UNIT 4

IOT COMMUNICATION AND OPEN PLATFORMS

1.1 IOT COMMUNICATION MODELS AND API

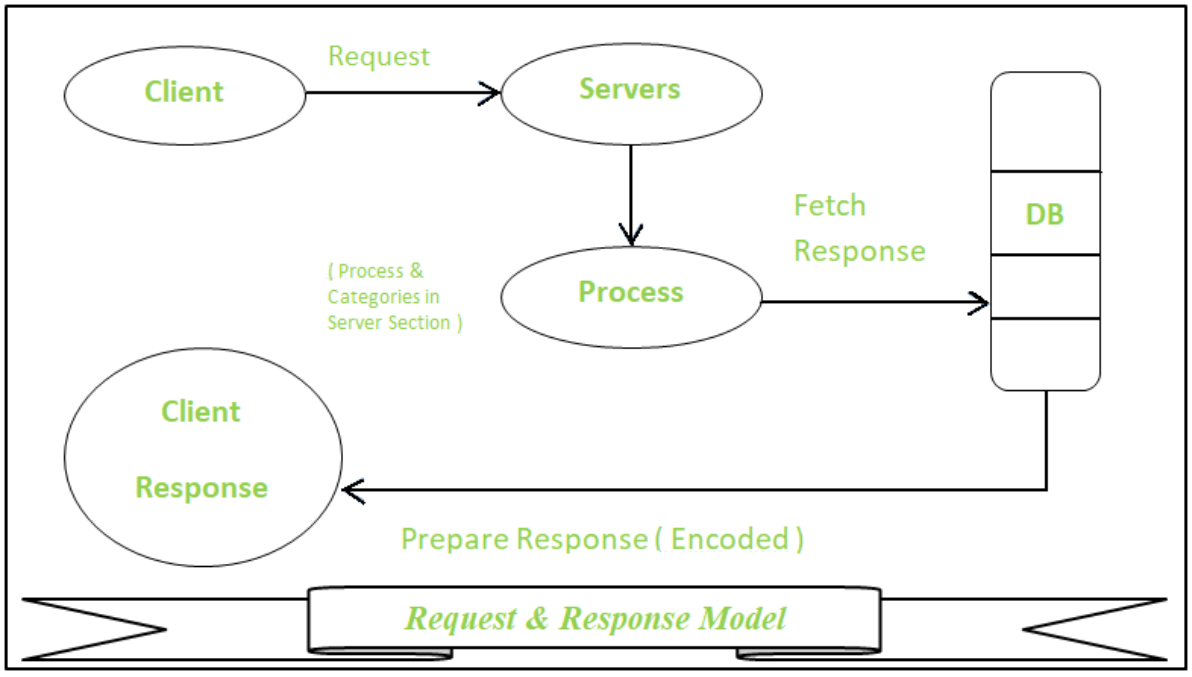
IoT devices are found everywhere and will enable circulatory intelligence in the future. For operational perception, it is important and useful to understand how various IoT devices communicate with each other. Communication models used in IoT have great value. The IoTs allow people and things to be connected any time, any space, with anything and anyone, using any network and any service.

Types of Communication Model:

1. **Request & Response Model –**

This model follows a client-server architecture.

- The **client**, when required, requests the information from the server. This request is usually in the encoded format.
- This model is stateless since the data between the requests is not retained and each request is independently handled.
- The server Categories the request, and fetches the data from the database and its resource representation. This data is converted to response and is transferred in an encoded format to the client. The client, in turn, receives the response.
- On the other hand — In **Request-Response** communication model client sends a request to the server and the server responds to the request. When the server receives the request it decides how to respond, fetches the data retrieves resources, and prepares the response, and sends it to the client.

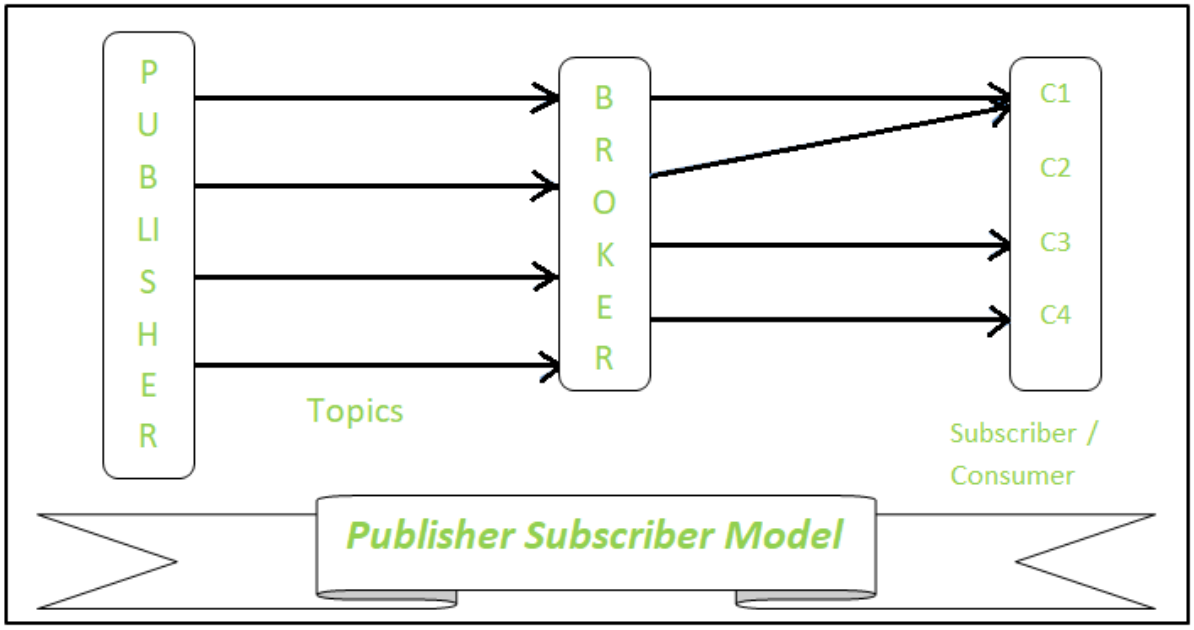


2. **Publisher-Subscriber Model –**

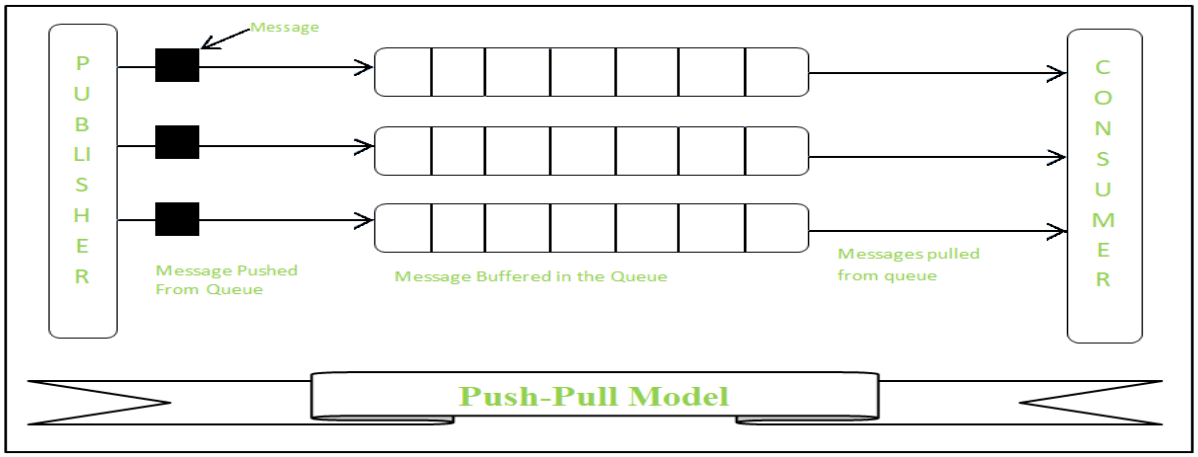
This model comprises three entities: Publishers, Brokers, and Consumers.

- **Publishers** are the source of data. It sends the data to the topic which are managed by the broker. They are not aware of consumers.
- **Consumers** subscribe to the topics which are managed by the broker.

- Hence, **Brokers** responsibility is to accept data from publishers and send it to the appropriate consumers. The broker only has the information regarding the consumer to which a particular topic belongs to which the publisher is unaware of.



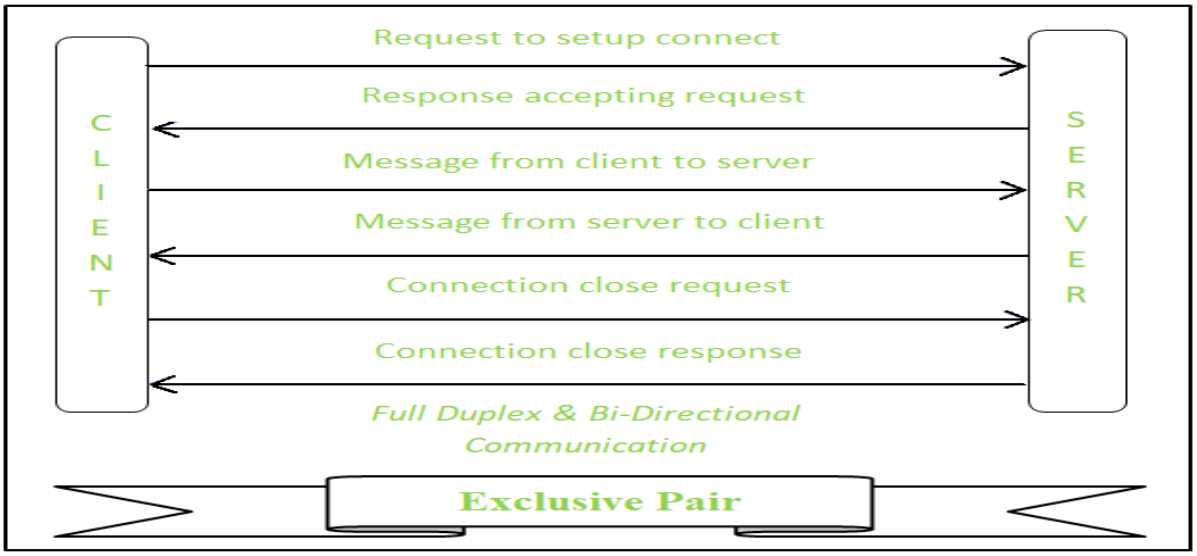
3. **Push-Pull Model –**
   The push-pull model constitutes data publishers, data consumers, and data queues.
   - **Publishers** and **Consumers** are not aware of each other.
   - Publishers publish the message/data and push it into the queue. The consumers, present on the other side, pull the data out of the queue. Thus, the queue acts as the buffer for the message when the difference occurs in the rate of push or pull of data on the side of a publisher and consumer.
   - **Queues** help in decoupling the messaging between the producer and consumer. Queues also act as a buffer which helps in situations where there is a mismatch between the rate at which the producers push the data and consumers pull the data.



4. Exclusive Pair –

- **Exclusive Pair** is the bi-directional model, including full-duplex communication among client and server. The connection is constant and remains open till the client sends a request to close the connection.
- The **Server** has the record of all the connections which has been opened.

- This is a state-full connection model and the server is aware of all open connections.
- WebSocket based communication API is fully based on this model.



IOT COMMUNICATION API

The Internet of Things (IoT) has changed how we interact with devices and created new opportunities for businesses to gather and analyze data. However, this has created a demand for device−to−device communication that is seamless. This is where IoT Communications APIs come to play a major role in communication systems.

IoT Communications APIs are a set of protocols and interfaces that allow IoT devices and platforms to communicate with one another. These APIs enable developers to create apps that interface with IoT devices over conventional web protocols including HTTP, MQTT, CoAP, and others.

Internet of Things Communications APIs standardizes how IoT devices communicate with one another and with other systems. They enable developers to design applications and services that can access and operate IoT devices without worrying about underlying communication protocols and interfaces.

So, let's go over some of the most crucial advantages of Communications APIs.

Benefits of Communication APIs in IoT

There are multiple benefits to adopting Internet of Things Communications APIs. One of the primary benefits is that they give a standardized method of connecting with IoT devices, making it simpler for developers to construct apps and services that work with many types of devices. This can save developers a lot of time and effort because they don't have to establish specific communication protocols for each device with which they want to interact.

Another benefit of adopting Communications APIs is that they might make the development process easier. Developers can utilize high−level APIs that abstract away the specifics of communication protocols and interfaces instead of writing low−level code to handle communication with IoT devices. This can assist minimize the amount of code required and make it easier to construct reliable and scalable systems.

IoT Communications APIs can also aid in the enhancement of security. APIs may ensure that devices communicate securely and that data is encrypted and secured from unauthorized access by offering a standardized communication method with IoT devices. This can aid in the prevention of data breaches and other security issues.

Internet of Things Communications APIs are also built to be extremely scalable. As more devices are added to an IoT system, the APIs can handle the increasing traffic while also ensuring that device communication stays reliable and consistent.

Another advantage of adopting communications APIs is that they make the building of IoT applications easier. Developers can utilize pre−built APIs to connect their apps to IoT devices and platforms, saving time and reducing development complexity.

Types of Communication APIs in IoT

There are numerous IoT communications APIs accessible, each with its own set of advantages and disadvantages. Among the most popular APIs are:

- **REST APIs**
  REST APIs are a popular choice for IoT communications because they offer a straightforward and scalable method of exposing IoT device capabilities over the web. REST APIs can be used to access and alter IoT data and devices, allowing developers to create novel IoT−powered apps.
  Internet of Things communications APIs that use REST APIs follow the same concepts as normal REST APIs but with an emphasis on IoT−specific resources and capabilities. An IoT Communications API, for example, may provide resources such as sensors, actuators, or data streams.
- **WebSocket**
  WebSocket is a protocol that provides for bidirectional, real−time communication between a client and a server. WebSocket is frequently used in the context of IoT Communications APIs to provide low−latency, real−time data streaming between IoT devices and cloud services.
  WebSocket, unlike HTTP, is a persistent connection between the client and server, allowing for real−time communication without the need for constant polling. This

makes it perfect for Internet of Things (IoT) applications requiring real−time data updates, such as monitoring and control systems.

- **CoAP**

  Constrained Application Protocol (CoAP) is a protocol used in Communications APIs for resource−constrained devices and networks. It is a simple protocol for exposing and manipulating IoT device resources via the internet.

  CoAP is a low−overhead protocol with tiny message sizes and minimum header information. As a result, it is well suited for use in IoT devices with low processing power and memory. CoAP is also intended to function with restricted networks, such as those based on the IEEE 802.15.4 or 6LoWPAN standards.

- **MQTT**

  MQTT (Message Queuing Telemetry Transport) is a lightweight messaging protocol used in IoT Communications APIs to link devices to the cloud. MQTT was created to be efficient, dependable, and simple to use, making it an excellent choice for IoT applications.

  MQTT is a publish or subscribe messaging model that allows devices to submit messages to particular topics and receive messages from topics to which they are subscribed.

  In addition to these APIs, there are IoT Communications Platforms that offer a complete endto−end IoT communication solution. These platforms often feature pre−built APIs, data visualization dashboards, and other tools that make the development process more manageable.

  AWS IoT, Azure IoT, and Google Cloud IoT are some popular IoT Communications Platforms.

1.2 IOT COMMUNCATION PROTOCOLS

The IoT Data Link communication protocol provides service to the Network Layer. There are various protocols and standard technologies specified by the different organization for data link protocols.

Bluetooth

Bluetooth is a short-range wireless communication network over a radio frequency. Bluetooth is mostly integrated into smartphones and mobile devices. The Bluetooth communication network works within 2.4 ISM band frequencies with data rate up to 3Mbps.

There are three categories of Bluetooth technology:

1. Bluetooth Classic
2. Bluetooth Low Energy

3. Bluetooth SmartReady

Properties of Bluetooth network

- **Standard:** Bluetooth 4.2
- **Frequency:** 2.4GHz
- **Range:** 50-150m
- **Data transfer rates:** 3Mbps

Advantages of Bluetooth network

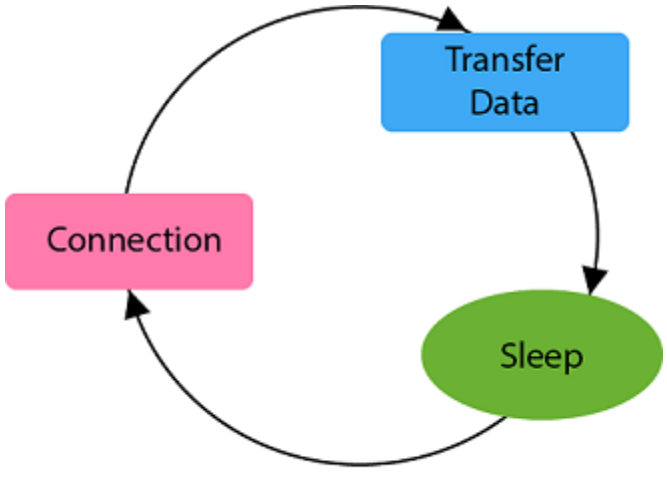
- It is wireless.
- It is cheap.
- It is easy to install.
- It is free to use if the device is installed with it.

Disadvantages of Bluetooth network

- It is a short-range communication network.
- It connects only two devices at a time.

Bluetooth Low Energy

Bluetooth low energy (BLE) is a short-range communication network protocol with PHY (physical layer) and MAC (Medium Access Control) layer. It is designed for low-power devices which uses less data. BLE always remain in sleep mode except when the connection between devices is initiated and data transmission occurs, due to this it conserves power of the device. Bluetooth low energy follows the master/slave architecture and offers two types of frames that are adverting and data frames. Slave node sent the advertising frame to discover one or more dedicated advertisement channels. Master nodes sense this advertisement channels to find slaves and connect them.



BLE Technology

Z-Wave

Z-Wave is a wireless communication protocol with the frequency of 900MHz. The ranges of Z-Wave lies between 30 meters to 100 meters with the data transfer rate of 100kbps so that it is suitable for small messages in IoT applications for home automation. This communication protocol operates on mesh network architecture with one and several secondary controllers.



Properties of Z-Wave protocol

- **Standard:** Z-Wave Alliance ZAD12837 / ITU-T G.9959
- **Frequency:** 908.42GHz
- **Range:** 30-100m
- **Data transfer rate:** 100kbps

Advantages of Z-Wave protocol

- Low power consumption
- Remote or local control
- Simple installation
- Interoperability

Application of Z-Wave protocol

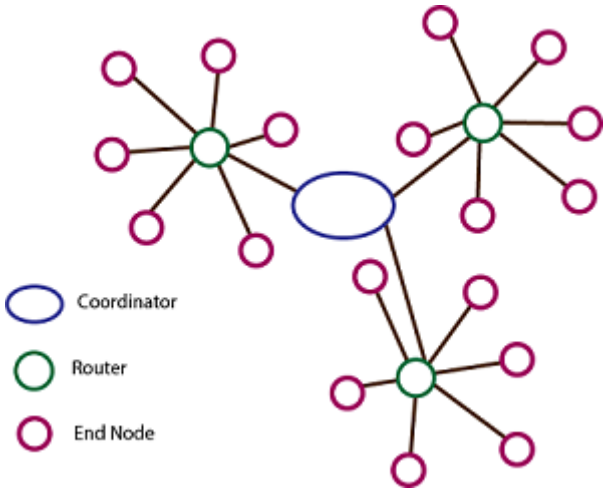
- Smart product and IoT based application
- Energy saving
- Home security

ZigBee Smart Energy

ZigBee is a low power, low data rate wireless personal area network communication protocol. It is mostly used in home automation and industrial settings. Since ZigBee is a low power communication protocol, the IoT power devices used with ZigBee technology. The ZigBee communication protocol is based on the IEEE 802.15.4 standard operating at the 2.4GHz frequency. The ZigBee protocol supports star, cluster or wireless mesh technology topology.

ZigBee uses the following devices in its network:

- Zigbee Coordinator
- Zigbee End Device
- Zigbee Router



Properties of ZigBee protocol

- **Standard:** ZigBee 3.0 based on IEEE802.15.4
- **Frequency:** 2.4GHz
- **Range:** 10-100m
- **Data transfer rate:** 250kbps

Advantages of ZigBee protocol

- Wireless
- Mesh networking
- Direct communication
- Low power consumption

Disadvantages of ZigBee protocol

- Costly
- Works with low speed within a small distance

Application of ZigBee protocol

- Commercial and residential control
- Personal and healthcare
- Home networking
- Industrial control and management
- Consumer electronics

LoRaWAN

LoRaWAN refers to Long Rage Wide Area Network which is a wide area network protocol. It is an optimized low-power consumption protocol design to support large-scale public networks with millions of low-power devices. A single operator operates the LoRaWAN. The LoRaWAN network is a bi-directional communication for IoT application with low cost, mobility, and security.

Properties of LoRaWAN protocol

- **Standard:** LoRaWAN
- **Frequency:** Various
- **Range:** 2-5km (urban environment), 15km (suburban environment)
- **Data Rates:** 0.3-50 kbps.

Wi-Fi

WiFi connectivity is one of the most popular IoT communication protocol, often an obvious choice for many developers, especially given the availability of WiFi within the home environment within LANs.
There is a wide existing infrastructure as well as offering fast data transfer and the ability to handle high quantities of data.
Currently, the most common WiFi standard used in homes and many businesses is 802.11n, which offers range of hundreds of megabit per second, which is fine for file transfers but may be too power-consuming for many IoT applications.

1.3 ZIGBEE

ZigBee is a Personal Area Network task group with low-rate task group 4. It is a technology of home networking. ZigBee is a technological standard created for controlling and sensing the network. As we know that ZigBee is the Personal Area Network of task group 4 so it is based on IEEE 802.15.4 and is created by Zigbee Alliance.

ZigBee is an open, global, packet-based protocol designed to provide an easy-to-use architecture for secure, reliable, low power wireless networks. Flow or process control equipment can be place anywhere and still communicate with the rest of the system. It can

also be moved, since the network doesn't care about the physical location of a sensor, pump or valve.

IEEE802.15.4 developed the PHY and MAC layer whereas, the ZigBee takes care of upper higher layers.

ZigBee is a standard that addresses the need for very low-cost implementation of Low power devices with Low data rates for short-range wireless communications.

General Characteristics of Zigbee Standard:

- Low Power Consumption
- Low Data Rate (20- 250 kbps)
- Short-Range (75-100 meters)
- Network Join Time (~ 30 msec)
- Support Small and Large Networks (up to 65000 devices (Theory); 240 devices (Practically))
- Low Cost of Products and Cheap Implementation (Open Source Protocol)
- Extremely low-duty cycle.
- 3 frequency bands with 27 channels.

Operating Frequency Bands (Only one channel will be selected for use in a network):

1. **Channel 0**: 868 MHz (Europe)
2. **Channel 1-10**: 915 MHz (the US and Australia)
3. **Channel 11-26**: 2.4 GHz (Across the World)

Features of Zigbee:

1. Stochastic addressing: A device is assigned a random address and announced. Mechanism for address conflict resolution. Parents node doesn't need to maintain assigned address table.
2. Link Management: Each node maintains quality of links to neighbors. Link quality is used as link cost in routing.
3. Frequency Agility: Nodes experience interference report to channel manager, which then selects another channel
4. Asymmetric Link: Each node has different transmit power and sensitivity. Paths may be asymmetric.
5. Power Management: Routers and Coordinators use main power. End Devices use batteries.

Advantages of Zigbee:

1. Designed for low power consumption.
2. Provides network security and application support services operating on the top of IEEE.
3. Zigbee makes possible completely networks homes where all devices are able to communicate and be
4. Use in smart home
5. Easy implementation
6. Adequate security features.
7. **Low cost**: Zigbee chips and modules are relatively inexpensive, which makes it a cost-effective solution for IoT applications.

8. **Mesh networking:** Zigbee uses a mesh network topology, which allows for devices to communicate with each other without the need for a central hub or router. This makes it ideal for use in smart home applications where devices need to communicate with each other and with a central control hub.
9. **Reliability:** Zigbee protocol is designed to be highly reliable, with robust mechanisms in place to ensure that data is delivered reliably even in adverse conditions.

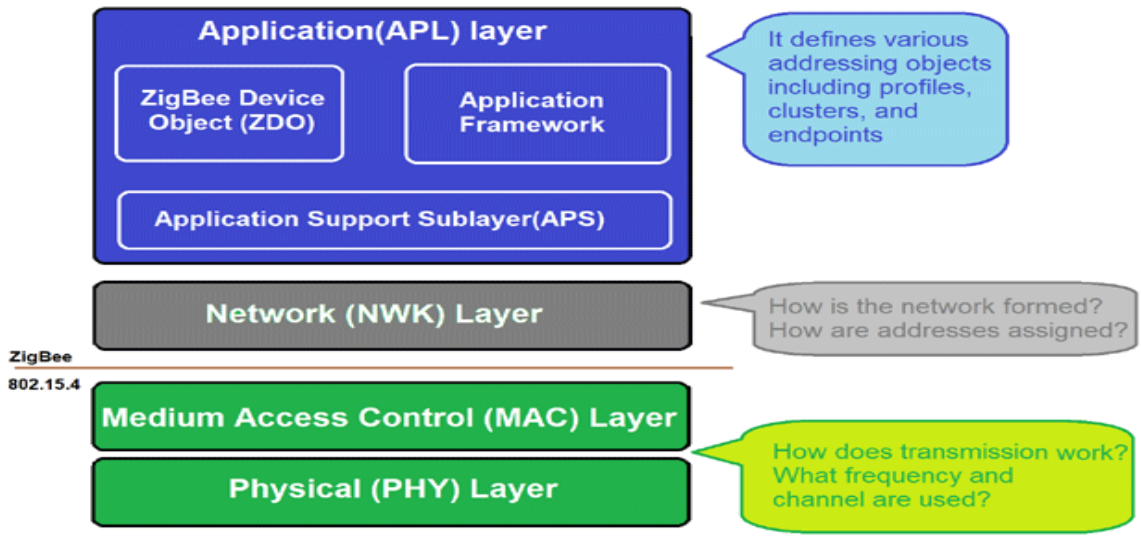
Disadvantages of Zigbee:

1. **Limited range:** Zigbee has a relatively short range compared to other wireless communications protocols, which can make it less suitable for certain types of applications or for use in large buildings.
2. **Limited data rate:** Zigbee is designed for low-data-rate applications, which can make it less suitable for applications that require high-speed data transfer.
3. **Interoperability:** Zigbee is not as widely adopted as other IoT protocols, which can make it difficult to find devices that are compatible with each other.
4. **Security:** Zigbee's security features are not as robust as other IoT protocols, making it more vulnerable to hacking and other security threats.

Zigbee Network Topologies:

- **Star Topology** (ZigBee Smart Energy): Consists of a coordinator and several end devices, end devices communicate only with the coordinator.
- **Mesh Topology** (Self Healing Process): Mesh topology consists of one coordinator, several routers, and end devices.
- **Tree Topology**: In this topology, the network consists of a central node which is a coordinator, several routers, and end devices. the function of the router is to extend the network coverage.

ZIGBEE ARCHITECTURE



Application layer defines various addressing objects including profiles, clusters, and endpoints. You can see the ZigBee stack layers in the figure above.

Network layer: It adds routing capabilities that allows RF data packets to traverse multiple devices (multiple "hops") to route data from source to destination (peer to peer).

MAC layer manages RF data transactions between neighboring devices (point to point). The MAC includes services such as transmission retry and acknowledgment management and collision avoidance techniques.

Physical layer: It defines how devices are connected to make a network; it defines the output power, number of channels and transmission rate. Most ZigBee applications operate on the 2.4 GHz ISM band at a 250kbps data rate.

Most XBee families have flow control, I/O, A/D and indicator lines built in which can be configured using appropriate commands. Analog samples are returned as 10-bit values. The analog reading is scaled such that 0x0000 represents 0V, and 0x3FF = 1.2V. (The analog inputs on the module cannot read more than 1.2V)

To convert the A/D reading to mV, do the following:

```
AD (mV) = (A/D reading * 1200mV) / 1023
```

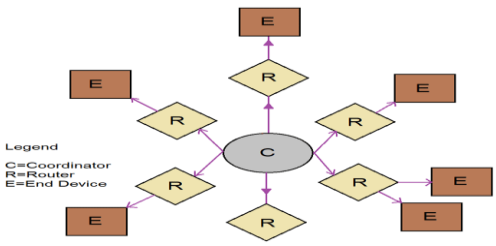
Data Transmission in ZigBee

You can call a network as combination of software and hardware which is capable of sending data from one location to another. Hardware is responsible for carrying the signals from one point of network to another. Software consists of instruction sets that make it possible to work as we expect.

Generally the data transmission ***by ZigBee packets can be done in two ways: unicast and broadcast.***

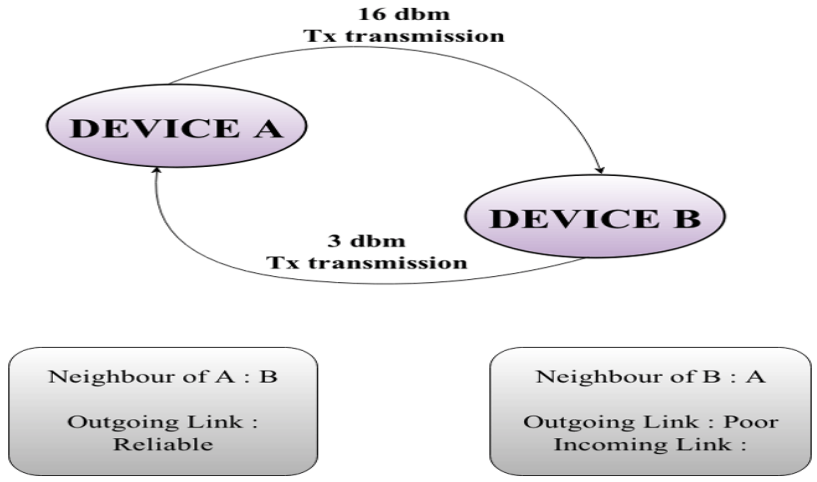
Broadcast Transmission:

In simple words Broadcast means the information/programme transmitted by radio or TV. In other words broadcast transmissions are sent to many or all devices in the network. Broadcast transmissions with the **ZigBee protocol** are propagated in the whole network such that all nodes receive the transmission. To accomplish this, the coordinator and all routers that receive a broadcast transmission will retransmit the packet three times.



Unicast Transmission:

Unicast transmissions in ZigBee route data from one source device to another destination device. The destination device could be an immediate neighbor of the source device, or it could have several hops in between the way. An example is shown below in the figure explaining mechanism for recognizing the reliability of the bi-directional link.



ZIGBEE-AODV

Routing under the AODV protocol is accomplished using tables in each node that store the next hop (intermediary node between source and destination nodes) for a destination node. If a next hop is not known, route discovery must take place in order to find a path. Since only a limited number of routes can be stored on a Router, route discovery will take place more often on a large network with communication between many different nodes.

Node	Destination Address	Next Hop Address
R3	Router 6	Coordinator
C	Router 6	Router 5
R5	Router 6	Router 6

When a source node must discover a route to a destination node, it sends a broadcast route request command. The route request command contains the source network address, the destination network address and a path cost field (a metric for measuring route quality). As the

route request command is propagated through the network (refer to the Broadcast Transmission), each node that re-broadcasts the message updates the path cost field and creates a temporary entry in its route discovery table.

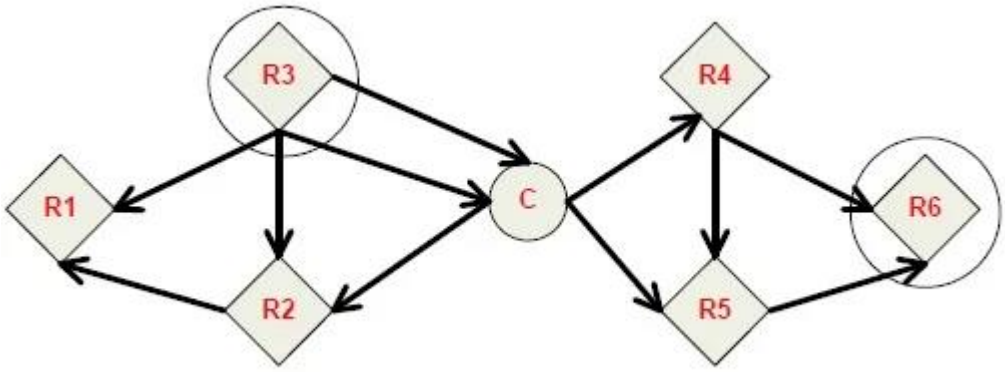


fig-1: Sample Route Request (Broadcast) Transmission Where R3 is Trying to Discover a Route to R6

When the destination node receives a route request, it compares the 'path cost' field against previously received route request commands. If the path cost stored in the route request is better than any previously received, the destination node will transmit a route reply packet to the node that originated the route request. Intermediate nodes receive and forward the route reply packet to the source node (the node that originated route request).

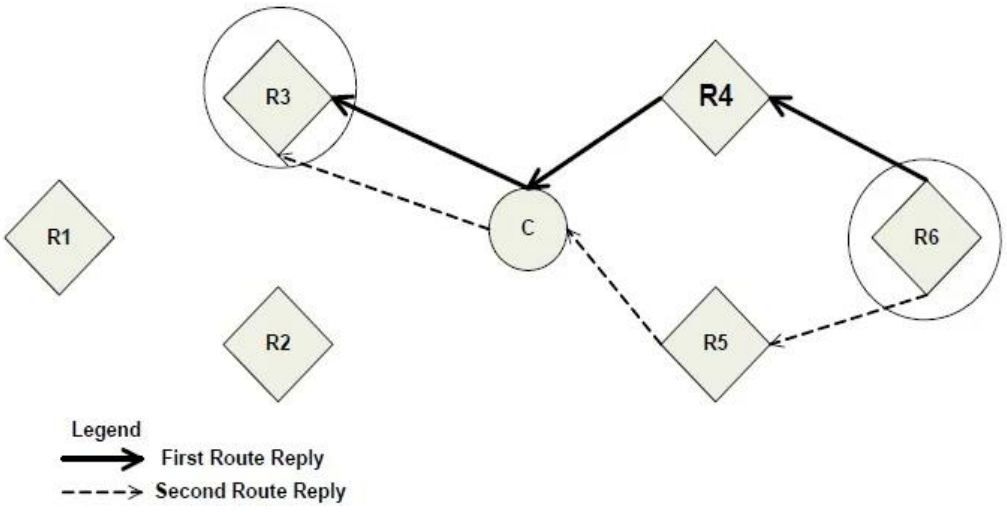
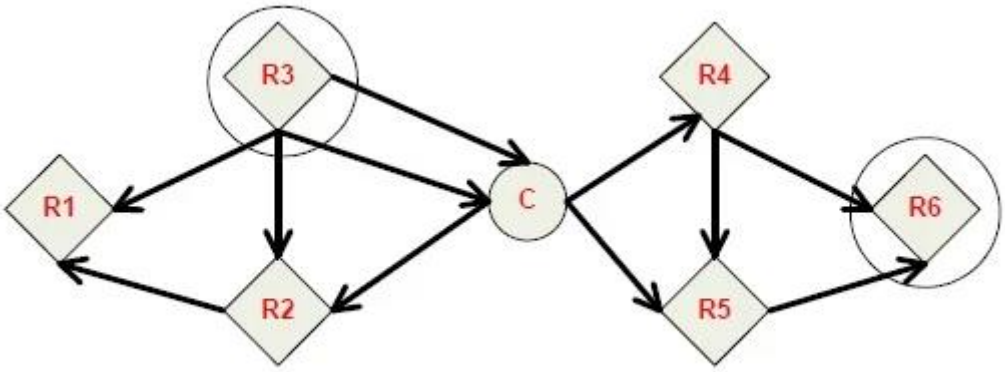


fig-2:

Retries and Acknowledgments: Here R6 can send multiple replies if it identifies a better route. ZigBee includes acknowledgment packets at both the Mac and Application Support (APS) layers. When data is transmitted to remote device, it may traverse multiple hops to reach the destination. As data is transmitted from one node to its neighbor, an acknowledgment packet (Ack) is transmitted in the opposite direction to indicate that the transmission was successfully received. If the Ack is not received, the transmitting device will retransmit the data, up to 4 times. This Ack is called the Mac layer acknowledgment.

fig.3:



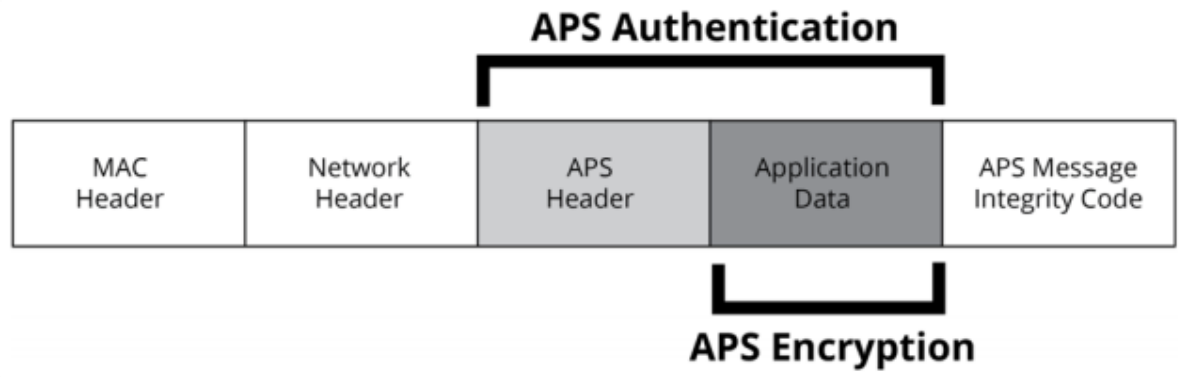
In addition, the device that originated the transmission expects to receive an acknowledgment packet (Ack) from the destination device. This Ack will traverse the same path that the data traversed, but in the opposite direction. If the originator fails to receive this Ack, it will retransmit the data, up to 2 times until an Ack is received. This Ack is called the ZigBee APS layer acknowledgment.

APS LAYER

APS layer security

APS layer security can be used to encrypt application data using a key that is shared between source and destination devices. Where network layer security is applied to all data transmissions and is decrypted and reencrypted on a hop-by-hop basis, APS security is optional and provides end-to-end security using an APS link key known by only the source and destination device. APS security cannot be applied to broadcast transmissions.

If you enable APS security, the APS header and data payload are authenticated with 128-bit AES as shown in the following image:



ZIGBEE APPLICATIONS

1. Home Automation
2. Medical Data Collection
3. Industrial Control Systems
4. meter reading system
5. light control system
6. Commercial
7. Government Markets Worldwide
8. Home Networking

1.4 GPS

Global Positioning System **(GPS)** is a navigation system based on satellite. It has created the revolution in navigation and position location. It is mainly used in positioning, navigation, monitoring and surveying applications.

The major **advantages** of satellite navigation are real time positioning and timing synchronization. That's why satellite navigation systems have become an integral part in most of the applications, where mobility is the key parameter.

A complete operational GPS space segment contains twenty-four satellites in MEO. These satellites are made into six groups so that each group contains four satellites. The group of four satellites is called as one **constellation**. Any two adjacent constellations are separated by 60 degrees in longitude.

The **orbital period** of each satellite is approximately equal to **twelve hours**. Hence, all satellites revolve around the earth two times on every day. At any time, the GPS receivers will get the signals from at least four satellites.

GPS Codes and Services

Each GPS satellite transmits two signals, **L_1 and L_2** are of different frequencies. **Trilateration** is a simple method for finding the position (Latitude, Longitude, Elevation) of GPS receiver. By using this method, the position of an unknown point can be measured from three known points

GPS Codes

Following are the two types of GPS codes.

- Coarse Acquisition code or C/A code
- Precise code or P code

The signal, L_1 is modulated with 1.023 Mbps pseudo random bit sequence. This code is called as Coarse Acquisition code or **C/A code** and it is used by the public.

The signal, L_2 is modulated with 10.23 Mbps pseudo random bit sequence. This code is called as Precise code or **P code** and it is used in military positioning systems. Generally, this P code is transmitted in an encrypted format and it is called as **Y code**

The P code gives better measurement accuracy when compared to C/A code, since the bit rate of P code is greater than the bit rate of C/A code.

GPS Services

Following are the two types of services provided by GPS.

- Precise Positioning Service (PPS)
- Standard Positioning Service (SPS)

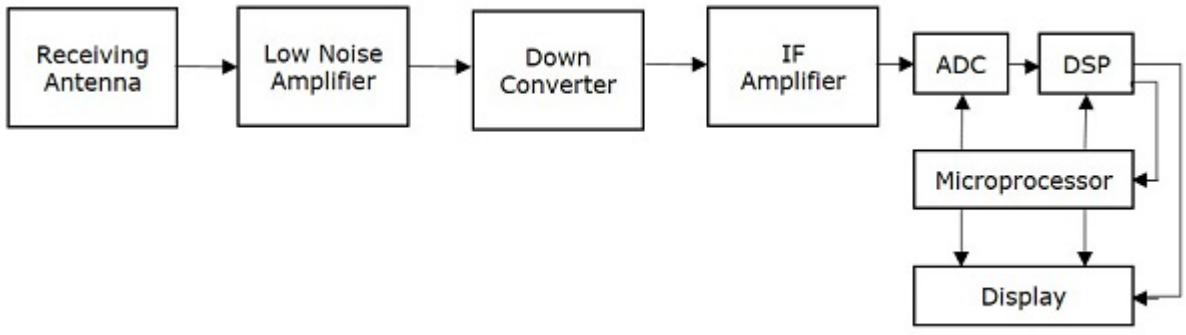
PPS receivers keep tracking of both C/A code and P code on two signals, L_1 and L_2. The Y code is decrypted at the receiver in order to obtain P code.

SPS receivers keep tracking of only C/A code on signal, L_1.

GPS Receiver

There exists only one-way transmission from satellite to users in GPS system. Hence, the individual user does not need the transmitter, but only a **GPS receiver**. It is mainly used to find the accurate location of an object. It performs this task by using the signals received from satellites.

The **block diagram** of GPS receiver is shown in below figure.



The function of each block present in GPS receiver is mentioned below.

- **Receiving Antenna** receives the satellite signals. It is mainly, a circularly polarized antenna.
- **Low Noise Amplifier** (LNA) amplifies the weak received signal
- **Down converter** converts the frequency of received signal to an Intermediate Frequency (IF) signal.
- **IF Amplifier** amplifies the Intermediate Frequency (IF) signal.
- **ADC** performs the conversion of analog signal, which is obtained from IF amplifier to digital. Assume, the sampling & quantization blocks are also present in ADC (Analog to Digital Converter).
- **DSP** (Digital Signal Processor) generates the C/A code.
- **Microprocessor** performs the calculation of position and provides the timing signals in order to control the operation of other digital blocks. It sends the useful information to Display unit in order to display it on the screen.

1.5 GSM MODULES

GSM (Global System for Mobile Communication) is used to transmit mobile data as well as voice services. This concept was implemented at Bell Laboratories using a mobile radio system in 1970. As the name suggests, it is the standardization group name that was established in the year1982 to make a general European mobile telephone standard. This technology owns above 70% of the market share of the digital cellular subscriber around the world. This technology was developed by using digital technology. At present, GSM technology supports above 1 billion mobile subscribers around the world in the above 210 countries. This technology provides voice and data services from fundamental to complex. This article discusses an overview of GSM technology.

What is GSM Technology?

GSM is a mobile communication modem; it stands for global system for mobile communication (GSM). The idea of GSM was developed at Bell Laboratories in 1970. It is a widely used mobile communication system in the world. GSM is an open and digital cellular technology used for transmitting mobile voice and data services operate at the 850MHz, 900MHz, 1800MHz, and 1900MHz frequency bands.

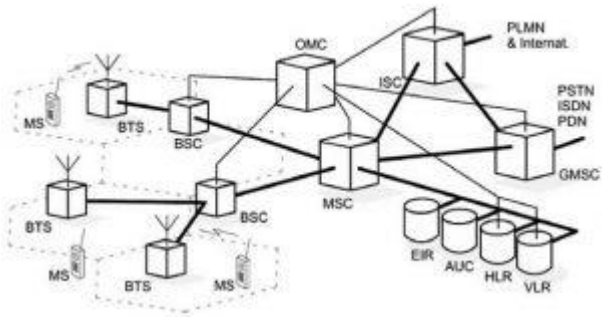
GSM technology was developed as a digital system using the time division multiple access (TDMA) technique for communication purposes. A GSM digitizes and reduces the data, then sends it down through a channel with two different streams of client data, each in its own particular time slot. The digital system has the ability to carry 64 kbps to 120 Mbps of data rates.

There are various cell sizes in a GSM system such as macro, micro, pico, and umbrella cells. Each cell varies as per the implementation domain. There are five different cell sizes in a GSM network macro, micro, pico, and umbrella cells. The coverage area of each cell varies according to the implementation environment.

The time division multiple access (TDMA) technique relies on assigning different time slots to each user on the same frequency. It can easily adapt to data transmission and voice communication and can carry 64kbps to 120Mbps of data rate.

GSM Technology Architecture

The main elements in the GSM architecture include the following.



- Network and Switching Subsystem (NSS)
- Base-Station Subsystem (BSS)
- The mobile station (MS)
- Operation and Support Subsystem (OSS)

Network Switching Subsystem (NSS)

In GSM system architecture, it includes different elements, which are frequently known as the core system/network. Here, it is basically a data network including a variety of units to provide the major control as well as interfacing of the entire mobile network system. The core network includes the major elements which are discussed below.

Mobile Switching Centre (MSC)

The Mobile Switching Centre or MSC is the key element in the core network region of the GSM network architecture. This mobile services switching center works like a standard switching node in an ISDN otherwise PSTN, however, it also gives extra functionality to allow the mobile user necessities to be supported like authentication, registration, inter-MSC handovers call location & routing of the call to a cell phone subscriber.

And, it also provides an edge toward the public switched telephone network so that the phone calls can be connected from the network of the mobile to a phone to a landline. Interfaces to other mobile switching center server are provided to allow mobile calls to be made to mobiles over dissimilar networks.

Home Location Register (HLR)

This HLR database includes the information regarding the administrative like every subscriber with their previous identified location. Like this, the GSM network is capable to connect the calls to the related base station for the mobile switch. Once an operator turns ON his/her phone, and then the phone registers through the network so that it is probable to decide which base transceiver station is communicating so that incoming calls can be connected properly.

Even once the mobile is switched on, but not active then it again registers to make sure that the HLR network is responsive to its most recent location. There is one HLR for each network, even though it may be dispersed across a variety of sub-centers for operational causes.

Visitor Location Register (VLR)

The VLR includes preferred information that is received from the HLR network to allow the preferred services for the separate subscriber. The visitor location register can be executed like a separate unit; however, it is usually realized like an essential element of the MSC, before an individual unit. Thus, access is finished quicker & more convenient.

Equipment Identity Register (EIR)

The EIR (Equipment Identity Register) is the unit that makes a decision whether specified mobile gear may be permitted over the network. Every mobile gear includes a number identified like the IMEI or International Mobile Equipment Identity.

So, this IMEI number is fixed within the mobile equipment & is verified through the network while registration. It mainly depends on the information that is held within the EIR, and the mobile device may be assigned one of 3 conditions which allowed over the network, barred access, otherwise watched in case its problems.

Authentication Centre (AuC)

The AuC (authentication center) is a protected file that includes the secret key in the SIM card of the user. The AuC is mainly used for verification & for coding on the radio channel.

Gateway Mobile Switching Centre (GMSC)

The GMSC/ Gateway Mobile Switching Centre is the end to which a ME finishing call is primarily connected without any information about the MS's place. The GMSC obtains the Mobile Station Roaming Number (MSRN) from the MSISDN based on HLR & connecting the call toward the exact visited MSC. The "MSC" division of the name GMSC is confusing as the gateway process does not need any linking toward an MSC.

SMS Gateway (SMS-G)

The SMS gateway or SMS-G is used jointly to explain two SMS-Gateways in the GSM standards. These gateways control messages which are directed in dissimilar ways.

The Short Message Service Gateway Mobile Switching Centre (SMS-GMSC) is used for short messages which are being transmitted to an ME. The Short Message Service Inter-Working Mobile Switching Centre (SMS-IWMSC) is used for short messages created through a mobile network. The main role of SMS-GMSC is related to GMSC, but the SMS-IWMSC offers a permanent access end to the SMS Centre.

These units were the major ones that are used in the network of GSM technology. They were normally co-located, however frequently the overall middle network was transmitted around

the country wherever the network was situated. In case of malfunction, it will give some flexibility.

Base Station Subsystem (BSS)

It acts as an interface between the mobile station and the network subsystem. It consists of the Base Transceiver Station which contains the radio transceivers and handles the protocols for communication with mobiles. It also consists of the Base Station Controller which controls the Base Transceiver station and acts as an interface between the mobile station and mobile switching center.

The network subsystem provides the basic network connection to the mobile stations. The basic part of the Network Subsystem is the Mobile Service Switching Centre which provides access to different networks like ISDN, PSTN, etc. It also consists of the Home Location Register and the Visitor Location Register which provides the call routing and roaming capabilities of GSM.

It also contains the Equipment Identity Register which maintains an account of all the mobile equipment wherein each mobile is identified by its own IMEI number. IMEI stands for International Mobile Equipment Identity.

The BSS or Base Station Subsystem section of the second generation GSM network architecture is basically connected with the mobiles over the network. This subsystem includes two elements which are discussed below.

Base Transceiver Station (BTS)

The BTS (Base Transceiver Station) which is utilized within a GSM network includes the radio Tx, Rx & their related antennas to transmit, receive & directly converse through the mobiles. This station is the important element for every cell and it converses with the mobiles & the interface among the two is identified like the Um interface with related protocols.

Base Station Controller (BSC)

The BSC (base station controller) is used to form the next phase reverse into the GSM technology. This controller is used to control a collection of base transceiver stations & it is frequently co-located through one of the transceiver stations within the group. This controller manages the resources of radio to control different items like handover in the collection of BTSs, assigns channels. It converses with the Base Transceiver Stations over Abis interface.

The subsystem element in the base station of the GSM network uses the radio allowable technology to allow a number of operators to right to use the system concurrently. Every channel supports up to 8 operators by allowing a base station to include different channels; a huge number of operators could be accommodated through every base station.

These are located carefully through the provider of the network to allow whole area coverage. This area can be enclosed with a base station that is often being called a cell. Because it is not achievable to stop the signals from overlapping into the nearby cells and channels which are used in single-cell are not utilized in the next.

Mobile Station

It is the mobile phone which consists of the transceiver, the display, and the processor and is controlled by a SIM card operating over the network.

The MS (Mobile stations) or ME (mobile equipment) are most generally identified through cell otherwise mobile phones which are the part of a GSM mobile communications n/w that the operator observes & operates. At present, their dimension has reduced radically whereas the functionality level has very much increased. And one more benefit is that the time among charges has drastically enlarged. There are different elements to the mobile phone, though the two essential elements are the hardware & the SIM.

The hardware includes the major elements of the mobile phone like the case, display, battery, & the electronics utilized to produce the signal & process the data receiver to be broadcasted. The mobile station includes a number called the IMEI. This can be set up on the mobile phone while manufacturing & it cannot be modified.

It is accessed by the network during registration to check whether the equipment has been reported as stolen.

The SIM (Subscriber Identity Module) card includes the data which gives the user identity toward the network. And also, it includes different information like a number called the IMSI (International Mobile Subscriber Identity). When this IMSI is used in the SIM card, the mobile user could simply change mobiles by moving the SIM from one mobile to another.

So mobile changing is easy without changing the same mobile number means that people would frequently improve, thus making a further income stream for the providers of network & serving to enhance the total financial victory of GSM.

Operation and Support Subsystem (OSS)

The operation support subsystem (OSS) is a part of the complete GSM network architecture. This is connected to the NSS & the BSC components. This OSS is mainly used to control the GSM network & the BSS traffic load. It should be noted down that when the number of BS enhances through the subscriber population scaling then some of the preservation tasks are moved to the base transceiver stations so that the ownership cost of the system can be reduced.

The GSM network architecture of 2G mainly follows a logical technique of operation. This is very simple as compared with present architectures of mobile phone network which utilize software-defined units to allow extremely supple operation. But the architecture of 2G GSM will demonstrate the voice & operational fundamental functions that are required & how they arranged together. When the GSM system is digital, then the network is a data network.

Features of GSM Module

The features of the GSM module include the following.

- Improved spectrum efficiency
- International roaming
- Compatibility with integrated services digital network (ISDN)
- Support for new services.
- SIM phonebook management
- Fixed dialing number (FDN)
- Real-time clock with alarm management
- High-quality speech
- Uses encryption to make phone calls more secure
- Short message service (SMS)

The security strategies standardized for the GSM system make it the most secure telecommunications standard currently accessible. Although the confidentiality of a call and secrecy of the GSM subscriber is just ensured on the radio channel, this is a major step in achieving end-to-end security.

GSM Modem

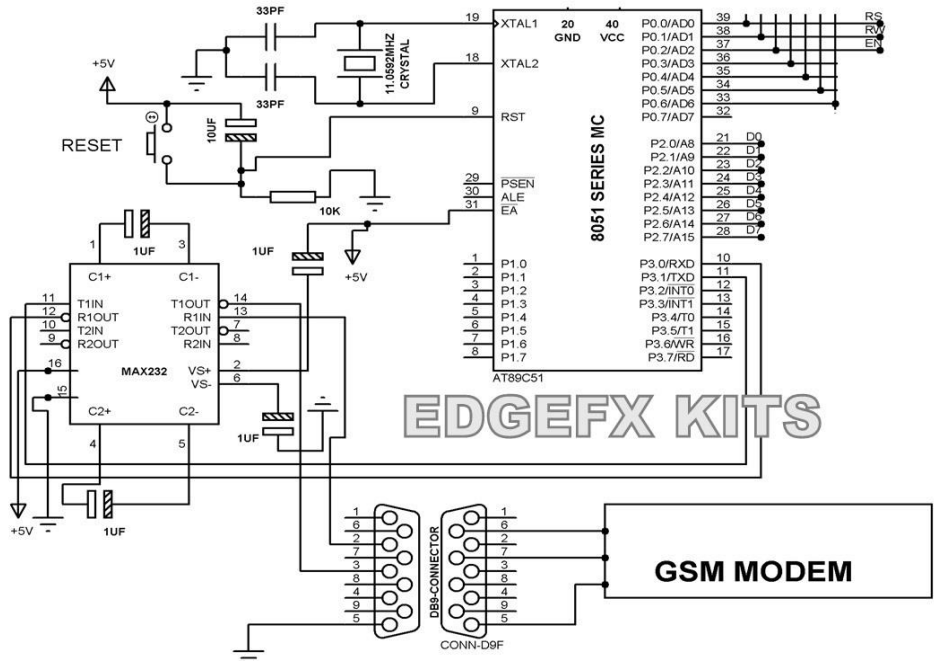
A GSM modem is a device that can be either a mobile phone or a modem device that can be used to make a computer or any other processor communicate over a network. A GSM modem requires a SIM card to be operated and operates over a network range subscribed by the network operator. It can be connected to a computer through serial, USB, or Bluetooth connection.

A GSM modem can also be a standard GSM mobile phone with the appropriate cable and software driver to connect to a serial port or USB port on your computer. GSM modem is usually preferable to a GSM mobile phone. The GSM modem has a wide range of applications in transaction terminals, supply chain management, security applications, weather stations, and GPRS mode remote data logging.

Working of GSM Module

From the below circuit, a GSM modem duly interfaced to the MC through the level shifter IC Max232. The SIM card mounted GSM modem upon receiving digit command by SMS from any cell phone sends that data to the MC through serial communication. While the program is executed, the GSM modem receives the command ‘STOP’ to develop an output at the MC, the contact point of which are used to disable the ignition switch.

The command so sent by the user is based on an intimation received by him through the GSM modem ‘ALERT’ a programmed message only if the input is driven low. The complete operation is displayed over a 16×2 LCD display.



GMS Modem Circuit

GSM Technology Applications

The applications of GSM technology include the following.

Intelligent GSM Technology for Automation and Security

These days, the GSM mobile terminal has become one of the items that are constantly with us. Just like our wallet/purse, keys or watch, the GSM mobile terminal provides us a communication channel that enables us to communicate with the world. The requirement for a person to be reachable or to call anyone at any time is very appealing.

This project, as the name says the project is based on GSM network technology for transmission of SMS from sender to receiver. SMS sending and receiving is used for ubiquitous access to appliances and allowing breach control at home. The system proposes two sub-systems. The appliance control subsystem enables the user to control home appliances remotely and the security alert subsystem gives automatic security monitoring.

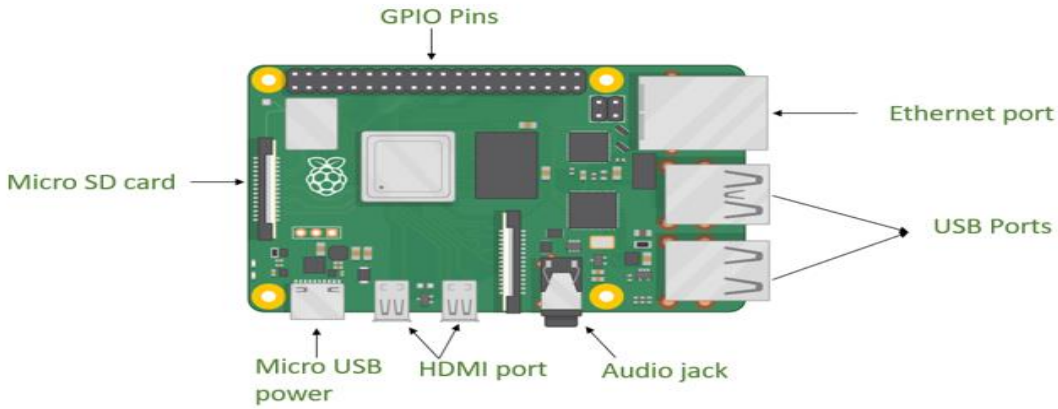
The system is capable enough to instruct users via SMS from a specific cell number to change the condition of the home appliance according to the user's needs and requirements. The second aspect is that of security alert which is achieved in a way that on the detection of intrusion, the system allows automatic generation of SMS thus alerting the user against security risk.

GSM technology will allow communication anywhere, anytime, and with anyone. The functional architecture of GSM employing intelligent networking principles, and its ideology, which provides the development of GSM is the first step towards a true personal communication system that enough standardization to ensure compatibility.

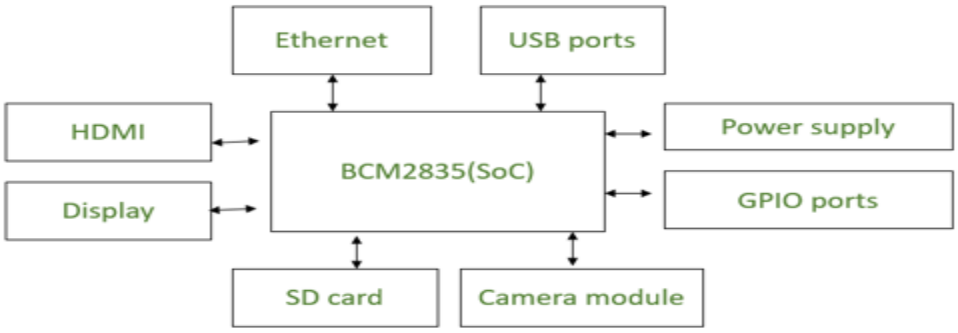
1.6 OPEN PLATFORM: RASBERRY PI ARCHITECTURE

Raspberry Pi is a small single-board computer (SBC). It is a credit card-sized computer that can be plugged into a monitor. It acts as a minicomputer by connecting the keyboard, mouse, and display. Raspberry Pi has an ARM processor and 512MB of RAM. The architecture of Raspberry Pi is discussed in this article.

The following diagram shows the architecture of Raspberry Pi:



The following diagram shows some main blocks of Raspberry Pi:



Raspberry Pi mainly consists of the following blocks:

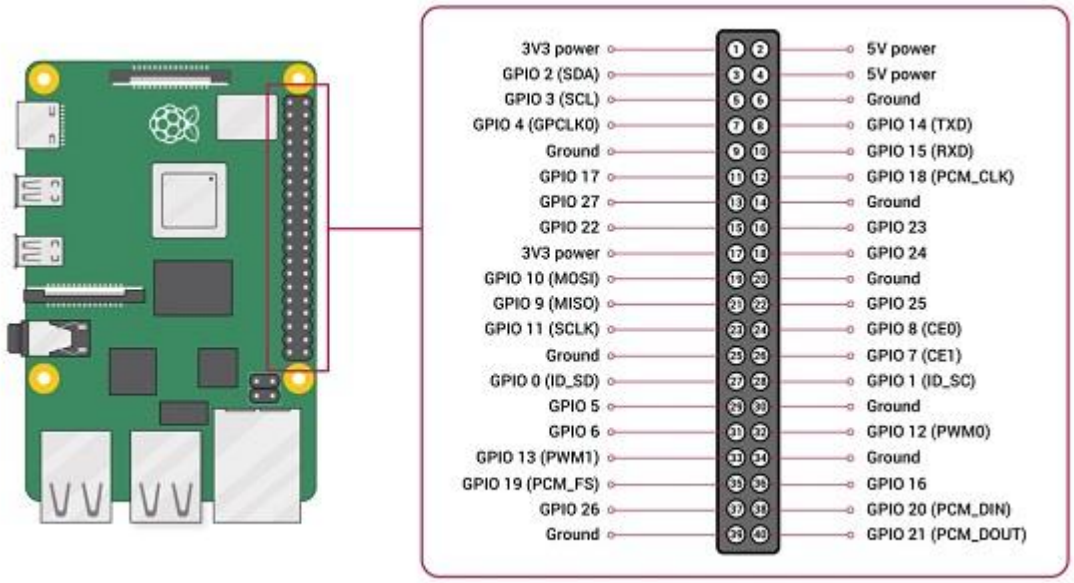
- **Processor:** Raspberry Pi uses Broadcom BCM2835 system on chip which is an ARM processor and Video core Graphics Processing Unit (GPU). It is the heart of the Raspberry Pi which controls the operations of all the connected devices and handles all the required computations.
- **HDMI:** High Definition Multimedia Interface is used for transmitting video or digital audio data to a computer monitor or to digital TV. This HDMI port helps Raspberry Pi to connect its signals to any digital device such as a monitor digital TV or display through an HDMI cable.
- **GPIO ports:** General Purpose Input Output ports are available on Raspberry Pi which allows the user to interface various I/P devices.

- **Audio output**: An audio connector is available for connecting audio output devices such as headphones and speakers.
- **USB ports:** This is a common port available for various peripherals such as a mouse, keyboard, or any other I/P device. With the help of a USB port, the system can be expanded by connecting more peripherals.
- **SD card: The** SD card slot is available on Raspberry Pi. An SD card with an operating system installed is required for booting the device.
- **Ethernet**: The ethernet connector allows access to the wired network, it is available only on the model B of Raspberry Pi.
- **Power supply:** A micro USB power connector is available onto which a 5V power supply can be connected.
- **Camera module:** Camera Serial Interface (CSI) connects the Broadcom processor to the Pi camera.
- **Display:** Display Serial Interface (DSI) is used for connecting LCD to Raspberry Pi using 15 15-pin ribbon cables. DSI provides a high-resolution display interface that is specifically used for sending video data.

GPIO Pinout

One of the powerful features of the Raspberry Pi is the row of GPIO (general-purpose input output) pins and the GPIO Pinout is an interactive reference to these GPIO pins.

Following diagram shows a 40-pin GPIO header, which is found on all the current Raspberry Pi boards −



The source of the diagram is www.raspberrypi.org

Voltages

From the above diagram, we can see that there are two 5V pins and two 3V3 pins on the board. It also has several ground pins (0V). All these pins are unconfigurable.

Outputs

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

Inputs

A GPIO pin can be designated as an input pin. The pin set as input pin can be read as 3V3(high) or 0V(low). You can use internal pull-up or pull-down resistors.

You can see in the above diagram, GPIO2 and GPIO3 pins have fixed pull-up resistors but for the other pins, you can configure it in software.

Alternative Functions

GPIO pins can be used with a variety of alternative functions. Among them, some are available on all pins and others on specific pins.

PWM: Pulse-width modulation

Software PWM are available on all the pins whereas Hardware PWM are available on GPIO12, GPIO13, GPIO18, and GPIO19.

SPI: Serial Peripheral Interface

The SPI are available on the following −

SPI0: MOSI (GPIO10); MISO (GPIO9); SCLK (GPIO11); CE0 (GPIO8), CE1 (GPIO7)
SPI1: MOSI (GPIO20); MISO (GPIO19); SCLK (GPIO21); CE0 (GPIO18); CE1 (GPIO17); CE2 (GPIO16)

I2C: Inter-integrated Circuit

The I2C are available on the following −

Data: (GPIO2); Clock (GPIO3)
EEPROM Data: (GPIO0); EEPROM Clock (GPIO1)

Serial

The serial function is available at the following −

TX(GPIO14)

RX(GPIO15)

Connect GPIO to Raspberry Pi

Following are some simple rules to reduce the risk of damaging your Raspberry Pi board, while using the GPIO connector −

- Do not try to put more than 3.3V on any GPIO pin.
- Do not try to draw more than 3mA per output. Although, you can draw more but to increase the life of your Pi Board, you should restrict upto 3mA.
- You should not poke the GPIO connector with a screwdriver when the Raspberry Pi board is powered up.
- 5V power is enough for your Raspberry Pi. Don't try to provide more power than that.
- You should not try to draw more than a total of 50mA from the 3.3V supply pins.

Output of GPIO pins

To set up the output of GPIO pins and to read the input values by using Python, you need to install **RPi.GPIO** python library.

Install RPi.GPIO python library

To install RPi.GPIO python library, type the following commands on terminal window of your Raspberry Pi −

sudo apt-get install python-dev

sudo apt-get install python-rpi.gpio

Almost all the latest versions of distributions have RPi.GPIO already installed. In that case, the above commands will update it to the latest version.

I2C Device

Let us check how we can make I2C work with Raspberry Pi.

Case 1: Using Adafruit Occidentalis 0.2 or later

In case, if you are using Adafruit Occidentalis, you don't need to do anything. Because, this distribution is preconfigured with I2C support.

Case 2: Using Raspbian

In case if you are using Raspbian, you need to do the following configuration changes −

First, edit the file **/etc/modules** by using the following command −

sudo nano /etc/modules

Now, we need to add the following lines to the end of this file −

i2c-bcm2708
i2c-dev

Next, we need to edit the file named **/etc/modprobe.d/raspi-blacklist.conf** and comment out the following line by adding a # −

blacklist i2c-bcm2708
#blacklist i2c-bcm2708

Once done, install the Python I2C library by using the following command −

sudo apt-get install python-smbus

Now, reboot your Raspberry Pi and it will be ready for I2C.

Find I2C Address

There is an I2C device attached to Raspberry Pi computer and you want to know its address.

For this, we need to install i2c-tools as follows −

sudo apt-get install i2c-tools

Once done, attach your I2C device to your Raspberry Pi board and run the following command −

sudo i2cdetect -y 1

Here, we need to take care about the following two things −

- First, if you have newer distributions then, it is quite possible that it has i2c-tools already installed.
- Second, if you have an older version 1 board in use, change 1 to 0 in the above code line.

1.7 RASBERRY PI INTERFACING

Serial

• The serial interface on Raspberry Pi has received (Rx) and transmit (Tx) pins for communication with serial peripherals.

SPI

- Serial Peripheral Interface (SPI) is a synchronous serial data protocol used for communicating with one or more peripheral devices.

- n the SPI connection, there is one master device and One and more peripheral devices.
- There are 5 pins on Raspberry Pi for SPI interface:-

MISO (**Master in and slave out**):- master line for sending data to the peripherals.
MOSI (**Master out slave in**):- slave line for sending data to the master.
SCK (**Serial clock**):- clock generated by Master to synchronize data transmission.
CE0 (**Chip enable 0**):- to enable or disable the device.
CE0 (**Chip enable 1**):- to enable or disable the device.

I2C

- The I2C interface pins on Raspberry Pi allow you to connect hardware modules.
- I2C interface allows synchronous data transfer with just two pins-SDA (data line) and SCL (clock line).

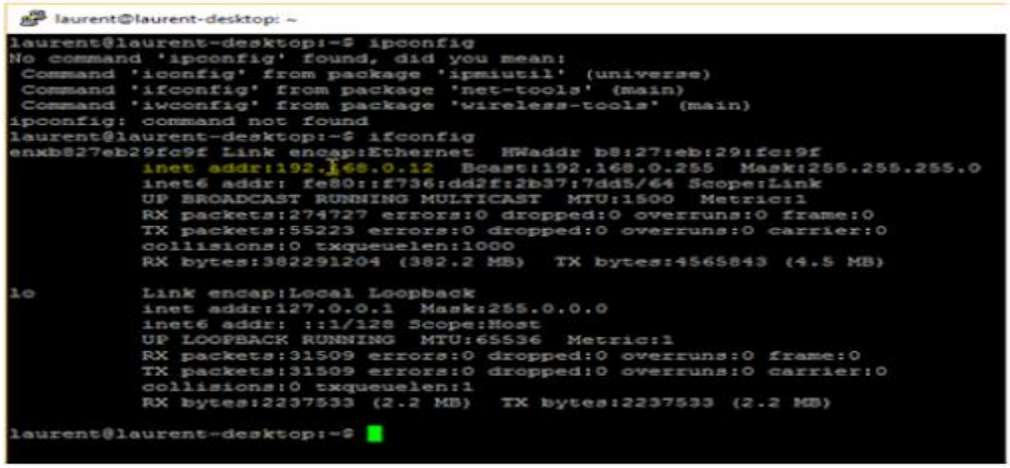
Rasbian Linux desktop



File explorer on Raspberry Pi



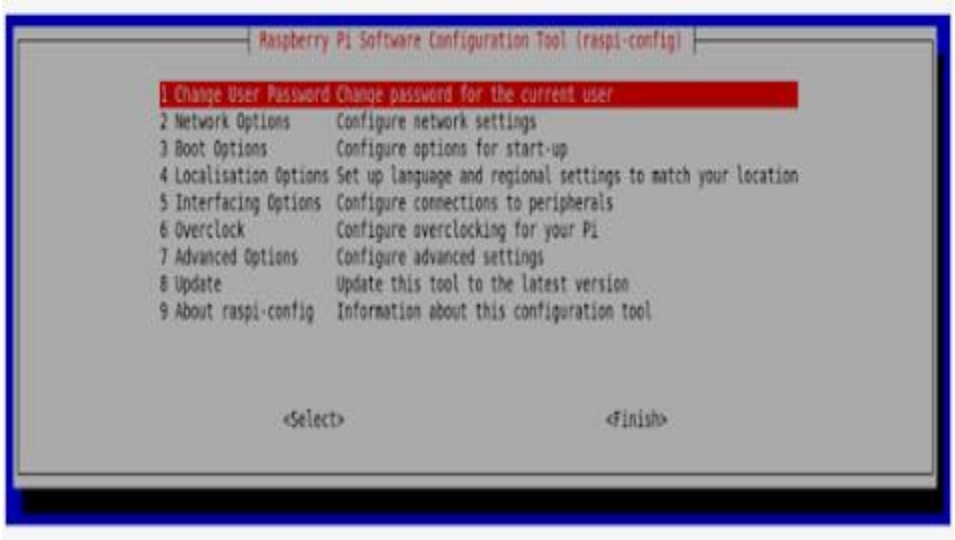
The console on Raspberry Pi



The browser on Raspberry Pi



Raspberry Pi configuration tools



Serial Peripheral Interface (SPI)

Let us check how we can use SPI (serial peripheral interface) bus with Raspberry Pi.

Case 1: Using Adafruit Occidentalis 0.2 or later

In case if you are using Adafruit Occidentalis, you don't need to do anything because this distribution is preconfigured with SPI support.

Case 2: Using Raspbian

In case if you are using Raspbian, you need to do the following configuration changes −

First, edit the file **/etc/modules** by using the following command −

```
sudo nano /etc/modules
```

Now, we need to add the following lines to the end of this file −

```
spidev
```

Next, we need to edit the file named **/etc/modprobe.d/raspi-blacklist.conf** and comment out the following line by adding a **#** −

```
blacklist spi-bcm2708
#blacklist spi-bcm2708
```

Once done, install the Python library to carry out communication from a Python program by using the following command −

```
cd ~
sudo apt-get install python-dev
git clone git://github.com/doceme/py-spidev
cd py-spidev/
sudo python setup.py install
```

Now, reboot your Raspberry Pi and it will be ready for SPI.

Serial Port

Suppose if you want to use the serial port i.e. Rx and Tx pins on your Raspberry Pi board but, it is used by Linux OS as a console connection. To disable this, we need to comment out a line in a file named **/etc/inittab**.

Open this file by using the following code line −

```
sudo nano /etc/inittab
```

Now, find the following line by scrolling down at the end of this file −

T0:23:respawn:/sbin/getty -L ttyAMA0 115200 vt100

Now, we need to use a # to comment out this line −

#T0:23:respawn:/sbin/getty -L ttyAMA0 115200 vt100

Now, save this file and reboot your Raspberry Pi.

Access serial port from Python

We can also use the serial port i.e. Rx and Tx pins on Raspberry Pi board by using Python. For this, we need to install **PySerial** library as follows −

sudo apt-get install python-serial

It is recommended to disable (as we did above) the Raspberry Pi's serial console before using **PySerial**.

Test the serial port

Once you start using the serial ports, you may want to send and receive the serial commands from a Terminal session. For this, we need to install **Minicom** as follows −

sudo apt-get install minicom

It is recommended to disable (as we did above) the Raspberry Pi's serial console before using **Minicom**.

As now Minicom is installed, we can start a serial communication session with a serial device connected to the RXD and TXD pins of the GPIO connector by using the following command −

minicom -b 9600 -o -D /dev/ttyAMA0

Here, in the above command, after -b is the baud rate and after -D is the serial port. We should use the same baud rate as set on the service we are communicating with.

1.8 RASBERRY PI PROGRAMMING

PIXEL Desktop Environment

PIXEL (Pi Improved Xwindows Environment, Lightweight) is a visual desktop environment which is a part of the recommended Raspbian Linux distribution. It is the quickest way to get started with Raspberry Pi and by default, it appears when our Raspberry Pi computer finishes starting up.

Some of the characteristics of PIXEL are as follows −

- It is based on LXDE (Lightweight X11 Desktop Environment) open-source desktop.
- Raspberry Pi foundation has redesigned LXDE and converted it into a PIXEL desktop environment.
- The PIXEL desktop environment works in a similar way to Mac OS and Windows OS.
- We can manage and find files by using mouse and icons.
- Using this desktop environment, it's really intuitive to navigate.

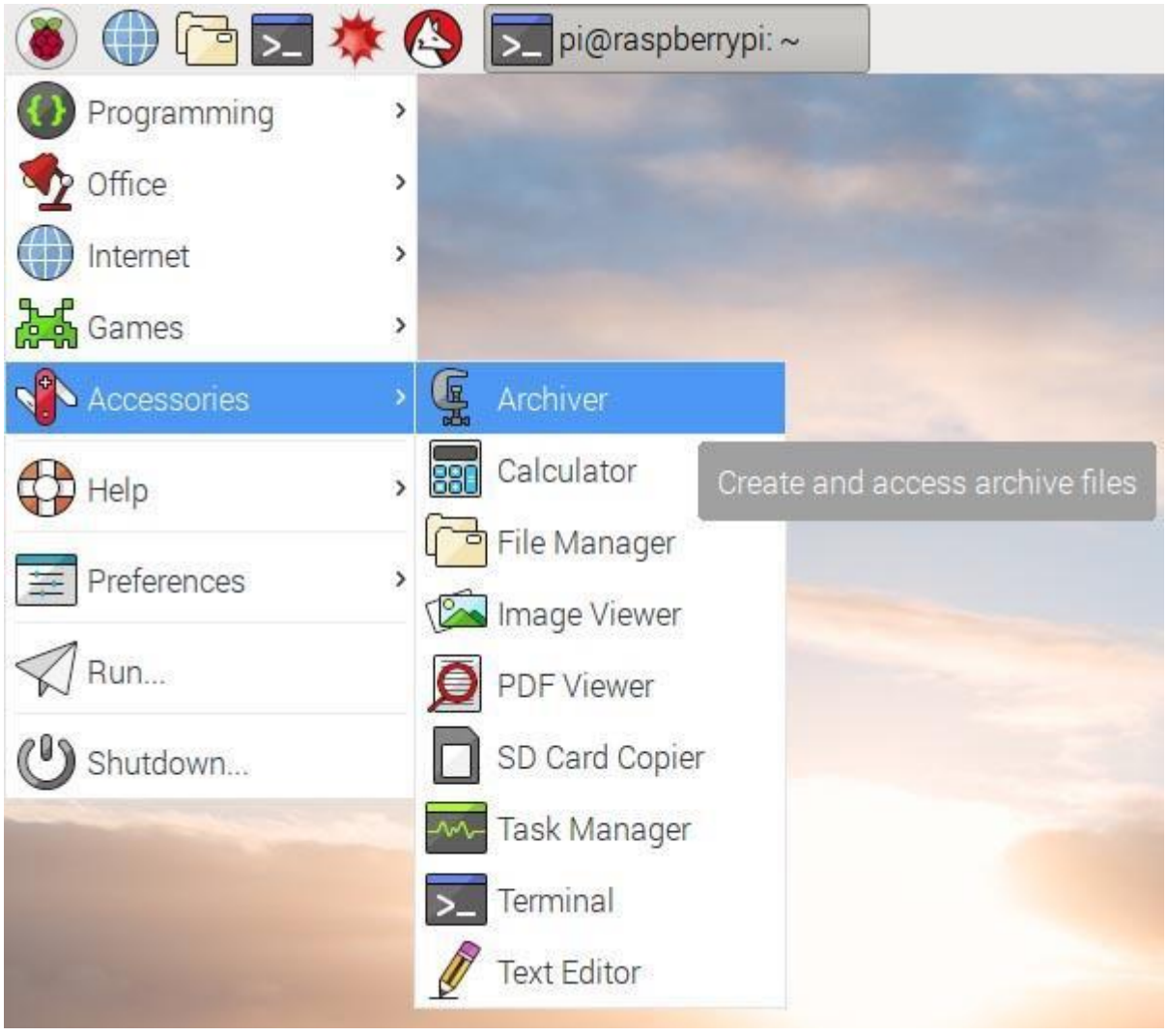
Navigate Desktop Environment

The image below is of the PIXEL desktop environment. You can see a taskbar (a strip along the top of the screen), which is usually visible in every program we will be using.

The Application Menu

For most of the programs, which we would like to run under PIXEL desktop or any other desktop environment, we need to use the application menu. You can get it by clicking the Raspberry Pi icon at the top left side of the desktop screen.

You will see the image as follows −



Submenu Programs

You will get the submenu program on the right, after moving the cursor over the categories of the programs. It will show the programs in that particular category.

You need to click on that category to start with that. If you want to add that category icon to the desktop, just right-click that program on the menu.

Following are the wealth of programs under submenu program −

Claws Mail

It is in the internet part of the Application with the help of which you can send or receive messages on your Raspberry Pi computer.

Debian Reference

As we have discussed earlier, the Raspbian version of Linux is the Pi-specific version of the Debian distribution. This icon will guide us how to use Linux on your Raspberry Pi computer.

This is a reference document, which is stored on your SD card and to find this, you need to go through the help section of the Application menu.

To get started with this, first, you need to click the icon and then, click the multi-files link (it is an HTML link) which is at the top of the screen.

LibreOffice

This is the most popular suite of productivity applications. It mainly includes word processing, spreadsheets, and presentations. You can get it from the office section of the Application menu.

Mathematica

Mathematica, under the programming section of the Application menu, is based on the Wolfram programming language. It is used for scientific and technical computing.

Minecraft Pi

We know about the world-building game called Minecraft. Similarly, Minecraft Pi is the Raspberry version of that. You can find it under the Game section of the Application program, and you can also program it by using the Python programming language.

Python 2 and Python 3

Raspberry Pi provides us the Python programming language, which can be found under Programming in the Application menu. We can also use the Thonny IDE (integrated development environment) which provides the Pi users, an alternative way of creating Python programs.

Python games

Raspberry Pi has games such as Reversi, Four in a Row, a sliding puzzle game as well as a snake game. These all are built in Python programming language and can be found in the Game section of the Application menu.

Scratch

Raspberry Pi foundation provides us a simple programming language called Scratch, which is approachable for the peoples of all ages. You can use it to create games and animations. It can also be used to manage electronic projects. You can find it under the Programming section of the Application menu.

Sense HAT emulator

As the name implies, it has some built-in sensors that can be used for creating experiments and other projects. It is an add-on for the Raspberry Pi users, which can be found under the Programming section of the Application menu.

Shutdown

Shutdown, a top-level option in the Application menu, can be used for switching off your Raspberry Pi, before you remove the power. With this, we will also get the options to log out as well as restart your Pi computer.

Sonic Pi

It is another programming language provided by Raspberry Pi foundation which is mainly used for creating music. You can also find it under the Programming section of the Application menu.

Terminal

Terminal is a window that let us issue the instructions from a command line without leaving your PIXEL desktop environment. There are two ways through which you can reach the terminal window. One is to get it in the Accessories part of the Programs menu and other is to use the button on the taskbar.

Wolfram

Wolfram is a programming language provided by Raspberry Pi foundation. It aims to incorporate knowledge, so that the programmers can get results quickly. You can get more information about this at www.wolfram.com/language. It is under the Programming section of the Application menu.

Running programs

Even after installing, some of the programs won't appear on the Application menu. You can use the **Run option** to run those programs.

Follow the below given steps −

Step 1 − First, we need to open the Application menu. For this, click the icon at top left of the desktop.

Step 2 − Now, we need to select the Run option from this menu.

Step 3 − Run option will give you a dialog box. You can type the name of the program, which you want to open, and then press Enter.

Close and Rearrange Programs

The controls for closing and rearranging the programs on the PIXEL desktop environment is like the ones in MS Windows. These controls enable us to close as well as resize (minimize and maximize) the programs.

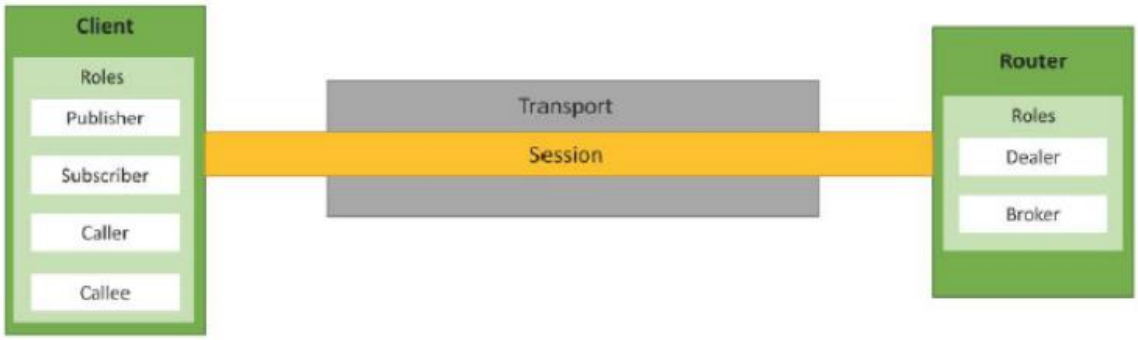
You can find these controls in the top right. They are explained below −

- **X button** − It is used for closing the programs/applications.
- **Maximize button** − As the name implies, this button will enlarge a particular application. Once used, the application will fill the screen.
- **Minimize button** − As the name implies, this button will reduce a particular application. It will hide the program from view but does not stop it from executing/running. We can return to the program by clicking the name of the program on the taskbar.

1.9 CONNECTING TO THE CLOUD

In truth, cloud computing and IoT are tightly coupled. The growth of IoT and the rapid development of associated technologies create a widespread connection of ―things. This has lead to the production of large amounts of data, which needs to be stored, processed and accessed. Cloud computing as a paradigm for big data storage and analytics. While IoT is exciting on its own, the real innovation will come from combining it with cloud computing. The combination of cloud computing and IoT will enable new monitoring services and powerful processing of sensory data streams. For example, sensory data can be uploaded and stored with cloud computing, later to be used intelligently for smart monitoring and actuation with other smart devices. Ultimately, the goal is to be able to transform data to insight and drive productive, cost-effective action from those insights. The cloud effectively serves as the brain to improved decision-making and optimized internet-based interactions. However, when IoT meets cloud, new challenges arise. There is an urgent need for novel network architectures that seamlessly integrate them. The critical concerns during integration are quality of service (QoS) and quality of experience (QoE), as well as data security, privacy and reliability. The virtual infrastructure for practical mobile computing and interfacing includes integrating applications, storage devices, monitoring devices, visualization platforms, analytics tools and client delivery. Cloud computing offers a practical utility-based model that will enable businesses and users to access applications on demand anytime and from anywhere.

WAMP – Auto Bahn for IoT Web Application Messaging Protocol (WAMP) is a sub-protocol of Web socket which provides publish-subscribe and remote procedure call (RPC) messaging patterns.



Transport: Transport is channel that connects two peers.

• Session: Session is a conversation between two peers that runs over a transport.

• Client: Clients are peers that can have one or more roles. In publish-subscribe model client can have following roles: – Publisher: Publisher publishes events (including payload) to the topic maintained by the Broker. – Subscriber: Subscriber subscribes to the topics and receives the events including the payload. In RPC model client can have following roles: – Caller: Caller issues calls to the remote procedures along with call arguments. – Callee: Callee executes the procedures to which the calls are issued by the caller and returns the results back to the caller.

• Router: Routers are peers that perform generic call and event routing. In publish-subscribe model Router has the role of a Broker: – Broker: Broker acts as a router and routes messages published to a topic to all subscribers subscribed to the topic. In RPC model Router has the role of a Broker:

– Dealer: Dealer acts a router and routes RPC calls from the Caller to the Callee and routes results from Callee to Caller.

• Application Code: Application code runs on the Clients (Publisher, Subscriber, Callee or Caller). In RPC model client can have following roles: – 1. Caller: Caller issues calls to the remote procedures along with call arguments. – Callee: Callee executes the procedures to which the calls are issued by the caller and returns the results back to the caller.

• Router: Routers are peers that perform generic call and event routing. In publish-subscribe model Router has the role of a Broker: – Broker: Broker acts as a router and routes messages published to a topic to all subscribers subscribed to the topic. In RPC model Router has the role of a Broker: – 1. Dealer: Dealer acts a router and routes RPC calls from the Caller to the Callee and routes results from Callee to Caller. 2. Application Code: Application code runs on the Clients (Publisher, Subscriber, Callee or Caller).

Xively Cloud for IoT Use of Cloud IoT cloud-based service

•The service provides for the data collection, data points, messages and calculation objects.

• The service also provisions for the generation and communication of alerts, triggers and feeds to the user.

• A user is an application or service. The user obtains responses or feeds from the cloud service. Pachube platform: for data capture in real-time over the Internet

• Cosm: a changed domain name, where using a concept of console, one can monitor the feeds

• Xively is the latest domain name. A commercial PaaS for the IoT/M2M
• A data aggregator and data mining website often integrated into the Web of Things
• An IoT PaaS for services and business services. Xively PaaS services:
• Data visualisation for data of connected sensors to IoT devices.
• Graphical plots of collected data. • Generates alerts. • Access to historical data • Generates feeds which can be real-world objects of own or others Xively HTTP based APIs
• Easy to implement on device hardware acting as clients to Xively web services • APIs connect to the web service and send data.
• APIs provides services for logging, sharing and displaying sensor data of all Xively Support •The platform supports the REST, WebSocket's and MQTT protocols and connects the devices to Xively Cloud Services • Native SDKs for Android, Arduino, ARM mbed, Java, PHP, Ruby, and Python languages • Developers can use the workflow of prototyping, deployment and management through the tools provided at Xively APIs
• Enable interface with Python, HTML5, HTML5 server, tornado
• Interface with WebSocket Server and WebSocket's
• Interface with an RPC (Remote Procedure Call). Xively PaaS services
• Enables services
• Business services platform which connects the products, including collaboration products
• Rescue, Bold chat, join.me, and operations to Internet
• Data collection in real-time over Internet Xively Methods for IoT Devices Data
•Concept of users, feeds, data streams, data points and triggers • Data feed typically a single location (e.g. a device or devices network),
• Data streams are of individual sensors associated with that location (for example, ambient lights, temperatures, power consumption).
• Pull or Push (Automatic or Manual Feed) Xively Data formats and Structures
• Number of data formats and structures enable the interaction, data collection and services
• Support exists for JSON , XML and CSV
• Structures: Tabular, spreadsheet, Excel, Data numbers and Text with a comma-separated values in file Xively Uses in IoT/M2M • Private and Public Data Access
• Data streams, Data points and Triggers
• Creating and Managing Feeds • Visualising Data

Django Django is an open source web application framework for developing web applications in Python.

• A web application framework in general is a collection of solutions, packages and best practices that allows development of web applications and dynamic websites.
• Django is based on the Model-Template-View architecture and provides a separation of the data model from the business rules and the user interface.
• Django provides a unified API to a database backend.
• Thus web applications built with Django can work with different databases without requiring any code changes.
• With this flexibility in web application design combined with the powerful capabilities of the Python language and the Python ecosystem, Django is best suited for cloud applications.
• Django consists of an object-relational mapper, a web templating system and a regular-expression based URL dispatcher. Django is Model-Template-View (MTV) framework. Model

• The model acts as a definition of some stored data and handles the interactions with the database. In a web application, the data can be stored in a relational database, non-relational database, an XML file, etc. A Django model is a Python class that outlines the variables and methods for a particular type of data. Template

• In a typical Django web application, the template is simply an HTML page with a few extra placeholders. Django's template language can be used to create various forms of text files (XML, email, CSS, JavaScript, CSV, etc.) View

• The view ties the model to the template. The view is where you write the code that actually generates the web pages. View determines what data is to be displayed, retrieves the data from the database and passes the data to the template.