NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY

(An Autonomous Institute, Affiliated to Visvesvaraya Technological University, Belgaum, Approved by AICTE &Govt. of Karnataka)

ACADEMIC YEAR 2019-2020

KNOWLEDGE ● CHARACTER ● UNITY

Internship Report on

"Home Automation and Security System"

Carried out at

Li2 - TECHNOLOGIES

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

BACHELOR OF ENGINEERING

Nikhil Kumar G **1NT16EC092**

Under the guidance of

Prof. Rudresha K J
Assistant Professor
Dept of ECE, NMIT
Internal Guide

Mr. Vijay Rao
Director
Li2-Technologies
External Guide

Department of Electronics and Communication Engineering
NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY
Yelahanka, Bangalore-560064

NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY

(An Autonomous Institute, Affiliated to VTU, Belgaum, Approved by AICTE & State Govt. of Karnataka),
Yelahanka, Bangalore-560064

Department Of Electronics And Communication Engineering

KNOWLEDGE ● CHARACTER ● UNITY

CERTIFICATE

Certified that the Internship entitled "**Home Automation and Security System**" carried out at **Li2-Technologies Pvt Ltd.** by **NIKHIL KUMAR G (1NT16EC092),** bonafide student of Nitte Meenakshi Institute of Technology in partial fulfillment for the award of **Bachelor of Engineering** in Electronics and Communication of the Visvesvaraya Technological University, Belagavi during the academic year 2019-2020.The internship report has been approved as it satisfies the academic requirement in respect of internship work for completion of autonomous scheme of Nitte Meenakshi Institute of Technology for the above said degree.

Signature of the Guide	**Signature of the HOD**
(Prof. Rudresha K J)	**(Dr. Ramachandra A. C)**

External Viva

Name of the Examiners	**Signature with Date**
1………………………………	
2………………………………	

ACKNOWLEDGEMENT

Behind every achievement there are people who inspire us to do it. To them we take the opportunity to lay the words of gratitude imprinted not just on the paper but deep in our heart.

I express my deepest thanks to **Dr N. R Shetty,** Director of **Nitte Meenakshi Institute of Technology** and our Principal **Dr H. C. Nagaraj** for following me to carry out the internship and supporting us throughout.

I am grateful to **Dr. Ramachandra A. C**, HOD, Department of Electronics and Communication Engineering, NMIT, Bangalore for granting me the permission and guiding me throughout.

I wish to convey my deep sense of gratitude to **Mr. Rudresha K J,** Assistant Professor, ECE Department, NMIT for his valuable guidance through the conduction of this internship

I wish to thank **Mr. Aravind Nadig** and **Mr. Vijay Rao** for giving me the opportunity to be a part of the **Li2 Innovations** team and for their constant support and resources throughout.

Finally, we thank all other unnamed who helped us in various ways to gain knowledge and have a good training.

ABSTRACT

Home Automation And Security System.

Home Automation is a technological solution that enables automating household appliance. It uses a combination of hardware and software technologies that enable control and management over appliances and devices within a home. Nowadays Home Automation starts with Wi-fi or Bluetooth Communication. We have a lot of android application available on the play store and app store to control home automatio. In this project we use an Iot based microcontroller **NodeMCU(ESP8266)** which supports Wi-fi. Wi-fi is controlled by using a **Ifttt** android application instead of using a button or a switch. Here we only turn on the button in Ifttt app to control the electrical appliances to turn on and off. So here, Iftt app is used as a transmitting device and uses Wi-fi module placed in the circuit as receiver. Iftt app will transmit voice command using wi-fi to the electrical appliances depends upon the required condition. When we talk about security systems here, we talk about image processing we made use of Raspberry Pi. Raspberry Pi is the name of series of small single-board computers developed by the Raspberry Pi foundation. OpenCV is a library containing programming functions which mainly aims at real-time computer vision and machine learning software. It was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Digital Image processing is growing fast, this due to increase in available machine learning techniques where the developers can access with cloud. People counter Is a measure of people traversing a certain passage or entrance which can be implemented in various domains such as in school and public libraries, airports, malls. In provides different applications of security and business management. We use Raspberry Pi with PiCamera module where it captures and streams video in which we keep track on count of number of people entering and leaving a particular area of interest. As a part of Security systems By making use of same libraries we also speak about face recognition by using by using effective algorithms of Digital Image Processing. In this we learn step by step to use Raspberry Pi and PiCamera module to recognize faces in real time.

ABOUT ORGANIZATION

Li2 Technologies Pvt Ltd. is a venture committed to empowering education through experiential learning solutions for schools and colleges. Li2 is the national leader in applied training on Robotics, Embedded Systems and Mobile Computing for students, Engineering graduates and professionals. The idea of starting Li2 germinated from the fact that Indian academic system is predominantly focused on rote classroom learning with little scope for the application of theory thus creating a huge gap between the academia and current industry needs.

Li2 is committed to filling this void in technology education through its innovative products and training services thus playing a significant role in our nation's strive to be a technological superpower. The courses with respect to Robotics, IoT, connected devices and more are conducted at this organization. Also, hands-on Workshop on various interesting topics such as 3D Printing, Robotics, IoT, Connected devices and more

Table of Contents:

Table of Figures:

INTRODUCTION TO HOME AUTOMATION AND SECURITY SYSTEMS

As we know, when we talk about home automation we mainly refer to controlling of home appliances with the help of google assistant by making use of the ESP8266 microcontroller and interfacing it with several electrical components like fan and bulb by connecting it with relay. This was our first requirement in the company where we completed our internship. Next, we were told to do projects on Digital Image Processing domain. So, the second project was Security systems, this is achieved by making use of Raspberry Pi and Open CV software. The first project in this was Real time face recognition and we concluded it counting the number of people as well. Here, for real time face recognition we made use of haar cascade algorithm which is effective when we use a Raspberry Pi board. And people counter was done by tracking the objects by the centroid script.

CHAPTER 1:

1 CONTROLLING HOME APPLIANCES USING GOOGLE ASSISTANT

1.1 Introduction

Google Assistant is an artificial intelligence powered virtual assistant developed by Google that is primarily available on mobile and smart home devices. Google assistant is AI based voice command service using which we interact with google assistant and it can search on the internet, control appliances, alarms etc,. Our main objective is to control smart home devices including lights, switches, fans and thermostats using Google Assistant. This application includes Google assistant along with Adafruit server and IFTTT service.

In this experiment we will control a light and fan connected to NodeMCU. The concept here is to use IFTTT which stands for 'If This Then That' helps us to create different commands to perform on the house which we call as applets. We will learn about creating applets and other stuff in the upcoming sections. Initially, IFTTT basically connects our Google assistant to io.adafruit.com. Weare going to use Google assistant to send commands and io.adafruit.com responds to them.

1.2 Hardware Requirements

1.2.1 NodeMCU ESP8266 Board

The ESP8266 is a series of WiFi chips produced by Espressif Systems. It is System-On-Chip(SOC) which integrates a 32-bit microcontroller, standard data peripheral interfaces, antenna switches, RF bauln, low noise amplifier, filters, power amplifier and power management modules into a small package.

NodeMCU is using the ESP-12 module, which can be programmed directly using the Arduino IDE. There is an on-board LED on the NodeMCU which is internally connected to the digital pin D0(GPIO16) so that we can switch LED ON and OFF by controlling the digital pin D0.

Features of NodeMCU (based on ESP-12):

- The ESP-12 has a WiFi standard of 802.11 b/g/n. It means the device is capable to work on 2.4 GHz bands of WiFi with support for 20/40 MHz channel width.
- The NodeMCU has 16 general purpose input output (GPIO) pins. But, many of them are used internally to interface internal components.

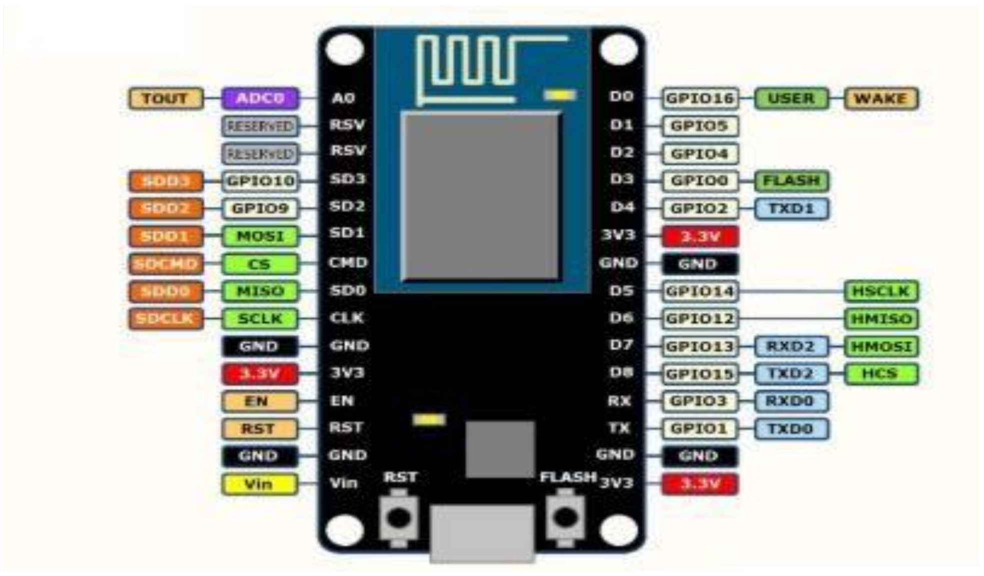


Figure 1. 1 Pin configuration of NodeMCU

- Two pins are reserved for TX and RX to communicate with a host (computer) from where a code will be uploaded to the chip. However, when no communication is taking place, these pins can be used as GPIO as well.

- 13 pins are available to use as digital GPIO pins.

- The NodeMCU also has a pin A0 connected to a 10-bit Analog-to-Digital Converter (ADC). This allows the pin to be used to take analog inputs from a sensor.

How to start with NodeMCU?

NodeMCU Development board is featured with wifi capability, analog pin, digital pins and serial communication protocols.To get start with using NodeMCU for IoT applications first we need to know about how to write/download NodeMCU firmware in NodeMCU Development Boards. And before that where this NodeMCU firmware will get as per our requirement.

There is online NodeMCU custom builds available using which we can easily get our custom NodeMCU firmware as per our requirement.

How to write codes for NodeMCU?

After setting up ESP8266 with Node-MCU firmware, let’s see the IDE (Integrated Development Environment) required for development of NodeMCU.

- **NodeMCU with ESPlorer IDE**

  Lua scripts are generally used to code the NodeMCU. Lua is an open source, lightweight, embeddable scripting language built on top of C programming language.

  For more information about how to write Lua script for NodeMCU refer getting started with NodeMCU using ESPlorerIDE

  NodeMCU with Arduino IDE

  Here is another way of developing NodeMCU with a well-known IDE i.e. Arduino IDE. We can also develop applications on NodeMCU using Arduino development environment. This makes easy for Arduino developers than learning new language and IDE for NodeMCU.

  For more information about how to write Arduino sketch for NodeMCU refer getting started with NodeMCU using ArduinoIDE

Difference in using ESPlorer and Arduino IDE

Well, there is a programming language difference we can say while developing application for NodeMCU using ESPlorer IDE and Arduino IDE.

We need to code in C\C++ programming language if we are using Arduino IDE for developing NodeMCU applications and Lua language if we are using ESPlorer IDE.

Basically, NodeMCU is Lua Interpreter, so it can understand Lua script easily. When we write Lua scripts for NodeMCU and send/upload it to NodeMCU, then they will get executes sequentially. It will not build binary firmware file of code for NodeMCU to write. It will send Lua script as it is to NodeMCU to get execute.

In Arduino IDE when we write and compile code, ESP8266 toolchain in background creates binary firmware file of code we wrote. And when we upload it to NodeMCU then it will flash all NodeMCU firmware with newly generated binary firmware code. In fact, it writes the complete firmware.

That’s the reason why NodeMCU not accept further Lua scripts/code after it is getting flashed by Arduino IDE. After getting flashed by Arduino sketch/code it will be no more Lua interpreter and we got error if we try to upload Lua scripts. To again start with Lua script, we need to flash it with NodeMCU firmware.

Since Arduino IDE compile and upload/writes complete firmware, it takes more time than ESPlorer IDE.

Setting up the NodeMCU:

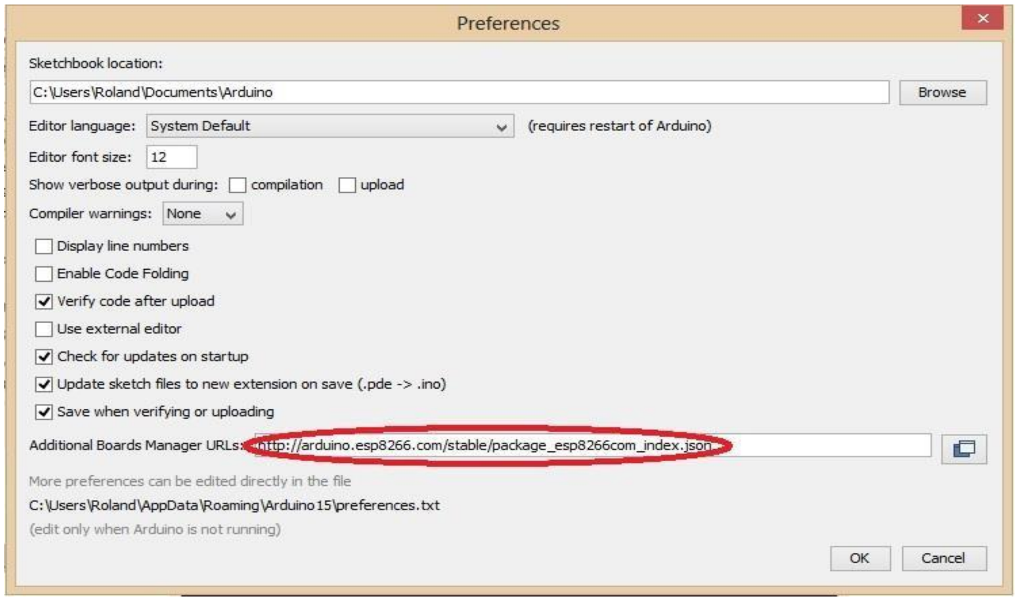
In order to turn ON the NodeMCU we will need a micro-USB cable. Then it should be connected wirelessly using WiFi connection.

Step 1: Connect your NodeMCU to your computer

You need a USB micro B cable to connect the board. Once you plugged it in, a blue LED will start flashing.

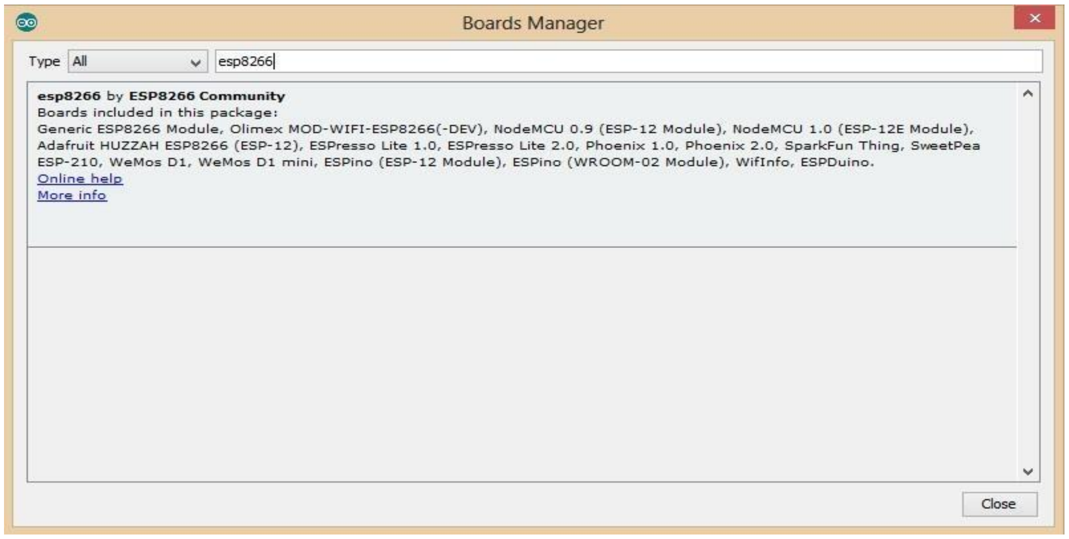
Step 2: Open Arduino IDE

You need to have at least Arduino IDE version 1.6.4 to proceed with this. Go to File > Preferences. In the "Additional Boards Manager URLs" field, type (or copy - paste) *http://arduino.esp8266.com/stable/package_esp8266com_index.json*. Don't forget to click OK



Sketchbook location: C:\Users\Roland\Documents\Arduino
Editor language: System Default (requires restart of Arduino)
Editor font size: 12
Show verbose output during: ☐ compilation ☐ upload
Compiler warnings: None
☐ Display line numbers
☐ Enable Code Folding
☑ Verify code after upload
☐ Use external editor
☑ Check for updates on startup
☑ Update sketch files to new extension on save (.pde -> .ino)
☑ Save when verifying or uploading
Additional Boards Manager URLs: http://arduino.esp8266.com/stable/package_esp8266com_index.json
More preferences can be edited directly in the file
C:\Users\Roland\AppData\Roaming\Arduino15\preferences.txt
(edit only when Arduino is not running)

Then go to Tools > Board > Board Manager. Type "esp8266" in the search field. The entry "esp8266 by ESP8266 Community" should appear. Click that entry and look for the install button on the lower right.



esp8266

esp8266 by **ESP8266 Community**
Boards included in this package:
Generic ESP8266 Module, Olimex MOD-WIFI-ESP8266(-DEV), NodeMCU 0.9 (ESP-12 Module), NodeMCU 1.0 (ESP-12E Module), Adafruit HUZZAH ESP8266 (ESP-12), ESPresso Lite 1.0, ESPresso Lite 2.0, Phoenix 1.0, Phoenix 2.0, SparkFun Thing, SweetPea ESP-210, WeMos D1, WeMos D1 mini, ESPino (ESP-12 Module), ESPino (WROOM-02 Module), WifInfo, ESPDuino.
Online help
More info

Once the download is complete, we can start coding.

1.2.2 Relay

Relays are switch that open and close circuits electromechanically or electronically. Relays control one electrical circuit by opening and closing contacts in another circuit. As relay diagrams show, when a relay contact is normally open (NO), there is an open contact when the relay is not energized. When a relay contact is Normally Closed (NC), there is a closed contact when the relay is not energized. In either case, applying electrical current to the contacts will change their state.

Relays are generally used to switch smaller currents in a control circuit and do not usually control power consuming devices except for small motors and Solenoids that draw low amps. Nonetheless, relays can "control" larger voltages and amperes by having an amplifying effect because a small voltage applied to a relays coil can result in a large voltage being switched by the contacts.

Protective relays can prevent equipment damage by detecting electrical abnormalities, including overcurrent, undercurrent, overloads and reverse currents. In addition, relays are also widely used to switch starting coils, heating elements, pilot lights and audible alarms.

Features of 5-Pin 5V Relay:

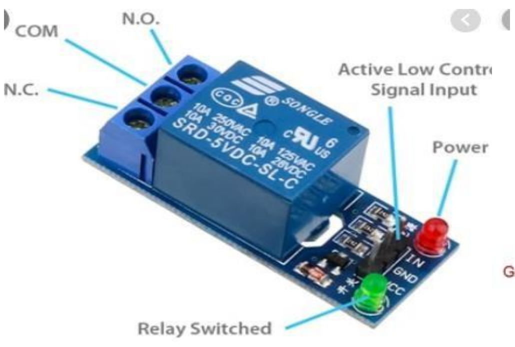
- Trigger Voltage (Voltage across coil): 5V DC
- Trigger Current (Nominal current): 70mA

- Maximum AC load current: 10A @ 250/125V AC
- Maximum DC load current: 10A @ 30/28V DC
- Compact 5-pin configuration with plastic moulding
- Operating time: 10msec Release time: 5msec
- Maximum switching: 300 operating/minute (mechanically)
-

Pin Configuration:

- + : 5V power supply
- - : Ground
- S : Signal from the Arduino
- NC : normally closed
- NO : normally open
- COMMON : common

Working of Relay: The two important parameters of the relay. One is the trigger voltage, this is the voltage required to turn on the relay that is to change the contact from Common->NC to Common->NO. The relay here has 5V trigger voltage, but we can also find relays of values 3V, 6V and even 12V so select one based on the available voltage in the project. The other parameter is the Load voltage and current, this is the amount of voltage or current that the NC ,NO or Common terminal of the relay could withstand.



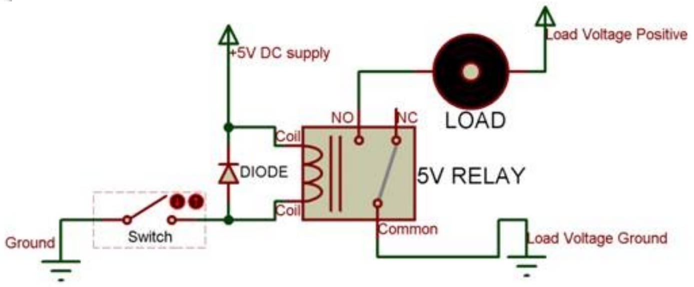


Figure 1. 2 Relay Configuration

Since the relay has 5V trigger voltage we have used a +5V DC supply to one end of the coil and the other end to ground through a switch. This switch can be anything from a small transistor to a microcontroller or a microprocessor which can perform switching operating.

Applications of Relay:

- Commonly used in switching circuits.
- For Home Automation projects to switch AC loads

- To Control (On/Off) Heavy loads at a pre-determined time/condition
- Used in safety circuits to disconnect the load from supply in event of failure
- Used in Automobiles electronics for controlling indicators glass motors etc.

1.2.3 Breadboard

An electronics breadboard (as opposed to the type on which sandwiches are made) is actually referring to a solderless breadboard. These are great units for making temporary circuits and prototyping, and they require absolutely no soldering.

Another common use of breadboards is testing out new parts, such as Integrated circuits (ICs). When you are trying to figure out how a part works and constantly rewiring things, you don't want to have to solder your connections each time.

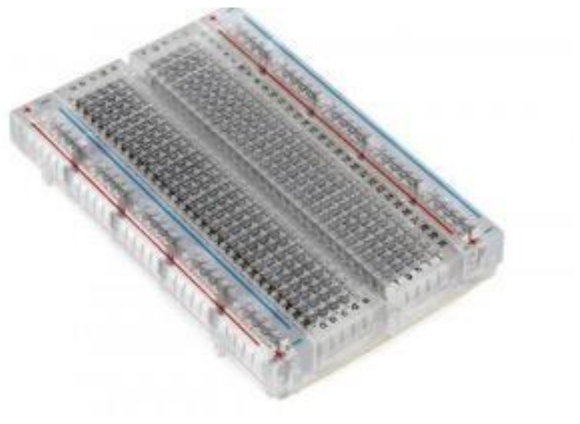


Figure 1. 3 Breadboard

1.2.4 USB Cable

USB, short for Universal Serial Bus, is an industry standard that was developed to define cables, connectors and protocols for connection, communication, and power supply between personal computers and their peripheral devices.

Figure 1. 4 Usb Cable

1.2.5 Connecting wires

Connecting wires allows an electrical current to travel from one point on a circuit to another because electricity needs a medium through which it can move. Most of the connecting wires are made up of copper or aluminum. Copper is cheap and good conductivity. Instead of the copper, we can also use silver which has high conductivity.

Figure 1. 5 Jumper cables

1.3 Software Requirements

To Build home automation application, we use three different platforms.

1. Google Assistant
2. Adafruit IO account
3. IFTTT account
4. Arduino IDE on your computer
5. Adafruit MQTT library

1.3.1 Google Assistant

Google Assistant is Google’s voice assistant. When it launched, Google Assistant was an extension of Google Now, designed to be personal while expanding on Google’s existing “OK Google” voice controls.

Originally, Google Now smartly pulled out relevant information for you. It knew where you worked, your meetings and travel plans, the sports teams you liked, and what interested you so that it could present you with information that mattered to you.

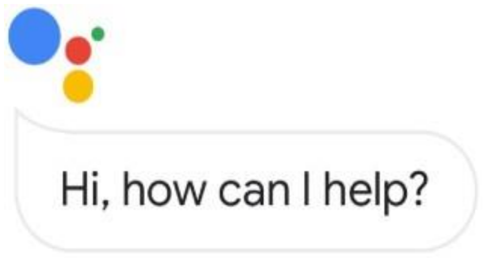
Google has long killed Google Now, but Assistant lives in the same space, fusing these personalized elements with a wide-range of voice control. Google Assistant supports both text or voice entry and it will follow the conversation whichever entry method you’re using.

Figure 1. 6 Google assistant

What can Google Assistant do?

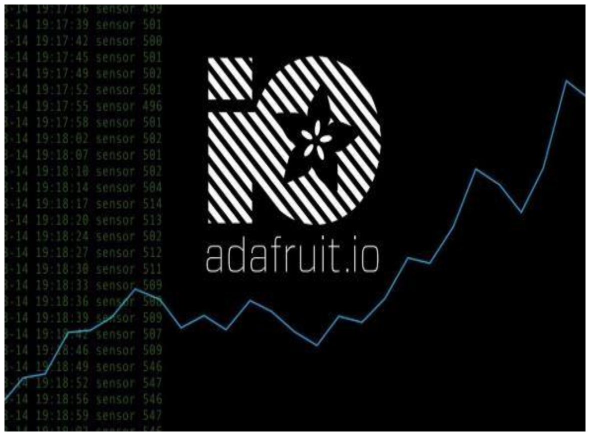
Google Assistant offers voice commands, voice searching, and voice-activated device control, letting you complete a number of tasks after you’ve said the “OK Google” or “Hey, Google” wake words. It is designed to give you conversational interactions.

Google Assistant will:



- Control your devices and your smart home
- Access information from your calendars and other personal information
- Find information online, from restaurant bookings to directions, weather and news
- Control your music
- Play content on your Chromecast or other compatible devices
- Run timers and reminders
- Make appointments and send messages
- Open apps on your phone
- Read your notifications to you
- Real-time spoken translations

1.3.2 Adafruit IO



Adafruit IO is a platform designed to display,respond and interact with the projects's data. Adafruit.io is a cloud service that means we can just run it anddon't have to manage it. We can connect to it over the Internet and is meant primarily for storing and then retrieving data but it can do a lot more than just that. Adafruit.IO can handle and visualize multiple feeds of data.

Figure 1. 7 Adafruit IO

What can Adafruit can do?

- Display your data in real-time, online
- Make your project internet-connected: Control motors, read sensor data, and more!
- Connect projects to web services like Twitter, RSS feeds, weather services, etc.
- Connect your project to other internet-enabled devices

Dashboards are a feature integrated into Adafruit IO which allow you to chart, graph, gauge, log, and display your data. And these dashboards can be accessed and viewed from anywhere in the world.

1.3.3 IFTTT(If This Then That)

IFTTT
Do more with the services you love
OnHub
nest
echo
EVERNOTE
Hundreds of apps and devices you use every day work with IFTTT.

If This Then That, also known as IFTTT is a free web-based service to create chains of simple conditional statements, and helps us to create different commands to perform on the house which we call as applets. An applet is triggered by changes that occur within other web services such as Gmail, Facebook, Telegram, Instagram, or Pinterest and more.

In addition to the web-based application, the service runs on iOS and Android. The automations are accomplished via applets which are sort of like macros that connect multiple apps to run automated tasks where we can can turn on or off an applet.

The IFTTT service is free for users and is simple to use. We can also create our own applets or make variations of existing ones via IFTTT's user-friendly, straightforward interface. We'll learn about creating applets and other stuff later. For now IFTTT basically connects our Google assistant to io.adafruit.com.

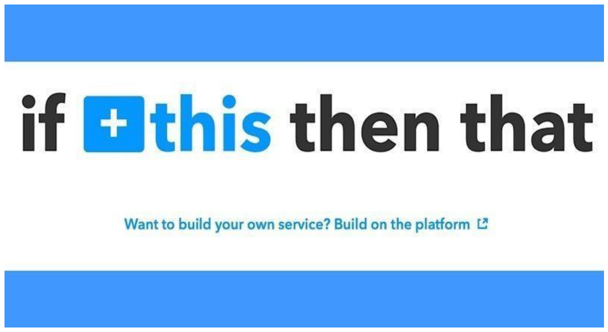
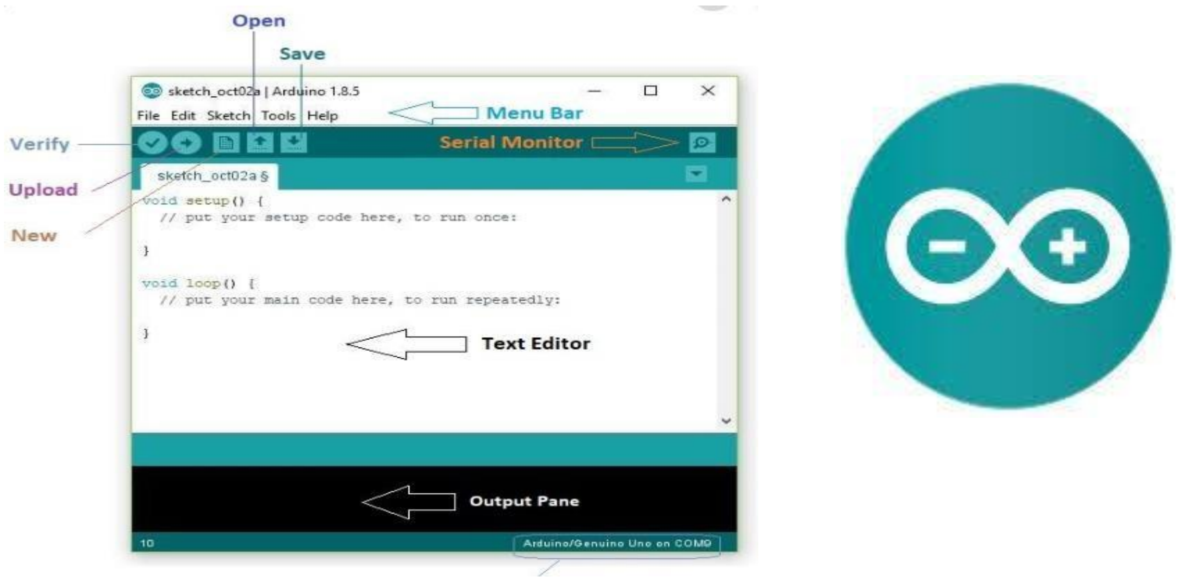


Figure 1. 8 IFTTT

1.3.4 Arduino IDE

The Arduino integrated development environment (IDE) is a cross platform application for Windows, macOS, Linux that is written in the programming language Java that is mainly used for writing, compiling and uploading the code in the Arduino Device. Almost all Arduino modules are compatible with this software that is an open source and is readily available to install and start compiling the code on the go.. It is used to write and upload programs to Arduino compatible boards, but also, with the help of 3rd party cores, other vendor development boards.


Open
Save
sketch_oct02a | Arduino 1.8.5
File Edit Sketch Tools Help
Menu Bar
Verify
Upload
New
Serial Monitor
sketch_oct02a §
Text Editor
Output Pane
Arduino/Genuino Uno on COM9

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

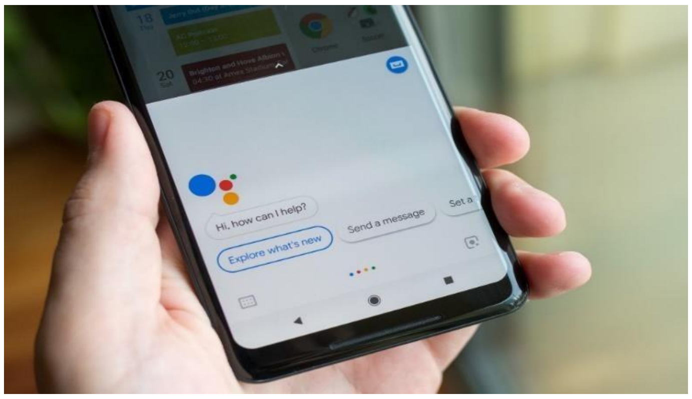
Figure 1. 9 Arduino IDE

The main code, also known as a sketch, created on the IDE platform will ultimately generate a Hex File which is then transferred and uploaded in the controller on the board.The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the wiring project, which provides many common input and output procedures. The IDE environment mainly contains two basic parts:Editor and Compiler where former is used for writing the required code and later is used for compiling and uploading the code into the given Arduino Module.

User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The Arduino IDE employs the program to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

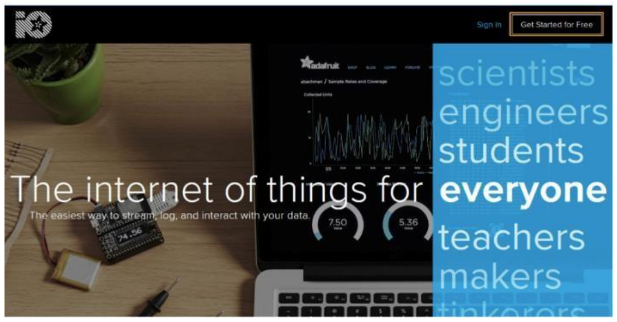
1.4 Procedure

Step 1: Install Google Assistant on a smart phone.

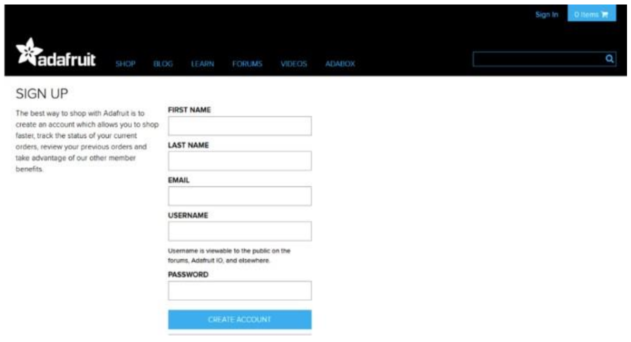


Step 2: Create an Adafruit account.

- Open a browser and go to io.adafruit.com. Click on Get started for free.


Sign In
Get Started for Free
scientists
engineers
students
The internet of things for everyone
The easiest way to stream, log, and interact with your data.
teachers
makers

- Fill in your details and click on Create Account. Note that the site would have navigated to **accounts.adafruit.com**.



SIGN UP

The best way to shop with Adafruit is to create an account which allows you to shop faster, track the status of your current orders, review your previous orders and take advantage of our other member benefits.

FIRST NAME

LAST NAME

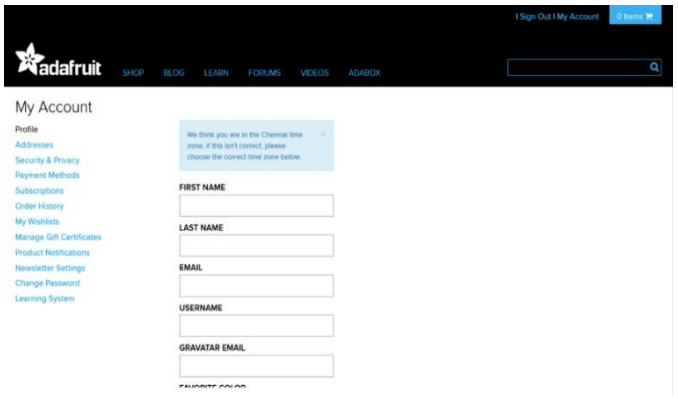
EMAIL

USERNAME

Username is viewable to the public on the forums, Adafruit IO, and elsewhere.

PASSWORD

- When once successfully logged in, this page will be displayed where we can change our details if wished.



My Account

We think you are in the Chennai time zone, if this isn't correct, please choose the correct time zone below.

FIRST NAME

LAST NAME

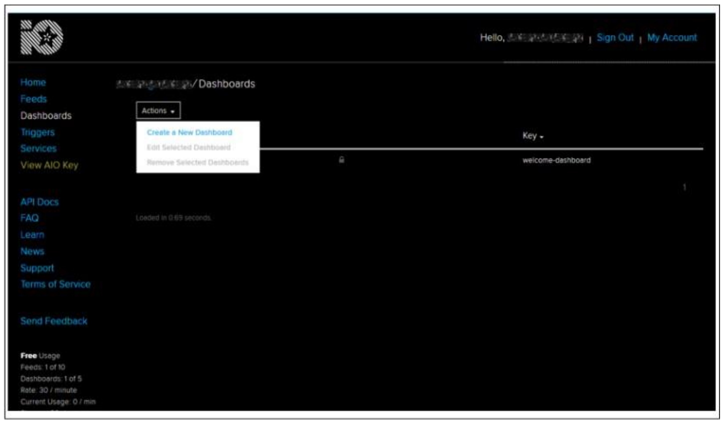
EMAIL

USERNAME

GRAVATAR EMAIL

FAVORITE COLOR

- Now, create dashboard at Adafruit and name the dashboard and press create. This dashboard is a user interface to control things remotely .



Hello, [redacted]

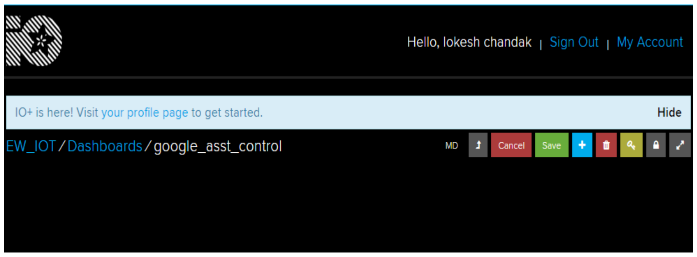
[redacted]/ Dashboards

welcome-dashboard

Loaded in 0.69 seconds.

Free Usage
Feeds: 1 of 10
Dashboards: 1 of 5
Rate: 30 / minute
Current Usage: 0 / min

- After following above steps, provide name to the dashboard and save it. We can see our dashboard as follow

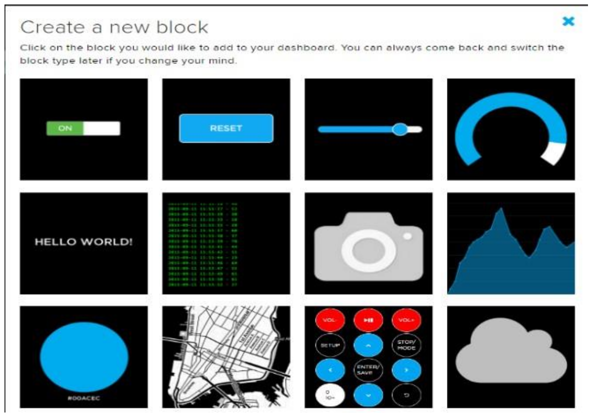


Hello, lokesh chandak | Sign Out | My Account

IO+ is here! Visit your profile page to get started. Hide

EW_IOT / Dashboards / google_asst_control

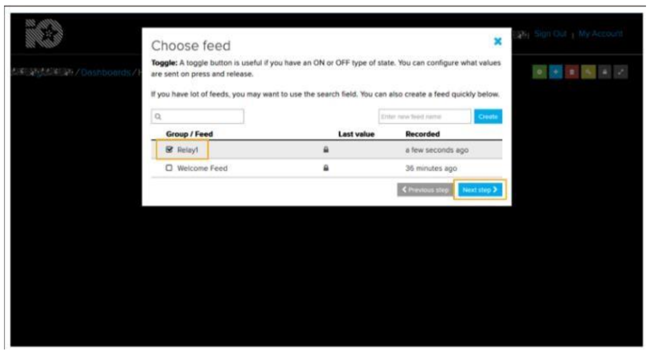
- Now, create feed (user interface) to control light On-Off. To create it, just click on '+' symbol and select toggle block shown below.



Create a new block

Click on the block you would like to add to your dashboard. You can always come back and switch the block type later if you change your mind.

- After selecting toggleenter name of new feed and create it. After creation, select the feed and then click on Next Step.



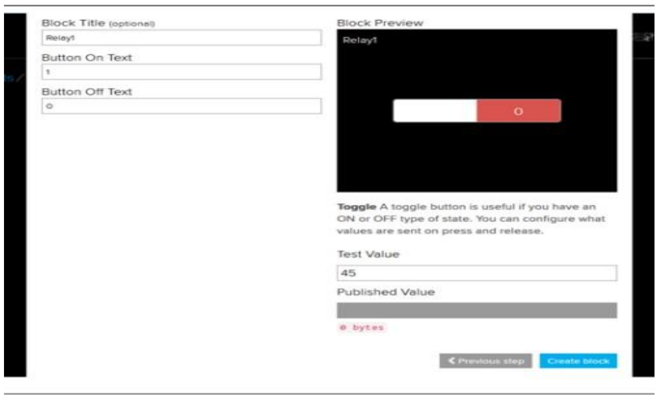
Choose feed

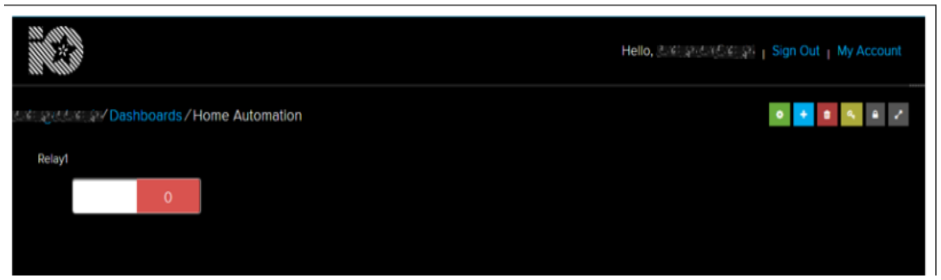
Toggle: A toggle button is useful if you have an ON or OFF type of state. You can configure what values are sent on press and release.

If you have lot of feeds, you may want to use the search field. You can also create a feed quickly below.

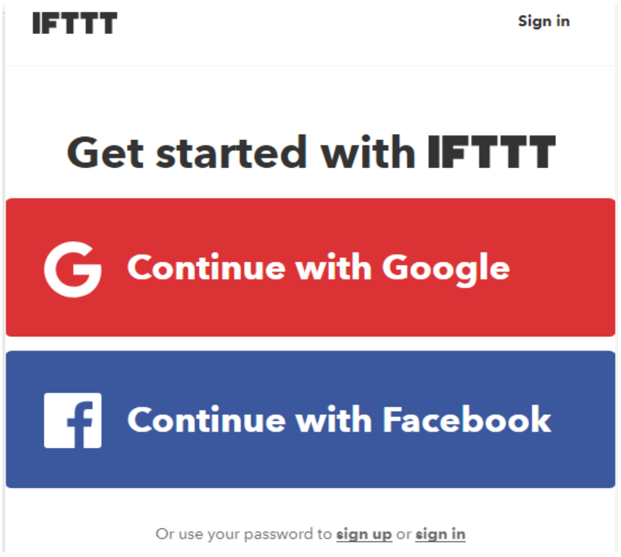
Group / Feed	Last value	Recorded
✓ Relay1	🔒	a few seconds ago
Welcome Feed	🔒	36 minutes ago

- Here, We have used 0(OFF) and 1(ON) text for button and then click on create. This will create toggle button on the dashboard which can be used to control things remotely. Now, my dashboard is ready for IoT application like home automation.


Block Title (optional)
Relay1
Button On Text
1
Button Off Text
0
Block Preview
Relay1
0
Toggle A toggle button is useful if you have an ON or OFF type of state. You can configure what values are sent on press and release.
Test Value
45
Published Value
0 bytes


Hello, [redacted] | Sign Out | My Account
[redacted] / Dashboards / Home Automation
Relay1
0

Step 3: Configure IFTTT.

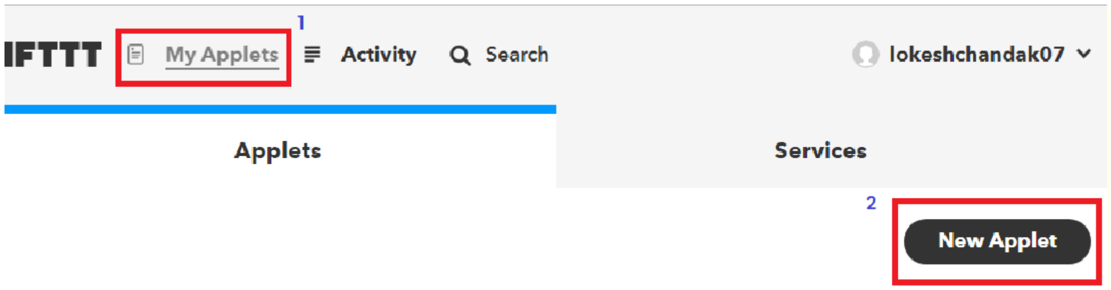


Get started with IFTTT

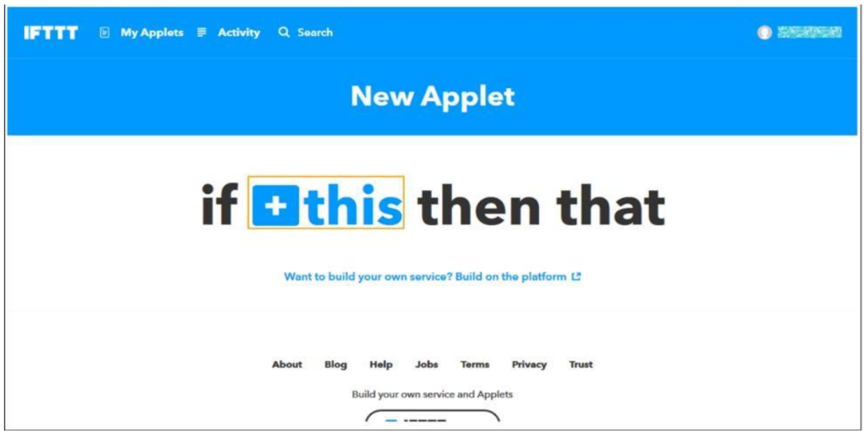
Or use your password to sign up or sign in

Note: Create account on IFTTT by using same e-mail id which you have used for Adafruit.

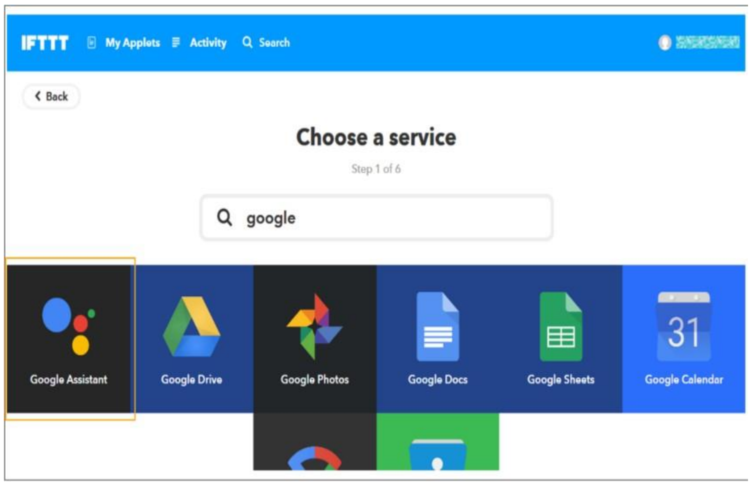
- After account creation, click on **My Applets** and then select **New Applet** shown below.

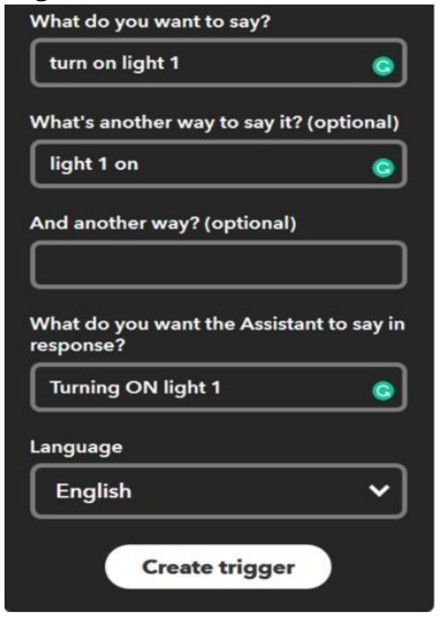


- After selecting a new applet, we get a new page in which we should click on to **THIS** as shown in the below image.



- Search for Google assistant and select on **Say a simple phrase**. Now enter the voice phrases which we will be using as a command for Google assistant.





What do you want to say?
turn on light 1

What's another way to say it? (optional)
light 1 on

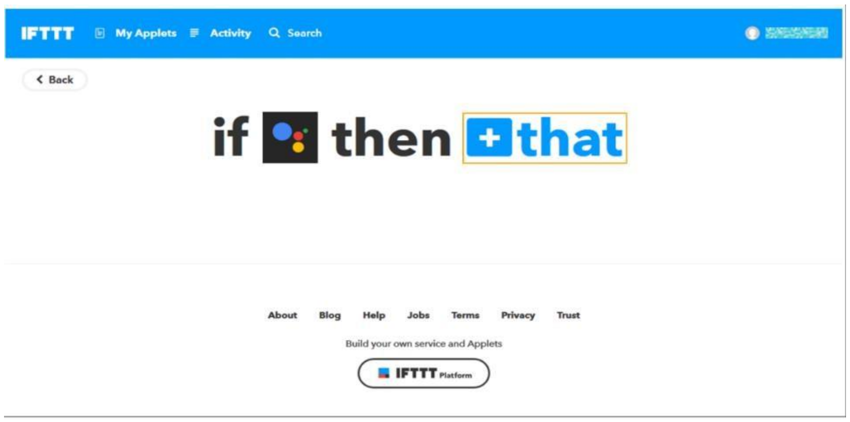
And another way? (optional)

What do you want the Assistant to say in response?
Turning ON light 1

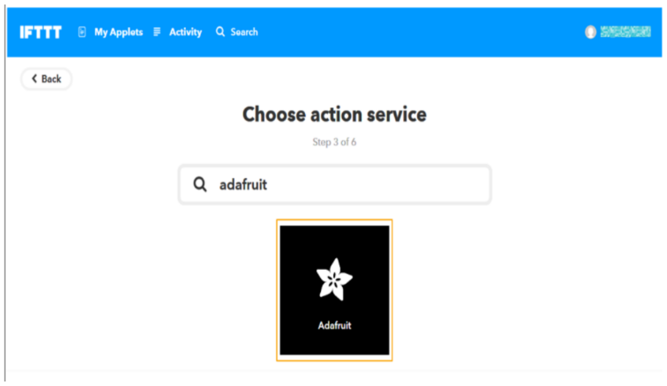
Language
English

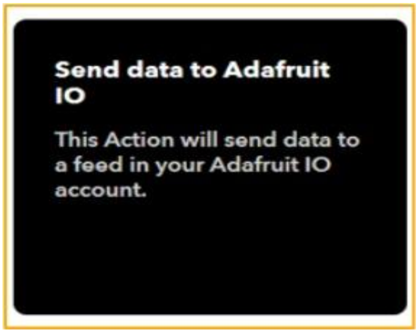
We can enter any phrase as per our application. As we can see, the phrases entered in the above fields is for making**turning on light 1.**

- Now for making light off, we have to create another applet with different phrases.Now, we get another page on which we have to click on **that** option which is used to connect Google Assistant with Adafruit.



- Search and select Adafruit. Press on **Send data to Adafruit IO**`and authorize the link to the Adafruit account.

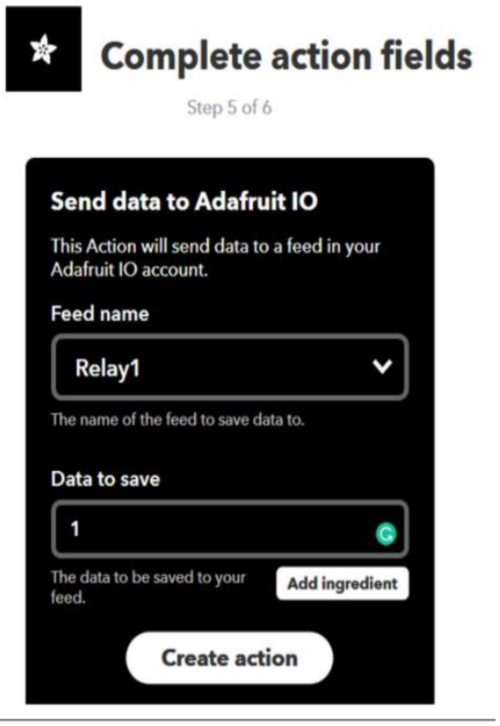




Send data to Adafruit IO

This Action will send data to a feed in your Adafruit IO account.

- Select the feed name and enter data to be send to the feed of Adafruit dashboard(for light we have given Relay1). Select Create action. Click on Finish.



Complete action fields

Step 5 of 6

Send data to Adafruit IO

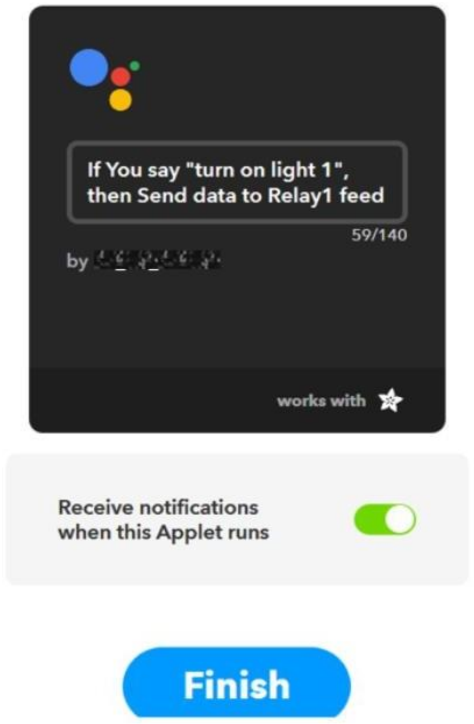
This Action will send data to a feed in your Adafruit IO account.

Feed name: Relay1

The name of the feed to save data to.

Data to save: 1

The data to be saved to your feed.



If You say "turn on light 1", then Send data to Relay1 feed

59/140

by [redacted]

1.5 Circuit Design and Implementation

So, when we use Google Assistant on mobile or any other smart device and give voice command as “Ok Google, Turn Light 1”, applet created in IFTTT receive this command and will send data ‘1’ to the Adafruit feed. This will trigger the event on Adafruitdashboard which is continuously monitored by the microcontroller (here NodeMCU). This microcontroller will take action as per the data change of the Adafruit dashboard.

ESP-WROOM-32
+
+
IFTTT
+
Google Assistant

Figure 1. 10 Implementation

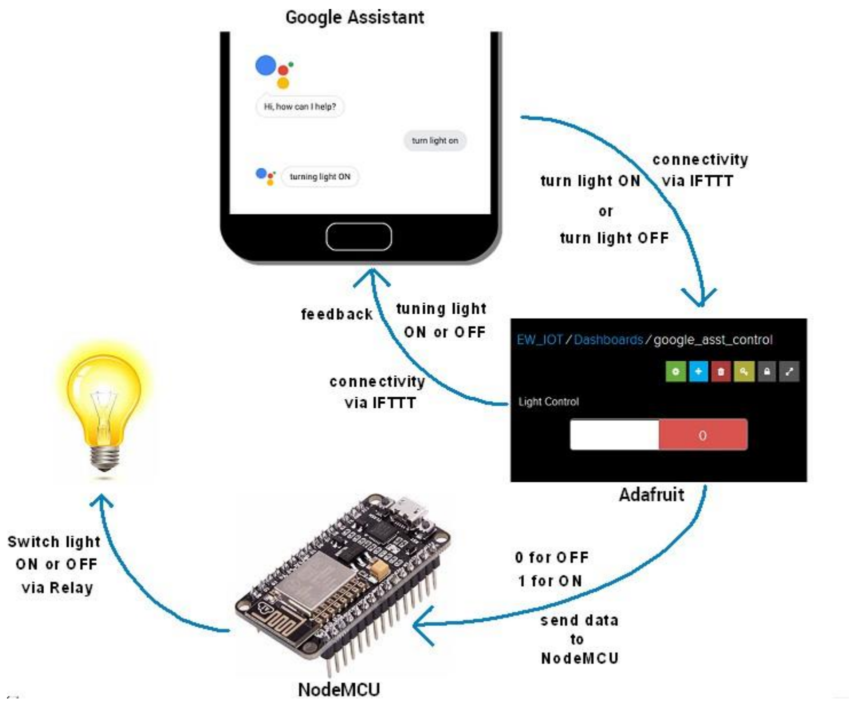


Figure 1. 11 Flow diagram

1.5.1 Circuit diagram

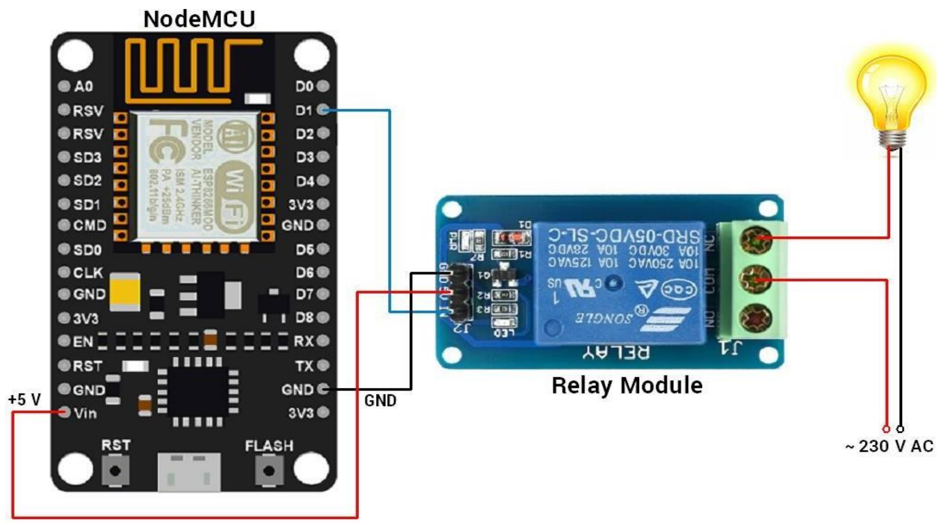


Figure 1. 12 Circuit

Library and packages:

Here, we used the Adafruit MQTT library for receiving data from the Adafruit server. To install this library, select option **Sketch -> Include Library -> Manage Libraries.**

In that library, search for Adafruit MQTT and installed it

Note: Fill in the WiFi SSID and Password to which the microcontroller(NodeMCU) to be connected. And also enter Adafruit account details in the code above(Username and AIO key generated by the Adafruit).

1.6 Result

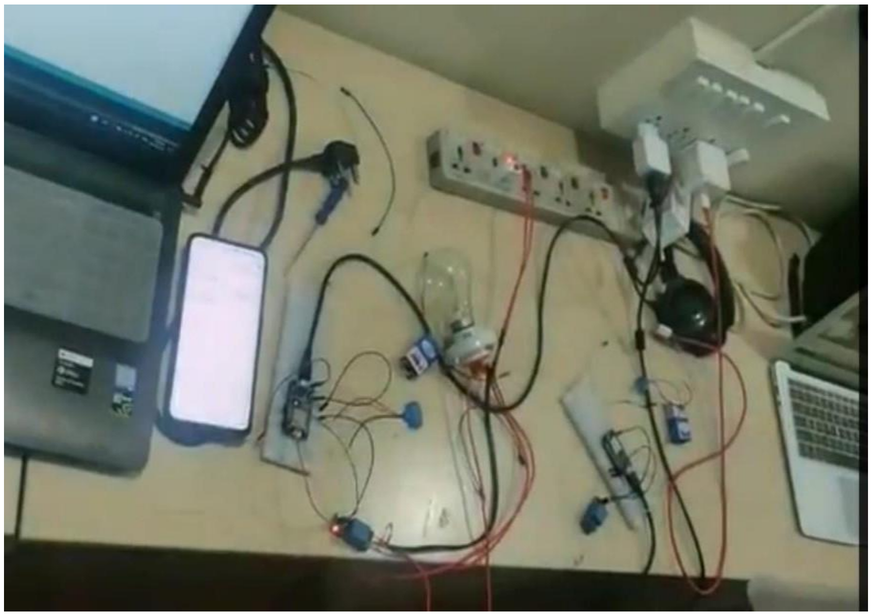


Figure 1. 13 Result1

Initially Electric bulb is Off before giving command to Google assistant.

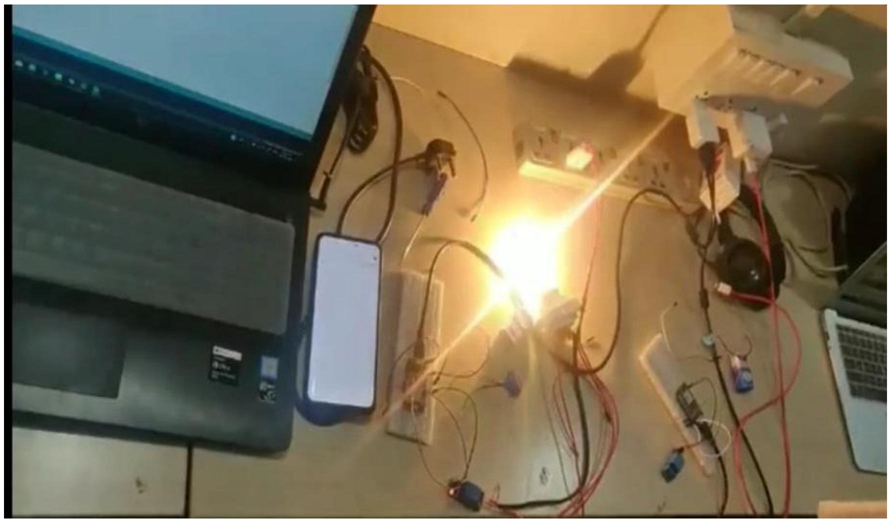


Figure 1. 14 Result2

Immediately after giving the command "OK Google. Turn on Light 1"switching on of electric bulb.

The result was positive and the system responded well. The diagram above shows the complete prototype implementation of the proposed system.

NOTE: 5V/1A Output, Mobile Chargers were used to power the NodeMCU and the Relay Board

1.6.1 Reasons for choosing NodeMCU

1. **Inexpensive:** NodeMCU is indeed a low cost microcontroller which comes upto Rs. 265.
2. **More compatible development :** Since more and more people knew, bought, and tried using ESP8266, they found the disadvantage of the chip and tried to improve it. For example, to develop ESP8266, people had to understand FreeRTOS and coded with the professional C language. But an Arduino IDE plug-in was then developed so that developers can use Arduino IDE to write ESP8266 command code to diminish the difficulty of development.
3. **Flexible Design and Enhanced Function:** Besides the software progressive contributed by the community, ESP8266 chip and board have been enhanced as well, like chip clock rate acceleration, a new analog digital converter (ADC) to improve the sensitivity.
4. **Convenient Application Development :** Various programming languages, thus can be used to develop ESP8266, such as JavaScript, Java, Python, Objective C(Common programming languageon Mac and iOS), PHP, Ruby, etc. API can be applied to smartphoneon Mac and iOS), PHP, Ruby, etc. API can be applied.

1.6.2 Advantages

1. **Safety.** The ability to control small appliances and lighting with your fingertips anywhere you are will add safety in your home.
2. **Security.** The ability to lock the door through your phone is one of the greatest benefits of home automation. This will give you peace of mind knowing that the door is close and not guessing.
3. **Convenience.** The ability to control everything with your fingertips is very convenient. You never leave the house without your wallet, keys and your smart phone. With our smart phone always with us, we can easily monitor our home and control everything with just touch of a finger.

4. **Saves Time.** Since we are living in a very fast-paced environment, we don't even have time to worry about our home.

5. **Save Money.** This is the biggest advantage of home automation. With the ability to control the light, whether dimming or turning on/off on specific time will saves homeowner a great ton of money.

1.6.3 Disadvantages

Now that you have known about the major benefits of smart homes, there are many disadvantages of smart homes as well.

1. **Cost**: The biggest problems, con or disadvantage of a smart home system is the cost. There are quite a number of companies that provide the smarty home system, but all of them are quite expensive. This is something that only a few can afford.

2. **Dependency on Internet**: The basic requirement for the smart home system is the internet. Without a good and strong internet connection, you will not be able to take control of this. If there is no internet connection for some reason, there is no other way through which you can access and control your system.

3. **Dependency on Professionals**: In case there is a problem with the smart home system, you cannot simply call a handyman or someone similar to repair or manage the bug. You will have to depend on the professionals. Only the company professionals can help you to handle the problem.

1.7 Conclusion and Future work

This project was a wonderful learning experience. Not only did it allow us to explore and learn about different topics such as NodeMCU, Ifttt, adafruit, etc., this was also a wonderful opportunity to experience working as a part of a company. The aim was to propose a cost effective voice controlled (Google Assistant) home automation controlling general appliances found in one's home. The approach discussed in the paper was successful as Google Assistant Controlled Home Automation design was successfully implemented. We, as a team were given a small glimpse of what it would be like to work in the industry. This project is far from complete.

We were able to fix the many problems or challenges we faced as much as possible. However, some obstacles we were not able to solve within our given month of internship.

1.7.1 Scope for improvement

The future scope foras Google Assistant Controlled Home Automation can be huge. There are many fators to improve on to make as Google Assistant Controlled Home Automation more powerful, intelligent, scalable, and to become better overall for home automation. To make the system respond more faster own private Blynk server can be made. It always has a scope for improvement. One just needs to put on a thinking cap and try and make the system more better.

Some of the main noticeable improvements that could be made as scope for future reference were

- Switching delay of the appliances.
- Improving the frame rate of the NodeMCU.
- More energy can be conserved by ensuring occupation of the house before turning on devices based and checking brightness and turning off the lights if not necessary.

CHAPTER 2:

2 PEOPLE COUNTER USING OPENCV AND PYTHON

2.1 Introduction

Digital Image processing is growing fast, this due to increase in available machine learning techniques where the developers can access with cloud. People counter Is a measure of people traversing a certain passage or entrance which can be implemented in various domains such as in school and public libraries, airports, malls. In provides different applications of security and business management. We use Raspberry Pi with PiCamera module where it captures and streams video in which we keep track on count of number of people entering and leaving a particular area of intertest. People counting is one of the widely studied and commercially exploited subject.

- **Line counting at the reference**
- **Counting within an area**
- **Crowd detection within an area**
- **Queue detection at the counter**

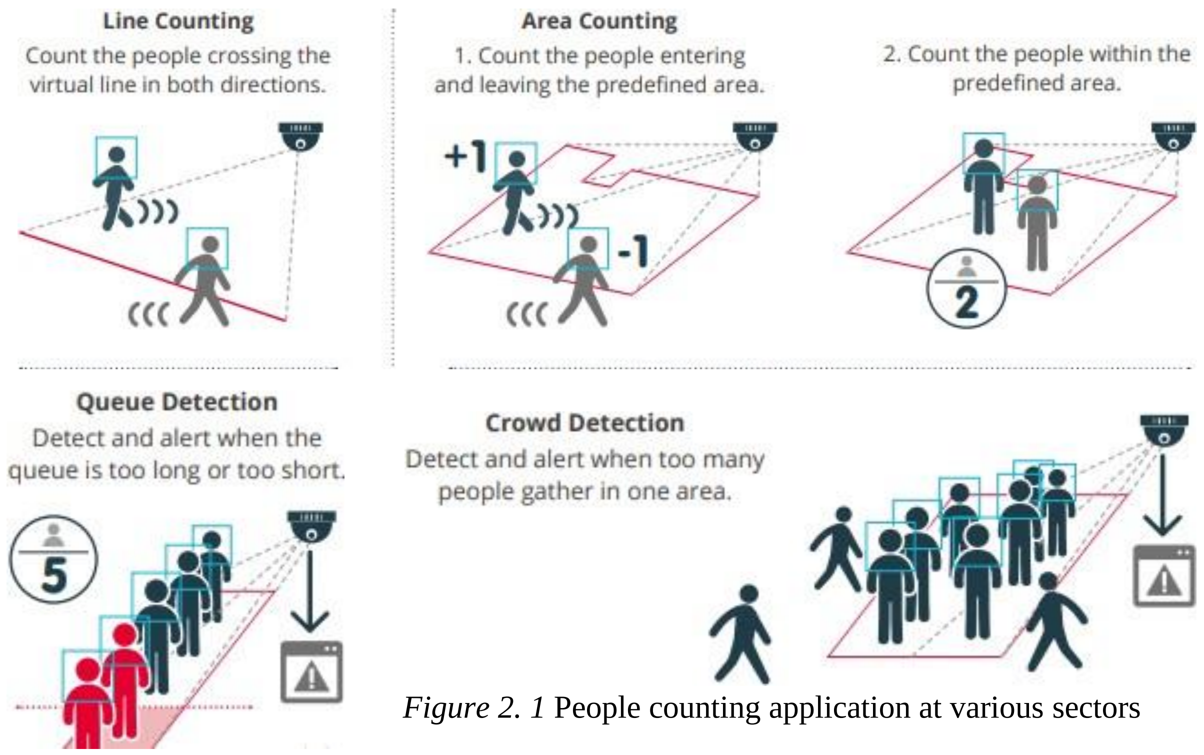

Line Counting
Count the people crossing the virtual line in both directions.
Area Counting
1. Count the people entering and leaving the predefined area.
+1
-1
2. Count the people within the predefined area.
2
Queue Detection
Detect and alert when the queue is too long or too short.
5
Crowd Detection
Detect and alert when too many people gather in one area.

Figure 2. 1 People counting application at various sectors

We shall build a People Counter using OpenCV and Python which will count number of people who are heading either upwards or downwards with respect to line of reference in real-time. So, we put forth a solution which is based on a single ceiling-mounted camera, which identifies people by background extraction of the camera image.

Here, to count the number of people entering from the door, Raspberry Pi board has been used which is SBC, on which we interfaced a Picamera. Picamera is used for detecting people on live video stream. The Raspberry Pi board is connected to the monitor through HDMI port, for getting the results. The monitor shows the number of people captured by Picamera.

2.2 Hardware Requirements

2.2.1 Raspberry Pi

Raspberry Pi is the name of series of small single-board computers developed by the Raspberry Pi foundation, a UK which aims to educate people in computing and create access to computing education. It is a little device that enables people to learn how to program in languages like Scratch and Python. Raspberry use projects.

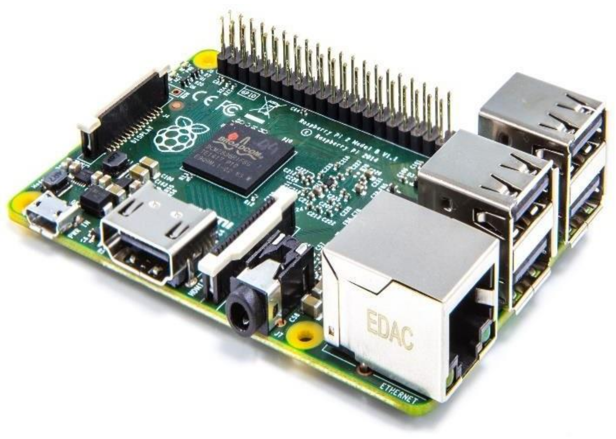


Figure 2. 2 Raspberry Pi

It is an economical computer that runs on linux, also provides a set of GPIO(general purpose input/output) pins that allow us to control electronic components for physical computing and explore the Internet of Things(IoT).

The board comes with a HDMI port and USB ports meant to be used for a HDMI monitor and USB mouse and keyboard. This is the normal way to use a Raspberry Pi. The most commonly used OS for the Raspberry Pi is Raspbian which is a conventional Linux based OS. Since it is Linux based, a Terminal (similar to Command Prompt on Windows) can be used to execute commands on it.

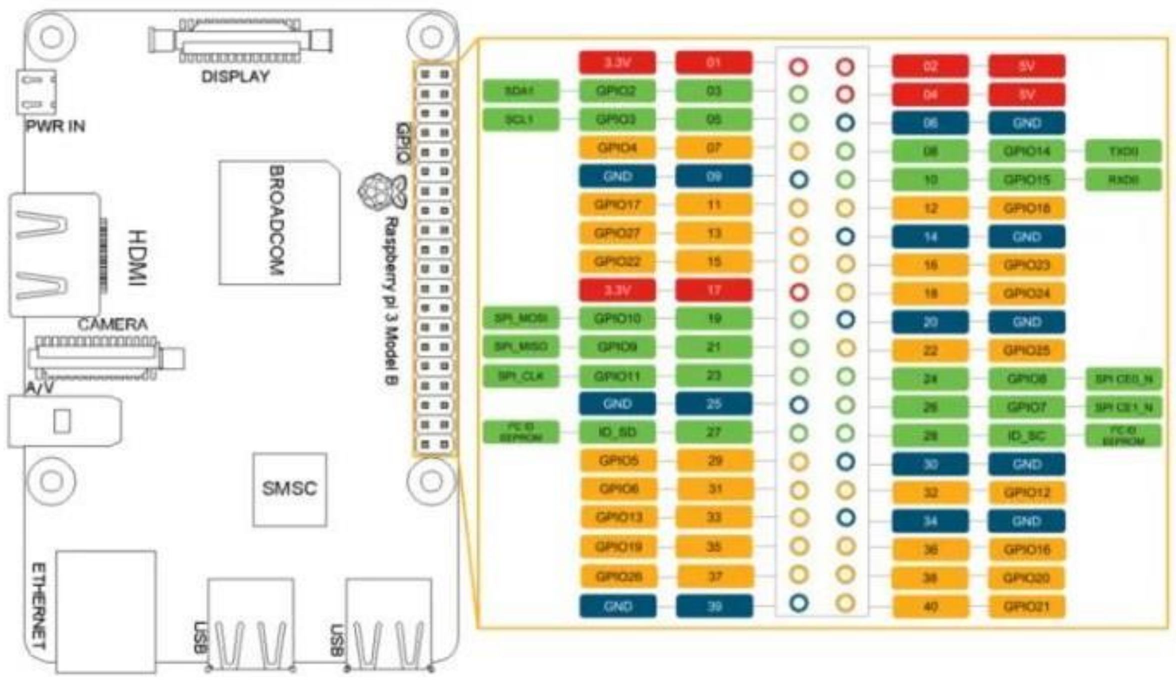


Figure 2. 3 Raspberry Pi 3B+ GPIO pinout

The Raspberry Pi 3 has a total of 40 pins categorized as follows.

- 5V (2 pins)
- 3.3V (2 pins)
- GND (2 pins)
- GPIO (34 pins)

GPIO (General Purpose Input Output) pins

The GPIO pins are used to interface external hardware. The voltage of these pins varies from 0 to 3.3V. Do ensure that the input voltage is not above 3.3V as it might cause permanent damage to the Raspberry pi board.

- Output function of GPIO – A GPIO pin designated as an output pin can be set to high (3V3) or low (0V).

- Input function of GPIO – 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 the software.

- Other functions of GPIO – Other than simple input and output devices, the GPIO pins can be used with a variety of alternative functions, some are available on all pins, others on 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); CE2 (GPIO16)
- I2C (Inter-Integrated Circuit)
- Data: (GPIO2); Clock (GPIO3)
- EEPROM Data: (GPIO0); EEPROM Clock (GPIO1)
- Serial
- TX (GPIO14); RX (GPIO15)

2.2.2 PiCamera Module

Connect the Raspberry Pi Camera Module to the Raspberry Pi and take pictures, record video, and apply image effects.

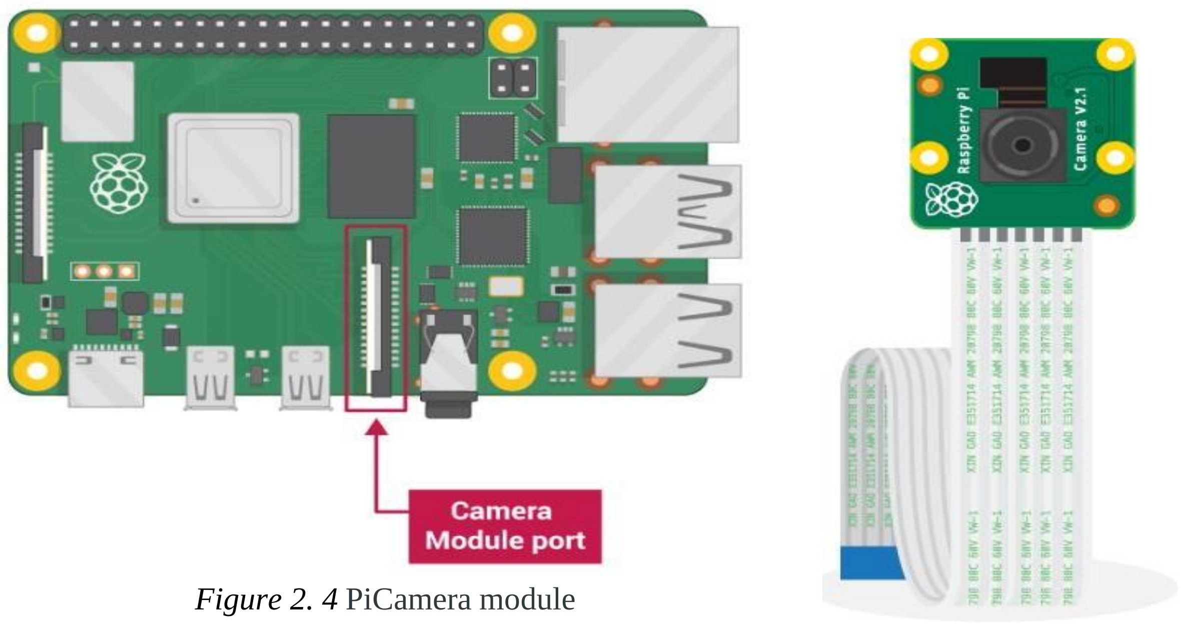


Figure 2. 4 PiCamera module

There are two versions of the Camera Module:

- The standard version(camera module V2), which is designed to take pictures in normal light
- The NoIR version(Pi NoIR Camera V2), which doesn’t have an infrared filter, so you can use it together with an infrared light source to take pictures in the dark

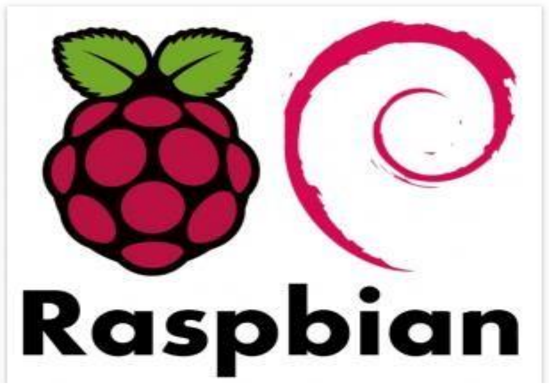
Features of Pi Camera module V2:

- Fixed focus lens on-board
- 8 megapixel native resolution sensor-capable of 3280 x 2464 pixel static images
- Supports 1080p30, 720p60 and 640x480p90 video
- Size 25mm x 23mm x 9mm
- Weight just over 3g
- Connects to the Raspberry Pi board via a short ribbon cable (supplied)
- Camera v2 is supported in the latest version of Raspbian, Raspberry Pi's preferredoperating system

2.3 Software Requirements

2.3.1 Raspbian

Raspbian is a free operating system based on Debian optimized for the Raspberry Pi hardware. An operating system is the set of basic programs and utilities that make your Raspberry Pi run. However, Raspbian provides more than a pure OS it comes with over 35,000 packages, pre-compiled software bundled in a nice format for easy installation on your Raspberry Pi.



The initial build of over 35,000 Raspbian packages, optimized for best performance on the Raspberry Pi, was completed in June of 2012. However, Raspbian is still under active development with an emphasis on improving the stability and performance of as many Debian packages as possible.

2.3.1 OpenCV (Open Source Computer Vision Library)

OpenCV is library containing programming functions which mainly aims at real-time computer vision and machine learning software. It was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products.

The library has numerous optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc.

Currently OpenCV supports a wide variety of programming languages like C++, Python, Java etc. and is available on different platforms including Windows, Linux, OS X, Android, iOS etc. Also, interfaces based on CUDA & OpenCL are also under the active development of high-speed GPU operations.

2.3.2 OpenCV-Python

Python is a general-purpose programming language which became very popular in short time mainly because of its simplicity and code readability. It enables the programmer to express their ideas in fewer lines of code without reducing any readability.

Python's standard library is very extensive, offering a wide range of facilities as indicated by the long table of contents listed below. The library contains built-in modules (written in C) that provide access to system functionality such as file I/O that would otherwise be inaccessible to Python programmers, as well as modules written in Python that provide standardized solutions for many problems that occur in everyday programming. Some of these modules are explicitly designed to encourage and enhance the portability of Python programs by abstracting away platform-specifics into platform-neutral APIs.

2.4 Installing Libraries and Packages:

In the above build project we have used Raspberry Pi model 3 B+ with latest version of Raspbian having an OpenCV inside virtual environment with OpenCV3 correctly installed in it.

In the new terminal lets enter into the virtual environment :

```
source ~/.profile
```

```
workon cv
```

Inside virtual environment enter Python interpreter andconfirm that you are running the 3.5 (or above) version.

```
python
```

Inside the interpreterimport the OpenCV library and check for OpenCV version which appears 3.3.0 or any superior version

```
import cv2
```

```
cv2.__version__
```

2.4.1 Python Libraries and Packages for People Counting:

To build people counter we require a number of different python packages, including:

- **NumPy**
- **OpenCV**
- **dlib**
- **imutils**

Install NumPy by giving the command:

```
pip install numpy
```

Also,
```
pip install scipy
```

Install the package OpenCV 3.0 or greater in the OS by running the command n the terminal:

```
pip install opencv-contrib-python``
```

When once Python3 and OpenCV are installed check the configuration by running:

```
import cv2
```

```
cv2.__version__
```

Install imutils:

```
pip install imutils
```

Or,
```
sudo pip3 install --upgrade imutils
```

2.4.2 Object detection *v/s* Object tracking

The basic difference between object detection vs object tracking is that we apply object detection for determining where in an image/frame an object is. An object detector is also typically more computationally expensive, and hence slower, than an object tracking algorithm. Examples of object detection algorithms include Haar cascades, HOG + Linear SVM, and deep learning-based object detectors such as Faster R-CNNs, YOLO, and Single Shot Detectors (SSDs).

Whereas an object tracker, on the other hand, assigns a unique ID to that particular object and track the object as it moves around a video stream, predicting a new object location in the next frame.

2.4.3 Combining both object detection and object tracking

The concept of object detection and object tracking into a single algorithm, typically divided into two phases:

- **Phase 1- Detecting***:*

During the detection phase the computation is more expensive object tracker to detect if new objects have entered our view, and (2) see if we can find objects that were "lost" during the tracking phase. For each detected object we create or update an object tracker with the new bounding box coordinates.We only run this phase once every N frames since object detector is more computationally expensive.

- **Phase 2-Tracking**:

When we are in the "tracking" phase for each of our detected objects, we create an object tracker to track the object as it moves around the frame. We'll continue tracking until we've reached the *N*-th frame and then re-run our object detector. The entire process then repeats.

2.4.4 Combining object tracking algorithms:

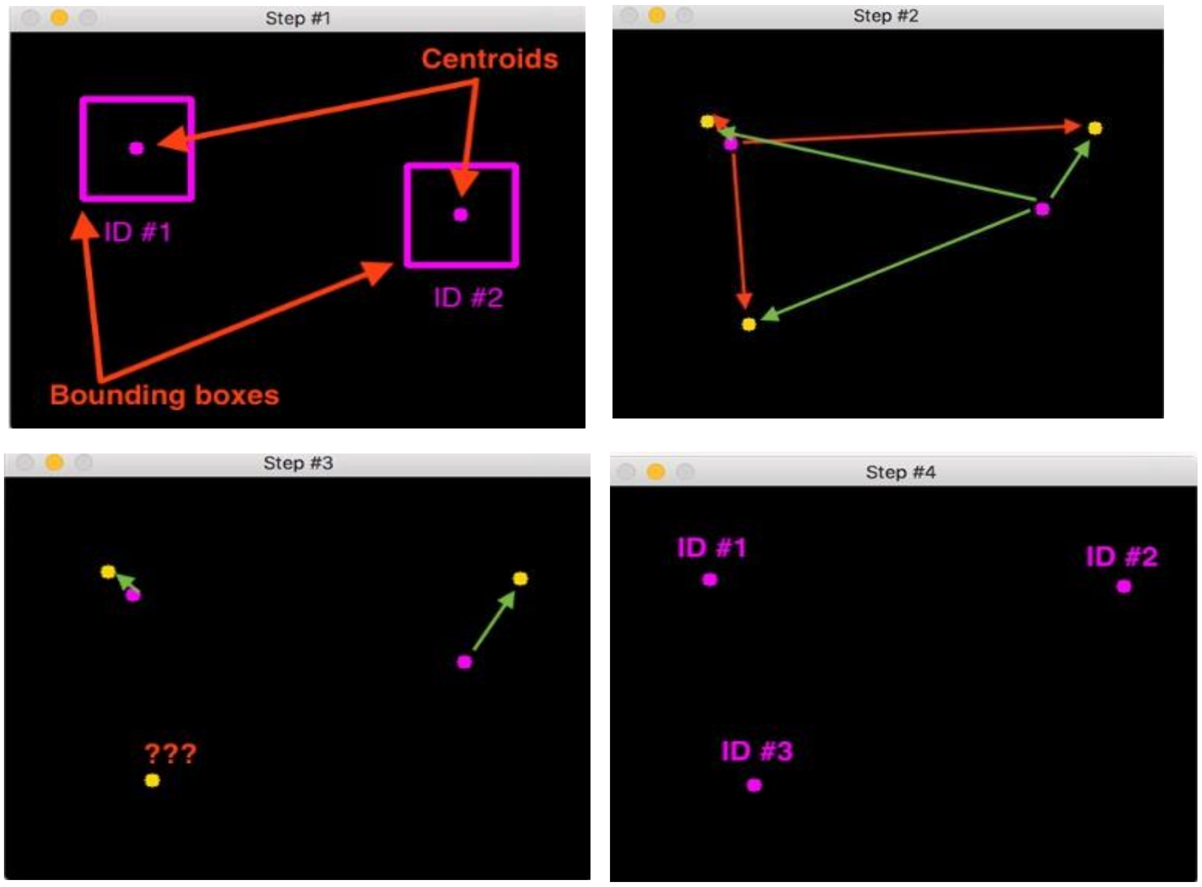


Figure 2. 5 Algorithms

Step 1: To build a simple object tracking via centroids script with Python, the first step is to accept bounding box coordinates and use them to compute centroids.

Step 2: Three objects are present in this image. We need to compute the Euclidean distance between each pair of original centroids(red) and new centroids(green).

Step 3: A simple centroid object tracking method has associated objects with minimized object distances. When once we have obtained the Euclidean distances we attempt to associate object IDs.

Step 4: In the above object tracking example, we come across a new object that wasn't matched with an existing object, so it is registered as object ID #3.

2.5 Implementation of People Counter with OpenCV & Python

2.5.1 Creating of Trackable Objects

In order to track and count an object in a video stream, we need an easy way to store information.

- It's object ID.
- It's previous centroids Whether or not the object has already been counted

2.6 People Counting Results

Here we are allowed to execute two different commands for two different scenarios to record the count.

- Open up a terminal and execute the following command to obtain the counting in the captured video:

```
python people_counter.py --prototxtmobilenet_ssd/MobileNetSSD_deploy.prototxt \
--model mobilenet_ssd/MobileNetSSD_deploy.caffemodel \
--input videos/example_01.mp4 --output output/output_01.avi
```

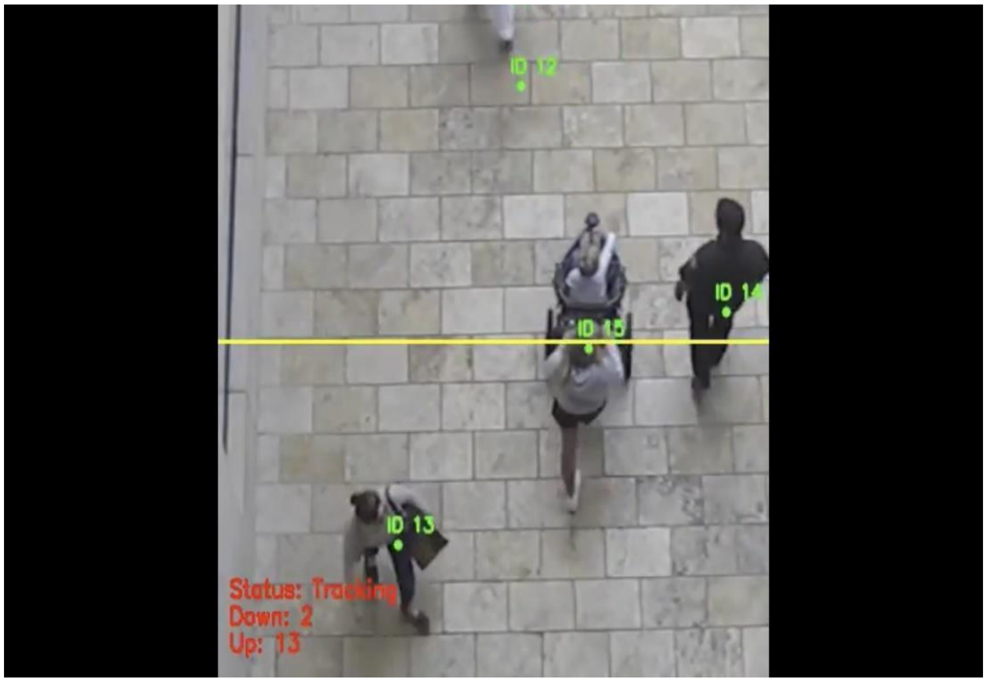


Figure 2. 6 Result1

Here you can see that our person counter is counting the number of people who:

1. Are entering the department store (down)
2. And the number of people who are leaving (up)

- Another example of people counting with OpenCV To obtain the counting in the live stream from the Picamera execute the below command:
  python people_counter.py --prototxt mobilenet_ssd/MobileNetSSD_deploy.prototxt
  **--model mobilenet_ssd/MobileNetSSD_deploy.caffemodel **
  --output output/webcam_output.avi

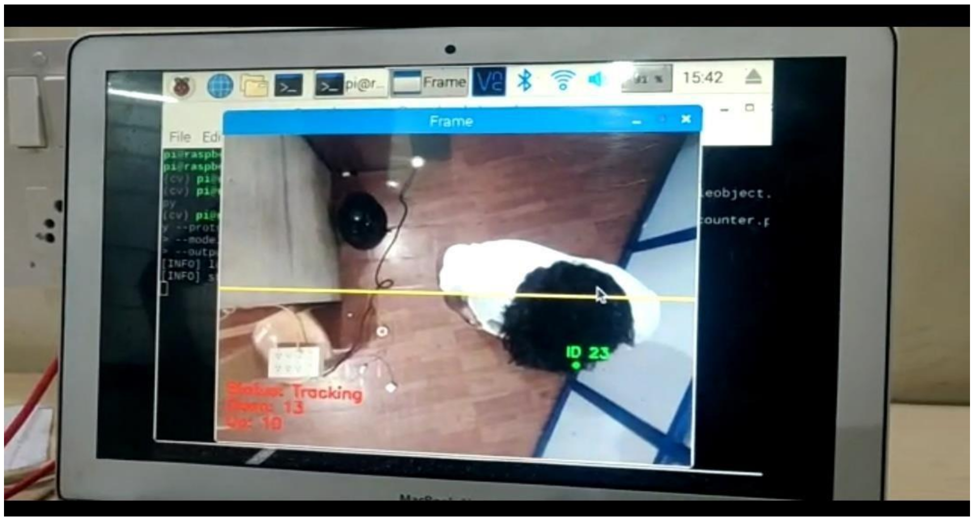


Figure 2. 7 Result2

Executed result of people counter using PiCamera focusing inside the office area where it could successfully count number of members entering inside(down count) as well as coming out (Up count)

2.6.1 Benefits of People Counting

1. **Real Time People Counting:** Count the number of people entering, exiting and passing by your store in real time.
2. **Conversion Rate**: Calculate the conversion rates of each store by knowing how many of your customers generate your sales.
3. **Benchmark**: Benchmark high-performing stores and optimize your lowest performing stores to reach their performance.
4. **Peak Hours**: Discover your power hours in which your stores generate most traffic, and have biggest sale opportunities.
5. **Staff Optimization**: Optimize staff operation in accordance to the number of visitors and their needs within your stores during power hours.
6. **Group Counting:** Purify your conversion rate by consolidating groups of visitors, families and couples as 1 potential buyer.

2.6.2 Use of People Counting in Different Industries

1. **People Counting In Retail:** Retail stores are dynamic environments with ever-changing customer behavior so people counting is vital for retailers for them to know the exact number of visitors, the exact number of visitors that turned into buyers and their optimal number of staff to customer ratio to maximize customer satisfaction.
2. **People Counting In Libraries & Museums**: Libraries and museums rely on government funding, In order to receive the necessary amount of grants, they need to be able to accurately judge their footfall. With people counting data, they will be able to know how many people visit the premises. Hence, they can receive their government grants accordingly.
3. **People Counting In Airports**: Airports are one of the busiest and most dynamic places in the world.The most important aspect for airports is to allocate staff accordingly, so people

counting information is crucial for them to know exactly how many people come in and out and also at the security check lines.

4. **People Counting In Events & Exhibitions**: Events and expos are important grounds to make lasting impressions and present their products and services. With people counting, sponsors and owners know the exact number of visitors, peak hours and staff allocation accordingly with the number of visitors.

5. **People Counting In Shopping Malls**: Shopping Malls draw lots of visitor traffic; not only because of shopping opportunities, but also for entertainment value. With people counting, you will be able to see which areas are most crowded and attractive, thus, shape your marketing attractions and planning accordingly.

2.7 Conclusion and Future Scope

We have learned how to build a people counter using OpenCV and Python.Uptill now people counting system is used for the purposes such as to facilitate security management as well as urban planning. In military application for instance in urban warfare, soldiers might not be able to check every room of building. Sending a camera into a room that could autonomously report how many people are present can help soldiers assess threat level. But apart from this we can use this system in shopping malls. We can count number of people going in particular section and if there is too much crowd in that section then we can segregate the crowd by applying some technique.
Our trained People counter is

- Capable of running in real-time on a standard CPU
- Leverages two separate object tracking algorithms, including both centroid tracking and correlation filters for improved tracking accuracy
- Applies both a “detection” and “tracking” phase, making it capable of detecting new people and picking up people that may have been “lost” during the tracking phase.

Practical Python and OpenCV is meant to be a gentle introduction to the world of computer vision and image processing. The same type of system for counting foot traffic with OpenCV can be used to count automobile traffic with OpenCV.
The benefit of this hybrid approach is that we can apply highly accurate object detection methods *without* as much of the computational burden. We will be implementing such a tracking system to build our people counter

CHAPTER 3:

3 A REAL TIME FACE RECOGNITION

3.1 Introduction

Face recognition is a method of identifying or verifying the identity of an individual using their face. It is a biometric technology used for mapping the facial features, patterns, and/or texture of an individual from a digital image or live video feed for the purpose of identity storage and verification.

In this we learn step by step to use Raspberry Pi and PiCamera module to recognize faces in real time. We utilize Open Source Computer Vision Library(OpenCV) where we will be concentrating on Raspberry Pi (Raspbian as OS)and Python.

3.2 Image Processing with OpenCV

Visual information one of the most important type of information perceived, processed and interpreted by human brain. Image processing is a method to perform some operations on an image, in order to extract some useful information out of it. An image is nothing more than a two-dimensional matrix (3-D in case of colored images).

Image processing with Python & Open-CV aims at providing an overview of Open-CV library, itsfunctions, application & capabilities along with getting your hands adept with it. Image processing basically means performing processes on an image with the help of software. The goal of applying processes like smoothing, sharpening, contrasting, stretching etc. on an image can be to increase its readability or to enhance its quality or even transform the image.

3.3 Installing Libraries and Packages

In the above build project we have used Raspberry Pi model 3 B+ with latest version of Raspbian having an OpenCV inside virtual environment with OpenCV3 correctly installed in it.

In the new terminal lets enter into the virtual environment:

```
source ~/.profile
 workon cv
```

Inside virtual environment enter Python interpreter andconfirm that you are running the 3.5 (or above) version.

```
python
```

Inside the interpreterimport the OpenCV library and check for OpenCV version which appears 3.3.0 or any superior version

```
import cv2
```

```
cv2.__version__
```

3.4 Phases for Face recognition:

To create and complete the Face Recognition process, we must work in 3 distinct phases :

- Face Detection and Data Gathering
- Train the Recognizer
- Face Recognition

3.4.1 Phase 1: Face Detection*:*

The most basic task on Face Recognition is of course, "Face Detecting". Before anything, you must "capture" a face (Phase 1) in order to recognize it, when compared with a new face captured on future (Phase 3).

The most common way to detect a face (or any objects), is using the Haar Cascade classifier

Object Detection using Haar feature-based cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones in their paper, "Rapid Object Detection using a Boosted Cascade of Simple Features" in 2001. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images.

Here we will work with face detection. Initially, the algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then we need to extract features from it. The good news is that OpenCV comes with a trainer as well as a detector. If you want to train your own classifier for any object like car, planes etc. you can use OpenCV to create one.

If you do not want to create your own classifier, OpenCV already contains many pre-trained classifiers for face, eyes, smile, etc.

This Phase involves capturing of a set of images which adds to the data set so as to compare the faces which are recognized in phase 3.The most common way to detect a face (or any objects), is using the Haar Cascade classifier.

Object Detection using haar feature-based cascade classifiers is an effective object detection method. It is a machine learning based approach where a cascade function is trained from a lot of positive & negative images andOpenCV comes with a trainer as well as a detector.

The program to detect a face using Python and OpenCV is **faceDetection.py**:

Initially create a directory where you develop your project

```
mkdir FacialRecognitionProject
```

In this directory we shall create our project and to be saved Facial Classifier before hand which is to be downloaded haarcascade_frontalface_default.xml.

Now create a subdirectory where we will store our facial samples and name it “dataset”:

```
mkdir dataset
```

```
faceCascade = cv2.CascadeClassifier('Cascades/haarcascade_frontalface_default.xml')
```

This is the line that loads the "classifier" (that must be in a directory named "Cascades/", under your project directory).

Then, we will set our camera and inside the loop, load our input video in grayscale mode (same we saw before).

Now we must call our classifier function, passing it some very important parameters, as scale factor, number of neighbors and minimum size of the detected face.

```
faces = faceCascade.detectMultiScale(        gray,            scaleFactor=1.2,
minNeighbors=5,              minSize=(20, 20) )
```

Where,

gray is the input grayscale image.

scaleFactor is the parameter specifying how much the image size is reduced at each image scale. It is used to create the scale pyramid.

minNeighbors is a parameter specifying how many neighbors each candidate rectangle should have, to retain it. A higher number gives lower false positives.

minSize is the minimum rectangle size to be considered a face.

The function will detect faces on the image. Next, we must "mark" the faces in the image, using, for example, a blue rectangle. This is done with this portion of the code:

```
for (x,y,w,h) in faces:     cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)     r
oi gray = gray[y:y+h, x:x+w]      roi color = img[y:y+h, x:x+w]
```

If faces are found, it returns the positions of detected faces as a rectangle with the left up corner (x,y) and having "w" as its Width and "h" as its Height ==> (x,y,w,h).

Once we get these locations, we can create an "ROI" (drawn rectangle) for the face and present the result with imshow() function.

Run the above python Script on your python environment, using the **Rpi Terminal:**

```
python faceDetection.py
```

Likewise for Data Gathering

What we do here, is starting from last step (Face Detecting), we will simply create a dataset, where we will store for each id, a group of photos in gray with the portion that was used for face detecting.

The code is very similar to the code that we saw for face detection. What we added, was an "input command" to capture a user id, that should be an integer number (1, 2, 3, etc)

```
face id = input('\n enter userid end press  ==>  ')
```

And for each one of the captured frames, we should save it as a file on a "dataset" directory:

```
cv2.imwrite("dataset/User." + str(face_id) + '.' + str(count) + ".jpg", gray[y
:y+h,x:x+w])
```

Note that for saving the above file, you must have imported the library "os". Each file's name will follow the structure:

```
User.face_id.count.jpg
```

For example, for a user with a face_id = 1, the 4th sample file on dataset/ directory will be something like:

```
User.1.4.jpg
```

Run the Python script and capture a few Ids. You must run the script each time that you want to aggregate a new user (or to change the photos for one that already exists).

3.4.2 Phase 2: Train the Recognizer

In the second phase, we must take all user data from our dataset and train the OpenCV Recognizer. This is done directly by a specific OpenCV function which results in a .yml file that will be saved on a "trainer/" directory.

Create another subdirectory to store the trained data:

```
mkdir trainer
```

second python script: face_training.py

Install PIL library on the Raspberry Pi

Confirm if you have the PIL library installed on your Rpi. If not, run the below command in Terminal:

```
pip install pillow
```

After compiling, a file named a file named “trainer.yml” will be saved in the trainer directory that was previously created.

We will use as a recognizer, the LBPH (LOCAL BINARY PATTERNS HISTOGRAMS) Face Recognizer, included on OpenCV package. We do this in the following line:

```
recognizer = cv2.face.LBPHFaceRecognizer_create()
```

The function "getImagesAndLabels (path)", will take all photos on directory: "dataset/", returning 2 arrays: "Ids" and "faces". With those arrays as input, we will "train our recognizer":

```
recognizer.train(faces, ids)
```

As a result, a file named "trainer.yml" will be saved in the trainer directory that was previously created by us.

That's it! I included the last print statement where I displayed for confirmation, the number of User's faces we have trained.

Every time that you perform Phase 1, Phase 2 must also be run.

3.4.3 Phase 3: Recognizer

The final phase of the project. Here, we capture face on our camera and if this person had his face captured and trained before, the recognizer will make a “prediction” returning its id and an index, shown how confident the recognizer is with its match.

3rd phase python script: face_recognition.py

Here we include a new array, so it will display names instead of numbered ids.

We are including here a new array, so we will display "names", instead of numbered ids:

```
names = ['None', 'Nikhil', 'ayub', 'Charan']
```

So, for example: Nikhil will the user with id = 1; ayub with id=2, etc.

Next, we will detect a face, same we did before with the haasCascade classifier. Having a detected face we can call the most important function in the above code:

```
id, confidence = recognizer.predict(gray portion ofthe face)
```

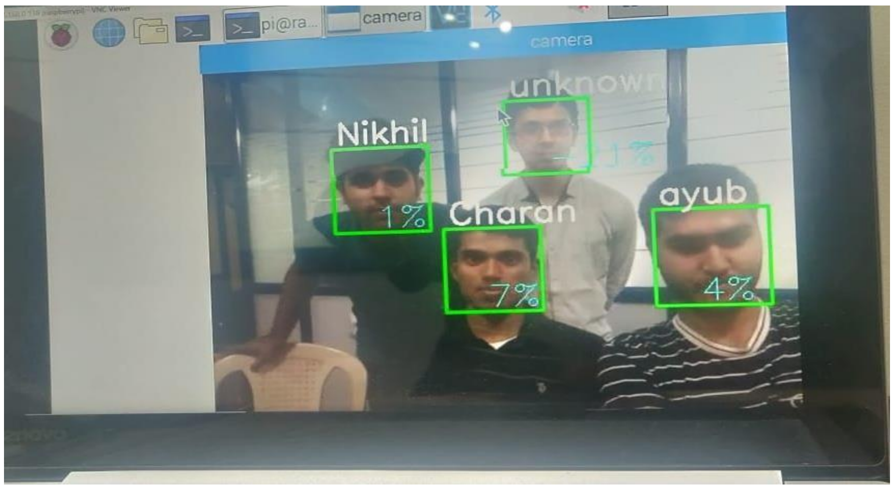
The recognizer.predict (), will take as a parameter a captured portion of the face to be analyzed and will return its probable owner, indicating its id and how much confidence the recognizer is in relation with this match.

Note that the confidence index will return "zero" if it will be cosidered a perfect match

And at last, if the recognizer could predict a face, we put a text over the image with the probable id and how much is the "probability" in % that the match is correct ("probability" = 100 - confidence index). If not, an "unknown" label is put on the face.

3.5 Result

The result of all the 3 phases of output has been obtained successfully .Here are some of the pictures of our project.



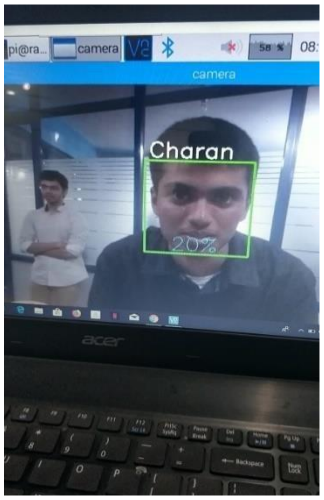


Figure 3. 1 Result snaps

3.5.1 Advantages

Facial recognition enables the computerized and automated processing of biometric data based on the digital image or live video feed of a person for a variety of purposes or applications.

1. **Security Through Biometric Authentication**: One of the benefits of facial recognition system is its application in biometrics. It can be used as a part of identification and access control systems in organizations, as well as personal devices, such as smartphones.

2. **Automated Image Recognition**: The system can also be used to enable automated image recognition capabilities. Through the social networking site which can recognize photos of its users and allow automated linking or tagging to individual user profiles.

3. **Deployment in Security Measures**: Facial recognition system also involves its application in law enforcement and security systems. Automated biometric identity allows less intrusive monitoring and mass identification.

4. **Human-Computer Interaction**: The system also supports virtual reality and augmented realityapplications. In both VR and AR applications, the system facilitates further human-computer interaction.

5. **Equips Devices with Added Functionalities**: It is also worth noting that equipping devices with facial recognition capabilities means expanding their capabilities.

3.5.2 Disadvantages

Facial recognition system has drawbacks and limitations revolving around concerns over its effectiveness and controversial applications.

1. **Issues About Reliability and Efficiency:** A notable disadvantage of facial recognition system is that it is less reliable and efficient than other biometric systems such as fingerprint.

2. **Further Reports About It Reliability:** Several reports have pointed out the ineffectiveness of some systems. For example, a report by an advocacy organization noted that the systems used by law enforcement agencies in the U.K. had an accuracy rate of only 2 percent. Applications in London and Tampa, Florida did not result in better law enforcement according to another report.

3. **Concerns About Racial Bias:** A study by the American Civil Liberties Union revealed that the Recognition technology developed by Amazon failed nearly 40 percent false matches in tests involving people of color. In general, the system has been criticized for perpetuating racial bias due to false matches.

4. **Issues with Privacy Laws:** Alleged conflict with privacy rights is another disadvantage of facial recognition. In Illinois, for example, its Biometric Information Privacy Act requires affirmative consent for companies to collect biometric data. The fact that the system enables less intrusive mass identification also translates to mass surveillance, which according to groups, is a violation of privacy rights.

3.6 Conclusion

Face recognition is an emerging technology that can provide many benefits. Face recognition can save resources and time, and even generate new income streams, for companies that implement it right. Face recognition technology has come a long way in the last decade. However, next generation face recognition systems are going to have widespread application in smart environments where these computers and machines are more like helpful assistants. To achieve this goal computers must be able to reliably identify nearby people within pattern of normal human interactions.

The computational models, which were implemented in this project, were chosen after extensive research, and the successful testing results. Using proposed method we improve the face recognition system under illumination variation and non-frontal view. (Proposed approach is very simple in term of calculation. (Improve Speed of recognition. (It required only one scanning without any need to a complicated analysis.

4 APPENDIX

4.1 Controlling of Home Appliances using Google Assistant

```
#include <ESP8266WiFi.h>

#include "Adafruit_MQTT.h"

#include "Adafruit_MQTT_Client.h"


#define Relay1          D1

#define Relay2          D2

#define Relay3          D3

#define Relay4          D5


#define WLAN_SSID       "xxxxxxxx"          // Your WiFi SSID

#define WLAN_PASS       "xxxxxxxx"      // Your WiFi password



#define AIO_SERVER      "io.adafruit.com" //Adafruit Server

#define AIO_SERVERPORT  1883

#define AIO_USERNAME    "xxxxxxxx"          // Adafruit Username

#define AIO_KEY         "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx" // AIO Key from Adafruit


//WIFI CLIENT

WiFiClient client;
```

```
Adafruit_MQTT_Client mqtt(&client, AIO_SERVER, AIO_SERVERPORT,
AIO_USERNAME, AIO_KEY);

Adafruit_MQTT_Subscribe Light1 = Adafruit_MQTT_Subscribe(&mqtt,
AIO_USERNAME"/feeds/Relay1"); // Feeds name should be same everywhere

Adafruit_MQTT_Subscribe Light2 = Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME
"/feeds/Relay2");

Adafruit_MQTT_Subscribe Light3 = Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME
"/feeds/Relay3");

Adafruit_MQTT_Subscribe Light4 = Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME
"/feeds/Relay4");

Adafruit_MQTT_Subscribe LightAll = Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME
"/feeds/RelayAll");


void MQTT_connect();


void setup() {

  Serial.begin(115200);


  pinMode(Relay1, OUTPUT);

  pinMode(Relay2, OUTPUT);

  pinMode(Relay3, OUTPUT);

  pinMode(Relay4, OUTPUT);


  // Connect to WiFi access point.

  Serial.println(); Serial.println();

  Serial.print("Connecting to ");

  Serial.println(WLAN_SSID);
```

```
WiFi.begin(WLAN_SSID, WLAN_PASS);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println();

  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());

  mqtt.subscribe(&Light1);
  mqtt.subscribe(&Light3);
  mqtt.subscribe(&Light2);
  mqtt.subscribe(&Light4);
  mqtt.subscribe(&LightAll);
}

void loop() {

  MQTT_connect();

  Adafruit_MQTT_Subscribe *subscription;
  while ((subscription = mqtt.readSubscription(20000))) {
    if (subscription == &Light1) {
```

```
Serial.print(F("Got: "));
      Serial.println((char *)Light1.lastread);
      int Light1_State = atoi((char *)Light1.lastread);
      digitalWrite(Relay1, Light1_State);


    }
    if (subscription == &Light2) {
      Serial.print(F("Got: "));
      Serial.println((char *)Light2.lastread);
      int Light2_State = atoi((char *)Light2.lastread);
      digitalWrite(Relay2, Light2_State);
    }
    if (subscription == &Light3) {
      Serial.print(F("Got: "));
      Serial.println((char *)Light3.lastread);
      int Light3_State = atoi((char *)Light3.lastread);
      digitalWrite(Relay3, Light3_State);
    }
    if (subscription == &Light4) {
      Serial.print(F("Got: "));
      Serial.println((char *)Light4.lastread);
      int Light4_State = atoi((char *)Light4.lastread);
      digitalWrite(Relay4, Light4_State);
}
     if (subscription == &LightAll) {
        Serial.print(F("Got: "));
        Serial.println((char *)LightAll.lastread);
        int LightAll_State = atoi((char *)LightAll.lastread);
        digitalWrite(Relay1, LightAll_State);
        digitalWrite(Relay2, LightAll_State);
        digitalWrite(Relay3, LightAll_State);
        digitalWrite(Relay4, LightAll_State);

    }
  }

}

void MQTT_connect() {
  int8_t ret;

  if (mqtt.connected()) {
    return;
  }

  Serial.print("Connecting to MQTT... ");
  uint8_t retries = 3;
  while ((ret = mqtt.connect()) != 0) {
    Serial.println(mqtt.connectErrorString(ret));
    Serial.println("Retrying MQTT connection in 5 seconds...");
    mqtt.disconnect();
    delay(5000);
    retries--;
    if (retries == 0) {
      while (1);
    }
  }
  Serial.println("MQTT Connected!");
```

4.2 People Counter Using Opencv And Python

- Open the **trackableobject.py** file and insert the following code:

```
class TrackableObject:
	def_init_(self, objectID, centroid):
		# store the object ID, then initialize a list of centroids
		# using the current centroid
		self.objectID = objectID
		self.centroids = [centroid]

		# initialize a boolean used to indicate if the object has
		# already been counted or not
		self.counted = False
```

Create a file with name **people_counter.py** with following code:

```
# import the necessary packages
from pyimagesearch.centroidtracker import CentroidTracker
from pyimagesearch.trackableobject import TrackableObject
from imutils.video import VideoStream
from imutils.video import FPS
import numpy as np
import argparse
import imutils
import time
import dlib
import cv2

# construct the argument parse and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-p", "--prototxt", required=True,
	help="path to Caffe 'deploy' prototxt file")
ap.add_argument("-m", "--model", required=True,
	help="path to Caffe pre-trained model")
ap.add_argument("-i", "--input", type=str,
	help="path to optional input video file")
ap.add_argument("-o", "--output", type=str,
	help="path to optional output video file")
ap.add_argument("-c", "--confidence", type=float, default=0.4,
	help="minimum probability to filter weak detections")
ap.add_argument("-s", "--skip-frames", type=int, default=30,
	help="# of skip frames between detections")
args = vars(ap.parse_args())

# initialize the list of class labels MobileNet SSD was trained to
# detect
CLASSES = ["background", "aeroplane", "bicycle", "bird", "boat",
	"bottle", "bus", "car", "cat", "chair", "cow", "diningtable",
	"dog", "horse", "motorbike", "person", "pottedplant", "sheep",
	"sofa", "train", "tvmonitor"]

# load our serialized model from disk
print("[INFO] loading model...")
net = cv2.dnn.readNetFromCaffe(args["prototxt"], args["model"])

# if a video path was not supplied, grab a reference to the webcam
if not args.get("input", False):
	print("[INFO] starting video stream...")
	vs = VideoStream(src=0).start()
	time.sleep(2.0)
# otherwise, grab a reference to the video file
else:
	print("[INFO] opening video file...")
	vs = cv2.VideoCapture(args["input"])
```

```
# initialize the video writer (we'll instantiate later if need be)
writer = None

# initialize the frame dimensions (we'll set them as soon as we read
# the first frame from the video)
W = None
H = None

# instantiate our centroid tracker, then initialize a list to store
# each of our dlib correlation trackers, followed by a dictionary to
# map each unique object ID to a TrackableObject
ct = CentroidTracker(maxDisappeared=40, maxDistance=50)
trackers = []
trackableObjects = {}

# initialize the total number of frames processed thus far, along
# with the total number of objects that have moved either up or down
totalFrames = 0
totalDown = 0
totalUp = 0

# start the frames per second throughput estimator
fps = FPS().start()
```

```
# loop over frames from the video stream
while True:
        # grab the next frame and handle if we are reading from either
        # VideoCapture or VideoStream
        frame = vs.read()
        frame = frame[1] if args.get("input", False) else frame

        # if we are viewing a video and we did not grab a frame then we
        # have reached the end of the video
        if args["input"] is not None and frame is None:
                break

        # resize the frame to have a maximum width of 500 pixels (the
        # less data we have, the faster we can process it), then convert
        # the frame from BGR to RGB for dlib
        frame = imutils.resize(frame, width=500)
        rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

                # if the frame dimensions are empty, set them
# start the frames per second throughput estimator
fps = FPS().start()

# loop over frames from the video stream
while True:
        # grab the next frame and handle if we are reading from either
        # VideoCapture or VideoStream
        frame = vs.read()
        frame = frame[1] if args.get("input", False) else frame

        # if we are viewing a video and we did not grab a frame then we
        # have reached the end of the video
        if args["input"] is not None and frame is None:
                break

        # resize the frame to have a maximum width of 500 pixels (the
        # less data we have, the faster we can process it), then convert
        # the frame from BGR to RGB for dlib
        frame = imutils.resize(frame, width=500)
        rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

        # if the frame dimensions are empty, set them
        if W is None or H is None:
        (H, W) = frame.shape[:2]
# if we are supposed to be writing a video to disk, initialize
        # the writer
        if args["output"] is not None and writer is None:
                fourcc = cv2.VideoWriter_fourcc(*"MJPG")
                writer = cv2.VideoWriter(args["output"], fourcc, 30,
                        (W, H), True)
```

```
	# initialize the current status along with our list of bounding
	# box rectangles returned by either (1) our object detector or
	# (2) the correlation trackers
	status = "Waiting"
	rects = []

	# check to see if we should run a more computationally expensive
	# object detection method to aid our tracker
	if totalFrames % args["skip_frames"] == 0:
		# set the status and initialize our new set of object trackers
		status = "Detecting"
		trackers = []

		# convert the frame to a blob and pass the blob through the
		# network and obtain the detections
		blob = cv2.dnn.blobFromImage(frame, 0.007843, (W, H), 127.5)
		net.setInput(blob)
detections = net.forward()

		# loop over the detections
		for i in np.arange(0, detections.shape[2]):
			# extract the confidence (i.e., probability) associated
			# with the prediction
			confidence = detections[0, 0, i, 2]
```

```
                    # filter out weak detections by requiring a minimum
                    # confidence
                    if confidence > args["confidence"]:
                        # extract the index of the class label from the
                        # detections list
                        idx = int(detections[0, 0, i, 1])

                        # if the class label is not a person, ignore it
                        if CLASSES[idx] != "person":
                            continue

                        # compute the (x, y)-coordinates of the bounding box
                        # for the object
                        box = detections[0, 0, i, 3:7] * np.array([W, H, W, H])
                        (startX, startY, endX, endY) = box.astype("int")
# construct a dlib rectangle object from the bounding
                        # box coordinates and then start the dlib correlation
                        # tracker
                        tracker = dlib.correlation_tracker()
                        rect = dlib.rectangle(startX, startY, endX, endY)
                        tracker.start_track(rgb, rect)

                        # add the tracker to our list of trackers so we can
                        # utilize it during skip frames
                        trackers.append(tracker)
```

```
        # otherwise, we should utilize our object *trackers* rather than
        # object *detectors* to obtain a higher frame processing throughput
        else:
            # loop over the trackers
            for tracker in trackers:
                # set the status of our system to be 'tracking' rather
                # than 'waiting' or 'detecting'
                status = "Tracking"

                # update the tracker and grab the updated position
                tracker.update(rgb)
                pos = tracker.get_position()
    # unpack the position object
                startX = int(pos.left())
                startY = int(pos.top())
                endX = int(pos.right())
                endY = int(pos.bottom())

                # add the bounding box coordinates to the rectangles list
                rects.append((startX, startY, endX, endY))

        # draw a horizontal line in the center of the frame -- once an
        # object crosses this line we will determine whether they were
        # moving 'up' or 'down'
```

```
		cv2.line(frame, (0, H // 2), (W, H // 2), (0, 255, 255), 2)


		# use the centroid tracker to associate the (1) old object
		# centroids with (2) the newly computed object centroids
		objects = ct.update(rects)


		# loop over the tracked objects
		for (objectID, centroid) in objects.items():
			# check to see if a trackable object exists for the current
			# object ID
			to = trackableObjects.get(objectID, None)
	# if there is no existing trackable object, create one
			if to is None:
				to = TrackableObject(objectID, centroid)


			# otherwise, there is a trackable object so we can utilize it
			# to determine direction
			else:
				# the difference between the y-coordinate of the *current*
				# centroid and the mean of *previous* centroids will tell
				# us in which direction the object is moving (negative for
				# 'up' and positive for 'down')
				y = [c[1] for c in to.centroids]
				direction = centroid[1] - np.mean(y)
				to.centroids.append(centroid)
		# check to see if the object has been counted or not
		if not to.counted:
			# if the direction is negative (indicating the object
			# is moving up) AND the centroid is above the center
			# line, count the object
			if direction < 0 and centroid[1] < H // 2:
				totalUp += 1
				to.counted = True
			# if the direction is positive (indicating the object
			# is moving down) AND the centroid is below the
			# center line, count the object
			elif direction > 0 and centroid[1] > H // 2:
				totalDown += 1
				to.counted = True

	# store the trackable object in our dictionary
	trackableObjects[objectID] = to


	# draw both the ID of the object and the centroid of the
	# object on the output frame
	text = "ID {}".format(objectID)
	cv2.putText(frame, text, (centroid[0] - 10, centroid[1] - 10),
		cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2)
	cv2.circle(frame, (centroid[0], centroid[1]), 4, (0, 255, 0), -1)
```

```
		# construct a tuple of information we will be displaying on the
		# frame
		info = [
			("Up", totalUp),
			("Down", totalDown),
			("Status", status),
		]
	# loop over the info tuples and draw them on our frame
		for (i, (k, v)) in enumerate(info):
			text = "{}: {}".format(k, v)
			cv2.putText(frame, text, (10, H - ((i * 20) + 20)),
				cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 0, 255), 2)

		# check to see if we should write the frame to disk
		if writer is not None:
			writer.write(frame)

		# show the output frame
		cv2.imshow("Frame", frame)
		key = cv2.waitKey(1) & 0xFF

		# if the `q` key was pressed, break from the loop
		if key == ord("q"):
			break
	# increment the total number of frames processed thus far and
	# then update the FPS counter
	totalFrames += 1
	fps.update()

# stop the timer and display FPS information
fps.stop()
print("[INFO] elapsed time: {:.2f}".format(fps.elapsed()))
print("[INFO] approx. FPS: {:.2f}".format(fps.fps()))

# check to see if we need to release the video writer pointer
if writer is not None:
	writer.release()

# if we are not using a video file, stop the camera video stream
if not args.get("input", False):
	vs.stop()

# otherwise, release the video file pointer
else:
	vs.release()

# close any open windows
cv2.destroyAllWindows()
```

4.3 A Real Time Face Recognition

Phase 1: Face Detection

The program to detect a face using Python and OpenCV is **faceDetection.py**

```
import numpy as np
import cv2
faceCascade = cv2.CascadeClassifier('Cascades/haarcascade_frontalface_default.xml')
cap = cv2.VideoCapture(0)
cap.set(3,640) # set Width
cap.set(4,480) # set Height
while True:
    ret,  img  =  cap.read()
    img = cv2.flip(img, -1)
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = faceCascade.detectMultiScale(
        gray,
        scaleFactor=1.2,
        minNeighbors=5,
        minSize=(20, 20)
    )
    for (x,y,w,h) in faces:
        cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)
        roi_gray = gray[y:y+h, x:x+w]
        roi_color = img[y:y+h, x:x+w]
    cv2.imshow('video',img)
    k = cv2.waitKey(30) & 0xff
    if k == 27: # press 'ESC' to quit
        break
```

```
cap.release()
cv2.destroyAllWindows()
```

In this directory we shall create our project and to be saved Facial Classifier before hand which is to be downloaded haarcascade_frontalface_default.xml.

Now create a subdirectory where we will store our facial samples and name it “dataset”:

```
import cv2
import os

cam = cv2.VideoCapture(0)
cam.set(3, 640) # set video width
cam.set(4, 480) # set video height
face_detector = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

# For each person, enter one numeric face id
face_id = input('\n enter user id end press <return> ==>  ')
print("\n [INFO] Initializing face capture. Look the camera and wait ...")

# Initialize individual sampling face count
count = 0
while(True):
   ret, img = cam.read()
img = cv2.flip(img, -1) # flip video image vertically
   gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
   faces = face_detector.detectMultiScale(gray, 1.3, 5)
   for (x,y,w,h) in faces:
      cv2.rectangle(img, (x,y), (x+w,y+h), (255,0,0), 2)
      count += 1
      # Save the captured image into the datasets folder
      cv2.imwrite("dataset/User." + str(face_id) + '.' +
               str(count) + ".jpg", gray[y:y+h,x:x+w])
```

```
        cv2.imshow('image', img)
    k = cv2.waitKey(100) & 0xff # Press 'ESC' for exiting video
    if k == 27:
        break
elif count >= 30: # Take 30 face sample and stop video
         break

# Do a bit of cleanup
print("\n [INFO] Exiting Program and cleanup stuff")
cam.release()
cv2.destroyAllWindows()
```

Phase 2: Train the Recognizer

Create another subdirectory to store the trained data and second python script**face_training.py**

```
import cv2
import numpy as np
from PIL import Image
import os

# Path for face image database
path = 'dataset'
recognizer = cv2.face.LBPHFaceRecognizer_create()
detector = cv2.CascadeClassifier("haarcascade_frontalface_default.xml");

# function to get the images and label data
def getImagesAndLabels(path):
imagePaths = [os.path.join(path,f) for f in os.listdir(path)]
faceSamples=[]
    ids = []

for imagePath in imagePaths:
PIL_img = Image.open(imagePath).convert('L') # grayscale

img_numpy = np.array(PIL_img,'uint8')
        id = int(os.path.split(imagePath)[-1].split(".")[1])
        faces = detector.detectMultiScale(img_numpy)
        for (x,y,w,h) in faces:
faceSamples.append(img_numpy[y:y+h,x:x+w])
ids.append(id)
    return faceSamples,ids

print ("\n [INFO] Training faces. It will take a few seconds. Wait ...")

faces,ids = getImagesAndLabels(path)
recognizer.train(faces, np.array(ids))

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

# Print the numer of faces trained and end program
print("\n [INFO] {0} faces trained. Exiting Program".format(len(np.unique(ids))))
```

Phase 3: Recognizer

3rd phase python script**: face_recognition.py**

```
import cv2

import numpy as np

import os

recognizer = cv2.face.LBPHFaceRecognizer_create()

recognizer.read('trainer/trainer.yml')

cascadePath = "haarcascade_frontalface_default.xml"

faceCascade = cv2.CascadeClassifier(cascadePath);

font = cv2.FONT_HERSHEY_SIMPLEX

#iniciate id counter

id = 0

# names related to ids: example ==> Nikhil: id=1, etc

names = ['None', 'Nikhil', 'ayub', 'Charan']
```

```
# Initialize and start realtime video capture
cam = cv2.VideoCapture(0)
cam.set(3, 640) # set video widht
cam.set(4, 480) # set video height
# Define min window size to be recognized as a face
minW = 0.1*cam.get(3)
minH = 0.1*cam.get(4)
while True:
    ret, img =cam.read()
    img = cv2.flip(img, -1) # Flip vertically
    gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)

    faces = faceCascade.detectMultiScale(
        gray,
        scaleFactor = 1.2,
        minNeighbors = 5,
        minSize = (int(minW), int(minH)),
       )
    for(x,y,w,h) in faces:
        cv2.rectangle(img, (x,y), (x+w,y+h), (0,255,0), 2)
        id, confidence = recognizer.predict(gray[y:y+h,x:x+w])
        # Check if confidence is less them 100 ==> "0" is perfect match
        if (confidence < 100):
            id = names[id]
            confidence = " {0}%".format(round(100 - confidence))
        else:
            id = "unknown"
            confidence = " {0}%".format(round(100 - confidence))

        cv2.putText(img, str(id), (x+5,y-5), font, 1, (255,255,255), 2)
        cv2.putText(img, str(confidence), (x+5,y+h-5), font, 1, (255,255,0), 1)

    cv2.imshow('camera',img)
    k = cv2.waitKey(10) & 0xff # Press 'ESC' for exiting video
    if k == 27:
        break
# Do a bit of cleanup
print("\n [INFO] Exiting Program and cleanup stuff")
cam.release()
cv2.destroyAllWindows()
```

REFERENCES

These are few reference papers ,

1. Jayavrinda Vrindavanam and Husain Al hayki, "**Efficient and Alternative Approach for Android Based Mobile Remote control in a Bluetooth Environment**, Springer, 2012.

2. Jayavrinda Vrindavanam, Husain Al Hayki,"**Android Based Mobile Remote Control for Office Electrical Devices in a Bluetooth Environment**" **National Conference on Advanced Technologies in Electrical and Mechanical Engineering 2013**, Al Musanna Institute of Technology, Muscat, April 16, 2013.

3. Mohamed Salman and Jayavrinda Vrindavanam, "**Efficient Interactive Control System based on GSM**" **International Journal of latest trends in Engineering and technology**, November 2013, Vol.3. Issue.2.

4. https://www.researchgate.net/publication/228437591_Realtime_people_counting_system_using_a_single_video_camera.

5. https://www.researchgate.net/publication/228437591_Realtime_people_counting_system_using_a_single_video_camera.

6. https://gist.github.com/li2hub/31c5ba8d8f324def419a7a4b77158348#file-google_assistant-ino

7. https://www.pyimagesearch.com/opencv-tutorials-resources-guides/