locking and unlocking system using face recognition:

The box like structure is holding the Raspberry Pi camera which is used for capturing pictures of faces to store in the database and then later detect those faces and recognize those faces which is being used for a door locking system for surveillance purposes. The door only opens after a successful reorganization process, when the face is stored previously in the database and that image has got the access to open the door then it will work otherwise it won’t open.

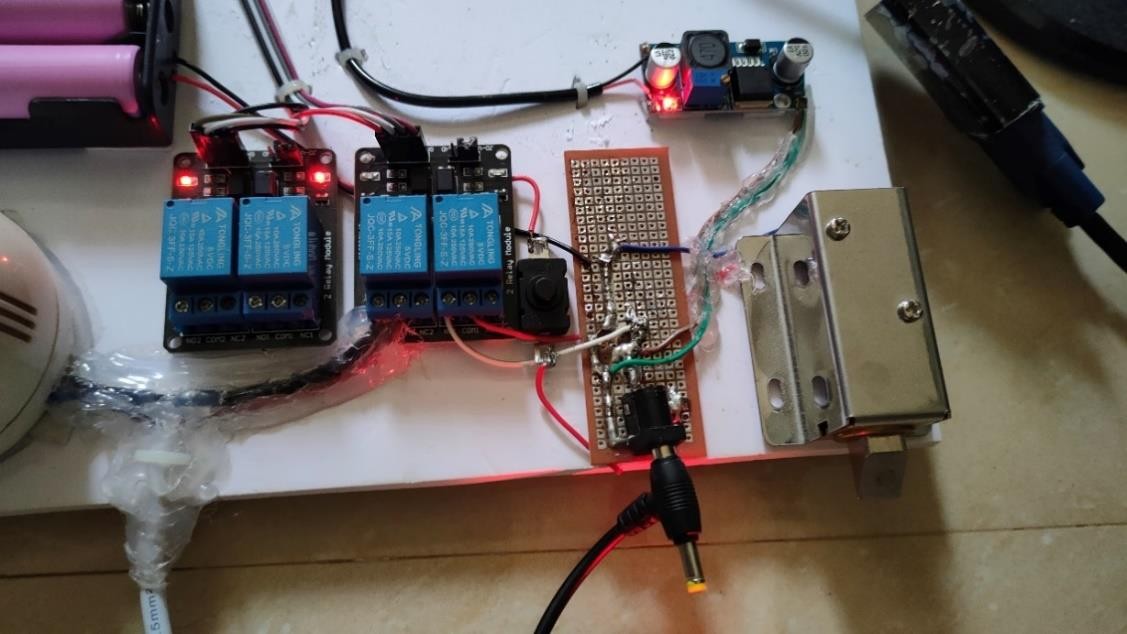


Figure 5.3: Door locking

The picture is taken by the Raspberry Pi camera which is used for the face recognition. Firstly taking the pictures and storing it in the database then during the recognition process the outcome shows the name of the identity if it is stored in the database and it also shows the matching index which is previously mentioned that if it is above the fixed threshold then, it is successful otherwise not. Here clearly it is not the case as the matching index is below threshold hence door is locked.

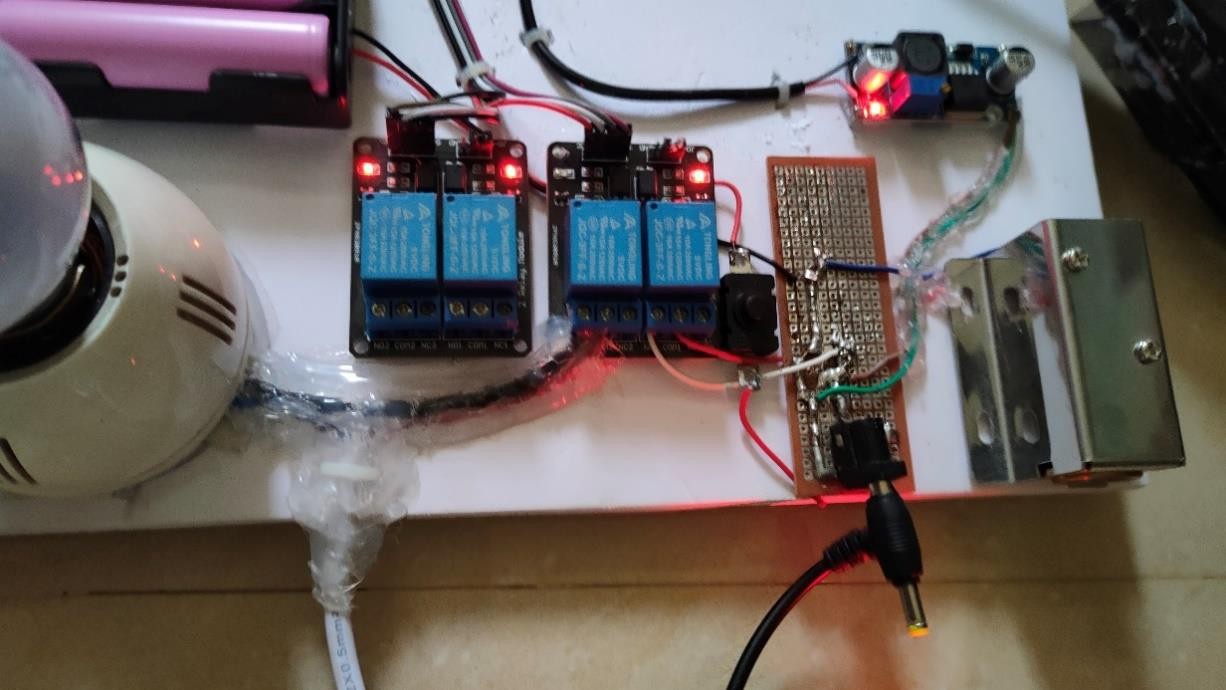


Figure 5.4 :Unlocking

Here the matching index is above the threshold hence the door is unlocked. This is the case of successful face recognition.

The main advantage of this system is that it’s very small in size and lightweight, thus can be easily integrated into wooden/metal doors. It has a good recognition rate with 95% accuracy and the recognition time is less than 0.5 second. It’s extremely power efficient requiring only 5 watts of power to operate. Low power consumption enables the system to run on batteries for hours at a stretch in case of power failure. Also, The low cost and long battery life increase usability and overall efficiency. The aim of this system is to assist people in maintaining security in their homes, offices and other important places. The system can efficiently manage the entry of people at a particular place and with the help of IoT it can also materialize remote controlling of entrances. Also, it saves valuable time of people and reduces the hassles of the security guards and staff, increasing the degree of security.

The above setup makes up an easy to use web app which switches on or off the Electrical devices connected to the Raspberry Pi Zero W and makes it remote accessible and thus fulfils the objectives.

# CHAPTER 6 CONCLUSION AND FUTRURE WORK

### Future Scope :

As the day by increment of technology automation around us, the Automation industry is growing rapidly. In such a scenario the use of this project will also increase. In the coming future there will be high demand of automated and long distance controllable devices over any type of network for the purpose of time reduction and increase in productivity. This project fulfils all the expected demands of automation easily. As the device used in this project, i.e., Raspberry Pi Zero W is a mini computer so it can be deployed for several different purposes at the same time. Alongside controlling the electricity devices, it can serve the multiple purposes, some of which are mentioned below:

* media streamer
* plantation
* hosting server: Due to its multipurpose use this system will be very effective and efficient in future.

### Conclusion :

In this modern era, machine learning and IoT have become two of the most prominent fields which have made our lives easier, safer and efficient through variety of their applications. In almost every aspect of our daily life, we can see the benefits of these fields. However, our effort was to develop a helping hand for maintaining security at important places. Faces were extracted out of images and the machine was trained with some positive and negative images.

Home Automation is the most trending technology currently. This form of technology includes a lot of prospects from a lot of topics or subject assembled together. Home automation here has almost brought a evolutionary change in handling or operating home appliances and equipment’s and made it easy and convenient to use home appliances. A lot of home appliances can be controlled through home automation such as lights, fans, TV, air conditioners, fridge etc. Home automation helps us in operating this appliances from far away. In this proposed scheme an effort has been made to build a home automation system

with the main virtue of Face Recognition. Face Recognition here has been done with the help of a Raspberry Pi camera which was used to take pictures of the faces and store it in the database where.

To better the Face recognition performance, there are miscellaneous things that can be improved here, some of them being fairly easy to go with. For example, you could add colour processing, edge detection, etc. Today, one of the fields that utilizes facial recognition the most is security. Facial recognition is an extremely viable apparatus that can help law masters perceive offenders and programming organizations are utilizing the innovation to enable clients to get to their innovation. Home Automation built upon IoT is a reliable, scalable and highly cost-efficient than the other different ways of Home-Automation System.

# REFERENCES

* + 1. D. Surie, O. Laguionie, T. Pederson, “Wireless sensor networking of everyday objects in a smart home environment”, Proceedings of the International Conference on Intelligent Sensors, Sensor Networks and Information Processing - ISSNIP - 2008, pp. 189 – 194.
    2. Sirsath N. S, Dhole P. S, Mohire N. P, Naik S. C & Ratnaparkhi N.S Department of Computer Engineering, 44, Vidyanagari, Parvati, Pune-411009, India University of Pune, “Home Automation using Cloud Network and Mobile Devices” , ITSI Transactions on Electrical and Electronics Engineering (ITSI- TEEE) ISSN (PRINT)

:2320 –8945, Volume -1, Issue-2, 2013 pg 93-97.

* + 1. “ Microcontroller based Home Security System with Remote Monitoring”, Nikhil Agarwal, Department of EC Engineering MIT, Manipal, Special Issue of International Journal of Computer Applications (0975 –8887) International Conference on Electronic Design and Signal Processing (ICEDSP) 2012 pg 38- 41.
    2. Pavithra.D, Ranjith Balakrishnan, “IoT based Monitoring and Control System for Home Automation” Proceedings of 2015 Global Conference on Communication Technologies(GCCT 2015) pg 169-173.
    3. M. Ibrahim, A. Elgamri, S. Babiker and A. Mohamed, "Internet of things based smart environmental monitoring using the Raspberry-Pi computer," Digital Information Processing and Communications (ICDIPC), 2015 Fifth International Conference on, Sierre, 2015, pp. 159- 164.doi: 10.1109/ICDIPC.2015.7323023 .
    4. R. R. Harmon, E. G. Castro-Leon and S. Bhide, "Smart cities and the Internet of Things," 2015 Portland International Conference on Management of Engineering and Technology (PICMET), Portland, OR, 2015, pp. 485-494.doi: 10.1109/PICMET.2015.7273174.
    5. J. Ye, Q. Xie, Y. Xiahou and C. Wang, "The research of an adaptive smart home system," Computer Science & Education (ICCSE), 2012 7th International Conference on, Melbourne, VIC, 2012, pp. 882-887.

# APPENDIX A – PROGRAM

{A}PROGRAM FOR FACE RECOGNITION ON RASPBERRY PI.

#This is a demo of running face recognition on a Raspberry Pi.

# This program will print out the names of anyone it recognizes to the console. # To run this, you need a Raspberry Pi 2 (or greater) with face\_recognition and # the picamera[array] module installed.

# You can follow this installation instructions to get your RPi set up:

# <https://gist.github.com/ageitgey/1ac8dbe8572f3f533df6269dab35df65> import face\_recognition

import picamera import numpy as np

import RPi.GPIO as GPIO import time

lock=4 GPIO.setmode(GPIO.BCM) GPIO.setup(lock,GPIO.OUT) GPIO.output(lock,True)

# Get a reference to the Raspberry Pi camera.

# If this fails, make sure you have a camera connected to the RPi and that you # enabled your camera in raspi-config and rebooted first.

camera = picamera.PiCamera() camera.resolution = (320, 240)

output = np.empty((240, 320, 3), dtype=np.uint8)

# Load a sample picture and learn how to recognize it. print("Loading known face image(s)")

obama\_image = face\_recognition.load\_image\_file("XYZ.jpg") obama\_face\_encoding =

face\_recognition.face\_encodings(obama\_image)[0] # Initialize some variables

face\_locations = [] face\_encodings = [] while True:

print("Capturing image.")

# Grab a single frame of video from the RPi camera as a numpy array camera.capture(output, format="rgb")

# Find all the faces and face encodings in the current frame of video face\_locations =

face\_recognition.face\_locations(output)

print("Found {} faces in image.".format(len(face\_locations))) face\_encodings =face\_recognition.face\_encodings(output, face\_locations) # Loop over each face found in the frame to see if it's someone we know. for face\_encoding in face\_encodings:

# See if the face is a match for the known face(s)

match = face\_recognition.compare\_faces([obama\_face\_encoding], face\_encoding) name = "<Unknown Person>"

if match[0]:

name = "ABC" GPIO.output(lock,False) time.sleep(10); GPIO.output(lock,True)

print("I see someone named {}!".format(name))

{B}PROGRAM FOR BLYNK APP:

#define BLYNK\_PRINT stdout #ifdef RASPBERRY

#include <BlynkApiWiringPi.h> #else

#include <BlynkApiLinux.h> #endif

#include <BlynkSocket.h>

#include <BlynkOptionsParser.h>

static BlynkTransportSocket \_blynkTransport; BlynkSocket Blynk(\_blynkTransport);

#include <BlynkWidgets.h>

unsigned int checkpin1; unsigned int checkpin2; unsigned int checkpin3; unsigned int checkpin4;

unsigned int pin1laststatus; unsigned int pin2laststatus; unsigned int pin3laststatus; unsigned int pin4laststatus;

void setup()

{

}

BLYNK\_WRITE(V1)

{

printf("virtual 1 changed\n"); digitalWrite (2,param[0]) ;

while(digitalRead(2)!=param[0]){digitalWrite (2,param[0]) ;}

}

BLYNK\_WRITE(V2)

{

printf("virtual 1 changed\n"); digitalWrite (3,param[0]) ;

while(digitalRead(3)!=param[0]){digitalWrite (3,param[0]) ;}

}

BLYNK\_WRITE(V3)

{

printf("virtual 1 changed\n"); digitalWrite (4,param[0]) ;

while(digitalRead(4)!=param[0]){digitalWrite (4,param[0]) ;}

}

BLYNK\_WRITE(V4)

{

printf("virtual 1 changed\n"); digitalWrite (14,param[0]) ;

while(digitalRead(14)!=param[0]){digitalWrite (14,param[0]) ;}

}

//..........................................................................

void loop()

{

Blynk.run();

checkpin1 = digitalRead(17); checkpin2 = digitalRead(27); checkpin3 = digitalRead(22);

checkpin4 = digitalRead(18);

if(checkpin1 != pin1laststatus){ pin1laststatus = checkpin1; printf("Button 1 changed\n");

Blynk.virtualWrite(V1, !digitalRead(2)); digitalWrite (2, !digitalRead(2)) ;

}

if(checkpin2 != pin2laststatus){ pin2laststatus = checkpin2; printf("Button 2 changed\n");

Blynk.virtualWrite(V2, !digitalRead(3)); digitalWrite (3, !digitalRead(3)) ;

}

if(checkpin3 != pin3laststatus){ pin3laststatus = checkpin3; printf("Button 3 changed\n");

Blynk.virtualWrite(V3, !digitalRead(4)); digitalWrite (4, !digitalRead(4)) ;

}

if(checkpin4 != pin4laststatus){ pin4laststatus = checkpin4; printf("Button 4 changed\n");

Blynk.virtualWrite(V4, !digitalRead(14)); digitalWrite (14, !digitalRead(14)) ;

}

}

int main(int argc, char\* argv[])

{

const char \*auth, \*serv; uint16\_t port;

parse\_options(argc, argv, auth, serv, port); Blynk.begin(auth, serv, port);

while(true) {

loop();

}

return 0;

}

{C}PROGRAM FOR TWO FACES:

import RPi.GPIO as GPIO import time GPIO.setwarnings(False) from threading import Thread

import face\_recognition import picamera

import numpy as np

ON = False OFF = True

lock=7 GPIO.setmode(GPIO.BOARD)

GPIO.setup(lock,GPIO.OUT) GPIO.output(lock,OFF)

def camera\_code():

camera = picamera.PiCamera() camera.resolution = (320, 240)

output = np.empty((240, 320, 3), dtype=np.uint8)

print("Loading known face image(s)")

obama\_image = face\_recognition.load\_image\_file("faces/kayum.jpg") obama\_face\_encoding = face\_recognition.face\_encodings(obama\_image)[0]

# Initialize some variables face\_locations = [] face\_encodings = []

obama\_image1 = face\_recognition.load\_image\_file("faces/XYZ.jpg") obama\_face\_encoding1 = face\_recognition.face\_encodings(obama\_image1)[0]

# Initialize some variables face\_locations1 = [] face\_encodings1 = []

while True: print("Capturing image.")

camera.capture(output, format="rgb")

face\_locations = face\_recognition.face\_locations(output) print("Found {} faces in image.".format(len(face\_locations)))

face\_encodings = face\_recognition.face\_encodings(output, face\_locations)

camera.capture(output, format="rgb")

face\_locations1 = face\_recognition.face\_locations(output) print("Found {} faces in image.".format(len(face\_locations1)))

face\_encodings1 = face\_recognition.face\_encodings(output, face\_locations1)

for face\_encoding in face\_encodings:

# See if the face is a match for the known face(s)

match = face\_recognition.compare\_faces([obama\_face\_encoding], face\_encoding) match1 = face\_recognition.compare\_faces([obama\_face\_encoding1],

face\_encoding1)

name = "<Unknown Person>"

if match[0]: name = "ABC"

print("I see someone named {}!".format(name)) GPIO.output(lock,ON)

time.sleep(10); GPIO.output(lock,OFF)

elif match1[0]: name = "Kayum"

print("I see someone named {}!".format(name)) GPIO.output(lock,ON)

time.sleep(10); GPIO.output(lock,OFF)

else:

print("I see someone named {}!".format(name))

get\_level\_thread = Thread(target = camera\_code) get\_level\_thread.daemon = True get\_level\_thread.start()

class keypad():

def \_init\_(self, columnCount = 3):

# CONSTANTS

if columnCount is 3: self.KEYPAD = [

[1,2,3],

[4,5,6],

[7,8,9], ["\*",0,"#"]

]

self.ROW = [40,38,36,37]

self.COLUMN = [35,33,31]

elif columnCount is 4: self.KEYPAD = [ [1,2,3,"A"],

[4,5,6,"B"],

[7,8,9,"C"],

["\*",0,"#","D"]

]

self.ROW = [28,27,26,23]

self.COLUMN = [22,21,30,21]

else:

return

def getKey(self):

# Set all columns as output low

for j in range(len(self.COLUMN)): GPIO.setup(self.COLUMN[j], GPIO.OUT)

GPIO.output(self.COLUMN[j], GPIO.LOW)

# Set all rows as input

for i in range(len(self.ROW)):

GPIO.setup(self.ROW[i], GPIO.IN, pull\_up\_down=GPIO.PUD\_UP)

# Scan rows for pushed key/button

# A valid key press should set "rowVal" between 0 and 3. rowVal = -1

for i in range(len(self.ROW)):

tmpRead = GPIO.input(self.ROW[i]) if tmpRead == 0:

rowVal = i

# if rowVal is not 0 thru 3 then no button was pressed and we can exit if rowVal <0 or rowVal >3:

self.exit() return

# Convert columns to input

for j in range(len(self.COLUMN)):

GPIO.setup(self.COLUMN[j], GPIO.IN, pull\_up\_down=GPIO.PUD\_DOWN)

# Switch the i-th row found from scan to output GPIO.setup(self.ROW[rowVal], GPIO.OUT) GPIO.output(self.ROW[rowVal], GPIO.HIGH)

# Scan columns for still-pushed key/button

# A valid key press should set "colVal" between 0 and 2. colVal = -1

for j in range(len(self.COLUMN)):

tmpRead = GPIO.input(self.COLUMN[j]) if tmpRead == 1:

colVal=j

# if colVal is not 0 thru 2 then no button was pressed and we can exit if colVal <0 or colVal >2:

self.exit() return

# Return the value of the key pressed

self.exit()

return self.KEYPAD[rowVal][colVal]

def exit(self):

# Reinitialize all rows and columns as input at exit for i in range(len(self.ROW)):

GPIO.setup(self.ROW[i], GPIO.IN, pull\_up\_down=GPIO.PUD\_UP) for j in range(len(self.COLUMN)):

GPIO.setup(self.COLUMN[j], GPIO.IN, pull\_up\_down=GPIO.PUD\_UP)

if \_name\_ == '\_main\_':

# Initialize the keypad class kp = keypad()

i=0 password=[1,2,3,4] checkpass=[]

# Loop while waiting for a keypress digit = None

while True:

digit = kp.getKey() if(digit!=None):

i+=1

checkpass.append(digit) print(digit) time.sleep(0.5) if(digit=="\*"):

i=0 checkpass=[]

if(i==4):

if((checkpass[0]==password[0]) and (checkpass[1]==password[1]) and (checkpass[2]==password[2]) and (checkpass[3]==password[3])):

print("lockopen") GPIO.output(lock,ON) time.sleep(10); GPIO.output(lock,OFF) print("lock close")

i=0 checkpass=[]

elif((checkpass[0]!=password[0]) or (checkpass[1]==password[1]) or (checkpass[2]!=password[2]) or (checkpass[3]!=password[3])):

print("Incorrect Password") i=0

checkpass=[] else:

GPIO.cleanup()

{D}PROGRAM FOR KEYPAD:

import RPi.GPIO as GPIO import time GPIO.setwarnings(False) from threading import Thread

import face\_recognition import picamera

import numpy as np

ON = False OFF = True lock=7

GPIO.setmode(GPIO.BOARD)

GPIO.setup(lock,GPIO.OUT) GPIO.output(lock,OFF)

def camera\_code():

camera = picamera.PiCamera() camera.resolution = (320, 240)

output = np.empty((240, 320, 3), dtype=np.uint8)

# Load a sample picture and learn how to recognize it. print("Loading known face image(s)")

obama\_image = face\_recognition.load\_image\_file("XYZ.jpg") obama\_face\_encoding = face\_recognition.face\_encodings(obama\_image)[0]

# Initialize some variables face\_locations = [] face\_encodings = []

while True: print("Capturing image.")

# Grab a single frame of video from the RPi camera as a numpy array camera.capture(output, format="rgb")

# Find all the faces and face encodings in the current frame of video face\_locations = face\_recognition.face\_locations(output) print("Found {} faces in image.".format(len(face\_locations)))

face\_encodings = face\_recognition.face\_encodings(output, face\_locations)

# Loop over each face found in the frame to see if it's someone we know. for face\_encoding in face\_encodings:

# See if the face is a match for the known face(s)

match = face\_recognition.compare\_faces([obama\_face\_encoding], face\_encoding) name = "<Unknown Person>"

if match[0]: name = "ABC"

print("I see someone named {}!".format(name)) GPIO.output(lock,ON)

time.sleep(10); GPIO.output(lock,OFF)

else:

print("I see someone named {}!".format(name))

get\_level\_thread = Thread(target = camera\_code) get\_level\_thread.daemon = True get\_level\_thread.start()

class keypad():

def \_init\_(self, columnCount = 3):

# CONSTANTS

if columnCount is 3: self.KEYPAD = [

[1,2,3],

[4,5,6],

[7,8,9], ["\*",0,"#"]

]

self.ROW = [40,38,36,37]

self.COLUMN = [35,33,31]

elif columnCount is 4: self.KEYPAD = [ [1,2,3,"A"],

[4,5,6,"B"],

[7,8,9,"C"],

["\*",0,"#","D"]

]

self.ROW = [28,27,26,23]

self.COLUMN = [22,21,30,21]

else:

return

def getKey(self):

# Set all columns as output low

for j in range(len(self.COLUMN)): GPIO.setup(self.COLUMN[j], GPIO.OUT)

GPIO.output(self.COLUMN[j], GPIO.LOW)

# Set all rows as input

for i in range(len(self.ROW)):

GPIO.setup(self.ROW[i], GPIO.IN, pull\_up\_down=GPIO.PUD\_UP)

# Scan rows for pushed key/button

# A valid key press should set "rowVal" between 0 and 3. rowVal = -1

for i in range(len(self.ROW)):

tmpRead = GPIO.input(self.ROW[i]) if tmpRead == 0:

rowVal = i

# if rowVal is not 0 thru 3 then no button was pressed and we can exit if rowVal <0 or rowVal >3:

self.exit() return

# Convert columns to input

for j in range(len(self.COLUMN)):

GPIO.setup(self.COLUMN[j], GPIO.IN, pull\_up\_down=GPIO.PUD\_DOWN)

# Switch the i-th row found from scan to output GPIO.setup(self.ROW[rowVal], GPIO.OUT) GPIO.output(self.ROW[rowVal], GPIO.HIGH)

# Scan columns for still-pushed key/button

# A valid key press should set "colVal" between 0 and 2. colVal = -1

for j in range(len(self.COLUMN)):

tmpRead = GPIO.input(self.COLUMN[j]) if tmpRead == 1:

colVal=j

# if colVal is not 0 thru 2 then no button was pressed and we can exit if colVal <0 or colVal >2:

self.exit() return

# Return the value of the key pressed self.exit()

return self.KEYPAD[rowVal][colVal]

def exit(self):

# Reinitialize all rows and columns as input at exit for i in range(len(self.ROW)):

GPIO.setup(self.ROW[i], GPIO.IN, pull\_up\_down=GPIO.PUD\_UP) for j in range(len(self.COLUMN)):

GPIO.setup(self.COLUMN[j], GPIO.IN, pull\_up\_down=GPIO.PUD\_UP)

if \_name\_ == '\_main\_':

# Initialize the keypad class kp = keypad()

i=0 password=[1,2,3,4]

checkpass=[]

# Loop while waiting for a keypress digit = None

while True:

digit = kp.getKey() if(digit!=None):

i+=1

checkpass.append(digit) print(digit) time.sleep(0.5) if(digit=="\*"):

i=0 checkpass=[]

if(i==4):

if((checkpass[0]==password[0]) and (checkpass[1]==password[1]) and (checkpass[2]==password[2]) and (checkpass[3]==password[3])):

print("lockopen") GPIO.output(lock,ON) time.sleep(10); GPIO.output(lock,OFF) print("lock close")

i=0 checkpass=[]

elif((checkpass[0]!=password[0]) or (checkpass[1]==password[1]) or (checkpass[2]!=password[2]) or (checkpass[3]!=password[3])):

print("Incorrect Password") i=0

checkpass=[] else:

GPIO.cleanup()

# APPENDIX B – BILL OF MATERIALS

|  |  |  |  |
| --- | --- | --- | --- |
| Sl.No | Components | Qty | Cost in Rs |
| 1. | Camera module | 1 | 850 |
| 2. | Camera cable | 1 | 100 |
| 3. | Lock | 1 | 550 |
| 4. | Keyboard | 1 | 150 |
| 5. | Jump wires | 5 | 100 |
| 6. | Normal wires | 5 | 50 |
| 7. | Programmer switch | 1 | 200 |
| 8. | Register and capacitor | 4 | 80 |
| 9. | Raspberry pi W | 1 | 950 |
| 10. | Buck converter | 1 | 130 |
| 11. | USB hub | 1 | 300 |
| 12. | Battery | 1 | 350 |
| 13. | Switch | 2 | 100 |
| 14. | Four channel relay | 4 | 300 |
| 15. | Wi-Fi Adaptor | 1 | 250 |
| TOTAL | |  | **4500** |