**“SMART CAMPUS PROJECT DESIGN AND IMPLEMENTATION ON IoT PLATFORM USING RASPBERRY PI BY PYTHON APPROACH”**

*A Dissertation Submitted in Partial Fulfillment of the Requirement for the*

*award of Bachelor of Technology*

*In*

*Electronics and Communication Engineering*

by

V.BILHANI (15K91A04F1)

V.SURESH GOPI (15K91A04E7)

S.HEMANTH SAI (15K91A04D6)

Under the guidance of

**K. PADMAJA DEVI**

Associate Professor



2018-2019

**TKR COLLEGE OF ENGINEERING AND TECHNOLOGY**

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

(Accredited by NBA &NAAC with ‘A’ grade approved by AICTE,

Affiliated to JNTUH Hyderabad)

Medbowli, Meerpet, Saroornagar, Hyderabad-500079, Telangana

**ACKNOWLEDGEMENT**

The satisfaction that accompanies the successful completion of the major project would be incomplete without the mention of the people who made it possible, without whose constant guidance and encouragement would have made efforts go in vain. We consider ourselves privileged to express gratitude and respect towards all those who guided us through the completion of this project.

We record our respect and gratitude to our internal guide Ms. K. PADMAJA DEVI, Associate Professor, Department of Electronics and Communication Engineering, for her valuable guidance and constant encouragement in carrying out the project.

We take this opportunity to thank Dr. D. NAGESHWAR RAO, Professor and Head of the Department, Electronics and Communication Engineering for his constant support.

We are extremely grateful to Professor Dr. D. V. RAVI SHANKAR, Principal of TKRCET who has been a constant source of inspiration, motivation and support.

Last but not least, we wish to thank the staff of ECE Department whose continuous support made us successful in the completion of the major project.

**DECLARATION**

We hereby declare that the Major Project Report entitled “**SMART CAMPUS DESIGN AND IMPLEMENTATION ON IoT PLATFORM USING RASPBERRY PI BY PYTHON APPROACH** ” is original and bonafide work of our own for the award of degree of **BACHELOR OF TECHNOLOGY** and submitted to the **Department of ECE, TKR COLLEGE OF ENGINEERING & TECHNOLOGY**, Hyderabad, under the guidance of Ms. K. PADMAJA DEVI, Associate Professor and has not been copied from any earlier report.

V.BILHANI 15K91A04F1

V.SURESH GOPI 15K91A04E7

S.HEMANTH SAI 15K91A04D6

**ABSTRACT**

A smart campus is intended to connect worldwide educational resources to faculty and students contributing in intellectual development. As student enters into the campus, he has to register himself so that it is used to show his presence in the campus and is used as attendance. Concerned authority may check presence of users at any time in the campus. Users can make audio and video calls to others users who are registered in the network by dialing to pre-assigned numbers.

Campus IVRS system is used to update the day-to-day academic activities such as any important events and announcements. Registered users can dial to the pre-defined numbers to access latest updates of campus related activities. Suitable menu based activity system is required to navigate it. Important information like circulars like exam schedule, campus interviews can be implemented by making outbound calls by using text to speech converters. These features are implemented by enabling the entire campus with Wi-Fi access to get the services by installing access at suitable locations with wired back bone network which is connected to the VoIP server. All features are initiated by using SIP protocol. Raspberry Pi is used as VoIP server, a series of small single board computers (SBC), which is economical and consumes less power when compared to traditional desktops/Servers. This project is also useful to make private branch exchange PBX network for small scale industries, apartments and college campuses.

**TABLE OF CONTENTS**

**S.NO CHAPTER PAGE. NO.**

1. INTRODUCTION 01

2. CHAPTER 1 03

1.1 VoIP 03

1.2 Why VoIP? 03

1.3 How does VoIP work? 04

1.4 VoIP configurations 04

1.5 Requirements, Availability and Service limitations 05

1.6 Network 06

1.7 Types of network 07

1.8 Topology 07

1.9 Methods of switching 10

1.10 Network components 10

1.11 Networking elements 10

1.12 IP Addressing 16

1.13 IP address hierarchy 17

1.14 Classful IP address 17

1.15 Network address 18

1.16 Subnetting and supernetting 19

1.17 IPV4 and IPV6 20

3. CHAPTER 2 21

2.1 OSI layer architecture 21

2.2 OSI layers 22

2.3 Transmission control protocol (TCP) / IP 26

2.4 RTP and RTCP 27

2.5 Session initiation protocol (SIP) 31

2.6 SIP entities 32

2.7 SIP requests 34

4. CHAPTER 3 37

3.1 IoT 37

3.2 Sensors 38

3.3 Connectivity 39

3.4 Benefits of IoT 39

3.5 Single Board Computer (SBC) 40

3.6 How to select SBC? 41

3.7 Raspberry Pi 42

3.8 Installing NOOBS in RPI 43

3.9 RPI commands 45

3.10 Virtual Network Computing (VNC) 46

5. CHAPTER 4 48

4.1Asterisk 48

4.2 Asterisk architecture 49

4.3 Development and scripting 50

4.4 Applications 51

4.5 Asterisk configuration 51

4.6 Asterisk commands 54

6. CHAPTER 5 55

5.1 Python 55

5.2 Python features 56

5.3Pycall 56

5.4Py\_Asterisk 57

5.5Py\_MySQL 58

5.6 MySQL 59

5.7 Installation & commands 60

6. OUTPUTS 65

7. ADVANTAGE S AND DISADVANTAGES 67

8. CONCLUSION 69

9. FUTURE SCOPE 70

10. REFERENCES 71

**LIST OF FIGURES**

**S.NO FIGURE NAME PAGE. NO.**

1. CHAPTER 1 03

1.1 Voice To/From VoIP 05

1.2 Block diagram of network 06

1.3 Bus topology 07

1.4 Ring topology 08

1.5 Star topology 08

1.6 Tree topology 09

1.7 Mesh topology 09

1.8 Hub 12

1.9 Bridges 12

1.10 Switch 13

1.11Two level classful& three level subnet hierarchy 19

1.12 Supernetting 20

2. CHAPTER 2 21

2.1OSI Reference model 21 2.2 Data encapsulation 22

2.3 Physical layer 22

2.4 Data link layer 23

2.5 Network layer 24

2.6 Transport layer 24

2.7 Session layer 25

2.8 Presentation layer 26

2.9 Application layer 26

2.10 RTSP 28

2.11 RTP 28

2.12 SIP 32

2.13 SIP URLs 32

2.14 SIP Architecture 33

2.15 SIP Responses 35

2.16 SIP Call flow 36

2.17 SIP Message format 36

3. CHAPTER 3 37

3.1 IoT 37

3.2 Sensors 38

3.3 SBC 40

3.4 Raspberry Pi 43

3.5 VNC Viewer 47

4. CHAPTER 4 48

4.1 Asterisk architecture 50

5. CHAPTER 5 55 5.1MySQL database 64

5.2 Tables in database 64

6. OUTPUTS 65

6.1 Register.py 65

6.2 Detect.py 65

6.3 Inbound.py 66

6.4 Outbound.py 66

**INTRODUCTION**

A Smart campus is intended to connect the worldwide educational resources to faculty and students contributing in intellectual development. A smart campus should providevarious services to students, which makes the students campus life easier, comfortable and attractive. These services are not just for academic life but also for socializing, moving around, sharing events. Student’s thinking is impacted at least in three directions: academic, practical and social ways. Existing campuses could think of alternatives for positively impacting the student in these three ways for a better socio-economic society tomorrow.

Campus is to be provided with the facilities with broader scope to the student by providing facility to listen, access and interact with the eminent faculty with broader backgrounds .As student enters into the campus he is connected to the campus network using wireless network by his smart phone with limited facilities. First he has to register himself so that it is used show his presence in the campus and also it used as attendance. Concerned authority may check presence of users at any time in the campus. Users can makes the audio and video calls to the other users who are registered in the network without any cost by using pre assigned numbers and they may have group discussion using conference and chatting.

Campus requires IVRS system to update the day-to-day academic activities instead of displaying in the notice board such as classes schedule, exam results and any important events and announcements. Registered user should dial the pre-defined number to access latest updates of campus related activities. Suitable menu based activity system required to navigate it. Sometimes, in the case of informing important circulars like exam schedule, social gathering and campus interviews, can be implemented by making outbound calls. This is done by dialing to the registered users and playing informative announcement by using text to speech converters. Lecturers can know their load such as classes on that date and other works assigned to him.

As laptops, iPads and Smartphones have become more prevalent, there is dramatic increase in WIFI usage. These features are implemented by enabling the entire campus with Wi-Fi access to get the services by installing access points at suitable locations with wired back bone network. Backbone network is connected to the VoIP server .All features are initiated by using SIP protocol. One of the main advantage of WIFI is having limited coverage suitable for near field communication, unlicensed spectrum i.e., no need to purchase the spectrum and one can have their own frequency planning and greater bandwidth.

In this project, Raspberry Pi is used as server. The Raspberry Pi is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation. This model became far more popular than anticipated, selling outside its target market for uses such as robotics. Here Raspberry PI is used as VoIP server, which is economical and consumes less power compared to the traditional Desktop/Servers.

This project is also useful to make Private branch exchange PBX network for Small Scale Industries, Apartments and College campuses.

**CHAPTER 1**

**VOIP AND NETWORKING**

**VOIP**

VoIP is the ability to make telephone calls and send faxes over IP based data networks with a suitable quality of service and superior cost / benefit .

Voice over Internet Protocol (also voice over IP, VoIP or IP telephony) is a methodology and group of technologies for the delivery of voice communications and multimedia sessions over

Internet Protocol (IP) networks, such as the Internet. The terms Internet telephony, broadband

telephony, and broadband phone service specifically refer to the provisioning of communications services (voice, fax, SMS, voice-messaging) over the public Internet, rather than via the public switched telephone network (PSTN).

The purest VoIP implementation uses IP capable end-user equipment such as IP phones or a computer and does not rely on a standard telephone switch. Figure is a simplified diagram of an IP telephone system connected to a wide area IP network. IP phones are connected to a LAN.

Voice calls can be made locally over the LAN. The IP phones include codecs that digitize and encode (as well as decode) the speech.

**Why VoIP?**

1. It is easier to install, configure and maintain.

2. It scales up or down easily.

3. A range of call features are supported.

4. Even older technology like fax is supported.

5. Hosted VoIP saves businesses money.

6. It integrates with other business systems.

7. It allows users to choose their hardware.

**How does VoIP work?**

To transport voice over a data network, the human voice must be packetized.

· This process contrasts significantly with the circuit-switching mechanism used in traditional

networks.

· Voice packetization involves appending headers with routing information to the voice data.

· Multiple voice samples are combined into a packet and the voice packet is switched hop-by hop through the network.

· To summarize, the voice signal is broken up into small pieces (packets) and sent though the

network one-by-one. The process of packetization compresses the callers voice signal,

transfers it over the IP network and it is then decompressed at the other end.

**VoIP configurations**

**Dedicated routers**

These devices allow you to use your traditional phone to place VoIP calls.

They are connected to cable/DSL modems (or any high-speed internet source) and allow you to attach an ordinary telephone.Once configured, and with an appropriate VoIP provider and service plan, these devices require no special software or interaction with a computer.

In fact, you only need to pick up your phone and dial a number at the dial tone.You also may bring your adapter with you when you travel and make calls wherever broadband

internet access is available.

**Adapters (USB)**

These devices also allow you to use a traditional phone to place VoIP calls.

They usually come in the form of USB adapters that are slightly larger than the typical thumb drive.They feature a standard modular phone jack to which you can attach an ordinary phone line.Once connected your phone behaves as if it were connected to standard phone service. Behind the scenes, however, the included software is actually setting up a VoIP call.

**Software-controlled VoIP applications**

There are many software applications (“softphones”) that allow you to place VoIP phone calls directly from an ordinary computer with a headset, microphone, and sound card.

Internet telephony service providers usually give away their softphones but require that you use their service. Together, these applications and services enable users to talk to other people using the same service at no cost, and to the rest of the world for a fee.

Software-based VoIP applications are quite attractive to consumers because they often already have most of the components necessary to get started at little to no cost.

**Dedicated VoIP phones**

A VoIP phone looks like an ordinary corded or cordless telephone, but it connects directly to a computer network rather than a traditional phone line.

**Requirements, Availability, and Service Limitations**

When considering VoIP service, you should not assume that its features, functionality, and options will equal those of traditional landlines; you should be familiar with the requirements, availability, and possible service limitations of VoIP service before switching to VoIP as either a primary means of communication or an enhancement to your current services.

**Requirements**

VoIP requires a connection to the Internet through an ISP, a VoIP service to extend the reach to traditional landlines, and VoIP software to actually place calls. Plain Old Telephone Service (POTS) requires none of these prerequisites. It is important to note that Digital Subscriber Line (DSL) internet service uses traditional phone lines for your internet connection; in this case, you already have telephone service to begin with. You may wish to weigh the expected benefits of VoIP against these costs given your current operating environment.

**Availability due to bandwidth**

VoIP communication nearly always requires a high-speed (broadband) internet connection for reliable functionality. Even given typical broadband connection speeds, though, service interruptions or degradation of quality is possible due to high internet traffic. For example, if you are trying to place a VoIP call while other people are using a lot of bandwidth on the same internet connection, the sound quality of your VoIP call or general VoIP availability may be affected.

S**ervices**

911 services are not guaranteed with a basic (VoIP to VoIP) setup. However, it is available with many of the interconnected services that extend VoIP connectivity to traditional landlines. You should not assume that 911 services are present and working (even with interconnected VoIP services) but should consult with the terms of your service agreement. The FCC has described some of the challenges of VoIP services and has provided tips for VoIP subscribers.

****

Fig 1.1Voice To/From IP

**NETWORK**

Interconnection of two or more computers physically and logically, in order to share the resources and Information.

****

Fig 1.2 Block diagram of network

**TYPES OF NETWORKS**

· Local Area Network (LAN)

· Campus Area Network(CAN)

· Metro Area Network(MAN)

· Wide Area Work(WAN)

**TOPOLOGY - PHYSICAL LAY OUT**

1. Bus Topology

2. Ring Topology

3. Star Topology

4. Mesh Topology

5. Tree Topology

**BUS TOPOLOGY**

Long cable to which the network devices are attached.

Advantages:

1. Easy to connect.

2. Less cable length

Disadvantages:

1. Entire network shut down

2. Terminators are required

3. Difficult to identify the problem

4. Not suitable for large building

****

Fig 1.3 Bus topology

**RING TOPOLOGY**

Advantages:

The response time predictable

Disadvantages:

1. More devices in the ring, the longer the n/w delays.

2. The entire n/w fails if one of work station fails.

****

Fig 1.4 Ring topology

**STAR TOPOLOGY**

Uses a central device with drop cables in all directions

Advantages:

1. Easy to install

2. No disruptions to the new

3. Easy to detect fault

Disadvantages:

1. More cable length
2. If the hub fails, node attached fails
3. More expensive

****

Fig 1.5 Star topology

**TREE TOPOLOGY**

Combined characteristics of Linear ,Bus and Star

Advantages:

1. Point to point wiring

2. Supported by several hardware and s/w vendors

Disadvantages:

1. If the back bone breaks the entire segment fails

2. More difficult to configure

****

Fig 1.6 Tree topology

**MESH TOPOLOGY**

Every node connected to every other node

****

Fig 1.7 Mesh topology

**METHODS OF SWITCHING**

1.Circuit Switching

- Dedicated path is established

- Ex: ATM, PSTN analog Dial-up line, Leased Line, T1

2.Message Switching

- Store & Forward - Ex: Telex, SMS

3.Packet Switching

- IP based

- Ethernet, FDDI, Frame Relay, X.25

**NETWORK COMPONENTS**

Network Hardware

1. Server

2. Work Stations

3. Network Interface card.

Network Software

1. Network Operating System.

Network Media

1. Cables

**NETWORKING ELEMENTS**

· Hosts

· Hubs

· Bridges

· Switches

· Routers

**HOST**

Provides users with connection to network

· Computers (both clients and servers)

· Printers

· Scanners

· Other user devices

Connects directly to network segment

Operates at all 7 layers of the OSI model

**NIC**-Physical connection between the network and computer work station

Functions:

· Host to card communication

· Buffering

· Frame formation- Header ,data ,trailer

· Parallel to serial converter

· Encoding -Decoding

· Cable access

· Handshake

· Transmission

**Types of NIC**

1. ARCnet

2. Ethernet

3. Token ring

4. FDDI

5. CDDI

**Network Interface Card**

· ARCnet( 2.5Mbps)

· Ethernet- 10Base5, 10Base2, 10BaseT,

1.Fast Ethernet: 100BaseTX , 100BaseFX, 100BaseT4

2.GE (1000BaseSX, 1000BaseLX, 1000BaseT)

· Token Ring (4 Mbps, 16Mbps)

· 100 VG Any LAN (100 Mbps)

· FDDI (100Mbps)

· CDDI (100Mbps)

· ATM ( 155 Mbps)

· WiFi (11Mbps, 54Mbps)

**HUB**

· Hub is a device to interconnect the network terminals.

· Hub is having multi port RJ 45 jacks to connect the network terminals.

· Normally used in Star Topology.

· Hub is working on the Physical Layer (Layer 1).

· Normally Hub works on broadcast mode.

· Hub is shared device since the bandwidth is shared by the network devices.

· It receives the data in one port and simply broadcast in all other ports.

· All the stations will receive the packets and check the destination address.

· The host which is having the MAC address same as the destination address, will take the

packet.

· All other stations will discard the packet



Fig 1.8 Hub

**BRIDGES**

· Bridge connects the two LAN segment at Data Link Layer of OSI Model.

· Generally Bridges works on MAC.

(The OSI Model consists of 2 sub layers MAC and LLC.)



Fig 1.9Bridge

· Bridges have the intelligence to understand the source and destination hardware addresses of network devices.

· A bridged network are physically separate networks but logically a single network.

**SWITCHES**

· A LAN switch is an internet working device that works on layer 2 having multiple RJ 45 connector for connecting network system.

· The earliest LAN switches were developed in 1990.

· A LAN switch providing dedicated bandwidth to devices.

· Switch preserve the bandwidth on your network using segmentation

· A switch will keep track of the MAC address attached to each of its ports and route the traffic to the to the appropriate port only.

· A LAN switch is also called a frame switch because it forwards Layer 2 frames, whereas an ATM switch forwards cells.

· Although Ethernet LAN switches are most common, Token Ring and FDDI LAN switches are becoming more prevalent as network utilization increases.



Fig 1.10Switch

**LAN Switch Operation**

LAN switches are similar to transparent bridges in functions such as

- Learning the topology

- Flooding

- Forwarding and

- Filtering.

· These switches support several new and unique Features such as:

- Dedicated communication between devices

- Multiple simultaneous conversion.

- Full-duplex communication.

- Media-rate adoption. (Multiple bit rate)

**LAN Switching Forwarding**

LAN switches can be characterised how they forward the packets( i.e. by the forwarding method they support ).

· The store-and-forward switching method

· The cut-through switching method

**The store-and-forward switching method**

· The store-and-forward switching method, error checking is performed and erroneous frames

are discarded.

· With the store-and-forward switching method, the LAN switch copies the entire frame into its onboard buffers or memory prior to transmit.

· Ability to check the FCS (Frame Check Sequence) field for CRC errors.

**The store-and-forward switching method**

· The frame is discarded if it contains a CRC error or if it is a runt (less than 64 bytes including the CRC) or a giant (more than 1518 bytes including the CRC).

· If the frame does not contain any errors, the switch forwards the frame to destination.

**Cut-through switching method**

· The Switch copies only the destination address (first 14 Bytes of the frame).

· It will immediately determines the outgoing interface, and forwards the frame toward its

destination, resulting fast switching.

· Error checking is not there since the switch is not receiving the entire frame.

**LAN Switching Bandwidth**

· LAN switches also can be characterized according to the proportion of bandwidth allocated to each port.

· Symmetric switch

· Asymmetric switch

**Symmetric switch**

· Symmetric switching provides evenly distributed bandwidth to each port.

· A symmetric switch provides switched connections between ports with the same bandwidth,

such as all 10BaseT or all 100BaseT.

· Symmetric switching is optimized for a reasonably distributed traffic load, such as in a peer-to -peer desktop environment.

**Asymmetric switch**

· Asymmetric switching provides unlike, or unequal, bandwidth between some ports.

· An asymmetric LAN switch provides switched connections between ports of unlike

bandwidths, such as a combination of 10BaseT and 100BaseT.

· This type of switching is also called 10/100 switching.

**LAN Switch and the OSI Model**

· LAN switches can be categorized according to the OSI layer at which they are working

· These categories are:

- Layer 2

- Layer 2 with Layer 3 features, or multi-layer

**ROUTERS**

· Routers were originally invented to solve some of the problems that weren't addressed by

bridges.

· Like bridges, routers are used to segment a LAN in order to reduce excess broadcast traffic and latency.

· Routers can relay data transmissions between similar or dissimilar network topologies.

· Routers can translate between Ethernet, Token Ring, and other signaling and media-access

schemes.

· Routers work at the network layer - they actually understand the protocols being used to carry the data over the network.

· They can use rules to decide what to do with a specific piece of data. which means they

examine the logical network address (e.g, 191.29.21.100) and not the physical hardware

address.

· Routers are useful in linking networks that are used for different purposes or by different

organizations.

· One can apply rules or filters to let certain data in, but keep other data out.

· Route data serving one purpose over a certain set of network connections, while routing other data over other connections.

· A router translates information from one network to another.

· Routers are smarter than bridges because they know about different routing protocols, address schemes, frame sizes and data rates in order to make the best decision.

· Routers select the best path to route a message, based on the destination address and origin.

· The best path is determined by using routing tables and algorithms.

· The router can direct traffic to prevent head-on collisions, and is smart enough to know when to direct traffic along back roads and shortcuts.

**IP ADDRESSING**

**Internet Registry (IR)**

· An Internet Registry is an organization that is responsible for distributing IP address space to its members or customers and for registering those distributions.

· IRs can be classified as:

– RIRs (Regional Internet Registry)

– NIRs (National Internet Registry)

– LIRs (Local Internet Registry)

**What is an IP Address**

· IP Address is Internet Protocol Address

· IP was first standardized in September 1981.

· It is an unique identifier for a node or a host in an IP Network.

· An Internet address works like a postal address, allowing data to be routed to the chosen

destination.

· There are two versions

– IPV4 (Now in use) 32 Bits

– IPV6 (Yet to be implemented) 128 Bits

**Binary Notation**

· 32 Bits are divided in to 4 octets of 8 bits each.

– 00000000. 00000000. 00000000. 00000000

– 11111111. 11111111. 11111111. 11111111.

· IP addresses are often expressed as four decimal numbers, each separated by a dot.

00111101.00000001.11000111.10000110 61.1.197.134

· This format is called "dotted-decimal notation.“

· Possible IP address will be

– 0.0.0.0 to 255.255.255.255 (0.0.0.0 and 255.255.255.255 are not used)

**IP address Hierarchy**

· IP address consists of 2 parts one is Network Prefix and the other is the host portion.

1.Network Prefix

2. Host Number



**Classful IP Address**

· Class A

– Referred as"/8s" since they have a 8-bit network-prefix.

– Maximum Networks possible 27 -2 = 126.

– All 0’s ie 0.0.0.0 is used for default route and 127.0.0.0 is used for loop back

· Class B

– Referred as "/16s" since they have a 16-bit network-prefix.

– Total networks - 16,384 (214)

– Hosts per network - 65,534 (216-2)

· Class C

– Referred as "/24s" since they have a 24-bit network-prefix.

– Total networks - 2,097,152 (221)

– Hosts per network - 254 (28-2)

**Network Address**

· In the host portion all 0’s and all 1’s are not used.

· All 0’s used for Network and all 1’s for broadcast.

· In classes A, B and C, an address with a host id of all 0s is not assigned to any host; it is reserved to define the network address itself.

**Subnet Mask**

· Class-A Address

– Default Mask is /8

– 11111111.00000000.00000000.00000000

– 255.0.0.0

Class-B Address

– Default Mask is /16

– 11111111.11111111.00000000.00000000

– 255.255.0.0

· Class-C Address

– Default Mask is /24

– 11111111.11111111.11111111.00000000

– 255.255.255.0

**Limitation of Classful Address**

· Inefficient due to wastage of unused IP addresses

· Inefficient use of bandwidth due to large broadcast domain

· Not suitable for medium sized organizations

– /24 Network which supports 254 hosts is too small while a /16 which supports 65,534

hosts is too large.

· Premature depletion of the /16 network address space.

**Subnetting**

· A Larger Network Address can be divided into several smaller networks called subnets.

· Each subnet acts as a separate broadcast domain

· Wastage of IP addresses reduced.

· Instead of the classful two-level hierarchy, subnetting supports a three-level hierarchy.



Fig 1.11 Two level classful hierarchy & three level subnet hierarchy

**Limitation of Subnetting**

· Each subnet will have equal number of hosts.

· But practically, the subnets may be in need of variable number of hosts

– Eg. Need for subnets can be as follows:

– First subnet having Maximum 100 hosts.

– Second subnet having 60 hosts.

– Third subnet is having 28 hosts.

– Fourth sub net having 10 hosts

· The problem can be overcome by using VLSM.

**Variable Length Subnet Mask**

· A Variable Length Subnet Mask (VLSM) is a means of allocating IP addressing resources to subnets according to their individual need rather than some general network-wide rule .

· Each subnet may have different subnet mask .

· Benefits

– Efficient use of the organization’ s assigned IP address space.

**Supernetting**

· Several networks are combined to create a Supernetwork.

· In Supernetting, an organization can combine several Class ‘C’ addresses to create a larger

range of addresses. It is the idea of combining two or more blocks of IP address that together compose a continuous range of addresses.



Fig 1.12 Supernetting

**Private IP Address**

· The following IP addresses are used for Private Network and not for Internet Routing

– 10.0.0.0 /8

– 172.16.0.0 / 12

– 192.168.0.0 /16

· APIPA (Automatic Private IP Addressing) in IPv4 (169.254.x.x/16)

**IPv4 & IPv6**

· We’re running out of IPv4 address space.

· IPv6 must be adopted for continued Internet growth.

· IPv6 is not backward compatible with IPv4.

· We must maintain IPv4 and IPv6 simultaneously for many years.

**Features of IPv6**

· New header format.

· Large address space.

· Efficient and hierarchical addressing and routing infrastructure.

· Stateless and stateful address configuration Built-in security.

· Better support for QoS.

· New protocol for neighboring node interaction.

**CHAPTER 2**

**OSI LAYERS AND PROTOCOLS**

**OSI model**

· An open system is a model that allows any two different systems to communicate regardless of their underlying architecture.

· The purpose is to open communication between different systems without requiring changes to the logic of the underlying hardware and software.

· The OSI model is not a protocol: it is a model for understanding and designing a network

architecture that is flexible, robust and interoperable.

**Layered Architecture**

· The OSI layer Reference Model is in order to provide interoperability between the networks

products of different vendors.

· OSI describes how information or data makes its way from one application programmes to

another through a network medium .

· The OSI Reference Model, divide the Network architecture in to a set of seven layers

· This layers define the different stages that data must go through to travel from one device to

another over a network.

· The OSI model is a layered communication process with each layer performing a specific task.



Fig 2.1 OSI Reference Model

**Data Encapsulation**

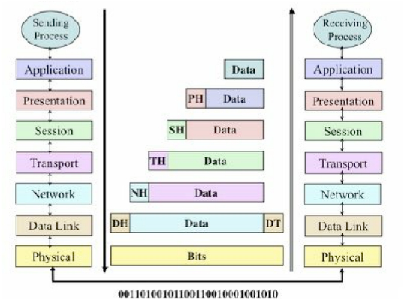


Fig 2.2 Data Encapsulation

**Physical Layer**

· The physical layer is mainly concerned with transmission of raw data over a channel.

· The physical layer coordinates the functions required to transmit a bit stream over a physical medium.

· It deals with mechanical and electrical specifications of the interface and transmission medium.

· The Physical Layer defines

1.Physical Characters of Interfaces and Media

2. Representation of Bits and encoding.

3.Transmission Rate

4. Synchronisation of Bits

5. Line Configuration ( Single, Multi)

6. Physical Topology

7. Transmission Mode ( Half duplex , Full duplex)

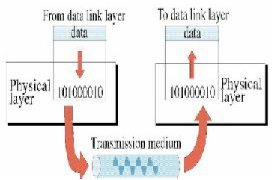


Fig 2.3 Physical layer

**Data Link Layer**

· Responsible of Data Link layer includes:

1.Framing

2.Physical Addressing

3. Flow Control

4. Error Control

5. Access Control

· Physical addresses are also known as hardware and BIA's (Burned In Addresses) but most

commonly as MAC addresses.

· The Data Link layer consists of two sub-layers:

1. LLC (Logical Link Control) Layer manages communication between devices over a single link of a network. Enables multiple higher-layer protocols to share a single physical data link.

2. MAC Layer

Manages protocol access to the physical network medium and determines hardware addresses.

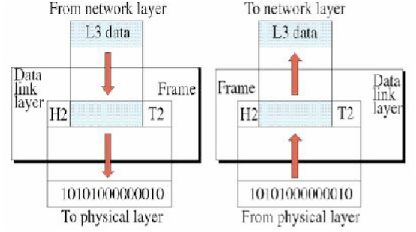


Fig 2.4 Data link layer

**Network Layer**

The network layer is responsible for the source-to destination delivery of a packet

· Defines logical addressing for nodes and networks/segments.

· It enables inter-networking, passing data from one network to another.

· Routers can determine how to forward packets through an internet- work.

· Routing occurs at this layer, hence Routed and Routing protocols reside on this layer.

· Routed protocols are used to encapsulate data into packets.

· The header added by the Network layer contains a network address so it can be routed through an internet-work.

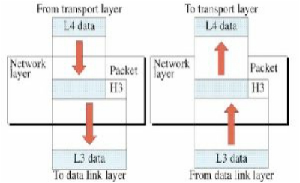


Fig 2.5 Network Layer

**Transport Layer**

· Responsible for source-to-destination(end-to-end)delivery of the entire message.

· The main purpose of this layers is making sure that the data is delivered error-free and in the

correct sequence.

· Mainly responsible for:

- Segmentation and reassembly

· Sequencing of data packets so they can be reassembled at destination

- Service-point addressing

· Identification of service addresses (ports) at the destination devices.

- Connection control

- Flow Control and Windowing

- Error detection and recovery

· When using a connection-oriented, reliable transport protocol, such as TCP:

- Data delivery guarantee is provided in the form of acknowledgments

- Synchronization of data delivery

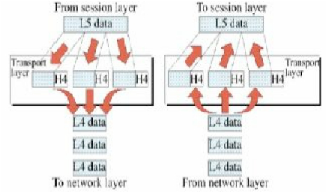


Fig 2.6 Transport layer

When using connection less transport protocol such as UDP no data delivery guarantee is

provided

· It provides multiplexing; the support of different flows of data to different applications on the same host.

**Session Layer**

· The session layer can be used to control the manner in which data are exchanged

· The session layer responsible for connection negotiation, log on procedure.

- Establishing the connection

- Maintaining the connection

- Synchronizing the connection

- Terminating the connection

· It provides error reporting for the Application and Presentation layer.

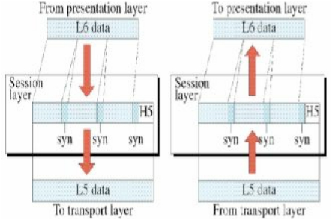


Fig 2.7 Session layer

**Presentation Layer**

· The presentation layer is concerned with the syntax and semantics of the information

exchanged between two systems

· Defines coding and conversion functions.

· Responsible for defining how information is presented to the user in the program used in the

application layer.

· It includes :

- Data representation formats

- Conversion of character representation formats

- Common data compression schemes

- Common data encryption schemes

The standards include ASCII, EBCDIC, BMP, GIF, JPEG, WAV, AVI, and MPEG.

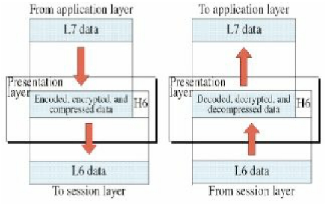


Fig 2.8 Presentation layer

**Application Layer**

· Application Layer provides network services directly to applications.

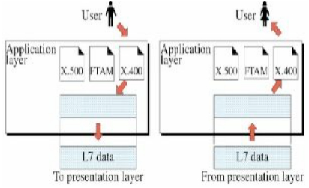


Fig 2.9 Application layer

· Contains the programs that invoke the services.

· Manages program requests that require access to services provided by the system.

· It determines the identity and availability of communication partners.

· It determines if sufficient resources are available to start program-to-program communication.

**TCP / IP**

· The TCP/IP protocol suite is so named for two of its most important protocols:

- Transmission Control Protocol (TCP)

- Internet Protocol (IP).

· The main design goal of TCP/IP was to build:

- An interconnection of networks, referred to as an internetwork, or internet,

- Providing universal communication services over heterogeneous physical networks.

**Internet Protocol**

· IP is responsible for :

- Moving packet of data from node to node.

- IP forwards each packet based on a four-byte destination address (the IP number)

**Transmission Control Protocol**

· The TCP protocol describes the host-to-host communication.

· TCP explains how two hosts can set up this communication and how they can stay in touch

with each other as data is being transferred.

· TCP is mainly responsible for :

- Data Concurrency

- Packet Sequencing

- Delivery guarantee

- Error control

- Retransmission

**REAL TIME PROTOCOL**

The Real-Time Transport Protocol (RTP) is an Internet protocol standard that specifies a way for programs to manage the real-time transmission of multimedia data over either unicast or multicast network services.

**What is RTP ?**

The Real-Time Transport Protocol (RTP) is an Internet protocol standard that specifies a way for programs to manage the real-time transmission either unicast or multicast network services. Originally specified in Internet Engineering Task Force (IETF) Request for Comments (RFC) 1889, RTP was designed by the IETF's Audio-Video Transport Working Group to support videoconferences with multiple ,geographically dispersed participants. RTP is commonly use in Internet telephony applications. RTP does not in itself guarantee real-time delivery of multimedia data (since this is dependent on network characteristics); it does, however, provide the wherewithal to manage the data as it arrives to best effect.

**Real time streaming**

The Real Time Streaming Protocol (RTSP) is a network control protocol designed for use in entertainment and communications systems to control streaming media servers. The protocol is used for establishing and controlling media sessions between end points

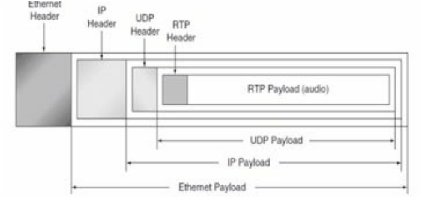


Fig 2.10 RTSP

The transmission of streaming data itself is not a task of RTSP. Most RTSP servers use the Real time Transport Protocol (RTP) in conjunction with Real-time Control Protocol (RTCP) for media stream delivery. However, some vendors implement proprietary transport protocols. The RTSP server software from Real Networks, for example, also used Real Networks' proprietary Real Data Transport (RDT).

**What is SIP and RTP used for?**

Sending Voice over IP (VoIP) requires two protocols: SIP and RTP (Real-time Transport Protocol).

SIP- Protocol used for establishing a session (call). IP address and port information is exchanged.

RTP -After SIP establishes a session, this protocol is used for exchanging voice packets.

· Sending Voice over IP (VoIP) requires two protocols: SIP and RTP.

· SIP- Protocol used for establishing a session. IP address and port information is exchanged.

· RTP – After SIP establishes a session, this protocol is used for exchanging voice packets.

You can think of SIP as a stage manager. SIP prepares the stage for RTP by setting up its

connection. Once RTP is finished with its stage (call), then SIP comes back to the stage to clean up after it.

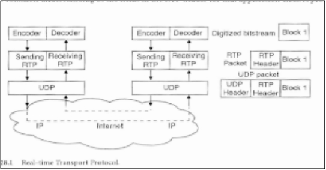


Fig 2.11 RTP

**Header Format**

The first 12 bytes are always present, whereas the contributing source identifiers are only used in certain circumstances. After this header, there may be optional header extensions. Finally, the RTP payload, whose format is determined by the application, follows the header. This header contains only the fields that are likely to be used by most of the multimedia applications. Information specific to a single application is carried in the RTP payload.

V — version: 2 bits: The first two bits are a version identifier, which contains the value 2 for the RTP version used at the time of this writing.

P — padding: 1 bit: If the padding bit is set, the packet contains one or more additional padding octets at the end, which are not part of the payload. The last octet of the padding contains a count of how many padding octets should be ignored, including itself. RTP data might be padded to fill up a block of certain size as required by an encryption algorithm.

X — extension: 1 bit: If the extension bit is set, exactly one extension header follows the fixed header.

CC — CSRC count: 4 bits: The CSRC count contains the number of CSRC identifiers that follow the fixed header. This number is more than one if the payload of the RTP packet contains data from several sources.

M — marker: I bit: The interpretation of the marker is defined by a profile. The marker is intended to allow significant events such as frame boundaries to be marked in the packet stream.

PT — payload type: 7 bits: PT identifies the format of the RTP payload and determines its

interpretation by the application. One possible use of this field would be to enable an application to switch from one coding scheme to another based on the information about resource availability on the network of feedback on application quality.

Sequence number: 16 bits: The sequence number increments by one for each RTP data packet sent, and can be used by the receiver to detect missing and misplaced packets. The initial value israndomly set to make crypto analysis attacks on possible encryption more difficult. RTP does not take any action when it detects a lost packet. It is left to the application to decide what to do when a packet is lost. For example, a video application could replay the previous frame if a frame is lost. Another application, however, could decide to

change encoding parameter in order to reduce the needed bandwidth. These decisions need some intelligence, which is left to the applications themselves.

Timestamp: 32 bits: The timestamp indicates the sampling instant of the first octet in the RTP data packet. Its function is to enable the receiver to play back samples at the appropriate intervals and to enable different media streams to be synchronized. It can also be used for calculations in jitter smoothing. The resolution of the clock used should be sufficient for the desired synchronization accuracy and for measuring packet jitter.

The initial value is randomly set. Because different applications may require different granularities of timing, RTP does not specify the units in which time is measured. The timestamp is just a counter of ticks, and the time between ticks is application specific. The clock granularity is specified in the RTP profile or payload format for an application.

SSRC — Synchronization source: 32 bits: This field identifies synchronization sources within the same RTP session. It is chosen randomly to avoid two sources in the same RTP session to have the same SSRC identifier. It indicates where the data were combined, or the source of the data if there is only one source. The sources could be in the same node or in different nodes, such as during a conference.

CSRC — Contributing source list. 0 to 15 items, 32 bits each: The CSRC list identifies the contributing sources for the payload contained in this packet. The CC field gives the number of identifiers.

**Real-time Transport Control Protocol — RTCP**

RTCP is the control protocol designed to work in conjunction with RT P. It is specified in RFC 3550. In an RTP session, participants periodically send RTCP packets to all the members in the same RTP session using IP multicast. RTCP packets contain sender and/or receiver reports that announce statistics such as the number of packets sent, number of packets lost, and inter-arrival jitter. This function may be useful for adaptive applications that can use this feedback to send high- or low qualitydata depending on the network congestion. Such applications will increase the compression ratio when there is little available bandwidth,

and will reduce the compression ratio, which will result in higher multimedia quality when there is more available bandwidth. This feedback information can also be used for diagnostic purposes to localize eventual problems.RTCP provides a way to correlate and synchronize different media streams that have come from the same sender. When collisions of SSRC occur, it is necessary to change the SSCR value of a given stream. This is done using RTCP.

In applications that involve separate multimedia streams, a common system clock is used for their synchronization. The system that initiates the session provides this function, and the RTCP messages enable all systems to use the same clock.

**RTP implementation**

RTP is an open protocol that does not provide pre implemented system calls. Implementation istightly coupled to the application itself. Application developers have to add the complete functionality in the application layer by themselves. It is, however, always more efficient to share and reuse code rather than starting from scratch. The RFC 3550 specification itself contains numerous code segments that can be used directly in the applications. There are some implementations with source code available on the web for evaluation and educational purposes. Many modules in the source code can be used with minor modifications.

**SESSION INITIATION PROTOCOL:**

· The Session Initiation Protocol (SIP) is a signalling protocol used for establishing sessions in an IP network.

· A session could be a simple two-way telephone call or it could be a collaborative multi-media conference session.

· The ability to establish these sessions means that a host of innovative services become

possible, such as voice-enriched e-commerce, web page click-to-dial, Instant Messaging and IP Centrex services.

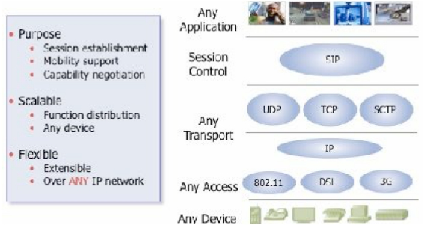


Fig 2.12 SIP

**User Mobility**

· SIP manages the user mobility by locating the users and by updating this information in the

SIP servers.

· Mobility management is achieved with the help of SIP URLs and registration of the users.



Fig 2.13 SIP URLs

**SIP Entities**

· User agents

· Proxy Servers

· Redirect Servers

· Registrar

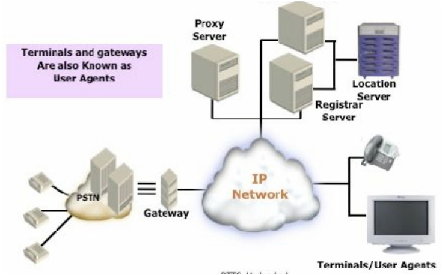


Fig 2.14 SIP Architecture

**User Agents**

· A user agent is an end system that acts on behalf of someone who wants to participate in calls.

· A user agent contains both a protocol client – called a user agent client (UAC) and a protocol server - called a user agent server (UAS).

· They can run, in a computer as one among many applications, or they can be implemented in dedicated device, such as SIP phone.

**Proxy Servers**

· Proxy servers make subsequent attempts on behalf of the user, rather than sending new contact information to the user.

**Redirect Servers**

· Redirect servers do not forward the request on behalf of UA but sends back the location

information to the UA.

· They help locate SIP UAs by providing alternative locations where the user can be reachable.

· It can be used to implement Group addresses.

**Registrar**

· Registrar refers to a SIP server accepting registrations.

· The contact information is then made available to other SIP servers within the same

administrative domain.

· A registrar is usually co-located with a redirect server or a proxy server.

**Location Server**

· They are not SIP entities.

· Location server stores and returns possible locations for users.

· Most registrars upload location updates to a location server

**SIP : Protocol Operation**

· Client / Server Protocol:

· SIP is based on web protocol HTTP and like HTTP it is a client / server protocol.

· A client is a SIP entity that generates requests.

· A server is a SIP entity that receiver requests and returns SIP responses.

· SIP Transaction=Request+Response.

**SIP Requests**

· The core SIP specification defines six types of SIP requests.

· Every SIP request contains a field, called a method, which denotes its purpose.

· There are six methods.

- INVITE

- ACK

- OPTIONS

- BYE

- CANCEL

- REGISTER.

**INVITE**

· INVITE requests invite users to participate in a session.

· The body of the INVITE requests contain the description of the session.

· SIP only handles the invitation to the user and the user’s acceptance of the invitation.

· Thus with a different session description, SIP can invite users to any type of session.

**ACK**

· ACK requests are used to acknowledge the reception of a final response to an INVITE. Thus, a client originating an INVITE request issues an ACK request when it receives a final

response for the INVITE.

**CANCEL**

· CANCEL request cancels pending transactions. Suppose, a SIP server has received an

INVITE but not yet returned a final response:

· It will stop processing the INVITE upon receipt of CANCEL.

· However, if the final response of INVITE has been returned, CANCEL would have no effect the transaction.

**BYE**

· BYE requests are used to abandon session. In two party sessions, abandonment by one of the parties implies that the session is terminated.

· For instance, what Mohit sends a bye to Neha, their session is automatically terminated

**SIP Responses**

· Upon reception of a request, a server issues one or several responses.

· Every response has a code that indicates the status of the transaction.

· Status codes are integers ranging from 100 to 699 and are grouped into classes.

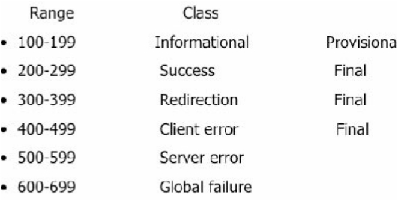


Fig 2.15 SIP Responses

**SIP Message response codes**

– 5xx: Server error

· 500 Server internal error

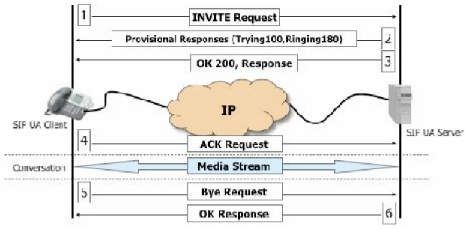
· 502 bad gateway

– 6xx: Global failure

· 600 busy

· 604 does not exist

**SIP Call flow**



2.16 SIP Call Flow

**SIP Message format**

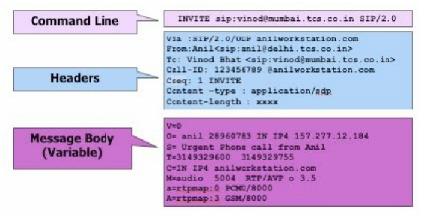


Fig 2.17 SIP Message format

**CHAPTER 3**

**IOT AND RASPBERRY PI**

**IoT**

IoT is a conceptual framework about enabling connectivity and embedded intelligence in devices. Some of these devices are connected today, but **MANY** are not. Not strictly machine-to-machine (M2M) – also machine-to-people, people-to-machine, machine-to-objects, people-to-objects. Creates the ability to collect data from a broad range of devices. Data can be accessed via the cloud and analyzed using “big data” techniques

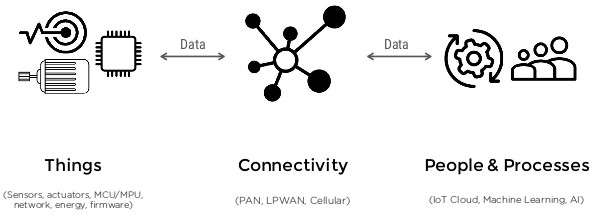


Fig 3.1 IoT

Iot  is the networking of physical objects that contain electronics embedded within their architecture in order to communicate and sense interactions amongst each other or with respect to the external environment. In the upcoming years, IoT-based technology will offer advanced levels of services and practically change the way people lead their daily lives. Advancements in medicine, power, gene therapies, agriculture, smart cities, and smart homes are just a very few of the categorical examples where IoT is strongly established.

**Sensors**

Industries and organizations have been using various kinds of sensors for a long time but the invention of the Internet of Things has taken the evolutions of sensors to a completely different level.

[IoT](https://www.finoit.com/iot-application-development-company/) platforms function and deliver various kind of intelligence and data using a variety of sensors. They serve to collect data, pushing it and sharing it with a whole network of connected devices. All this collected data makes it possible for devices to autonomously function, and the whole ecosystem is becoming “smarter” every day.

By combining a set of sensors and a communication network, devices share information with one another and are improving their effectiveness and functionality.

Take Tesla vehicles as an example. All of the sensors on a car record their perception of the surroundings, uploading the information into a massive database. The data is then processed and all the important new pieces of information are sent to all other vehicles. This is an ongoing process, through which a whole fleet of Tesla vehicles is becoming smarter every day.

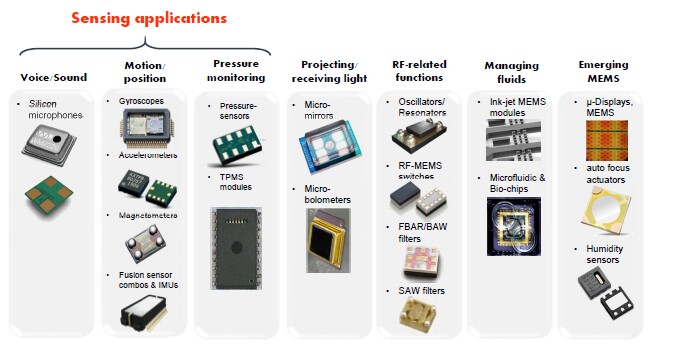


Fig 3.2 Sensors

**CONNECTIVITY**

Connectivity, it is one of the main things to keep in mind while developing any Internet-of-Things (IoT) project.

The various options that we have are:

* WiFi
* Thread
* ZigBee
* Bluetooth
* RFID and NFC

**People and Process**

Connected devices are definitely the things of the future, but when it comes to adoption and use, some industries are much further ahead than others.

According to [IDC](http://www.idc.com/getdoc.jsp?containerId=prUS42209117), companies in manufacturing and logistics top the connectivity spending list because they have direct connections between device data and business results on things like delivery times, product quality, predictive maintenance, and customer satisfaction.

These industries lead the way for Internet of Things (IoT) adoption and show what can be achieved when we connect people, processes, and things.

Here are some examples of how I believe businesses can use IoT to provide real-time visibility and drive informed decisions.

**Benefits of IoT**

Since IoT allows devices to be controlled remotely across the internet, thus it created opportunities to directly connect & integrate the physical world to the computer-based systems using sensors and internet. The interconnection of these multiple embedded devices will be resulting in automation in nearly all fields and also enabling advanced applications. This is resulting in improved accuracy, efficiency and economic benefit with reduced human intervention. It encompasses technologies such as smart grids, smart homes, intelligent transportation and smart cities. The major benefits of IoT are

* **Improved Customer Engagement** – IoT improves customer experience by automating the action. For e.g. any issue in the car will be automatically detected by the sensors. The driver, as well as the manufacturer, will be notified about it. Till the time driver reaches the service station, the manufacturer will make sure that the faulty part is available at the service station.
* **Technical Optimization**– IoT has helped a lot in improving technologies and making them better. The manufacturer can collect data from different car sensors and analyze them to improve their design and make them much more efficient.
* **Reduced Waste** – Our current insights are superficial, but IoT provides real-time information leading to effective decision making & management of resources. For example, if a manufacturer finds fault in multiple engines, he can track the manufacturing plant of those engines and can rectify the issue with manufacturing belt.

**SINGLE BOARD COMPUTER**

A single-board computer (SBC) is a complete computer built on a single circuit board, with microprocessor(s), memory, input/output (I/O) and other features required of a functional computer. Single-board computers were made as demonstration or development systems, for educational systems, or for use as embedded computer controllers. Many types of home computers or portable computers integrate all their functions onto a single printed circuit board. Unlike a desktop personal computer, single board computers often do not rely on expansion slots for peripheral functions or expansion. Single board computers have been built using a wide range of microprocessors.

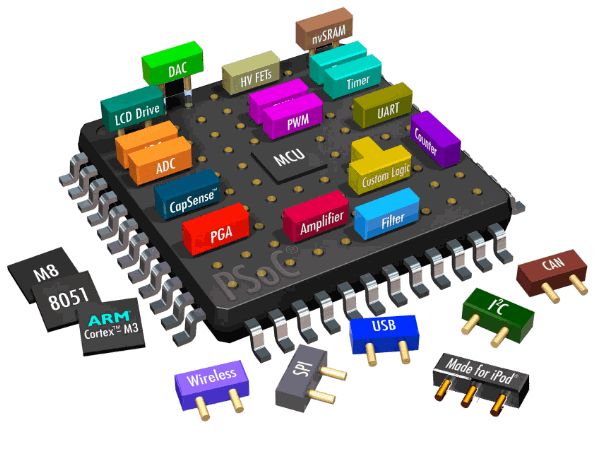


Fig 3.3 SBC

A system on a chip or system on chip that integrates all components of a computer or other electronic system. These components typically include a central processing unit (CPU), memory, input/output ports and secondary storage – all on a single substrate.

It may contain digital, analog, mixed-signal, and often radio frequency signal processing functions, depending on the application. As they are integrated on a single electronic substrate, SoCs consume much less power and take up much less area than multi-chip designs with equivalent functionality. Because of this, SoCs are very common in the mobile computing and edge computing markets. Systems on chip are commonly used in embedded systems and the Internet of Things.

Systems on Chip are in contrast to the common traditional motherboard-based PC architecture, which separates components based on function and connects them through a central interfacing circuit board.[nb 2] Whereas a motherboard houses and connects detachable or replaceable components, SoCs integrate all of these components into a single integrated circuit, as if all these functions were built into the motherboard. An SoC will typically integrate a CPU, graphics and memory interfaces,[nb 3] hard-disk and USB connectivity,[nb 4] random-access and read-only memories and secondary storage on a single circuit die, whereas a motherboard would connect these modules as discrete components or expansion cards.

**How to select SBC**

* Proccessor
* Clock speed
* Memory,Memory size
* GPU
* Wifi ,Bluetooth
* GPIO pins
* Display interface
* HDMI Output
* Power consumtion
* Cost,Size,OS
* USB
* PCIE
* GPS
* SIM

**RASPBERRY PI**

The Raspberry Pi is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation to promote the teaching of basic computer science in schools and in developing countries. The original model became far more popular than anticipated, selling outside of its target market for uses such as robotics. Over 5 million Raspberry Pis have been sold before February 2015, making it the best-selling British computer. By November 2016 they had sold 11 million units.

The first generation (Raspberry Pi 1 Model B) was released in February 2012. It was followed by a simpler and inexpensive model Model A. In 2014, the foundation released a board with an improved design in Raspberry Pi 1 Model B+. These boards are

approximately credit-card sized and represent the standard mainline form-factor. Improved A+ and B+ models were released a year later. A"compute module" was released in April 2014 for embedded applications, and a Raspberry Pi Zero with smaller size and reduced input/output (I/O) and general-purpose input/output (GPIO) capabilities was released in November 2015 for US$5.

The Raspberry Pi 2 which added more RAM was released in February 2015. Raspberry Pi 3 Model B released in February 2016, is bundled with on-board WiFi, Bluetooth and USB boot capabilities. As of January 2017, Raspberry Pi 3 Model B is the newest

mainline Raspberry Pi. Raspberry Pi boards are priced between US$5–35. As of 28 February 2017, the Raspberry Pi Zero W was launched, which is identical to the Raspberry Pi Zero, but has the Wi-Fi and Bluetooth functionality of the Raspberry

Pi 3 for US$10.



Fig 3.4 Raspberry Pi

**Features of RPI**

All models feature a Broadcom system on a chip (SoC), which includes an ARM compatible central processing unit (CPU) and an on-chip graphics processing unit (GPU, a Video Core IV). CPU speed ranges from 700 MHz to 1.2 GHz for the Pi 3 and on board memory range from 256 MB to 1 GB RAM. Secure Digital (SD) cards are used to store the operating system and program memory in either the SDHC or Micro SDHC sizes.

**Installing NOOBS in RPI**

To get started with Raspberry Pi, you need an operating system. NOOBS (New Out Of Box Software) is an easy operating system install manager for the Raspberry Pi.

How to get and install NOOBS

Buy preinstalled SD card

SD cards with NOOBS preinstalled are available from many of our distributors and independent retailers, such as [Pimoroni](https://shop.pimoroni.com/products/noobs-8gb-sd-card" \t "_blank), [Adafruit](https://www.adafruit.com/products/1583" \t "_blank) and The Pi Hut. For older models of Raspberry Pi, you’ll need a full-size SD card; for the Pi Zero, A+, B+, Pi 2 and Pi 3 you’ll need a micro SD card.

**Download**

We recommend using an SD card with a minimum capacity of 8GB.

1. Using a computer with an SD card reader, visit the Downloads page.
2. Click on NOOBS, then click on the Download ZIP button under ‘NOOBS (offline and network install)’, and select a folder to save it to.
3. Extract the files from the zip.

**Format your SD card**

It is best to format your SD card before copying the NOOBS files onto it. To do this:

1. Visit the SD Association’s website and download SD Formatter 4.0 for either Windows or Mac.
2. Follow the instructions to install the software.
3. Insert your SD card into the computer or laptop’s SD card reader and make a note of the drive letter allocated to it, e.g. G:/
4. In SD Formatter, select the drive letter for your SD card and format it.

**Drag and drop NOOBS files**

1. Once your SD card has been formatted, drag all the files in the extracted NOOBS folder and drop them onto the SD card drive.
2. The necessary files will then be transferred to your SD card.
3. When this process has finished, safely remove the SD card and insert it into your Raspberry Pi.

**First boot**

1. Plug in your keyboard, mouse, and monitor cables.
2. Now plug the USB power cable into your Pi.
3. Your Raspberry Pi will boot, and a window will appear with a list of different operating systems that you can install. We recommend that you use Raspbian – tick the box next to Raspbian and click on Install.
4. Raspbian will then run through its installation process. Note that this can take a while.
5. When the install process has completed, the Raspberry Pi configuration menu (raspi-config) will load. Here you are able to set the time and date for your region, enable a

Raspberry Pi camera board, or even create users. You can exit this menu by using Tab on your keyboard to move to Finish

**RPI commands**

* **ifconfig**

Use ifconfig to display the network configuration details for the interfaces on the current system when run without any arguments

* **ping**

The ping utility is usually used to check if communication can be made with another host.

e.g. ping 8.8.8.8

* **sudo**

The sudo command enables you to run a command as a superuser, or another user.

* **ssh**

ssh denotes the secure shell. Connect to another computer using an encrypted network connection

* **chmod**

You would normally use chmod to change the permissions for a file. The chmodcommand can use symbols u (user that owns the file), g (the files group) , and o (other users) and the permissions r (read), w (write), and x (execute). Using chmodu+x \*filename\* will add execute permission for the owner of the file.

* **cat**

You can use cat to list the contents of file(s), e.g. cat this File will display the contents of this File.

* **cp**

Using cp makes a copy of a file and places it at the specified location (this is similar to copying and pasting). For example, cp ~/fileA /home/otherUser/

* **mv**

The mv command moves a file and places it at the specified location (so where cp performs a 'copy-paste', mv performs a 'cut-paste').

* **ls**

The ls command lists the content of the current directory (or one that is specified)

* **cd**

Using cd changes the current directory to the one specified.

* **Pwd**

The pwd command displays the name of the present working directory: on a Raspberry Pi, entering pwd will output something like /home/pi.

**Virtual Network Computing (VNC)**

In computing, Virtual Network Computing (VNC) is a graphical desktop-sharing system that uses the Remote Frame Buffer protocol (RFB)to remotely control another computer. It transmits the keyboard and mouse events from one computer to another, relaying the graphical-screen updates back in the other direction, over a network.

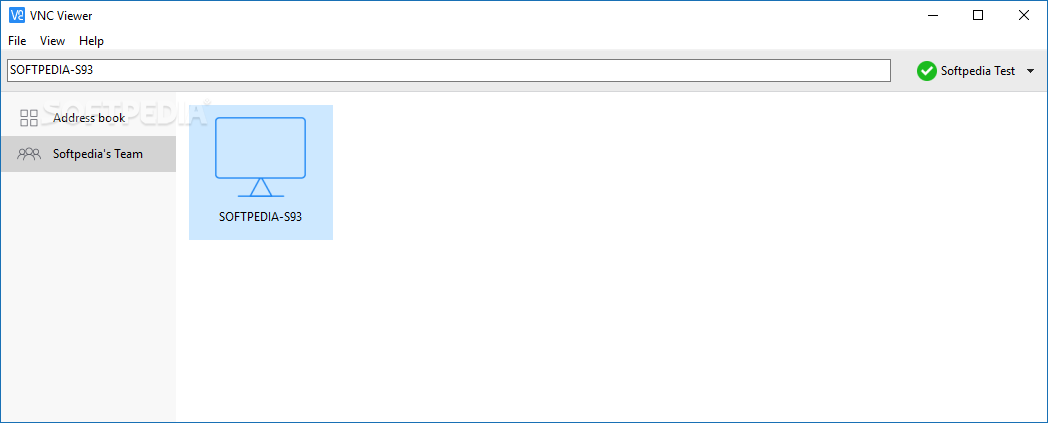


Fig 3.5 VNC Viewer

* Connection to a data-processing system from a remote location, for example, through a remote access service or virtual private network
* Remote desktop software, software allowing applications to run remotely on a server while displaying graphical output locally
* Terminal emulation – when used to interface with a remote system. May use standard tools like:
  + telnet – software used to interact over a network with a computer system
  + Ssh – secure shell: often used with remote applications
* Activation of features of a business telephone system from outside the business's premises
* Remote Access, a DOS-based bulletin-board system
* Remote Database Access, a protocol standard for database access

**CHAPTER 4**

**ASTERISK**

**OPERTION OF VOIP AT ASTERISK**

Asterisk is open source PBX (Private Branch Exchange) telephony system software. it is runs on virtually any Operating System (OS) like ubuntu, centos, Fedora etc. It supports most of the VOIP protocols like SIP,H.323 etc. It is also support many different hardware telephony cards.

* Voice over IP (VOIP) allows telephone conversations to travel over a LAN or the Internet instead of traditional telephone wiring.
* Asterisk is a "phone system" that can connect to many different devices, either traditional phone hardware or VOIP hardware.

**Telephone Terminology**

Foreign exchange office (FXO) is a port that connects to the phone company.

Foreign exchange system (FXS) is a port that connects to a telephone.

Analog telephone adapter (ATA) is a device with an Ethernet port and a telephone port (FXS).

Phone System Example

1.**Basic Phone**:

* Phone Line <-> Phone
* Telephone <-> ATA <-> Internet <-> VOIP Provider

2. **Phone System**:

* Phone Line <-> phone system <-> Telephone wires <-> Phones
* Phone Line <-> FXO <-> phone system <-> LAN <-> Phones

3. **VOIP Phone System**:

* VOIP Provider <-> Internet <-> Phone System <->Lan<-> SIP Phones

**Protocol Terminology**

* Session Initiation Protocol (SIP) manages a telephone connection between two parties.
* Session Description Protocol (SDP) provides the parameters for communicating between two parties
* Real time Transport Protocol (RTP) carries the voice data itself.
* Coder/Decoder (coDEC) converts voice into various data formats.simpleTunneling of UDP through NAT (STUN) determines how your firewall will interact with the communication.
* Inter Asterisk exchange (IAX) is an Asterisk protocol.

**ASTERISK ARCHITECTURE**

Telephony applications: Asterisk applications connect any phone, phone line or packet voice connection to another interface or service. Asterisk easily and reliably scales from very small to very large systems Asterisk supports high density, redundant applications.

Asterisk supports every possible kind of telephone technology. The technologies include VolP, SIP, H.323, IAX, and BGCP (for gateways and phone.) Asterisk can interoperate with almost all standards-based telephony equipment. Hardware to connect your Asterisk system is inexpensive. Asterisk supports traditional telephone technologies like ISDN PRI and T-Carrier including T1 and E-1. Telephony applications include calling, conferencing, call bridging, voicemail, auto attendant, custom Interactive Voice Response scripting. call parking, intercom and many others.

An Asterisk server connected to a local area network can control phones connected to that local area network. These phones can call each other through the Asterisk server. The Asterisk server can control phones connected to other networks or the Internet. even if those phones or the Asterisk server are behind firewalls. With Digium FXS interface cards, an Asterisk server can control local analog telephones. FXO and T-carrier interface boards from Digium can connect an Asterisk server to the PSTN. This allows calls to be made to and from the PSTN. PSTN users calls can be switched from one Asterisk server to another Asterisk server. A telephone controlled by an asterisk server can call a telephone controlled by a second Asterisk server. A call from a telephone controlled by one Asterisk server can be switched to a second Asterisk server and then on to the PSTN.

Asterisk contains engines that perform critical functions. When Asterisk starts, the Dynamic Module Loader loads and initializes drivers. The drivers provide channel drivers, file formats, call detail recording back ends, codec’s, and applications among others.

The Asterisk PBX Switching Core accepts telephone calls from the interfaces. The Switching Core handles calls according to the instructions found in a dial plan. The PX Swicting Core uses the Application Launcher to ring phone.

The PBX Switching Core includes a Scheduler and I/O manager that is available to drivers and applications. The Codec Translator seamlessly connects channels that compressed with differentcodec's. Most of Asterisk's flexibility comes from the applications, codec's, channel drivers, file formats and other facilities interaction with the various programming interfaces.

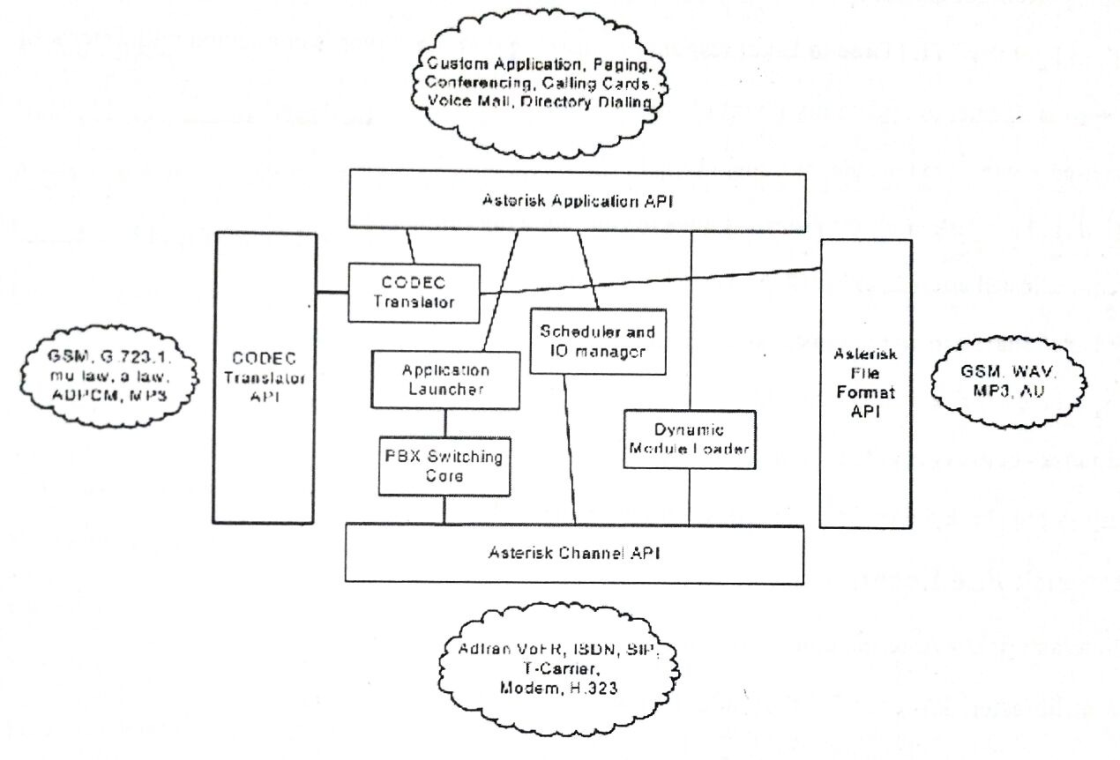


Fig 4.1 Asterisk Architecture

**Development and scripting**

If you want to add on to Asterisk, there are many ways to add functionality.

· Using the applications in the dial plan to build solutions. There are scripting commands

Like gotoif, variables to test and set and string handling functions to control what happens when a user dials an extension.

· Asterisk AGI: The application interface for extending the dialplan with your functionality in the language you choose – PHP, Perl, C, Java, Unix Shell and others

· manager: The manager API for connecting to the PBX from your application

· And the C API, documented in the source codes and documentation you generate from within the source code tree

Note that Asterisk is licensed by Digium with the GPL license. This means that you are free to modify the code and distribute your solution to customers, but you must also make the

modifications available under the GPL. If you want your modifications to become part of the

official Asterisk code repository, you need to waive the rights to the code to Digium.

**Applications**

To connect incoming calls to outbound connections or other local users asterisk consist of many applications, command you use to create a working PBX. From simple logic like Asterisk cmdGoto to more complex applications like Asterisk cmdVoiceMail and Asterisk cmdMeetMe.

**Starting Asterisk**

sudosafe\_asterisk ;for safe opening of asterisk interface

sudo asterisk –r ;for opening the asterisk command line interface

**Asterisk File Locations**

/etc/asterisk/- Asterisk configuration files

/var/lib/asterisk/- contains the astdb, fimware and keys

/usr/share/asterisk/sounds-in built asterisk sound prompts

/var/spool/asterisk/- temporary files and voicemail files

var/log/asterisk/- Asterisk log files

/var/log/asterisk/cdr-csv/- Asterisk call detail records

**ASTERISK CONFIGURATIONS DETAILS**

Text based configuration files:

1.Sip.conf

2.extensions.conf

**SIP.Conf**

These are two extensions 400&401 and a trunk under

Cd /etc/asterisk

Vim sip.conf

[general]

context=default ; Default context for incoming calls

videosupport=yes

disallow=all

allow=all

allow=g729

allow=gsm

allow=ulaw

allow=alaw

allow=g711u

allow=h263p

allow=h263

allow=h261

[400]

type=friend

user=400

secret=phone1

host=dynamic

context=public

[401]

type=friend

user=401

secret=phone2

host=dynamic

context=public

**extensions.conf**

To give the incoming and outgoing route for the extension we have to create and need to add thesefiles are given below at extensions.conf file:

#vim extensions.conf

[general]

[public]

exten => 435,1,Ringing()

exten => 435,2,Answer()

exten => 435,3,Dial(SIP/400)

exten => 435,4,SendText()

exten => 435,5,Hangup

exten => 999,1,Ringing()

exten => 999,2,Answer()

exten => 999,3,Dial(SIP/401)

exten => 435,4,SendText()

exten => 999,4,Hangup

exten => 123,1,Ringing()

exten => 123,2,Answer()

exten => 123,3,Playback(welcome1)

exten => 123,4,Playback(check-number-dial-again)

exten => 123,5,Playback(check-number-dial-again)

exten => 123,6,SendText()

exten => 123,7,Hangup()

exten => 400,1,Set(from=${MESSAGE(from)})

exten => 400,2,Set(MESSAGE(From)=${MESSAGE(from)})

exten => 400,3,Set(MESSAGE(body)=${MESSAGE(body)})

exten => 400,4,MessageSend(sip:400,${from})

exten => 401,1,Set(from=${MESSAGE(from)})

exten => 401,2,Set(MESSAGE(From)=${MESSAGE(from)})

exten => 401,3,Set(MESSAGE(body)=${MESSAGE(body)})

exten => 401,4,MessageSend(sip:401,${from})

**ASTERISK COMMANDS**

|  |  |
| --- | --- |
| sip set debug | Turns on sip debugging. |
| Reload | Reload the sip channel module. This is the equivalent of performing a reload chan\_sip. Reloading the SIP channel is required to load changes to sip.conf and sip\_notify.conf into memory. Active SIP channel are not dropped during a sip reload. |
| sip show channel | Displays extended information about an active SIP channel. See also sip show channels. |
| sip show history | Displays detailed information about a peer configured in sip.conf |
| sip show peers | Lists and displays the status of all SIP peers. |
| sip show registry | Lists and displays the status of all peers with whom you are registered. |

**CHAPTER 5**

**PYTHON AND MYSQL**

**PYTHON**

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

* **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
* **Python is Interactive** − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
* **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
* **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

**HISTORY OF PYTHON**

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL). Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

**PYTHON FEATURES**

Python's features include:

* **Easy-to-learn**: Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
* **Easy-to-read**: Python code is more clearly defined and visible to the eyes.
* **Easy-to-maintain**: Python's source code is fairly easy-to-maintain.
* **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
* **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
* **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
* **Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
* **Databases** − Python provides interfaces to all major commercial databases.
* **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
* **Scalable** − Python provides a better structure and support for large programs than shell scripting.

**Pycall**

**Call Files**

Since pycall is a python library for “creating and using Asterisk call files.

Call files are specially formatted text files that Asterisk uses to initiate outbound calls automtically. In a nutshell, Asterisk call files allow developers to automatically generate calls and launch programs through Asterisk, without any sort of complex network API.

**Use pycall**

* Simple pycall makes building telephony applications easy.
* Flexible pycall gives you a lot of flexibility. If you’re Asterisk environment is setup in a non-standard way, pycall won’t mind.

Pycall can be installed just like any other python program, through [pypi](http://pypi.python.org/pypi). To install it, simply run:

$ pip install pycall

If you’d like to install the latest development version of pycall (not recommended), then [download the latest release](https://github.com/rdegges/pycall) and run:

$ python setup.py install

**Spooling Directory**

If you’re Asterisk install doesn’t spool to the default /var/spool/asterisk/outgoing directory, you can override it with the spool\_dir attribute

: cf = CallFile(..., spool\_dir=’/tmp/outgoing’)

**PY\_ASTERISK**

The Python Asterisk package (codenamed py-Asterisk) is an attempt to produce high quality, well documented Python bindings for the Asterisk Manager API.The eventual goal of the package is to allow rich specification of the Asterisk configuration in Python rather than in the quirky, unstructured, undocumented mess that we call the Asterisk configuration files.

**Working Functionality**

* Python package implementing a manager client and event dispatcher.
* User-oriented command line interface to manager API.
* Asterisk module allowing dialplan configuration via the manager API (see [PbxConfigMgr my pbx\_config\_mgr page]).
* Objects to represent the standard applications, for specification of dialplan configuration in a friendlier Python syntax.

For installing following command is used

pip install py-Asterisk

wherepy-asterisk automatically calls asterisk commands of operating system.

**PY-MYSQL**

MySQL is an [open source](https://en.wikipedia.org/wiki/Open-source_software) [relational database management system](https://en.wikipedia.org/wiki/Relational_database_management_system) (RDBMS).Its name is a combination of "My" the name of co-founder [Michael Widenius](https://en.wikipedia.org/wiki/Michael_Widenius)'s daughter and "[SQL](https://en.wikipedia.org/wiki/SQL)", the abbreviation for [Structured Query Language](https://en.wikipedia.org/wiki/Structured_Query_Language).

MySQL is [free and open-source software](https://en.wikipedia.org/wiki/Free_and_open-source_software) under the terms of the [GNU General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License), and is also available under a variety of [proprietary](https://en.wikipedia.org/wiki/Proprietary_software) licenses. MySQL was owned and sponsored by the [Swedish](https://en.wikipedia.org/wiki/Sweden) company [MySQL AB](https://en.wikipedia.org/wiki/MySQL_AB), which was bought by Sun Microsystems (now [Oracle Corporation](https://en.wikipedia.org/wiki/Oracle_Corporation)). In 2010, when Oracle acquired Sun, Widenius [forked](https://en.wikipedia.org/wiki/Fork_(software_development)) the [open-source](https://en.wikipedia.org/wiki/Open-source) MySQL project to create [MariaDB](https://en.wikipedia.org/wiki/MariaDB" \o "MariaDB).

MySQL is a component of the [LAMP](https://en.wikipedia.org/wiki/LAMP_(software_bundle)) [web application](https://en.wikipedia.org/wiki/Web_application) [software stack](https://en.wikipedia.org/wiki/Software_stack) (and [others](https://en.wikipedia.org/wiki/List_of_AMP_packages)), which is an acronym for [Linux](https://en.wikipedia.org/wiki/Linux), [Apache](https://en.wikipedia.org/wiki/Apache_HTTP_Server), MySQL, [Perl](https://en.wikipedia.org/wiki/Perl)/[PHP](https://en.wikipedia.org/wiki/PHP)/[Python](https://en.wikipedia.org/wiki/Python_(programming_language)). MySQL is used by many database-driven web applications, including [Drupal](https://en.wikipedia.org/wiki/Drupal" \o "Drupal), [Joomla](https://en.wikipedia.org/wiki/Joomla" \o "Joomla), [phpBB](https://en.wikipedia.org/wiki/PhpBB" \o "PhpBB), and [WordPress](https://en.wikipedia.org/wiki/WordPress" \o "WordPress). MySQL is also used by many popular [websites](https://en.wikipedia.org/wiki/Website), including [Google](https://en.wikipedia.org/wiki/Google), [Facebook](https://en.wikipedia.org/wiki/Facebook), [Twitter](https://en.wikipedia.org/wiki/Twitter), [Flickr](https://en.wikipedia.org/wiki/Flickr), and [YouTube](https://en.wikipedia.org/wiki/YouTube).

The last stable release is available on PyPI and can be installed with pip:

$ python3 -m pip install PyMySQL

In python to access to mysql below command is used:

connection=pymysql.connect(host='localhost',user='user, password='passwd', db='db')

**MySQL**

A database is a separate application that stores a collection of data. Each database has one or more distinct APIs for creating, accessing, managing, searching and replicating the data it holds.MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons

* MySQL is released under an open-source license. So you have nothing to pay to use it.
* MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
* MySQL uses a standard form of the well-known SQL data language.
* MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
* MySQL works very quickly and works well even with large data sets.
* MySQL is very friendly to PHP, the most appreciated language for web development.
* MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).
* MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

**Installing MySQL**

If you don't have MySQL installed on your droplet, you can quickly download it.

Ubuntu:

sudo apt-get install mysql-server

**Access the MySQL shell**

Access the MySQL shell by typing the following command into terminal:

mysql -u root -p

After entering the root MySQL password into the prompt ,you will be able to start building your MySQL database.

* All MySQL commands end with a semicolon; if the phrase does not end with a semicolon, the command will not execute.
* Also, although it is not required, MySQL commands are usually written in uppercase and databases, tables, usernames, or text are in lowercase to make them easier to distinguish. However, the MySQL command line is not case sensitive.

**Create and Delete a MySQL Database**

MySQL organizes its information into databases each one can hold tables with specific data.

You can quickly check what databases are available by typing:

SHOW DATABASES;

Your screen should look something like this:mysql> SHOW DATABASES;

+--------------------+

| Database |

+--------------------+

| information\_schema |

| mysql |

| performance\_schema |

| test |

+--------------------+

4 rows in set (0.01 sec)

For Creating a database :

CREATE DATABASE database name;

In this case, for example, we will call our database "events."

mysql> SHOW DATABASES;

+--------------------+

| Database |

+--------------------+

| information\_schema |

| events |

| mysql |

| performance\_schema |

| test |

+--------------------+

5 rows in set (0.00 sec)

In MySQL, the phrase most often used to delete objects is Drop. You would delete a MySQL database with this command:

DROP DATABASE database name;

**Access a MySQL Database**

Once we have a new database, we can begin to fill it with information.

The first step is to create a new table within the larger database.

Let’s open up the database we want to use:

USE events;

In the same way that you could check the available databases, you can also see an overview of the tables that the database contains.

SHOW tables;

Since this is a new database, MySQL has nothing to show, and you will get a message that says, “Empty set”

**Create a MySQL Table**

Let’s imagine that we are planning a get together of friends. We can use MySQL to track the details of the event.

Let’s create a new MySQL table:

CREATE TABLE potluck (id INT NOT NULL PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(20),

food VARCHAR(30),

confirmed CHAR(1),

signup\_date DATE);

This command accomplishes a number of things:

* It has created a table called potluck within the directory, events.
* We have set up 5 columns in the table—id, name, food, confirmed, and signup date.
* The “id” column has a command (INT NOT NULL PRIMARY KEY AUTO\_INCREMENT) that automatically numbers each row.
* The “name” column has been limited by the VARCHAR command to be under 20 characters long.
* The “food” column designates the food each person will bring. The VARCHAR limits text to be under 30 characters.
* The “confirmed” column records whether the person has RSVP’d with one letter, Y or N.
* The “date” column will show when they signed up for the event. MySQL requires that dates be written as yyyy-mm-dd

Let’s take a look at how the table appears within the database using the "SHOW TABLES;" command:

mysql> SHOW TABLES;

+------------------+

| Tables\_in\_events |

+------------------+

| potluck |

+------------------+

1 row in set (0.01 sec)

**Adding Information to a MySQL Table**

We have a working table for our party. Now it’s time to start filling in the details. Use this format to insert information into each row:

INSERT INTO potluck VALUES (1, "John", "Casserole","Y", '2012-04-11');

Once you input that in, you will see the words:

Query OK, 1 row affected (0.00 sec)

Let’s add a couple more people to our group:

INSERT INTO potluck VALUES (2, "Sandy", "Key Lime Tarts","N", '2012-04-14');

INSERT INTO potluck VALUES (3, "Tom", "BBQ","Y", '2012-04-18');

INSERT INTO potluckVALUES (4, "Tina", "Salad","Y", '2012-04-10');

We can take a look at our table:

mysql> SELECT \* FROM potluck;

+----+-------+----------------+-----------+-------------+

| id | name | food | confirmed | signup\_date |

+----+-------+----------------+-----------+-------------+

| 1 | John | Casserole | Y | 2012-04-11 |

| 2 | Sandy | Key Lime Tarts | N | 2012-04-14 |

| 3 | Tom | BBQ | Y | 2012-04-18 |

| 4 | Tina | Salad | Y | 2012-04-10 |

+----+-------+----------------+-----------+-------------+

4 rows in set (0.00 sec)

**To Update Information in the Table**

Now that we have started our potluck list, we can address any possible changes. For example: Sandy has confirmed that she is attending, so we are going to update that in the table.

UPDATE potluck SET confirmed = 'Y' WHERE name = ‘Sandy';

**Add and Delete a Column**

We are creating a handy chart, but it is missing some important information: our attendees’ emails.We can easily add this:

ALTER TABLE potluck ADD email VARCHAR(40);

This command puts the new column called "email" at the end of the table by default, and the VARCHAR command limits it to 40 characters.

However, if you need to place that column in a specific spot in the table, we can add one more phrase to the command.

ALTER TABLE potluck ADD email VARCHAR(40) AFTER name;

Now the new “email” column goes after the column “name”.

Just as you can add a column, you can delete one as well:

ALTER TABLE potluck DROP email;

I guess we will never know how to reach the picnickers.

**Delete a Row**

If needed, you can also delete rows from the table with the following command:

DELETE from [table name] where [column name]=[field text];

mysql> DELETE from potluck where name='Sandy';

Query OK, 1 row affected (0.00 sec)

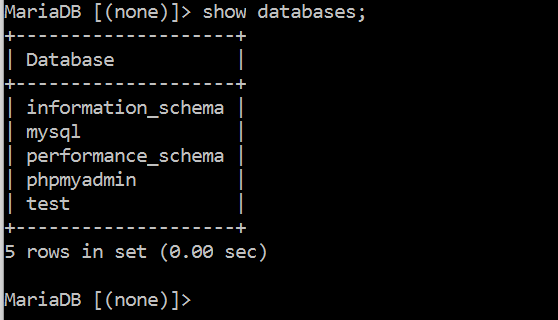
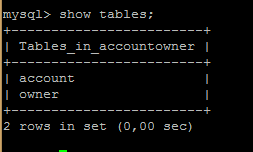
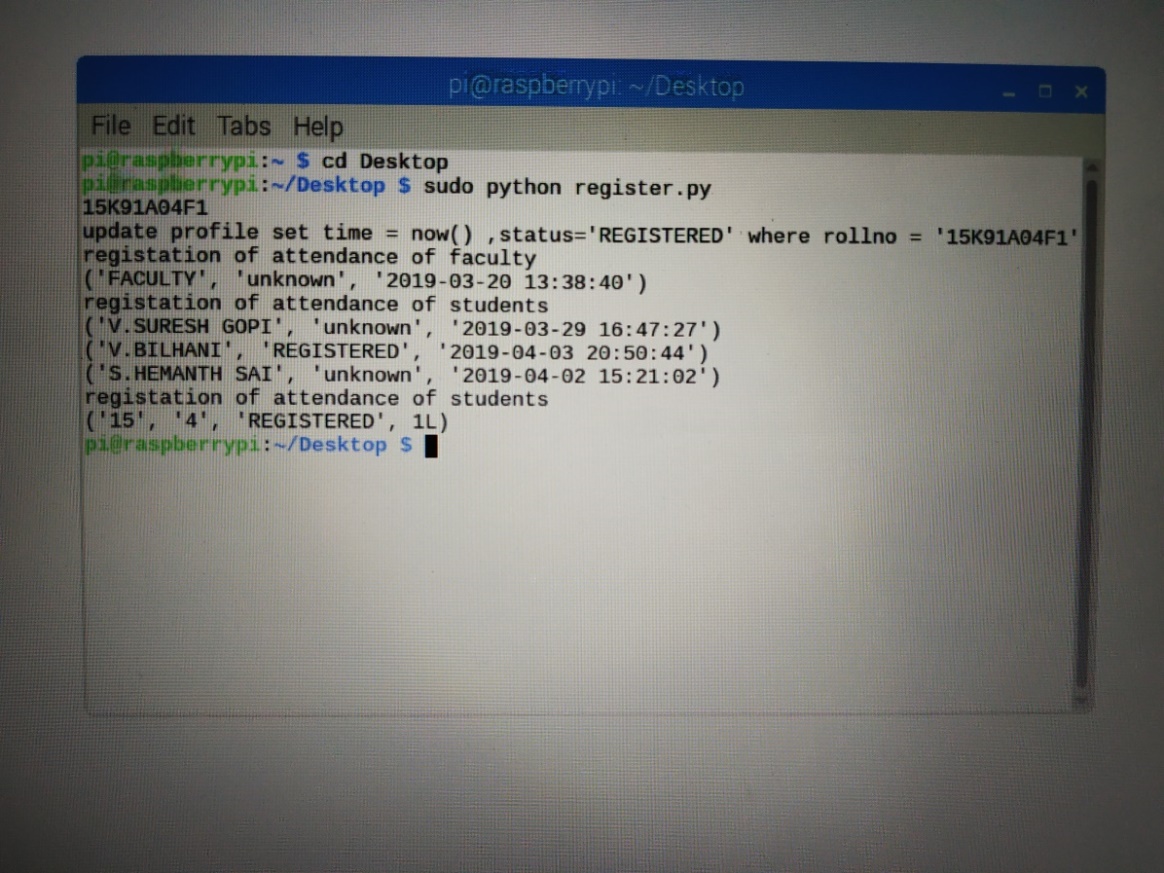
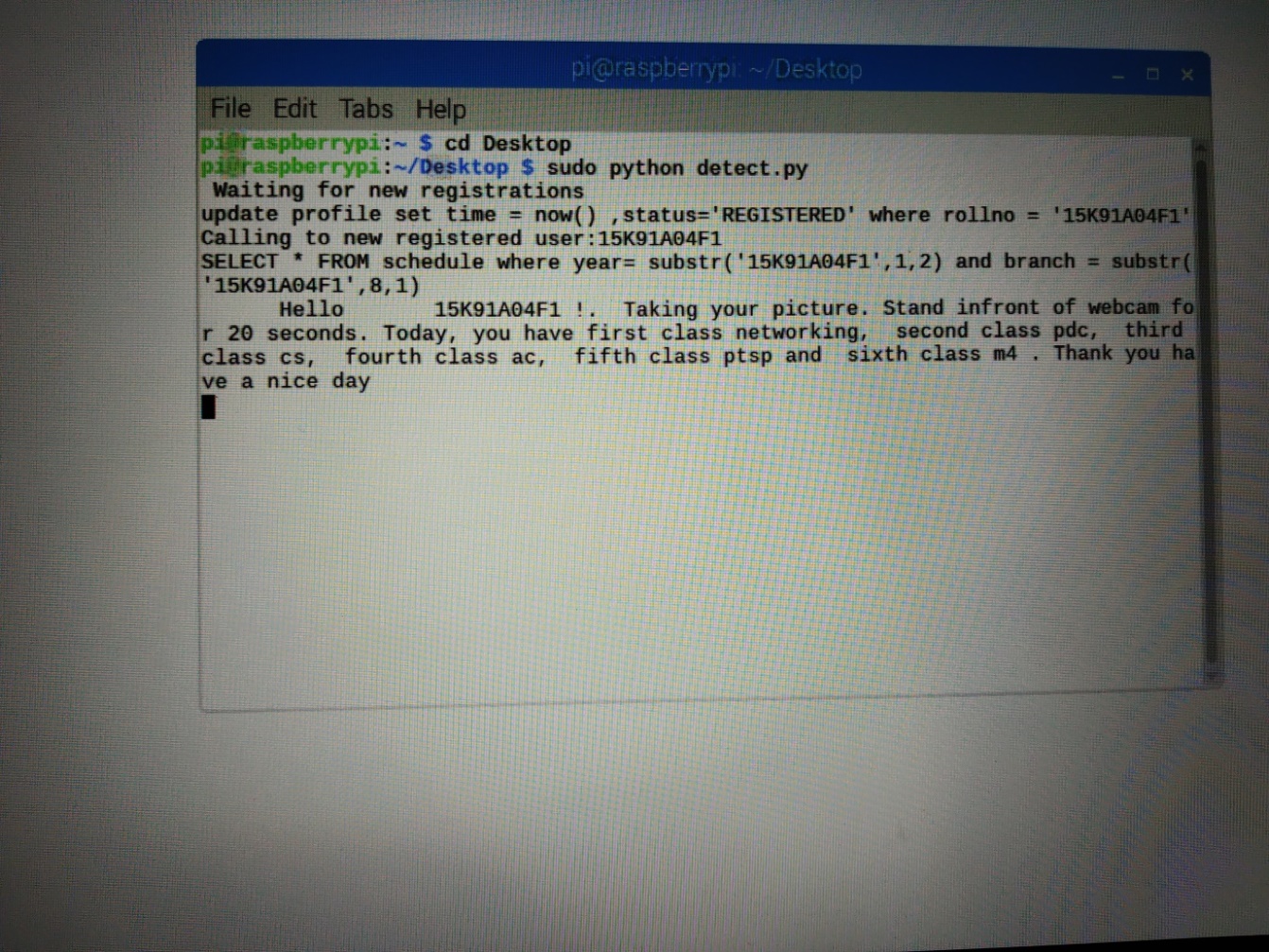
 

Fig 5.1 MySQL database Fig 5.2 Tables in database

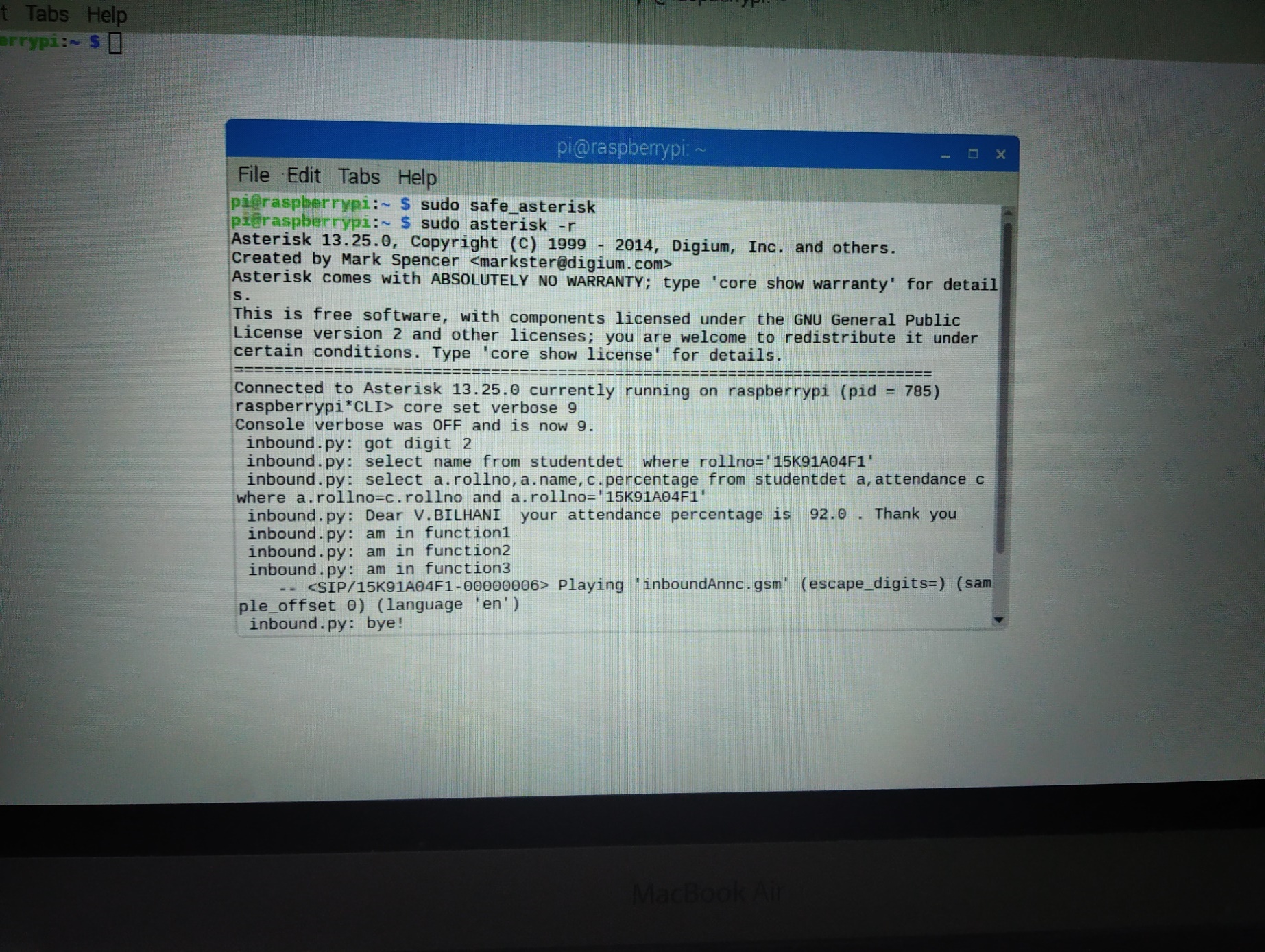
**OUTPUTS**

****

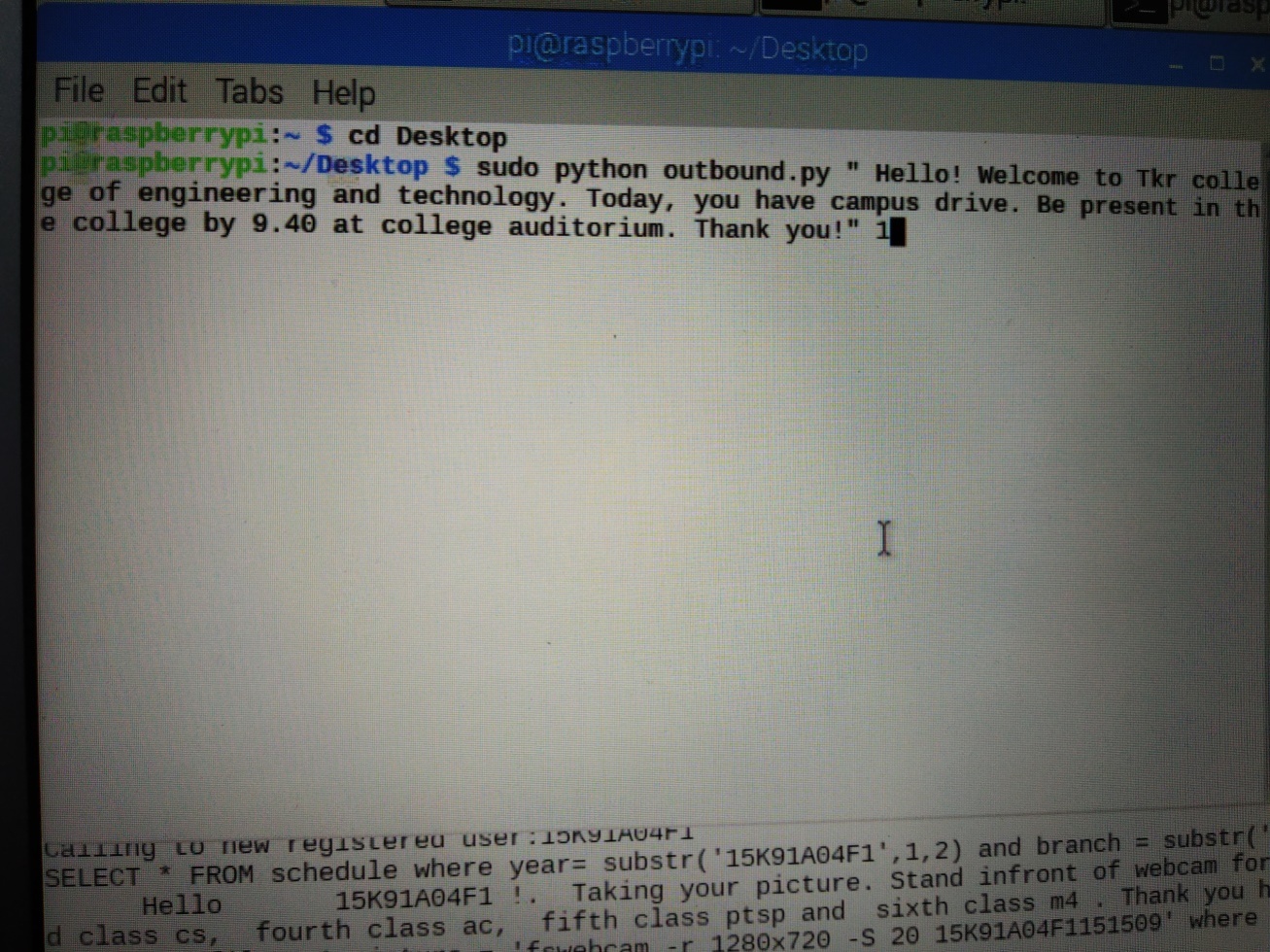
**Fig 6.1 REGISTER.PY**

****

**Fig 6.2 DETECT.PY**

****

**Fig 6.3 INBOUND.PY**

****

**Fig 6.4 OUTBOUND.PY**

**ADVANTAGES**

The advantages of smart campus are as given below:

***EASY TO INSTALL AND MAINTAIN***

Smart Campuses are easier to install. There is not much technical knowledge needed for this. There are no extra wires or cables required.

***LOWER COST***

VOIP service can be used at a very low cost. If you have an internet connection, you can almost consider using VOIP as free. All you have to pay is the internet charges as VOIP involves no extra charge for hardware, wires, cables etc.

***PORTABLE NUMBER***

A VOIP number, generally known as virtual number, is completely portable. The same number can be used wherever you go. This is especially useful for people who travel frequently. Even if the address is changed they can be reached in the VOIP number.

***LOWER TAXES***

If you take a look at your telephone bill, you can see that you pay a lot of money as tax. But the government has not imposed heavy taxes on VOIP phone services. Thus, choosing VOIP over the traditional phone service is a great money saver.

***ADDITIONAL FEATURES***

Apart from announcements and calls, group discussions can be done and attendance could also be marked, which makes the work easier.

**DISADVANTAGES**

Every project has its own demerits. So, smart campus also has it. They are,

***LESS RELIABLE***

The quality of the calls largely depends on the internet connection. If the traffic on the network is high, the voice quality drops. This is commonly seen in long distance or international calls where the voice appears distorted. This may lead to a problem, mainly for business calls where communication needs to be quick and action must be taken on the basis of the response.

***AFFECTED BY POWER CUTS***

This service requires regular power supply. Calls are not possible if there is an interruption in the power supply. Power supply should always be given to the server i.e., Raspberry Pi.

***SECURITY***

Internet services have always faced the issue of security. Same is the case with this also.

Telephone hacking has been a major concern in this regard. Due to the less time given for scanning data packets, the performance of firewalls may be less than satisfactory. VOIP is also affected by worms and virus. All these pose a threat to the security.

***LIMITED COVERAGE AREA***

Since the connectivity used is Wi-Fi, communications can be made within the limited area that is, activities can happen only within the Wi-Fi covered area.

**CONCLUSION**

A Smart campus intended connecting the worldwide educational resources to faculty and students contributing in intellectual development. A smart campus provides various services to students, which makes the students campus life easier, comfortable and attractive. These services are not just for academic life but also for socializing, moving around, sharing events. Student’s thinking is impacted at least in three directions: academic, practical and social ways. Existing campuses could think of alternatives for positively impacting the student in these three ways for a better socio-economic society tomorrow.

Campus to be provided with the facilities with broader scope to the student by providing facility to listen, access and interact with the eminent faculty with broader background.As laptops, iPads and Smartphones have become more prevalent, there are dramatic increases in WIFI usage. These features implemented by enabling the entire campus with Wi-Fi access to get the services by installing access points at suitable locations with wired back bone network. Backbone network connected to the VoIP server .All features are initiated by using SIP protocol. The Raspberry Pi which is a series of small single-board computers is used as VoIP server, which is economical and consumes less power compared to the traditional Desktop/Servers.This project in also useful to make Private branch exchange PBX network for Small Scale Industries, Apartments and College Campus.

**FUTURE SCOPE**

Smart campus is the wave of future. In the scope of smart campus these contextual factors can mean many things and thus lead to many applications and use cases, often also with the word smart in them and always with the [IoT](https://www.i-scoop.eu/internet-of-things-guide/)as one core element: [smart buildings](https://www.i-scoop.eu/internet-of-things-guide/facility-management-iot-smart-buildings/), smart transportation systems, smart governance, [smart healthcare](https://www.i-scoop.eu/internet-of-things-guide/internet-things-healthcare/), smart infrastructure, smart cities, smart interactive information panels to perform myriad tasks.While the future of smart campus technologies might look bright and bring us to next stages of what is possible but most of all feasible and useful, many smart campus projects really have also shown to offer little value and, keeping the terrible events in some cities in mind, the ethical dimensions, regulations, prioritizations with regards to how technologies can make a real difference and revisiting the principles of good governance and looking at what is wrong in so many cities today must be part of any smart campus future discussion

Smart buildings, which use automated processes to control a variety of operations such as tracking and managing energy, environment, security and other key features, are expected to see stronger demand in the coming years. Since the connectivity used here is Wi-Fi, the coverage area is limited within LAN. But by using other wireless technologies like Wi-Max the smart campus could further be converted into a smart city which allows the users to access various facilities. As the near future is going to change to IPV6, Internet of Things is going to have a great scope which would become the major platform for all the developments taking place.

**REFERENCES**

• https://www.voip-info.org/asterisk/

• https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT

• “VOIP for dummies”, Timothy V. Kelly

• https://www.realvnc.com/en/raspberrypi/

• Rahaman MM. VoIP implementation uisng Asterisk PBX” IOSR

• https://www.raspberrypi.org/

• Journal of Business and Management(IOSR-JBM) e-ISSN: 2278-487X, p-ISSN: 2319-7668.2014; 15(6):47-53.www.iosrjournals.org

• Kwon, T. T., Gerla, M., & Das, S. (2002). Mobility management for VoIP service: Mobile IP vs.SIP. Wireless Communications, IEEE, 9(5), 66-75.

• B. Khasnabish, “Implementing Voice over IP”, Wiley-Inter Science, Lexington,

Massachusetts,USA.

• PPTs and PDFs provided by BSNL

• http://www.ijircce.com/upload/2016/october154\_31\_IMPLEMENTATION.pdf