1 Introduction to Internet of Things

Syllabus

Application areas of IoT, Characteristics of IoT, Things in IoT, IoT stack, Enabling technologies, IoT challenges, IoT levels, IoT and cyber physical system, IoT and WSN.

Contents

1.1 Evolution of Internet of Things

GTU : Summer-17, 18

- The Internet of Things (IoT) refers to the capability of everyday devices to connect to other devices and people through the existing Internet infrastructure. Devices connect and communicate in many ways.
- Examples of this are smartphones that interact with other smartphones, vehicle-to-vehicle communication, connected video cameras, and connected medical devices.
- They are able to communicate with consumers, collect and transmit data to companies, and compile large amounts of data for third parties.
- Things are objects of the physical world (physical things) or of the information world (virtual world) which are capable of being identified and integrated into communication networks. Things have associated information, which can be static and dynamic.
- Physical things exist in the physical world and are capable of being sensed, actuated and connected. Examples of physical things include the surrounding environment, industrial robots, goods and electrical equipment.
- Virtual things exist in the information world and are capable of being stored, processed and accessed. Examples of virtual things include multimedia content and application software.
- A device is a piece of equipment with the mandatory capabilities of communication and optional capabilities of sensing, actuation, data capture, data storage and data processing.
- The devices collect various kinds of information and provide it to the information and communication networks for further processing. Some devices also execute operations based on information received from the information and communication networks.
- Fig. 1.1.1 shows evolutionary phase of internet.

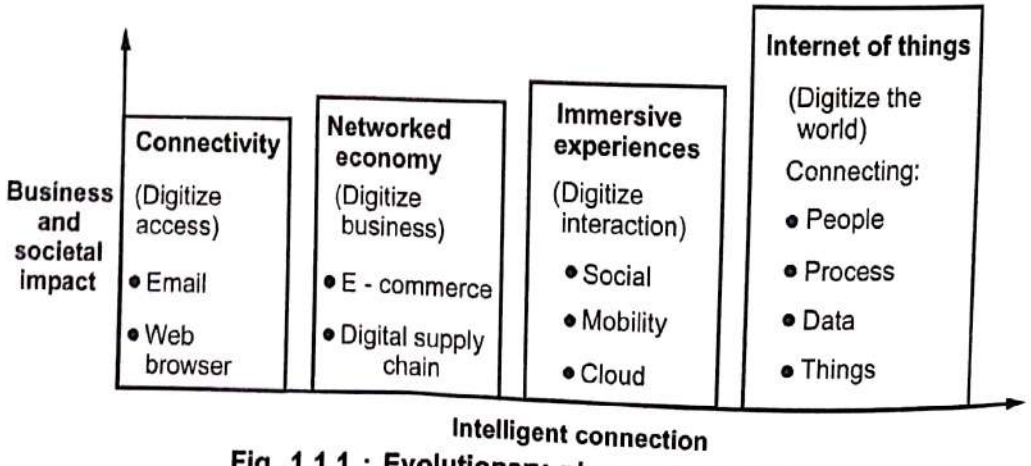


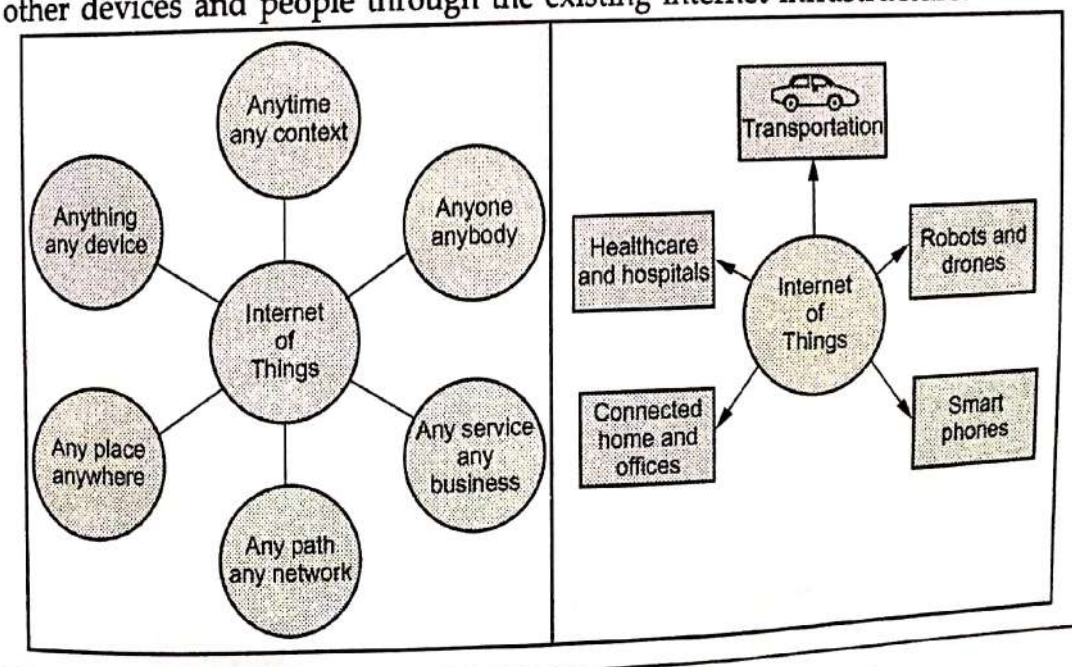
Fig. 1.1.1 : Evolutionary phase of internet

- Evolutionary phase of internet is Connectivity, Networked Economy, Immersive Experiences and IoT.

1. Connectivity: in the phase, peoples are connected to email, web services and searches the information.
2. Networked Economy : this phase support e-commerce and supply chain enhancements along with collaborative engagement to drive increased efficiency in business processes.
3. Immersive Experiences: this phase extended the Internet experience to encompass widespread video and social media while always being connected through mobility.
4. Internet of Things: it adds connectivity to objects and machines in the world around us to enable new services and experiences.

1.1.1 Definition of IoT

- The Internet of Things (IoT) is the network of physical objects i.e. devices, vehicles, buildings and other items embedded with electronics, software, sensors, and network connectivity that enables these objects to collect and exchange data.
- Wikipedia definition : The Internet of Things, also called the Internet of Objects, refers to a wireless network between objects, usually the network will be wireless and self-configuring, such as household appliances.
- WSIS 2005 Definition : By embedding short-range mobile transceivers into a wide array of additional gadgets and everyday items, enabling new forms of communication between people and things, and between things.
- The Internet of Things refers to the capability of everyday devices to connect to other devices and people through the existing internet infrastructure.



- Devices connect and communicate in many ways. Examples of this are smart phones that interact with other smart phones, vehicle-to-vehicle communication, connected video cameras, and connected medical devices. They are able to communicate with consumers, collect and transmit data to companies, and compile large amounts of data for third parties.
- IoT data differs from traditional computing. The data can be small in size and frequent in transmission. The number of devices, or nodes, that are connecting to the network are also greater in IoT than in traditional PC computing.
- Machine-to-Machine communications and intelligence drawn from the devices and the network will allow businesses to automate certain basic tasks without depending on central or cloud based applications and services.
- IoT impacts every business. Mobile and the Internet of Things will change the types of devices that connect into a company's systems. These newly connected devices will produce new types of data.
- The Internet of Things will help a business gain efficiency, harness intelligence from a wide range of equipment, improve operations and increase customer satisfaction.
- Ubiquitous computing, pervasive computing, Internet Protocol, sensing technologies, communication technologies, and embedded devices are merged together in order to form a system where the real and digital worlds meet and are continuously in symbiotic interaction.
- The smart object is the building block of the IoT vision. By putting intelligence into everyday objects, they are turned into smart objects able not only to collect information from the environment and interact /control the physical world, but also to be interconnected, to each other, through internet to exchange data and information.
- The expected huge number of interconnected devices and the significant amount of available data open new opportunities to create services that will bring tangible benefits to the society, environment, economy and individual citizens.
- However, the IoT is still maturing, in particular due to a number of factors, which limit the full exploitation of the IoT. Some of the factors are listed below :

1. There is no unique identification number system for object in the world.
2. IoT uses Architecture Reference Model (ARM) but there is no further development in ARM.
3. Missing large-scale testing and learning environments

4. Difficulties in exchanging of sensor information in heterogeneous environments.
5. Difficulties in developing business which embraces the full support of the Internet of Things.

1.1.2 IoT Characteristics

1. Interconnectivity : Everything can be connected to the global information and communication infrastructure.
2. Heterogeneity : Devices within IoT have different hardware and use different networks but they can still interact with other devices through different networks.
3. Things-related services : Provides things-related services within the constraints of things, such as privacy and semantic consistency between physical and virtual thing.
4. Dynamic changes: The state of a device can change dynamically, thus the number of devices can vary.
5. Integrated into information network: IoT devices are integrated with information network for communication purpose. It will exchange data with other devices.
6. Self-adapting: Self-adaptive is a system that can automatically modify itself in the face of a changing context, to best answer a set of requirements.
7. Self-configuration primarily consists of the actions of neighbour and service discovery, network organization, and resource provisioning.

1.1.3 Component of IoT

- The hardware utilized in IoT systems includes devices for a remote dashboard, devices for control, servers, a routing or bridge device, and sensors. These devices manage key tasks and functions such as system activation, action specifications, security, communication, and detection to support-specific goals and actions.
- Major components of IoT devices are as follows :

1. **Control units** : A small computer on a single integrated circuit containing processor core, memory and a programmable I/O peripheral. It is responsible for the main operation.
2. **Sensor :** Devices that can measure a physical quantity and convert it into a signal, which can be read and interpreted by the microcontroller unit. These devices consist of energy modules, power management modules, RF modules, and sensing modules. Most sensors fall into 2 categories : digital or analog. An analog data is converted to digital value that can be transmitted to the internet.

a. Temperature sensors : accelerometers

b. Image sensors : gyroscopes

c. Light sensors : acoustic sensors

d. Micro flow sensors : humidity sensors

e. Gas RFID sensors : pressure sensors

3. **Communication modules :** These are the part of devices and responsible for communication with rest of IoT platform. They provide connectivity according to wireless or wired communication protocol they are designed. The communication between IoT devices and the internet is performed in two ways:

a) There is an internet-enable intermediate node acting as a gateway;

b) The IoT device has direct communication with the internet.

- The communication between the main control unit and the communication module uses serial protocol in most cases.

4. **Power Sources :** In small devices the current is usually produced by sources like batteries, thermocouples and solar cells. Mobile devices are mostly powered by lightweight batteries that can be recharged for longer life duration.

- **Communication Technology and Protocol :** IoT primarily exploits standard protocols and networking technologies. However, the major enabling technologies and protocols of IoT are RFID, NFC, low-energy Bluetooth, low-energy wireless, low-energy radio protocols, LTE-A, and WiFi-Direct. These technologies support the specific networking functionality needed in an IoT system in contrast to a standard uniform network of common systems.

1.1.4 Working of IoT

- Fig 1.1.2 shows working of IoT.

1. **Collect and transmit data :** The device can sense the environment and collect information related to it and transmit it to a different device or to the internet.
2. **Actuate device based on triggers :** It can be programmed to actuate other devices based on conditions set by user.
3. **Receive information :** Device can also receive information from the network.
4. **Communication assistance :** It provides communication between two devices of same network or different network.

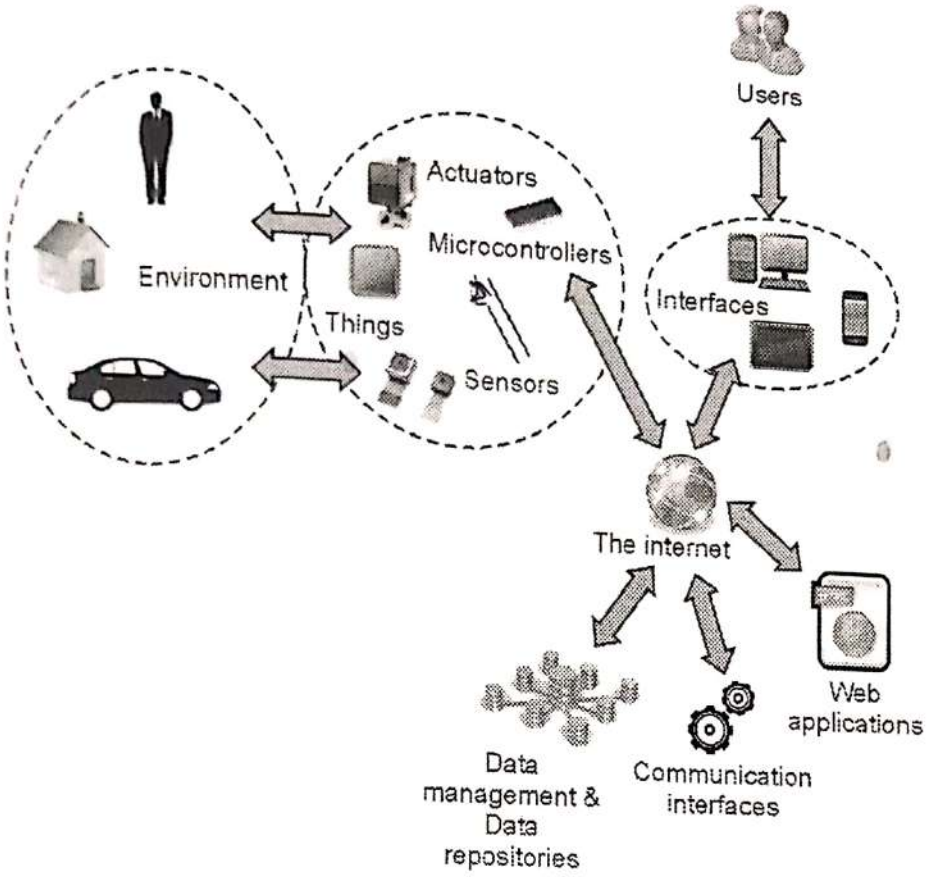


Fig. 1.1.2 : Working of IoT

- Sensors for various applications are used in different IoT devices as per different applications such as temperature, power, humidity, proximity, force etc.
- Gateway takes care of various wireless standard interfaces and hence one gateway can handle multiple technologies and multiple sensors.
- The typical wireless technologies used widely are 6LoWPAN, zigbee, Zwave, RFID, NFC etc. Gateway interfaces with cloud using backbone wireless or wired technologies such as WiFi, mobile , DSL or fibre.

1.1.5 Advantages and Disadvantages

Advantages of IoT

1. Improved customer engagement and communication.
2. Support for technology optimization
3. Support wide range of data collection
4. Reduced waste

Disadvantages of IoT

1. **Loss of privacy and security :** As all the household appliances, industrial machinery, public sector services like water supply and transport, and many other devices all are connected to the internet, a lot of information is available on it. This information is prone to attack by hackers.
2. **Flexibility :** Many are concerned about the flexibility of an IoT system to integrate easily with another
3. **Complexity :** The IoT is a diverse and complex network. Any failure or bugs in the software or hardware will have serious consequences. Even power failure can cause a lot of inconvenience.
4. **Compatibility :** Currently, there is no international standard of compatibility for the tagging and monitoring equipment
5. Save time and money.

1.1.6 Applications of IoT

1. **Home :** Buildings where people live. It controls home and security systems.
2. **Offices :** Energy management and security in office buildings; improved productivity, including for mobile employees.
3. **Factories :** Places with repetitive work routines, including hospitals and farms operating efficiencies, optimizing equipment use and inventory.
4. **Vehicles :** Vehicles including cars, trucks, ships, aircraft, and trains condition-based maintenance, usage-based design, pre-sales analytics
5. **Cities :** Public spaces and infrastructure in urban settings; adaptive traffic control, smart meters, environmental monitoring, resource management.
6. **Worksites :** It is custom production environments like mining, oil and gas construction; operating efficiencies, predictive maintenance, health and safety.

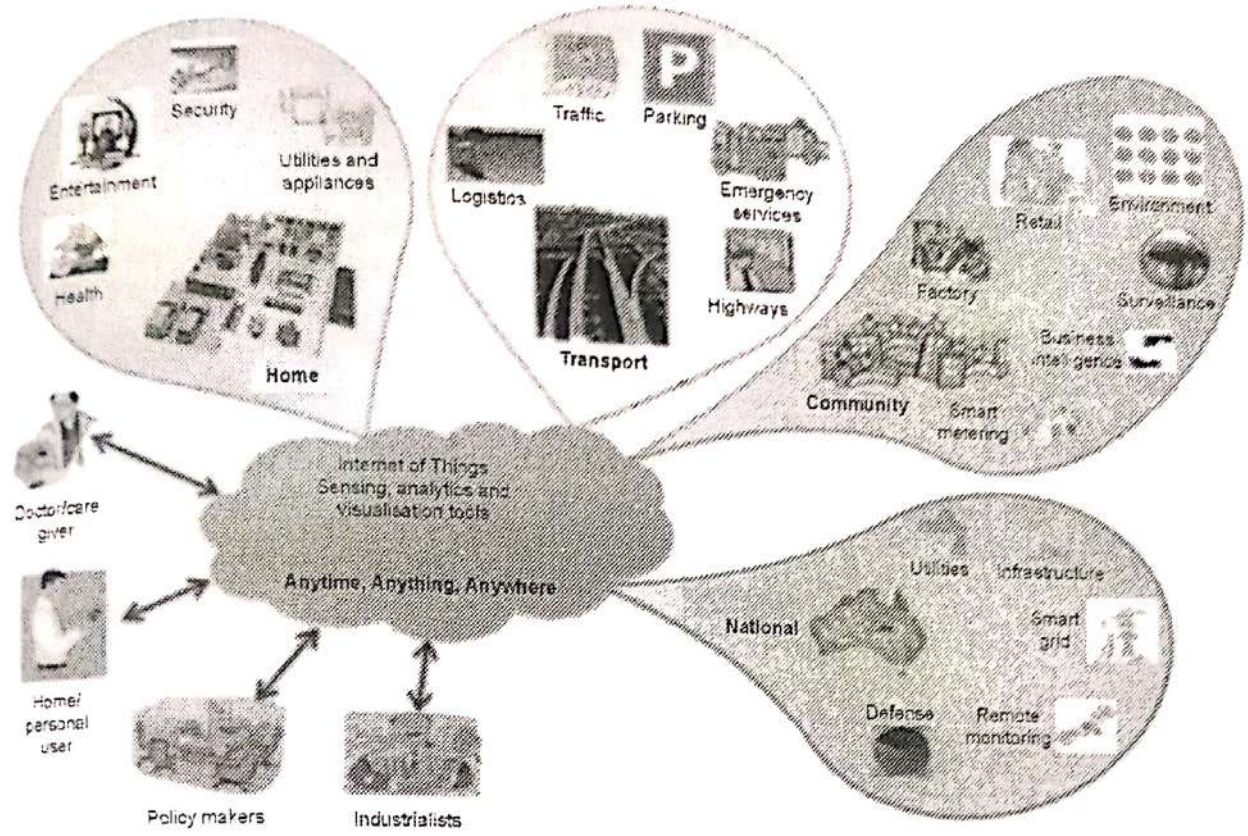


Fig. 1.1.3 IoT applications

1.2 Things in IoT

- IoT devices have unique identity and they are refer as "things" in IoT. Device can perform remote sensing, actuating and monitoring. IoT devices can exchange data between them and process data or send to centralized location for processing and storage.
- The Internet of Things refers to the set of devices and systems that interconnect real-world sensors and actuators to the Internet. This includes many different types of systems, such as:
  1. Mobile devices
  2. Smart meters and objects
  3. Wearable devices including clothing, health care implants, smart watches, and fitness devices
  4. Internet-connected automobiles
  5. Home automation systems, including thermostats, lighting, and home security
  6. Other measuring sensors for weather, traffic, ocean tides, road signals, and more
  7. Security camera

- Here "Things" is combination of software, hardware and services. Fig. 1.2.1 shows IoT devices.

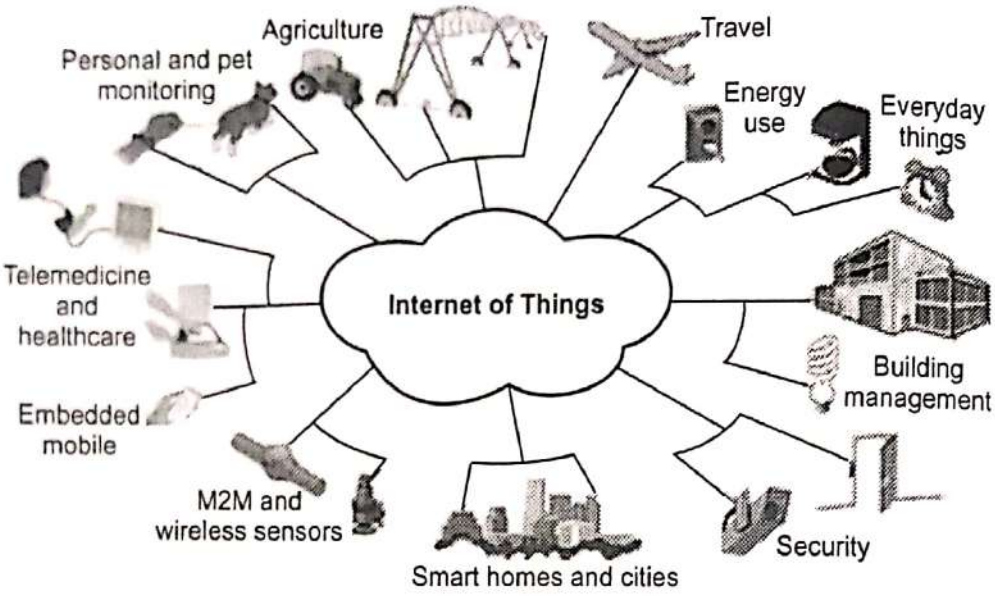


Fig. 1.2.1 : IoT devices

- IoT devices provide interface to various wire and wireless devices. Interface includes memory interface, I/O interface for sensors, Internet connectivity interface, storage interface etc.
- Using sensors, IoT collects various information like temperature, light intensity, humidity, air pressure. Some application used cloud-based storage. Collected information is stored in cloud and transmitted to other devices.
- Various types of IoT devices are smart clothing, smart watch, wearable sensors, LED lights, automobile industry etc.

1.3 IoT Stack

- Fig. 1.3.1 shows IoT stack.

Layer 7 : Application Layer
Layer 6 : User experience layer
Layer 5 : Session or message layer
Layer 4 : RF layer
Layer 3 : Hardware interface layer
Layer 2 : Processing and control action layer
Layer 1 : Physical or sensor layer

Fig. 1.3.1 : IoT Stack

- IoT stack contains seven layers:
  1. Layer 1 : Physical or sensor layer
  2. Layer 2 : Processing and control action layer
  3. Layer 3 : Hardware interface layer
  4. Layer 4 : RF layer
  5. Layer 5 : Session or message layer
  6. Layer 6 : User experience layer
  7. Layer 7 : Application Layer

Sr. No.	Layer	Functions and Applications
1.	Physical or sensor	• Sensor is main component of this layer. • Various sensor like temp, heat, humidity, pressure etc.
2.	Processing and control action	• Microcontroller or processor are used in this layer. • Microcontroller receive data from sensor. • Operating system paly important role in this layer.
3.	Hardware interface	• Components for hardware and serial communication. • RS232, CAN, SPI etc are used.
4.	RF	• Protocol used for communication and transport data using short and long rage communication.
5.	Session or message	• It uses messaging protocol for broadcasting messages. • Protocol : MQTT, FTP, HTTP, CoAP and SSH
6.	User experience	• It is related to user interface after product is designed. • It uses object and procedure-oriented technologies. • It uses various database and SQL
7.	Application	• Simple application • Smart city application • Smart Home • Smart Agriculture

1.4 Enabling Technologies

- IoT is enabled by several technologies including wireless sensor networks, cloud computing, Big data analytics, Embedded Systems, Security Protocols and architectures, communication protocols, web services, Mobile Internet, and Semantic Search engines.

1.4.1 Sensor

- Sensor converts a physical quantity into a corresponding voltage. Sensor is a device that when exposed to a physical phenomenon (temperature, displacement, force, etc.) produces a proportional output signal (electrical, mechanical, magnetic, etc.).
- The term transducer is often used synonymously with sensors. Sensor is a device that responds to a change in the physical phenomenon. On the other hand, a transducer is a device that converts one form of energy into another form of energy. Sensors are transducers when they sense one form of energy input and output in a different form of energy.
- Sensors can also be classified as passive or active. In passive sensors, the power required to produce the output is provided by the sensed physical phenomenon itself whereas the active sensors require external power source.
- In embedded system, sensor and actuators are used for controlling the system. Sensors are connected to input port. Actuators are connected to output port.
- Sensor captures the changes in the environmental variable. Middle system process the information. Actuators are changed according to the input variable. It displays the output.
- Example of control is air conditioner system. It controls the room temperature to a specified limit.

1.4.2 Cloud Computing

- Cloud computing has the almost unlimited capacity of storage and processing power which is a more mature technology at least to a certain extent to solve the problem of most of the Internet of things.
- Cloud computing is a pay-per-use model for enabling available, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, services) that can be rapidly provisioned and released with minimal management effort or service-provider interaction.

- Cloud storage services may be accessed through a web service API, a cloud storage gateway or through a web-based user interface.
- **Cloud computing services are offered to users in different forms :** Infrastructure-as- a- Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS).
- **Software as a Service (SaaS) :** Model in which an application is hosted as a service to customers who access it via the Internet. The provider does all the patching and upgrades as well as keeping the infrastructure running. The traditional model of software distribution, in which software is purchased for and installed on personal computers.
- **Platform as a Service (PaaS) :** Platform as a service is another application delivery model and also known as cloud-ware. Supplies all the resources required to build applications and services completely from the Internet, without having to download or install software. Services includes application design, development, testing, deployment, and hosting, team collaboration, web service integration, database integration, security, scalability, storage, state management, and versioning. PaaS is closely related to SaaS but delivers a platform from which to work rather than an application to work with.
- **Infrastructure as a Service (IaaS) :** SaaS and PaaS are providing apples to customers, IaaS doesn't. It offers the hardware so that your organization can put whatever they want onto it. Rather than purchase servers, software, racks, and having to pay for the datacenter space for them, the service provider rents for resources like server space, network equipment, memory etc.

1.4.3 Big Data Analytics

- A category of technologies and services where the capabilities provided to collect, store, search, share, analyze and visualize data which have the characteristics of high-volume, high-velocity and high-variety.
- Examples of big data generated by IoT systems :

a) Weather Monitoring Stations

b) Machine sensor data from industrial systems

c) Health and fitness data

d) Location and tracking systems

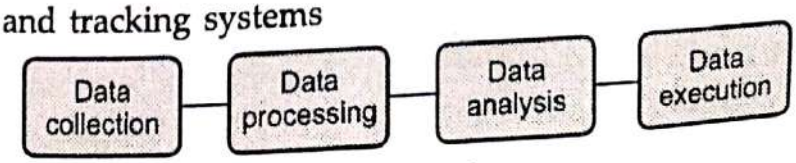


Fig. 1.4.1

1.4.4 Communication Protocol

- Communication protocols are used as a backbone of IoT systems. It enable network connectivity and coupling to applications. Communication protocols allow devices to exchange data over the network.
- Communication protocol also performs error correction and detection, flow control, data encoding, addressing mechanism etc.
- Sequence control, lost of packet, retransmission are the other functions of communication protocol.

1.4.5 Embedded System

- A system is a set of interacting or interdependent component parts forming a complex unit. It is a fixed plan to perform one or many task.
- Embedded system is an electronic system which is designed to perform one or a limited set of functions using software and hardware.
- General definition of embedded systems is : embedded systems are computing systems with tightly coupled hardware and software integration that are designed to perform a dedicated function. The word embedded reflects the fact that these systems are usually an integral part of a larger system, known as the embedding system. Multiple embedded systems can coexist in an embedding system.
- An embedded system has three main components: Hardware, Software and time operating system.
- Hardware parts includes power supply, processor, memory, times & communication ports, system application specific circuit etc.
- Software parts includes the application software is required to perform the series of tasks. An embedded system has software designed to keep in view of three constraints :
  a. Availability of System Memory
  b. Availability of processor speed
  c. The need to limit power dissipation when running the system continuously in cycles of wait for events, run , stop and wake up.
- Demand for low cost and higher density platform requires the integration of devices. As integration levels increases, more and more logic is added to the processor die, creating families of applications specific service processors.
- System-on-Chip (SoC) designs increasingly become the driving force of a number of modern electronics systems. A number of key technologies integrate together in forming the highly complex embedded platform.

1.5 IoT Challenges

GTU : Winter-18

- From the beginning, IoT devices present inherited challenges since they are constrained devices with low memory, processing, communication and energy capabilities.

1. The first key challenge for a ubiquitous deployment is the integration of multi-technology networks in a common all-IP network to ensure that the communication network is reliable and scalable. For this purpose, IoT relies on the connectivity and reliability for its communications on Future Internet architecture.

2. The second key challenge is to guarantee security, privacy, integrity of information and user confidentiality. The majority of the IoT applications need to take into considerations the support of mechanisms to carry out the authentication, authorization, access control, and key management.

- In addition, due to the reduced capabilities from the constrained devices enabled with Internet connectivity, a higher protection of the edge networks needs to be considered with respect to the global network.

3. The third key challenge is to offer support for the mobility, since the Future Internet presents a more ubiquitous and mobile Internet. Mobility support increases the applicability of Internet to new areas.

- The most present nowadays are mobile platforms such as smart phones and tablets which enable a tremendous range of applications based on ubiquitous location, context awareness, social networking, and interaction with the environment.

- Future Internet potential is not limited to mobile platforms, else IoT is another emerging area of the Future Internet, which is offering a high integration of the cybernetic and physical world.

4. Other challenges are also arising from the application, economical, and technological perspectives. For example, from an application point of view are the requirements for processing large amounts of data for a growing number of devices, it is called Big Data

1.6 IoT Levels

- Levels of IoT system are IoT Level-1, IoT Level-2, IoT Level-3, IoT Level-4, IoT Level-5, IoT Level-6.
- IoT system consists of following components

1. Device : IoT device performs identification, remote monitoring, sensing and actuating functions.

2. Resource : IoT device used software components as resources for accessing, processing, and storing sensor information. It also controls actuators.
3. Controller Service : It sends data from the device to the web service and receives commands from the application for controlling the device
4. Database : IoT generates large amount of local database and cloud database.
5. Web Service - Web services act as a link between the IoT device, application, database and analysis components
6. Analysis Component : It is responsible for analyzing the IoT data and generate results
7. Application : User use this interface for controlling and monitoring various IoT system.

1.6.1 IoT Level 1

- Physical devices and controllers that might control multiple devices. These are the "things" in the IoT, and they include a wide range of endpoint devices that send and receive information.
- Fig. 1.6.1 shows IoT level 1.

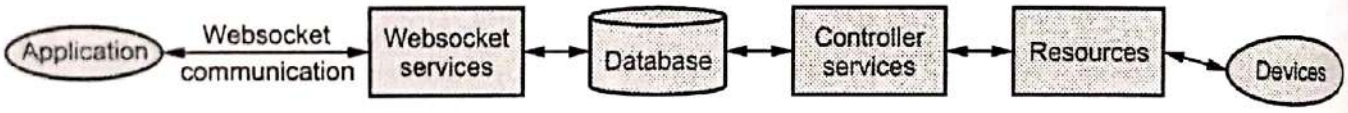


Fig. 1.6.1 IoT level 1

- Level 1 IoT systems are suitable for modeling low-cost and low complexity solutions where the data involved is not big and the analysis requirements are not computationally intensive
- Eg: Home automation
- The system consists of a single node that allows controlling the lights and appliances in a home remotely. The device used in the system interfaces with the light and appliances using electronic relay switches. The status information of each light or appliance is maintained in the local database. The controller service continuously monitors the state of each light or appliance and triggers the relay switches accordingly.

1.6.2 IoT Level 2

- In level 2, single node performs sensing and local analysis. Fig. 1.6.2 shows block diagram.

- Both data and application is stored on the cloud. This type of system is suitable for big data used for analysis and that to performed on local side.

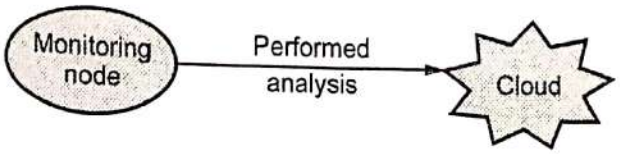


Fig. 1.6.2 Level 2

- Smart irrigation is an example of level 2 IoT system. System uses single sensor for monitoring the soil moisture level and controls the irrigation system.
- Controller service monitors continuously the moisture level. If the moisture level drops below a threshold, the irrigation system is turned ON.

1.6.3 IoT Level 3

- Single node is used in level 3 internet of things. Collected data is stored on the cloud and processed on the cloud. Application is also stored on the cloud.
- This is suitable for huge data and analysis requirement are computationally intensive.

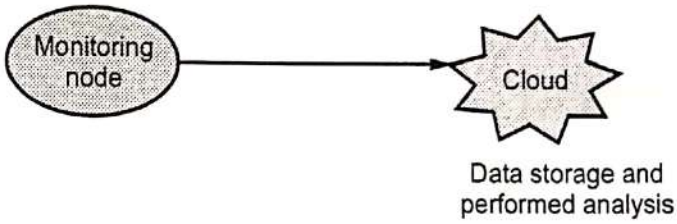


Fig. 1.6.3 Level 3

- REST or websocket is used for communicating with client machine. Consider an example of tracking package handling.
- Single node monitors the vibration level for the package being shipped. For monitoring vibration level, accelerometer and gyroscope sensor is used.
- Accelerometer and gyroscope sensor is used. System allows shippers tracking cargo to communicate with the devices, changing reporting frequencies and investigating alerts generated by attached sensors.
- Controller device sends the sensor data to the cloud in real time using WebSocket service. Data is stored on the cloud and analysis is also performed on the cloud.
- Cloud can send alerts if the vibration levels is greater than threshold level.

1.6.4 IoT Level 4

- Level 4 IoT system uses multiple nodes (sensors) and processing is performed on local node. Data is stored on the cloud and application is stored on the cloud storage.
- Fig. 1.6.4 shows block diagram of Level 4 IoT.

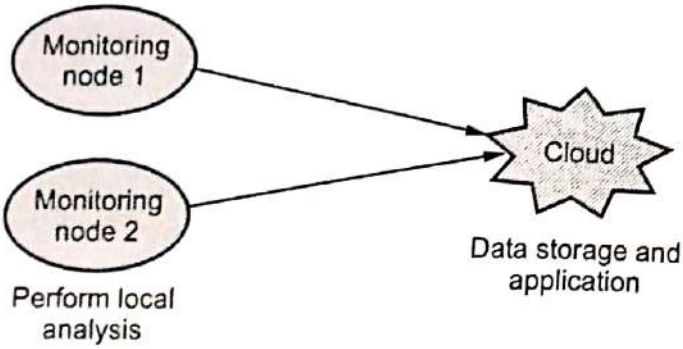


Fig. 1.6.4 Level 4

- Level 4 IoT uses two observer sensor for local and cloud. These sensor subscribe and receive information form the IoT device to cloud .
- The observer sensor can process information and used for various purposes. Control function is not performed by this observer sensor.
- Noise monitoring system is used IoT level 4. Sensors are not depends upon each other. Each sensor runs its own controller service that sends the data to the cloud.

1.6.5 IoT Level 5

- It contains multiple end sensor and one coordinator sensor. Fig. 1.6.5 shows IoT level 5 block diagram.
- The end sensor perform sensing and/or actuation. Coordinator sensor collect data from the end sensor and sends to the cloud.

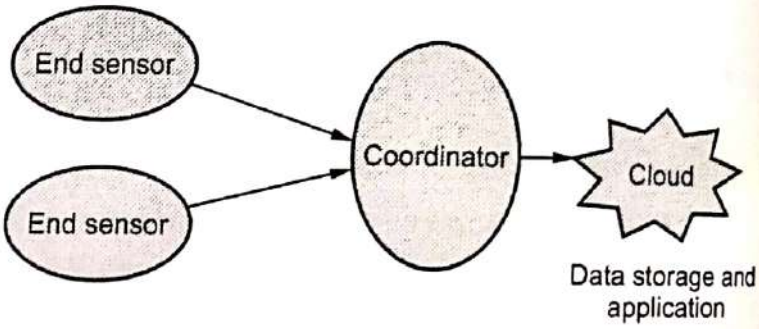


Fig. 1.6.5 Level 5

- Data is stored and analysis in the cloud and application is also cloud based. Forest fire detection system uses level 5 IoT system.
- Multiple sensor are kept at different location to monitor temperature, humidity, carbon dioxide level. Coordinator node (sensor) collect data for end node and these nodes act as gateway and provides Internet services to the IoT systems.

1.6.6 IoT Level 6

- It contains multiple independent end nodes and it performs sensing and/or actuation function. It sends data to the cloud. Fig. 1.6.6 shows block diagram of level 6 IoT.

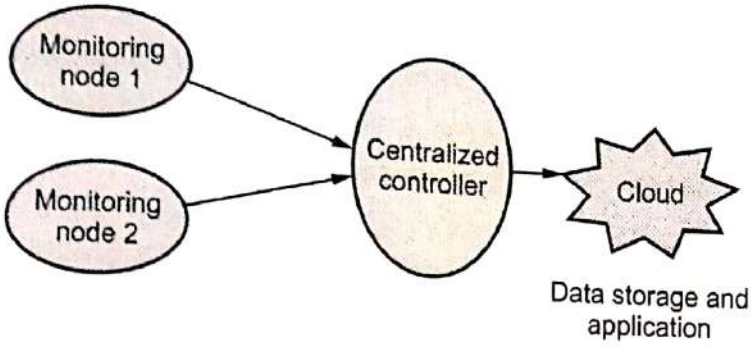


Fig. 1.6.6 Level 6

- Data is stored on the cloud and application is also cloud based. Result is displayed on the cloud. Centralized controller knows the status of monitoring node and sends control commands to the nodes.
- Weather monitoring system uses Level 2 IoT based system. Multiple nodes are kept at different location to monitor temperature, humidity level.
- End node send real time data to cloud using WebSocket service. Cloud database is used for storing data. Data analysis is performed on cloud side. Cloud based application is used for display data.

1.7 IoT and Cyber Physical System

- Cyber-physical systems integrate sensing, computation, control and networking into physical objects and infrastructure, connecting them to the Internet and to each other.
- Cyber-physical systems (CPS) are smart systems that include engineered interacting networks of physical and computational components.
- IoT refers to any systems of interconnected people, physical objects, and IT platforms, as well as any technology to better build, operate, and manage the physical world via pervasive data collection, smart networking, predictive analytics, and deep optimization.
- CPS includes software, hardware, sensors, actuators, and embedded systems, and is connected to human-machine interfaces and multiple systems.
- A number of sensors, actuators, and control devices are connected by a network to form a complex system for acquiring, processing, calculating, and analysing physical environment information and applying the results to the physical environment.
- The CPS is a technology closely related to the IoT, and a next-generation network-based distributed control system that combines a physical system with sensors and actuators and a computing element that controls it.

- The CPS is divided into three layers: the perceptual layer, the data transmission layer, and the application layer.
- The first layer or perception layer, includes the recognition and the sensor, and consists of the global positioning system (GPS), RFID, sensor, actuator, camera, and IoT.
- The collected data can be composed of sound, light, mechanical, chemical, thermal, electrical, biology and location data, and the sensor can generate real-time data through node collaboration in wide-area and local network domains.
- Thus, the perception layer recognizes and collects data, sends it to the communication layer, and collaborates between the IoT nodes in the network.
- The communication layer is responsible for exchanging and processing data between the sensor and the application in communication. This layer communicates using various technologies such as wire network devices and wireless. This is one of the key elements of the CPS, which typically has a wide range from local to global.
- The communication layer is also responsible for reliability and supports real-time transmission.
- The application layer can be applied and interacted with various fields, and is sometimes referred to by a different name depending on the application one is using.
- For example, a typical CPS is the Supervisory Control and Data Acquisition (SCADA) system used in critical infrastructures such as the Smart Grid and the industrial control system (ICS).
- This layer processes the information received from the data transport layer and includes the commands to be executed by the physical sensors and actuators, and it controls the commands to be used in each field.
- In addition, data aggregation of different resources, intelligent processing of large amounts of data, object control and management are performed.

1.8 IoT and WSN

- A wireless sensor network (WSN) is a network formed by a large number of sensor nodes where each node is equipped with a sensor to detect physical phenomena such as light, heat, pressure, etc.
- WSNs nowadays usually include sensor nodes, actuator nodes, gateways and clients. A large number of sensor nodes deployed randomly inside of or near the monitoring area , form networks through self-organization.

- Sensor nodes monitor the collected data to transmit along to other sensor nodes by hopping. During the process of transmission, monitored data may be handled by multiple nodes to get to gateway node after multi- hop routing, and finally reach the management node through the internet or satellite.
- Standards for WSN technology have been well developed, such as Zigbee (IEEE802.15.4). The IEEE 802.15.4 is simple packet data protocol for lightweight wireless networks.
- It works well for long battery life, selectable latency for controllers, sensors, remote monitoring and portable electronics.
- WSN is more for sensing and information-collecting purposes. Other networks include body sensor network (BSN), visual or video sensor network (VSN), vehicular sensor networks, underwater (acoustic) sensor networks, interplanetary sensor networks, fieldbus networks, and others.
- The extended scope of WSN is the USN, or ubiquitous sensor network, a network of intelligent sensors that could one day become ubiquitous.
- WSN developed at Linköping University in Sweden is based on the ZigBee specification and is an IoT solution. It provides nodes for monitoring temperature, relative humidity, carbon dioxide, VOC, PM and electricity consumption as well as for automatic control.
- A wireless sensor network is a network formed by a large number of sensor nodes where each node is equipped with some sensors to detect physical phenomena. In IoT, the sensor nodes and devices are interconnected to transmit useful measurement information via distributed sensor networks.
- Other networks are body sensor network (BSN), video sensor network (VSN), vehicular sensor network (V2V), interplanetary sensor network etc.
- VSN devices come with image sensors, adequate processing power and memory. They use wireless communication interfaces to collaborate and jointly solve tasks such as tracking persons within the network. In all applications, VSNs monitor a potentially large group of people and record sensitive image data which might contain identities of persons, their behaviour, interaction patterns or personal preferences.
- A central idea of VSNs is to keep data processing local to reduce the amount of transmitted data. Fig. 1.8.1 shows VSN.

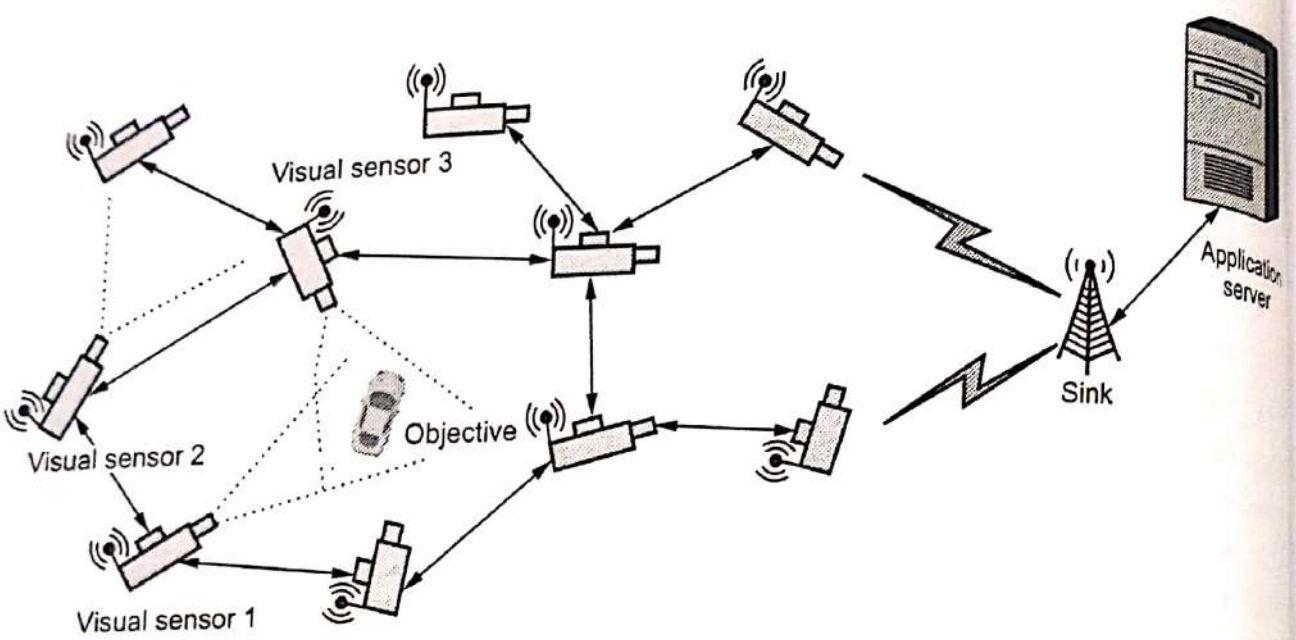


Fig. 1.8.1 : Example of wireless visual sensor networks

- A single VSN device has only a limited field of view but VSNs are typically designed to cover large areas. Therefore, multiple spatially distributed nodes are required. To avoid centralized control and data processing, VSNs use peer to peer communication for coordination, configuration, data exchange, handover of tracked objects or data fusion.
- To simplify deployment of spatially distributed VSNs, they rely no longer only on dedicated communication networks but make use of existing infrastructure which is not under full control of the VSN operators.
- Other VSN applications include environmental monitoring, smart homes and meeting rooms, entertainment and virtual reality as well as elderly care and assisted living.
- Body sensor network (BSN) is one of application of wearable computing device to enable wireless communication between several miniaturized body-sensor units and a single body central unit worn on the human body to transmit vital signs and motion readings to medical practitioners.
- Fig. 1.8.2 shows sources of body sensor network.

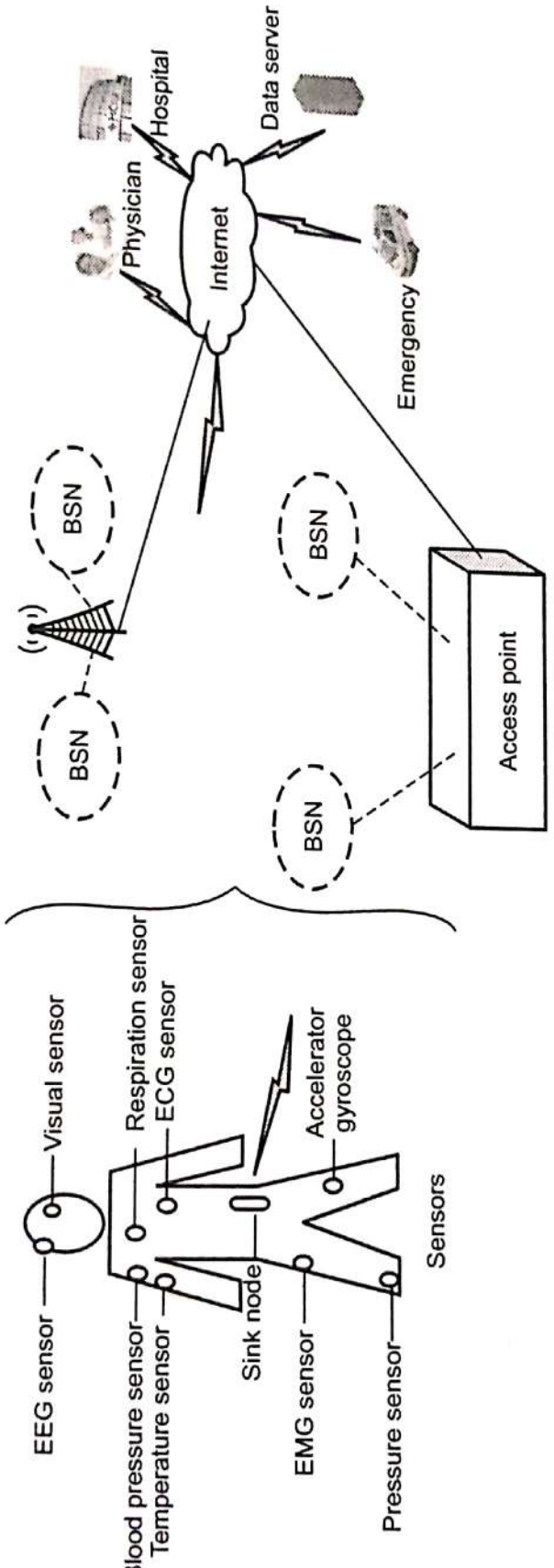


Fig. 1.8.2 : Source of body sensor network

- Sensor node electronics should be miniaturized, low power and detect medic signals such as electrocardiogram (ECG), photoplethysmogram (PPG), pulse rat blood flow, pressure and temperature. The collected data from the control device are then transferred to remote destinations in a wireless body area network fo diagnostics.
- Most pupolar wireless technologies used for medical monitoring are ZigBee WLAN, GSM, Bluetooth.
- Sensor nodes are designed to collect raw signals from a human body. A sensor node undertakes three functions: detecting signal via a front end digitizing/coding/controlling for a multi access communication and finally wireless transmission.
- In data acquisition and processing, the microcontroller maintains a power management scheme to control the distribution of the energy from battery in an optimized manner. The signal from a human body is usually weak and coupled with noise.
- The development of WSNs was motivated by military applications such as battlefield surveillance. Fig. 1.8.3 shows typical sensor network.

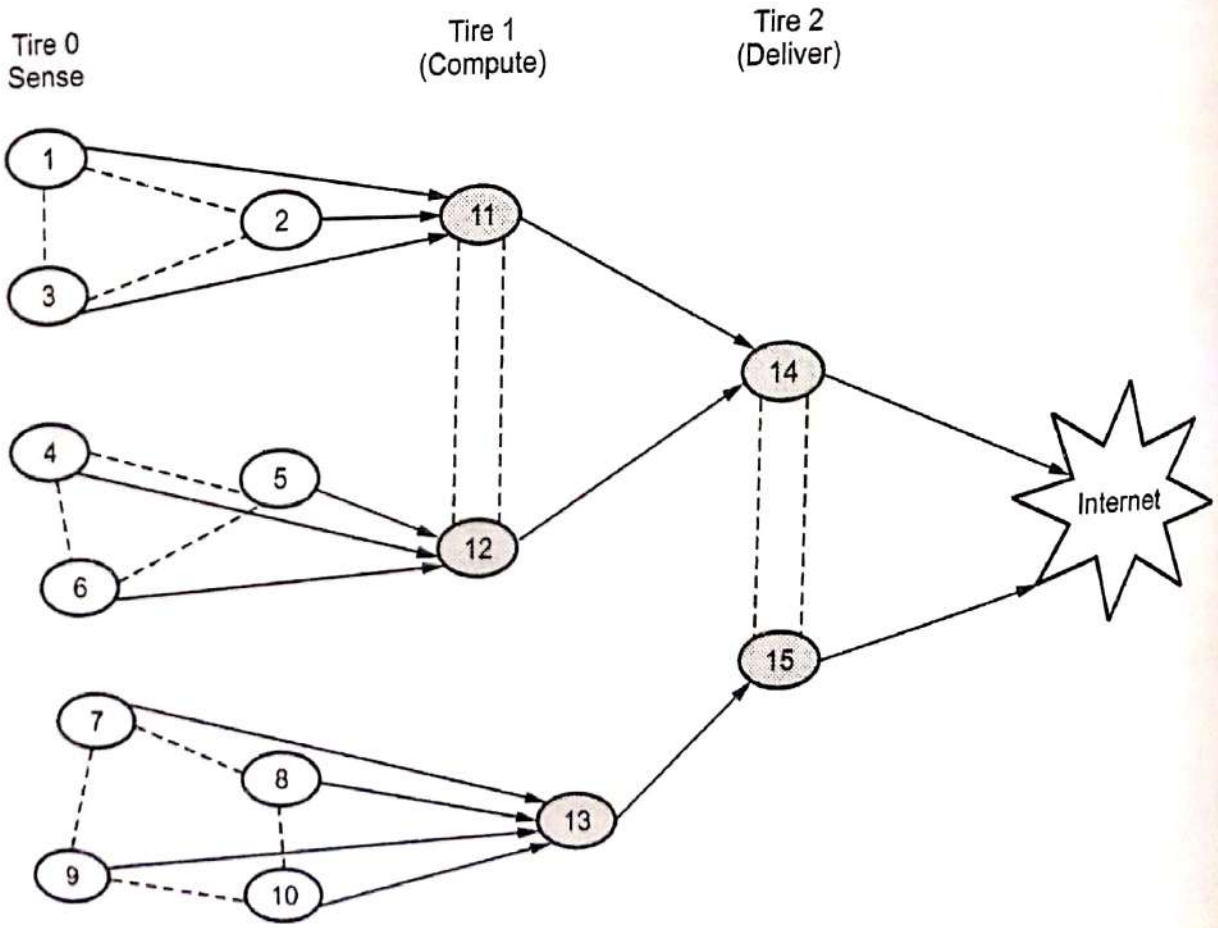


Fig. 1.8.3 : Sensor network architecture

- WSN is built of nodes, each node connected to one or more sensors. Each such sensor network node has typically several parts: a radio transceiver with an antenna, a microcontroller, an electronic circuit for interfacing with the sensors, and an energy source, usually a battery or an embedded form of energy harvesting.
- Topology of sensor network may be star topology or mesh topology. Following are the components used by sensor networks:
  1. Sensor node: sense target events, gather sensor readings, manipulate information, send them to gateway via radio link
  2. Base station/ sink: communicate with sensor nodes and user/ operator
  3. Operator/ user: task manager, send query
- For reliable data transmission in a WSN, routing method is used. Routing protocols are distributed and reactive. The nodes in the system start looking for a route only when they have application data to transmit.
- Commonly used routing algorithm for WSN are dynamic source routing (DSR) and Ad hoc on- demand distance vector (AODV).
- Energy is the scarcest resource of WSN nodes, and it deter-mines the lifetime of WSNs. WSNs are meant to be deployed in large numbers in various environments, including remote and hostile regions, with ad hoc communications as key. For this reason, algorithms and protocols need to address the following issues :
  1. Lifetime maximization
  2. Robustness and fault tolerance
  3. Self- configuration

1.9 Short Questions and Answers

Q.1 Define IoT.

Ans. : • By embedding short-range mobile transceivers into a wide array of additional gadgets and everyday items, enabling new forms of communication between people and things, and between things.

- The Internet of Things (IoT) is the network of physical objects i.e. devices, vehicles, buildings and other items embedded with electronics, software, sensors, and network connectivity that enables these objects to collect and exchange data.

2 Sensors, Microcontrollers and their Interfacing

Syllabus

Sensor interfacing, Types of sensors, Controlling sensors, Microcontrollers, ARM.

Contents

2.1 Sensors

- Sensor converts a physical quantity into a corresponding voltage. Sensor is a device that when exposed to a physical phenomenon (temperature, displacement, force, etc.) produces a proportional output signal (electrical, mechanical, magnetic, etc.).
- The term transducer is often used synonymously with sensors. Sensor is a device that responds to a change in the physical phenomenon. On the other hand, a transducer is a device that converts one form of energy into another form of energy. Sensors are transducers when they sense one form of energy input and output in a different form of energy.
- Sensors can also be classified as passive or active. In passive sensors, the power required to produce the output is provided by the sensed physical phenomenon itself whereas the active sensors require external power source.
- In embedded system, sensor and actuators are used for controlling the system. Sensors are connected to input port. Actuators are connected to output port.
- Sensor captures the changes in the environmental variable. Middle system process the information. Actuators are changed according to the input variable. It displays the output.
- Example of control is air conditioner system. It controls the room temperature to a specified limit.
- Deflection : The signal produces some physical (deflection) effect closely related to the measured quantity and transuded to be observable.
- Null : The signal produced by the sensor is counteracted to minimize the deflection. That opposing effect necessary to maintain a zero deflection should be proportional to the signal of the measurand.
- Here, the output is usually an 'electrical quantity' and measurand is a 'physical quantity, property or condition which is to be measured'.
- A sensor is a device that responds to a physical stimulus, measures the physical stimulus quantity and converts it into a signal usually electrical, which can be read by an observer or by an instrument.
- A sensor can be very small and itself can be a trackable devices. The sensor itself, if not connected is not part of the IoT or WSN value chain.

Specifications of Sensor :

1. **Accuracy** : Error between the result of a measurement and the true value being measured.
2. **Resolution** : The smallest increment of measure that a device can make.

3. **Sensitivity :** The ratio between the change in the output signal to a small change in input physical signal. Slope of the input-output fit line.
4. **Repeatability/Precision :** The ability of the sensor to output the same value for the same input over a number of trials.
5. **Bandwidth :** the frequency range between the lower and upper cut-off frequencies, within which the sensor transfer function is constant gain or linear.

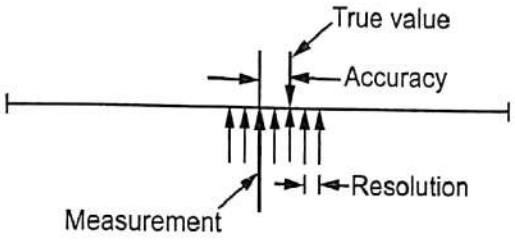


Fig. 2.1.1 : Accuracy vs. Resolution

2.1.1 Types of Sensors

- Mechanical sensor : Any suitable mechanical / electrical switch may be adopted but because a certain amount of force is required to operate a mechanical switch it is common to use micro-switches.
- Pneumatic sensor : These proximity sensors operate by breaking or disturbing an air flow. The pneumatic proximity sensor is an example or a contact type sensor. These cannot be used where light components may be blown away.
- Optical sensor : In there simplest form, optimcal proximity sensors operate by breaking a light beam which falls onto a light sensitive device such as a photocell. These are examples of non contact sensors. Care must be exercised with the lighting environment of these sensors for example optical sensors can be blinded by flashes from are welding processes, airborne dust and smoke clouds may impede light transmission etc.
- Electrical sensor : Electrical proximity sensors may be contact or non-contact. Simple contact sensors operate by making the sensor and the component complete an electrical circuit. Non-contact electrical proximity sensors rely on the electrical principles of either induction for detecting metals or capacitance for detecting non metals as well.
- Range sensing : Range sensing concerns detecting how near or far a component is from the sensing position, although they can also be used as proximity sensors. Distance or range sensors use non-contact analog techniques. Short range sensing, between a few millimetres and a few hundred millimetres is carried out using transmitted energy waves of various types e.g. radio waves, sound waves and lasers.

2.1.2 Sensor Data Communication Protocols

1. Direct transmission protocols :
   - In direct communication protocol, each sensor sends its data directly to the base station. If the base station is far away from the nodes, direct communication will require a large amount of transmit power from each node.
   - This will quickly drain the battery of the nodes and reduce the system lifetime. However, the only receptions in this protocol occur at the base station, so if either the base station is close to the nodes, or the energy required receiving data is large, this may be an acceptable method of communication.
2. Minimum transfer energy protocols :
   - In these protocols, nodes act as routers for other nodes' data in addition to sensing the environment. These protocols differ in the way the routes are chosen.
   - Some of these protocols only consider the energy of the transmitter and neglect the energy dissipation of the receivers in determining the routes.
   - Depending on the real time costs of the transmit amplifier and the radio electronics, the total energy expended in the system might actually be greater using MTE routing than direct transmission to the base station.
   - It is clear that in MTE routing, the nodes closest to the base station will be used to route a large number of data messages to the base station.
   - Thus, these nodes will die out quickly, causing the energy required to get the remaining data to the base station to increase and more nodes to die.

2.1.3 Actuators

- A device or mechanism capable of performing a physical action. Actuators interact with the world. Sensors capture information from the world.
- The interface between the microcontroller and the sensors or the actuators is either analog or digital.
- An actuator requires a control signal and a source of energy. An actuator is the mechanism by which a control system acts upon an environment. The control system can be simple, software-based, a human, or any other input.
- When the actuation is a motion, motor have to be used for rotational or linear motion.
- The selection of the proper actuator is more complicated than selection of the sensors, primarily due to their effect on the dynamic behaviour of the overall

system. Furthermore, the selection of the actuator dominates the power needs and the coupling mechanisms of the entire system.

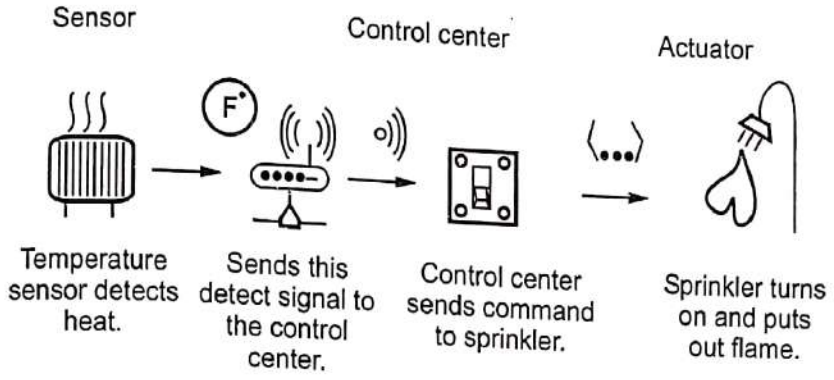


Fig. 2.1.2

- In typical IoT systems, a sensor may collect information and route to a control centre where a decision is made and a corresponding command is sent back to an actuator in response to that sensed input.

2.2 Sensor Interfacing

- The concept of interfacing sensors is giving input from sensors to microcontroller or input systems in a way which they can understand and act accordingly.
- Most of the sensors give output in analog form but the microcontroller or microprocessor needs input as digital so now comparators act as interfacing sensors where they convert analog signals to digital signals.
- Interfacing differs by the type of sensor and by the type of system you're interfacing to device or microprocessor / microcontroller.

2.3 Types of Sensors

2.3.1 MQ-02/05 Gas Sensor Interfacing with Arduino

- The MQ series of gas sensors use a small heater inside with an electro-chemical sensor. They are sensitive for a range of gasses and are used indoors at room temperature. The output is an analog signal and can be read with an analog input of the Arduino.
- The MQ-2 gas sensor module is useful for gas leakage detecting in home and industry. It can detect LPG, i-butane, propane, methane, alcohol, hydrogen and smoke.

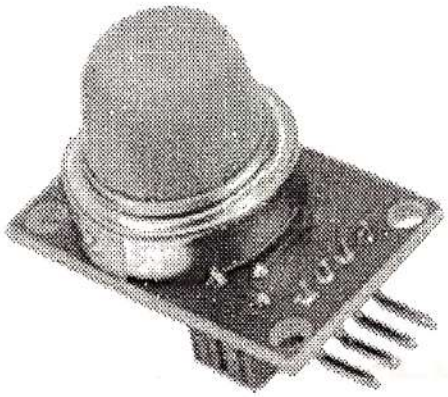


Fig. 2.3.1 : MQ-02 gas sensor

- Some modules have a built-in variable resistor to adjust the sensitivity of the sensor. Fig. 2.3.1 shows MQ-02 gas sensor.
- The analog gas sensor - MQ2 is used in gas leakage detecting equipment's in consumer and industry markets, this sensor is suitable for detecting LPG, i-butane, propane, methane, alcohol, Hydrogen, smoke. It has a high sensitivity and fast response time.
- MQ-2 Features
  a. With signal output command.
  b. Dual signal output (analog output and high/low digital output).
  c. 0 to 4.2 V analog output voltage, the higher the concentration, the higher the voltage.
  d. It has higher sensitivity to natural gas, natural gas and city gas.
  e. Long service life, stable and reliable.
  f. Fast response and recovery features.
- Fig. 2.3.2 shows MQ-02 gas sensor interfacing with Arduino.

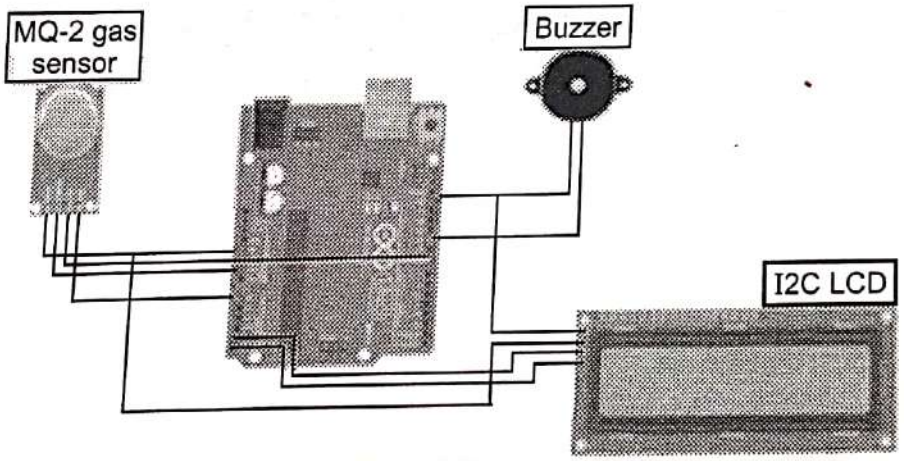


Fig. 2.3.2 : MQ-02 gas sensor interfacing with Arduino

Arduino	MQ-02
A0	A0
5 V	VCC
GND	GND

- The voltage that the sensing element outputs changes consequently to the smoke/gas level that exists within the atmosphere. The sensing element outputs a

voltage that's proportional to the concentration of smoke/gas. In different words, the link between voltage and gas concentration is that the following :

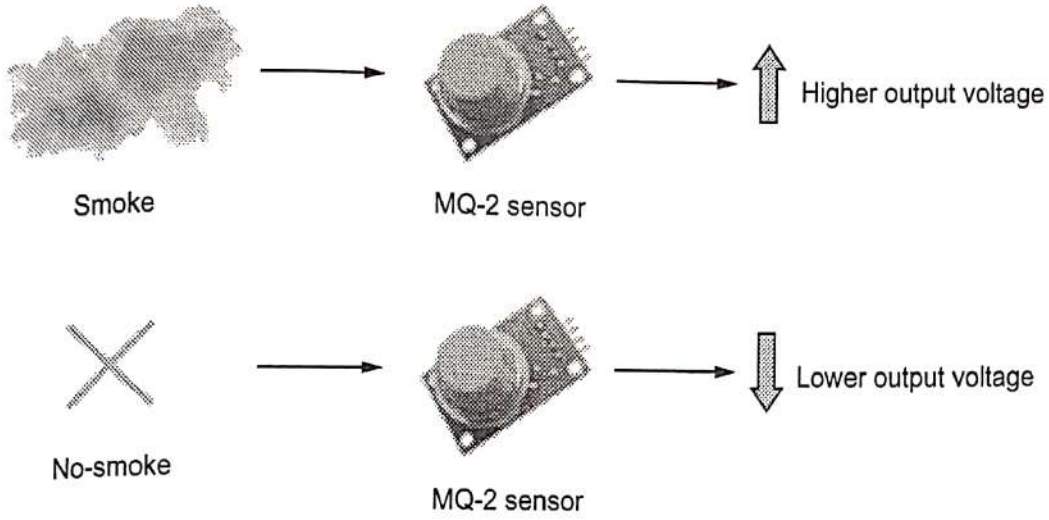


Fig. 2.3.3 : Functioning of MQ-2

2.3.2 Interfacing the Obstacle Sensor

- An object can be detected with an infrared system consisting of an infrared transmitter and a receiver.
- Infrared obstacle sensor module has built in IR transmitter and IR receiver that sends out IR energy and looks for reflected IR energy to detect presence of any obstacle in front of the sensor module. The module has on board potentiometer that lets user adjust detection range. The sensor has very good and stable response even in ambient light or in complete darkness.
- Fig. 2.3.4 shows obstacle sensor.

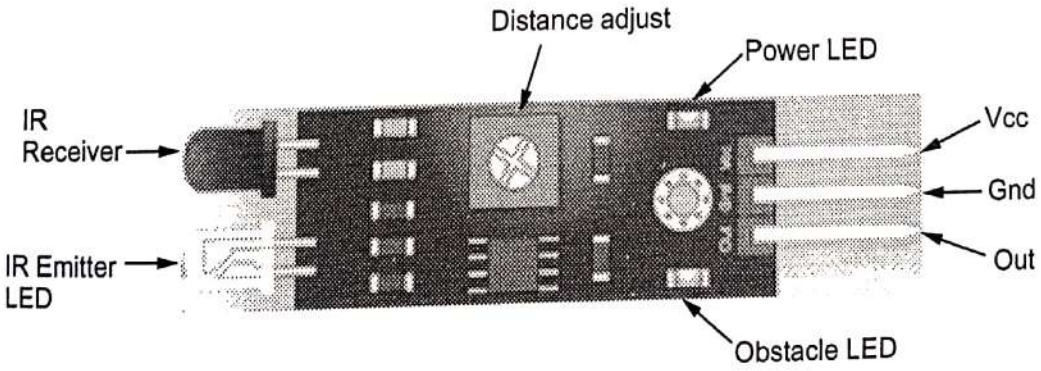


Fig. 2.3.4 : Obstacle sensor

- IR sensor has three terminals, two terminals for power supply and one terminal for output. Additionally, there is a potentiometer to calibrate the sensor.
- The IR transmitter sends an infrared signal that, in case of a reflecting surface (e.g. white color), bounces off in some directions including that of the IR receiver that captures the signal detecting the object.

- The module uses an infrared led (IR) and a photo transistor to detect the pulse of the finger and whenever a pulse is detected, red led flashes. There will be led on the light side of the finger and a photo transistor on the other side of the finger. Photo transistor is used to obtain the flux emitted. The resistance of the photo resistor will change when the pulses will change.
- In a transmissive sensor, the light source and the detector are place facing each other and the finger of the person must be placed in between the transmitter and receiver.
- Reflective sensor, on the other hand, has the light source and the detector adjacent to each other and the finger of the person must be placed in front of the sensor.
- A simple heartbeat sensor consists of a sensor and a control circuit. The sensor part of the heartbeat sensor consists of an IR LED and a photo diode placed in a clip.
- The control circuit consists of an op-amp IC and few other components that help in connecting the signal to a microcontroller.
- First, in order to display the heartbeat readings in bpm, we have to connect a 16×2 LCD display to the Arduino UNO.
- The 4 data pins of the LCD module (D4, D5, D6 and D7) are connected to pins 1, 1, 1 and 1 of the Arduino UNO. Also, a 10 KΩ potentiometer is connected to Pin 3 of LCD.
- The RS and E (Pins 3 and 5) of the LCD are connected to pins 1 and 1 of the Arduino UNO.
- Next, connect the output of the heartbeat sensor module to the analog input pin (Pin 1) of Arduino.

2.3.4 Interfacing Ultrasonic Sound Sensor

- Ultrasonic sensors work by emitting sound waves at a frequency too high for humans to hear. They then wait for the sound to be reflected back, calculating distance based on the time required.
- If you need to measure the specific distance from your sensor, this can be calculated based on this formula :

Distance L = ½ T × C

Where T = Time and C = The speed of sound

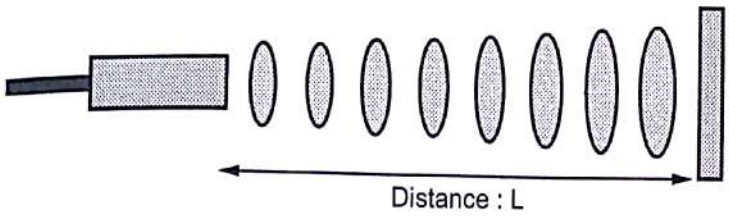


Fig. 2.3.6

- At 20 °C (68 °F), the speed of sound is 343 meters/second (1125 feet/second), but this varies depending on temperature and humidity.
- Fig. 2.3.7 shows ultrasonic sensors.

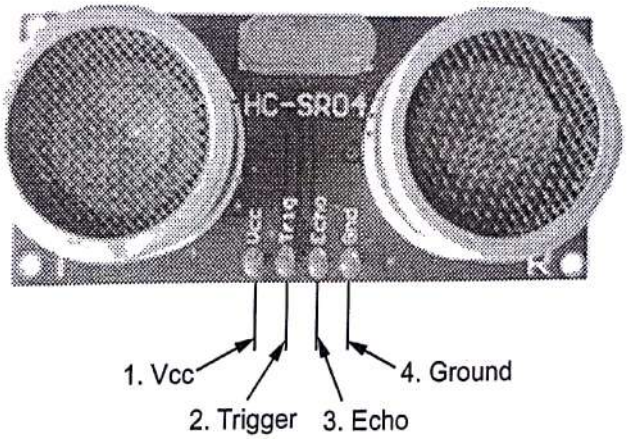


Fig. 2.3.7 : Ultrasonic sensors

Ultrasonic sensor pin configuration

Pin name	Description
Vcc	The Vcc pin powers the sensor, typically with +5 V
Trigger	Trigger pin is an input pin. This pin has to be kept high for 10 us to initialize measurement by sending US wave.
Echo	Echo pin is an output pin. This pin goes high for a period of time which will be equal to the time taken for the US wave to return back to the sensor.
Ground	This pin is connected to the ground of the system.

2.3.5 Gyro Sensor

- Gyroscope sensor is a device that can measure and maintain the orientation and angular velocity of an object.

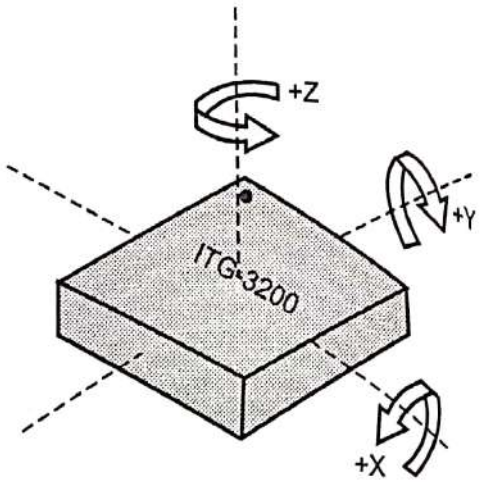


Fig. 2.3.8

- When things rotate around an axis they have what's called angular velocity. Angular velocity is generally expressed in degrees per second (°/s) or Revolutions Per Second (RPS).
- There are three basic types of gyroscope rotary (classical) gyroscopes, vibrating structure gyroscope and optical gyroscopes.
- Gyroscope sensors are used whenever rate of turn sensing is required without a fixed point of reference. A gyroscope sensor reports the rate of rotation of the device around the three sensor axes.
- If you attach the sensor to the wheel shown above, you can measure the angula velocity of the z axis of the gyro. The other two axes would not measure an rotation.
- Imagine if the wheel spins once per second. It would have an angular velocity o 360 degrees per second. The spinning direction of the wheel is also important. Is i clockwise around the axis, or is it counter-clockwise ?
- Rotation is positive in the counter clockwise direction (right-hand rule). That is, a observer looking from some positive location on the x, y or z axis at a devic positioned on the origin would report positive rotation if the device appeared t be rotating counter clockwise.
- Fig. 2.3.9 shows gyroscope sensor interfacing with Arduino-UNO.

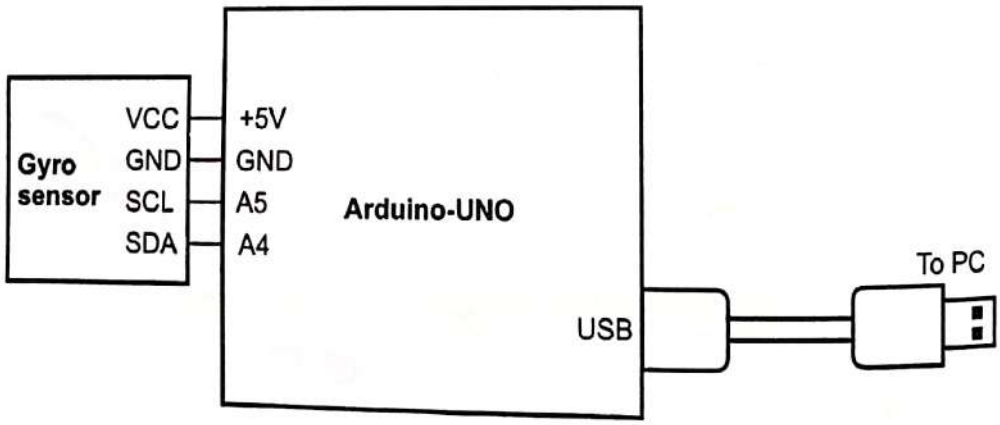


Fig. 2.3.9 : Gyroscope sensor interfacing with Arduino-UNO

- A gyroscope would be used in an aircraft to help in indicating the rate of rotatio around the aircraft roll axis. As an aircraft rolls, the gyroscope will measur non-zero values until the platform levels out, whereupon it would read a zer value to indicate the direction of "down."

- A traditional, mechanical gyroscope is a simple wheel that's mounted on 2-3 gimbals.

2.3.6 Light Dependent Resistor (LDR) Sensor

- Light Dependent Resistor (LDR) is a resistor whose resistance decreases with increasing incident light intensity; in other words, it exhibits photo conductivity. LDR sensor is also known as photoresistor.
- Fig. 2.3.10 shows LDR sensor.

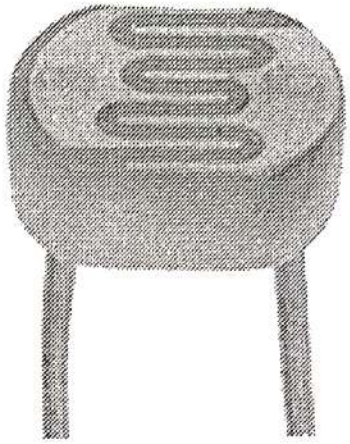


Fig. 2.3.10 LDR sensor

- A photo-resistor is made of a high resistance semiconductor. If light falling on the device is of high enough frequency, photons absorbed by the semiconductor give bound electrons enough energy to jump into the conduction band. The resulting free electron conduct electricity, thereby lowering resistance.
- The construction of an LDR includes a light-sensitive material that is placed on an insulating substrate like as ceramic. The material is placed in a zigzag shape in order to get the required power rating and resistance. The area of zigzag separates the metal placed areas into two regions.
- When the light is absorbed by the material then the conductivity of the material reduces. When the light falls on the LDR, then the electrons in the valence band of the material are eager to the conduction band. But, the photons in the incident light must have energy superior than the bandgap of the material to make the electrons jump from one band to another band.
- LDR's are cheap and are readily available in many sizes and shapes. Light-dependent resistors are simple and low-cost devices.
- Disadvantage : Highly inaccurate with a response time of about tens or hundreds of milliseconds.
- These devices are used where there is a need to sense the presence and absence of light is necessary. These resistors are used as light sensors and the applications of LDR mainly include alarm clocks, street lights, light intensity meters, burglar alarm circuits.

2.3.7 GPS Sensor

- GPS sensors are receivers with antennas that use a satellite-based navigation system with a network of 24 satellites in orbit around the earth to provide position, velocity and timing information.
- GPS modules contain tiny processors and antennas that directly receive data sent by satellites through dedicated RF frequencies.
- GPS is made up of three parts : Satellites, ground stations and receivers.
- GPS receivers come in all different shapes and sizes, are widespread and are affordable. Today, GPS receivers can be found in watches, phones, tablets, computers, cars and a wide variety of other devices.
- The satellite signals : The digital signals from the GPS satellites are emitted at two frequencies (1228 and 1575 MHz). They are received by the GPS receiver and contain much detailed information.
- In addition to the timing signal, there are also data for identification of the satellite (by its number), about the status of the satellite clock, the satellite orbit, the current status of the satellite and various correction data. The data is divided into frames of 1500 bits; one frame is transmitted in about 30 seconds.

2.4 Microcontrollers

- The 8051 microcontroller is one of the basic types of microcontroller, designed by Intel in 1980's. This microcontroller was based on harvard architecture.
- The 8051 microcontroller has two buses and two memory spaces of 64K × 8 size for program and data units. It has an 8-bit processing unit and 8-bit accumulator units.
- **Features of 8051 family :**
  1. 4096 bytes program memory on-chip.
  2. 128 bytes data memory on-chip.
  3. Four register banks.
  4. 128 user-defined software flags.
  5. 64 kilobytes each program and external RAM addressability.
  6. Two-level prioritized interrupt structure.
  7. Full depth stack for subroutine return linkage and data storage.
  8. Direct byte and bit addressability.
  9. Binary or decimal arithmetic.
  10. Signed-overflow detection and parity computation.

11. Hardware multiple and divide in 4 μsec.
12. Integrated boolean processor for control applications.

- Fig. 2.4.1 shows block diagram of 8051.

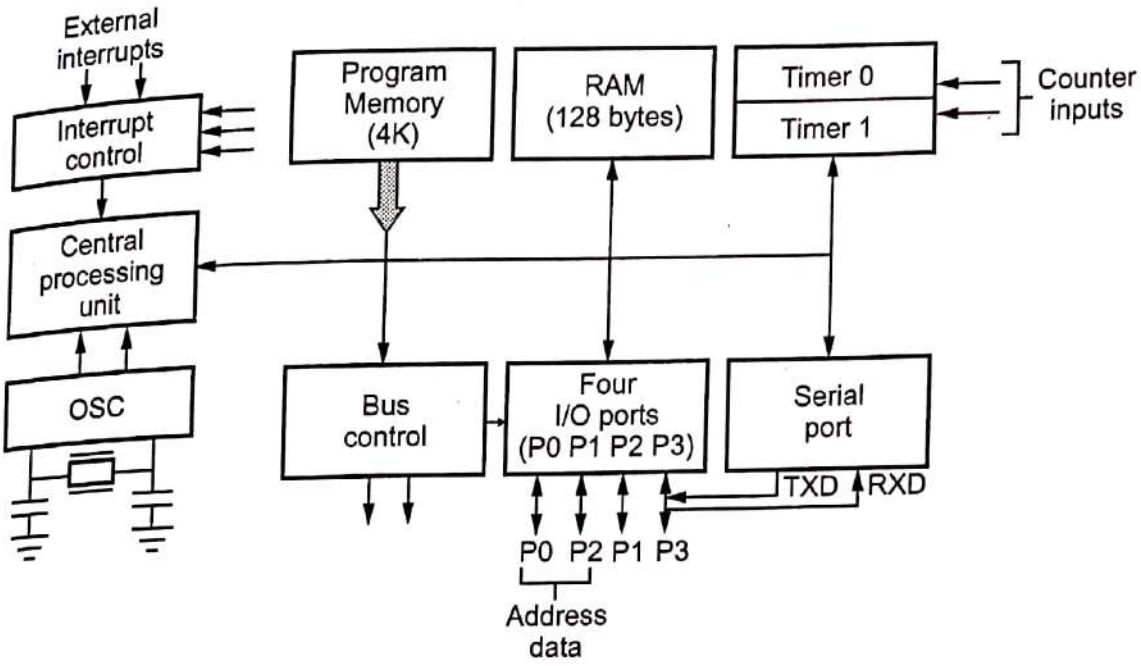


Fig. 2.4.1 : Architectural layout of 8051 microcontroller

- The various components/modules of microcontroller are processor core, memory, interrupt controller, timer/counter, digital I/O, analog I/O, interfaces and watchdog timer.

1. Processor core : It is the central processing unit of the microcontroller and consists of Arithmetic Logic Unit (ALU). This unit is responsible for the execution of arithmetical and logic operations.
2. Control unit : It controls the operation of MC and is represented by STATUS register. It helps in transfer of data to other registers.
3. Memory : There are two types of data memories in a MC. They are Static RAM (SRAM) and EEPROM/Flash.
   - SRAM consists of Special Function Registers (SFR) and General Purpose Registers (GPR). SRAM are 8 bit memory cells that provide link between the microcontroller hardware and software.
   - Special Function Registers (SFR) controls the functional blocks whereas the General Purpose Registers (GPR) stores the values of variables and constants.
   - EEPROM/Flash : It is a type of non-volatile electronic memory, which is used for storage of data. EEPROM memory is comparatively slow and has limited number of write cycles and hence used to store the data that needs to be saved when power supply of MC is switched off.

4. Counter/Timer module : This module has Timer/Counter which measures the time and calculate the events. Usually a MC contains 2-3 Timer/Counters and it operates in timer or counter mode and calculates the number of received pulses. Timer also generates PWM signal which is converted to DC voltage using low pass filters.
5. Digital module : This module consists of digital I/O. It helps the microcontroller to detect and output the logic states (High or Low).
6. Analog module : This module consists of analog I/O. It helps in converting voltage level to digital value. Most MC's have integrated analog-digital converters. To generate analog output RC low pass filters are used to convert PWM outputs to DC voltage.
7. Serial interface module : They are designed to receive or send the digital signals to other devices. Most controllers are provided with SPI, SCI, Ethernet, USB interfaces.
8. Watchdog timer : It observes the execution of the MC program. It generates 'RESET' signal if the execution process fails.

2.4.1 Microcontroller 8051 Family

	Features	8052	8051	8031
1.	Number of I/O pins	32	32	32
2.	On-chip program memory	8K	4K	No
3.	RAM	128 bytes	128 bytes	128 bytes
4.	Number of serial ports	1	1	1
5.	Number of timers	3	2	2
6.	Interrupt sources	8	6	8

2.4.2 Register Set of 8051

1. Register A :

It is also called as accumulator. It is an 8-bit register. Accumulator is used in all mathematical and arithmetical operations. It holds a source operand. The register A is located at the address E0H in the SFR memory space.

2. Register B :

Register B is located at the address F0H of the SFR address space. The B register is used along with the ACC in multiplication and division operations. These two operations are performed on data that are stored only in registers A and B. During multiplication operation, one of the operand is stores in B register and also the higher byte of the result. In case of division operation, the B register holds the divisor and also the remainder of the result. It can also be used as a general-purpose register for normal operations and is often used as an auxiliary register by programmers to store temporary results.

3. Program Status Word (PSW) :

- The 8051 has a 8-bit PSW register which is also known as flag register. In the 8-bit register only 6-bits are used by 8051. Two unused bits are user definable bits.
- In the 6-bits four of them are conditional flags. They are carry - CY, Auxiliary Carry - AC, Parity - P and Overflow - OV. These flag bits indicate some conditions that resulted after an instruction was executed.

B_7	B_6	B_5	B_4	B_3	B_2	B_1	B_0
CY	AC	F0	RS1	RS0	OV	-	P

CY Bit-7 - Carry flag

AC Bit-6 - Auxiliary carry flag for BCD operations

F0 Bit-5 - User defined flag (Flag zero)

RS1, RS0 Bit-4-3 - Select the working register banks as follows :

RS1	RS0	Bank selection	
0	0	00H - 07H	Bank 0
0	1	08H - 0FH	Bank 1
1	0	10H - 17H	Bank 2
1	1	18H - 1FH	Bank 3

OV Bit-2 - Overflow flag

\- Bit-1 - Reserved

P Bit-0 - Parity flag (1 = Even parity)

4. Stack Pointer Register (SP) :

- 8-bit register which stores the address of stack top. The stack pointer is used to indicate where the next value to be removed from the stack should be taken from. When a value is pushed onto the stack, the 8051 first increments the value of SP and then stores the value at the resulting memory location.
- Similarly when a value is popped off the stack, the 8051 returns the value from the memory location indicated by SP and then decrements the value of SP.
- Since the SP is only 8-bit wide it is incremented or decremented by two.
- SP is modified directly by six instructions : PUSH, POP, ACALL, LCALL, RET and RETI. It is also used intrinsically whenever an interrupt is triggered.

5. Data Pointer (DPTR) :

Data pointer (DPTR) consists of a high byte and low byte. Its function is to hold a 16-bit address. It is used by a number of commands which allow the 8051 to access external memory. DPTR can also be used as two 8-registers DPH and DPL.

6. Program Counter (PC) :

The 8051 has a 16-bit program counter. It is used to hold the address of memory location from which the next instruction is to be fetched. PC is automatically incremented to point the next instruction in the program sequence after execution of the current instruction.

7. Serial data buffer :

The serial buffer is actually two separate registers, a transmit buffer and a receive buffer. When data is moved to SBUF, it goes to the transmit buffer and is held for serial transmission. When data is moved from SBUF, it comes from the receive buffer.

8. Timer registers :

Register pairs (TH0, TL0) and (TH1, TL1) are the 16-bit counting registers for Timer/Counters 0 and 1, respectively.

9. Control register :

Special function registers IP, IE, TMOD, TCON, SCON and PCON contain control and status bits for the interrupt system, the Timer/Counters and the serial port.

2.5 ARM

- ARM means **Advanced RISC Machines.** ARM machines have a 32-bit Reduced Instruction Set Computer (RISC) Load Store Architecture. It is first RISC microprocessor for commercial use and market-leader for low-power and cost-sensitive embedded applications.

- The processor originated in England in 1984. At its inception ARM stood for Acorn RISC Machine. The first ARM reliant systems include the Acorn : BBC Micro, Masters and the Archimedes. During this early period they were used mostly for British educational systems and therefore, were not widely available or known outside England. However in 1987 the ARM became the first commercial RISC processor.
- The ARM is a Von Neumann, load/store architecture i.e. only 32 bit data bus for both instruction and data. Also for the load/store instruction access memory.
- Licenses ARM core designs to semiconductor partners who fabricate and sell to their customers. ARM does not fabricate silicon itself
- **First models had only a 26-bit program counter, limiting the memory space to 64 MB**
- In 1990, the research section of Acorn separated from the parent company and formed : Advanced RISC Machines Limited.
- The ARM is a **32-bit architecture.** When used in relation to the ARM :
  1. **Byte** means **8 bits**
  2. Halfword means **16 bits.**
  3. Word means **32 bits.**
- Most ARM's implement two instruction sets
  1. 32-bit ARM Instruction Set
  2. 16-bit Thumb Instruction Set
- Memory is addressed as a 32 bit address space. Data type can be 8 bit bytes, 16 bit half-words or 32 bit words and may be seen as a byte line folded into 4-byte words
- **ARM1** processor was the first commercialized RISC processor and it contained 25,000 transistors.

ARM processor	Features
ARM1	• First version of ARM processor. • 26-bit addressing, no multiply/ coprocessor.
ARM2	• ARM2, First commercial chip. • Included 32-bit result multiply instructions/coprocessor support.

ARM2a	• ARM3 chip with on-chip cache. • Added load and store. • Cache management.
ARM3	• ARM6, 32 bit addressing, virtual. • Memory support.
ARM7	• Most popular used today. • Suitable for DSP work.
ARM8	• It Includes a five stage pipeline. • Speculative instruction fetcher. • Processor to allow a higher clock speed.
ARM9	• Uses five stage pipeline. • Support Harvard Architecture chip.

2.5.1 Features

1. Thumb Set designed for 16-bit word lengths and instructions, which internally executes by same 32-bit core.
2. ARM views memory as a linear collection of bytes numbered upwards from zero. It contains memory management unit and memory protection unit.
3. Most operations are executed over registers.
4. All instructions can be conditional.
5. It uses Big-endian and Little-endian method.
6. The ARM processor supports 25 different instruction.
7. ARM provides no explicit return instruction.
8. The Software Interrupt (SWI) instruction is the only way an ARM processor can access resources controlled by the operating system.
9. Many Thumb data processing instructions use a 2-address format.
10. Jazelle Instruction Set : Introduces technological infrastructure for running Java code.
11. The ARM architecture has a large variety of addressing modes.

2.5.2 ARM Cortex

- **The ARM Cortex-A profile :** It is application profile. It is a family of applications processors for complex OS and user applications. The Cortex-A family processors support the ARM and Thumb instruction sets.
- **The ARM Cortex-R profile :** It is real time profile. Used in high end applications. Automatic breaking system is example of R-profile.
- **The ARM Cortex-M profile :** It is microcontroller profile. Used in industrial control applications.

A profile	R Profile	M profile
It is application profile	It is real time profile	It is microcontroller profile
Uses 32-bit and 64-bit registers	Uses 32-bit registers width	Uses 32-bit registers width
A32 (ARM), T32 (Thumb) and A64 instruction sets	ARM and Thumb instructions	Thumb instruction set only
Virtual memory support	Protected memory support	Protected memory support
Runs rich operating systems	Runs real-time OS	Microcontroller applications
Used in mobile phones and video systems	Automatic breaking systems	Industrial control applications

2.5.3 Operating Modes

- ARM processor has 7 modes of operation. Switching between modes can be done manually through modifying the mode bits in the CPSR register. Most application programs execute in user mode.
- The privileged modes are entered to serve interrupts or exceptions. The system mode is special mode for accessing protected resources. It does not use registers used by exception handlers, so it cannot be corrupted by any exception handler error.
- Switching between these modes requires saving/loading register values.

1. User mode : It is main execution mode of the processor. Processor runs application software in user mode and provides protection and isolation. All other execution modes are privileged and are therefore only used to run system software.
2. Fast interrupt processing mode : ARM processor entered in this mode when it receives an interrupt signal from the designated fast interrupt source.
3. Normal interrupt processing mode : ARM processor entered in this mode when it receives an interrupt signal from the any other interrupt source.

4. Software interrupt mode : While executing program, it processor found a software interrupt instruction then it enter in this mode.
5. Undefined instruction mode : Is entered when the processor attempts to execute an instruction that is supported neither by the main integer core or by one of the coprocessors. This mode can be used to implement coprocessor emulation.
6. System mode : Operating system executes privileged tasks.
7. Abort mode : This mode is related to memory faults. It is entered in response to memory faults.

2.5.4 Register Set and CSPR

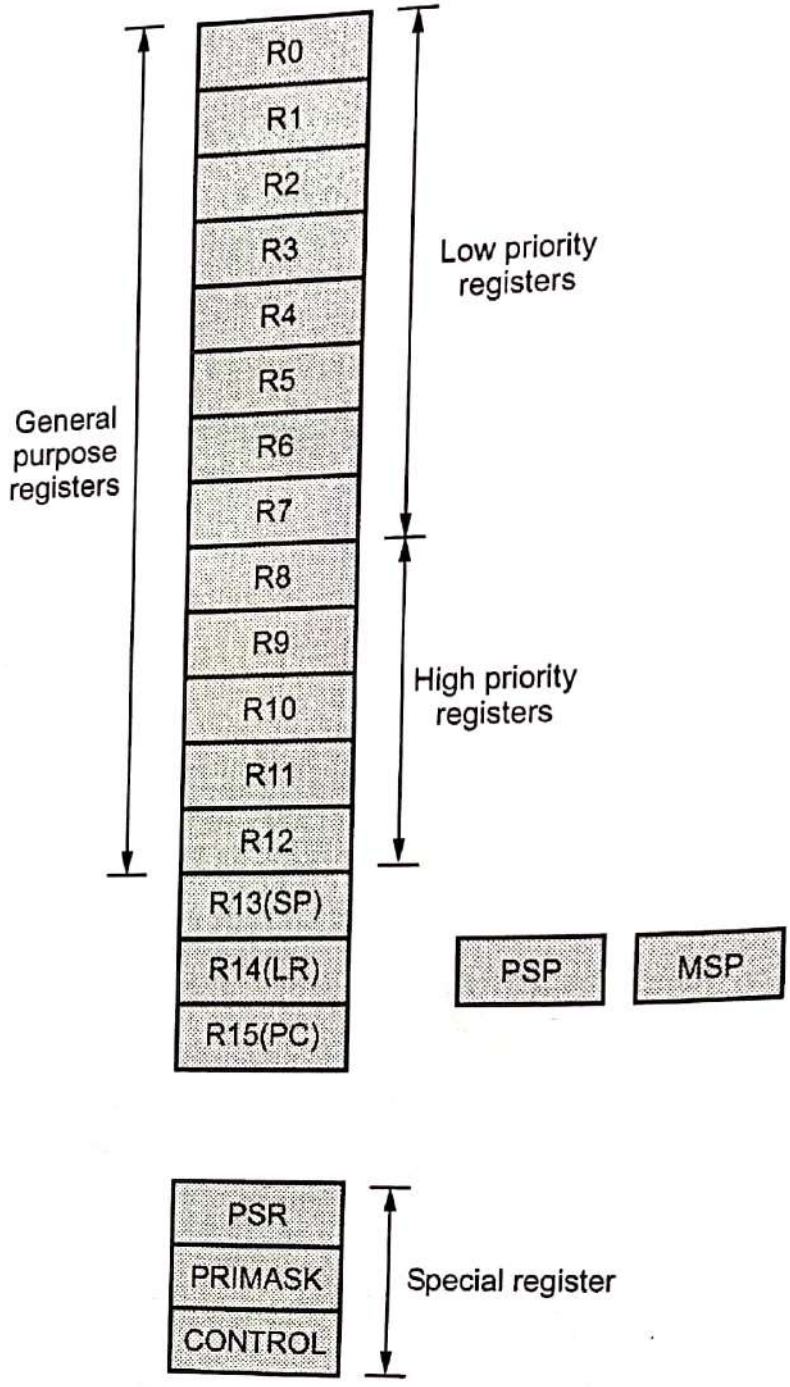


Fig. 2.5.1 Core register set

- The ARM processor has a total of 37 registers made up of 31 general 32 bit registers and 6 status registers. The 16 general registers and one or two status registers are accessible at any time. In privileged modes, mode-specific banked registers become available.
- Fig. 2.5.1 shows core register sets.
- **Registers R0 to R12 :** These are general-purpose registers used to hold either data or address values. All registers are general purpose and may be used to hold data or address values, except for R15 and R14.
- **Program counter register is R15 register.** This register also called Processor Status Register (PSR). R15 contains 24 bits of Program Counter (PC) and 8 bits of Processor Status Register (PSR).
- **Register R14 is used as subroutine link register.** It stores result of branch and link instruction when executes.
- Sometimes R13 used for special purpose. In processor mode, it is used as private stack pointer. R13 is banked for the exception modes. This means that an exception handler can use a different stack to the one in use when the exception occurred.

2.5.5 Current Program Status Register (CSPR)

- ARM core uses CPSR to monitor and control internal operations. The CPSR is a dedicated 32-bit register and resides in the register file.
- CPSR fields are divided in to four fields, each 8-bit wide : Flags, status, extension, and control. In current designs status and extension fields are reserved for future purpose. In some ARM processor cores have extra bits allocated J bit (available only on Jazelle enabled processing which execute 8-bit instructions).
- Fig. 2.5.2 shows CSPR.

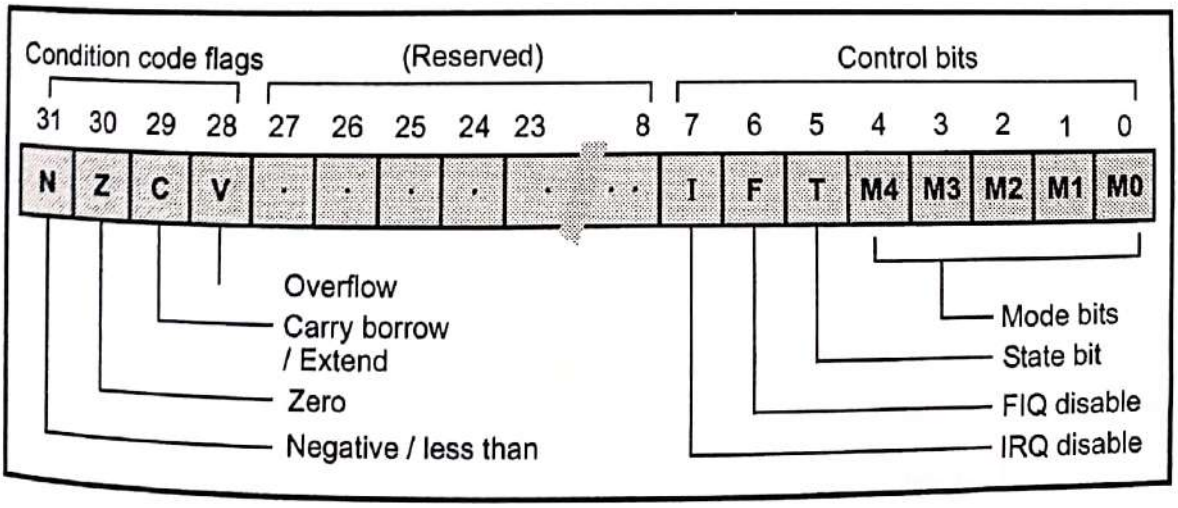


Fig: 2.5.2 CSPR bit configuration

Interrupt Disable Bits

I = 1, disables the IRQ.

F = 1, disables the FIQ.

T Bit (Architecture v4T only)

T = 0, Processor in ARM state

T = 1, Processor in thumb state

Condition Code Flags

N = Negative result from ALU flag

Z = Zero result from ALU flag

C = ALU operation Carried out

V = ALU operation oVerflowed

- **Details of CSPR bit configuration :**

Bits numbers	Name	Remarks
0 to 4	Mode	It specify the current mode of operation
5	T	Status bit T = 1 ARM T = 0 Thumb
6	F	Disables FIQ
7	I	Disables RIQ
8 to 23	Undefined	Undefined
24	J	Jazelle state
25 to 26	Undefined	Undefined
27	Q	Overflow flag
28	V	Conditional flag
29	C	Conditional flag
30	Z	Conditional flag
31	N	Conditional flag

2.5.6 Special Features of ARM Processor

1. Data bus width : ARM processor support 32-bit data width. Processor can rea or write 32 bits data in one cycle.
2. Computational capability : It provides very good computational capability.

3. Lower power : In embedded system, power saving is main constraints. Most of the components operate on the battery power. So designing lower power processor cores is thus a matter of high priority.

4. Pipelining : Pipelining increases the speed of the processor. Simple pipelining is three stage pipelining. In three stage pipeline, stages are fetch, decode and execute. Fig 2.5.3 shows three stage pipeline.

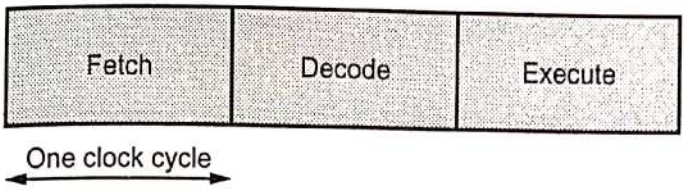


Fig. 2.5.3 Three stage pipelining

- **Instruction fetch** gets the next instruction from memory and increments the program counter. It also stores the new address back into PC register.
- **Instructions decode** takes the instruction from the fetch stage and breaks it down to control signals that are ready to be executed.
- **Execution** retrieves the operands from the register file, which keeps track of registers and the data in each, performs ALU operations, reads from or writes to memory and writes back to register file if that is necessary.
- At any time, 3 different instructions may occupy each of the the 3 - stages of pipeline. It may take three cycles to complete a single-cycle instruction. This is said to have three cycle **latency**. Once a pipeline fills, the processor completes a single-cycle instruction every clock cycle. Therefore the **throughput** is one instruction per cycle.
- Fig 2.5.4 shows three stage pipeline of ARM7 processor.

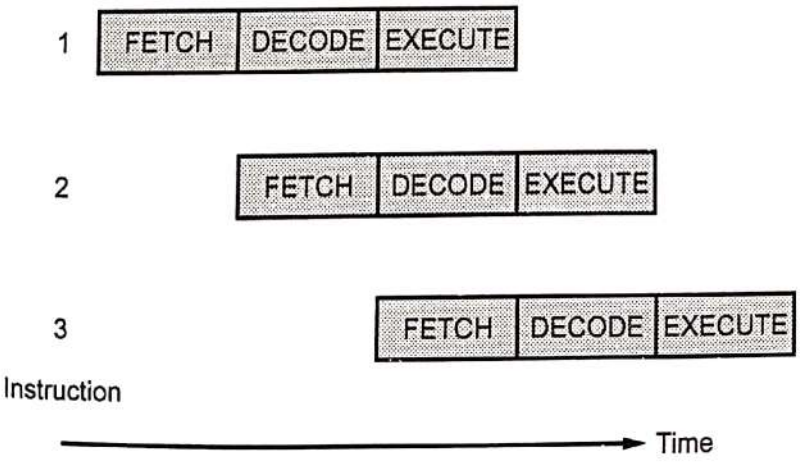


Fig. 2.5.4 Three stage pipeline in ARM7

2.5.7 Data Flow Model of ARM Core

- When an instruction is decoded inside the ARM core and how a partic instruction is executed by interacting with the internal registers file and then s result out of the registers.
- Fig. 2.5.5 shows data flow model of ARM core. Instruction decoder transl instructions before they are executed.

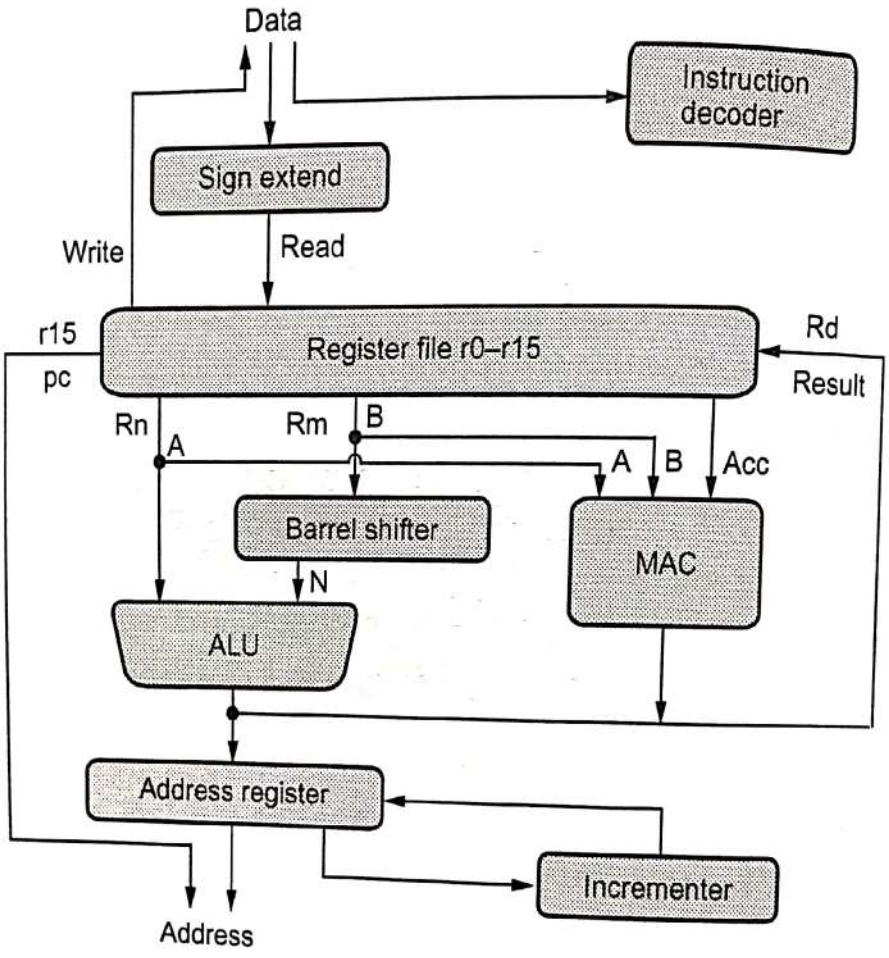


Fig. 2.5.5 : Data flow model of ARM core

- Load instruction : Copy data from memory to register.
- Store instruction : Copy data from register to memory.
- There are no data processing instructions that directly manipulate data in memory. Data items are placed in the register file - a storage bank made up of 32-bi registers.
- ARM instructions typically have two source registers Rn and Rm and one destination register Rd.
- ALU and MAC (Multiply-accumulate) unit takes the register values Rn and Rm from A and B buses and computes a result.

- Load and store instructions use the ALU to generate an address to be held in the address register and broadcast on the address bus.

Barrel shifter

- The ARM does not have actual shift instructions. Instead it has a barrel shifter which provides a mechanism to carry out shifts as part of other instructions.
- Barrel shifter is a unit that can perform more than one bit of shift/rotation, to the right or to the left on an operand.
- One operand to ALU is routed through the barrel shifter. Thus, the operand can be modified before it is used.
- The ARM instruction set has the capability to combine shift and rotate operations along with arithmetic, logical, compare, load and store operations in a single instruction. This is achieved through the barrel shifter.

2.6 Short Questions and Answers

Q.1 What is Barrel shifter in ARM ?

Ans. : Barrel shifter is a unit that can perform more than one bit of shift/rotation, to the right or to the left on an operand. One operand to ALU is routed through the Barrel shifter. Thus, the operand can be modified before it is used. The ARM instruction set has the capability to combine shift and rotate operations along with arithmetic, logical, compare, load and store operations in a single instruction. This is achieved through the barrel shifter.

Q.2 Define microcontroller.

Ans. : A microcontroller is a highly integrated chip which includes, on one chip, all or most of the parts needed for a controller. The microcontroller could be called a "one-chip solution". It typically includes CPU, RAM, ROM, I/O, timers and interrupt controller.

Q.3 How does the ARM know whether the instruction it's running is a thumb instruction or a regular ARM instruction ?

Ans. : The ARM chip contains a special state bit that tells the CPU whether to expect a compressed thumb instruction or a standard ARM instruction. This bit is toggled with its own instruction, BX, which must be inserted into the code every time a programmer or compiler wishes to switch between thumb mode and standard ARM mode. An obvious result of this is that there is some overhead to switching between modes. Thus it is probably not a good idea to switch to thumb unless it will save you more than two instructions of equivalent ARM code.

3 Protocols for IoT

Syllabus

Messaging protocols, Transport protocols, IPv4, IPv6, URI.

Contents

3.1 Messaging Protocol

- Internet of Things solutions employ some kind of messaging protocol for each individual IoT device to communicate in the system. These messaging protocols are used to transmit device telemetry or messages from the IoT devices to the IoT Messaging Hub.
- Messaging protocols are the rules, formats, and functions for messages sent between machines. Essentially, everyone has agreed on the types of information to include with data packets and the way of formatting that information so everyone can read it.

3.1.1 CoAP

- Constrained Application Protocol (CoAP) is a specialized web transfer protocol for use with constrained nodes and constrained (e.g., low-power, lossy) networks.
- CoAP is designed for simplicity, low overhead and multicast support in resource-constrained environments.
- CoAP is a web protocol that runs over the UDP for IoT. Datagram Transport Layer Security (DTLS) is used to protect CoAP transmission.
- The protocol is designed for machine- to-machine (M2M) applications such as smart energy and building automation.
- CoAP provides a request/response interaction model between application endpoints, supports built-in discovery of services and resources, and includes key concepts of the Web such as URIs and Internet media types.
- CoAP is designed to easily interface with HTTP for integration with the Web while meeting specialized requirements such as multicast support, very low overhead, and simplicity for constrained environments.
- CoAP is particularly targeted for small low power sensors, switches, valves and similar components that need to be controlled or supervised remotely, through standard Internet networks.
- CoAP is an application layer protocol that is intended for use in resource-constrained internet devices, such as wireless sensory network (WSN) nodes.
- The key features of CoAP are :
  1. CoAP is a RESTful protocol.
  2. Four methods similar to HTTP: Get, Put, Post and Delete.
  3. Four different message types : Confirmable, Non-Confirmable, Acknowledgement and Reset (Nack).

4. It support synchronous message exchange
5. Easy to proxy to and from HTTP.
6. Constrained web protocol fulfilling M2M requirements.
7. UDP binding with optional reliability supporting unicast and multicast requests. Confirmable and Acknowledgement / Reset messages to provide optional reliability when required. Low header overhead and reduced parsing complexity.
8. Simple proxy and caching capabilities.

- A stateless HTTP mapping, allowing proxies to be built providing access to CoAP resources via HTTP in a uniform way or for HTTP simple interfaces to be realized alternatively over CoAP.
- CoAP provides a request/response interaction model between application endpoints, supports built-in discovery of services and resources, and includes key concepts of the Web such as URIs and Internet media types.
- CoAP is designed to easily interface with HTTP for integration with the Web while meeting specialized requirements such as multicast support, very low overhead, and simplicity for constrained environments.
- CoAP is designed to interoperate with HTTP and the RESTful web through simple proxies, making it natively compatible to the Internet. CoAP is based on REST architecture, which is a general design for accessing Internet resources.
- Fig. 3.1.1 shows CoAP protocol stack.
- CoAP is based on the exchange of compact messages that, by default, are transmitted over UDP. Message of CoAP uses simple binary format.

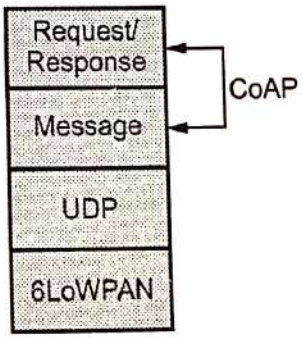


Fig. 3.1.1 CoAP protocol stack

- Message Layer supports 4 types message: CON (confirmable), NON (non-confirmable), ACK (Acknowledgement), RST (Reset)
- Reliable message transport: Keep retransmission until get ACK with the same message ID. Using default time out and decreasing counting time exponentially when transmitting CON. If recipient fail to process message, it responses by replacing ACK with RST.
- Unreliable message transport: transporting with NON type message. It doesn't need to be ACKed, but has to contain message ID for supervising in case of retransmission. If recipient fail to process message, server replies RST.

- Piggy-backed : Client sends request using CON type or NON type message and receives response ACK with confirmable message immediately. For successful response, ACK contain response message (identify by using token), for failure response, ACK contain failure response code.
- Separate response: If server receive a CON type message but not able to response this request immediately, it will send an empty ACK in case of client resend the message.
- When servers ready to response this request, it will send a new CON to client and client reply a confirmable message with acknowledgment. ACK is just to confirm CON message, no matter CON message carry request or response
- Fig. 3.1.2 shows CoAP message format.

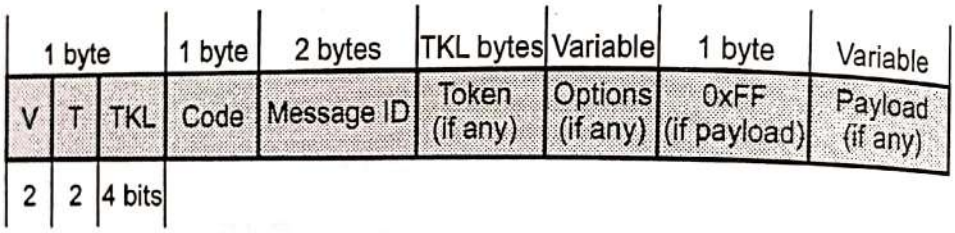


Fig. 3.1.2 CoAP message format

- **Version (V) :** A 2-bit unsigned integer indicating the CoAP version number. Current version is 1. Other values are reserved for future versions.
- **Type (T) :** A 2-bit unsigned integer indicating if this message is of type Confirmable (0), Non-Confirmable (1), Achnowledgement (2) or Reset (3).
- **Token Length (TKL) :** A 4-bit unsigned integer indicating the length of the variable-length Token field (0-8 bytes). Lengths 9-15 are reserved.\
- **Code :** An 8-bit unsigned integer indicating if the message carries a request (1-31) or a response (64-191), or is empty (0). (All other code values are reserved). In case of a request, the Code field indicates the Request Method (1 : GET; 2 : POST; 3 : PUT; 4 : DELETE); in case of a response a Response Code. Possible values are maintained in the CoAP Code Registry (see section 12 of the draft).
- **Message ID :** A 16-bit unsigned integer in network byte order used for the detection of message duplication and to match messages of type Acknowledgment/Reset to messages of type Confirmable/Non-confirmable.

Advantages :

- It runs over UDP and avoids overhead of TCP
- It is easy to do HTTP - CoAP translation
- It is a lightweight application layer protocol designed for constrained devices and constrained networks

Disadvantages :

- Constraints associated with DTLS
- No standardized framework for authorization and access control for CoAP exists as of now
- No explicit support for real-time IoT application at present.

3.1.2 MQTT

- Message Queue Telemetry Transport (MQTT) is Open Connectivity for Mobile, M2M and IoT.
- MQTT is designed for high latency, low-bandwidth or unreliable networks. The design principle minimizes the network bandwidth and device resource requirements.
- MQTT is a lightweight broker-based publish/subscribe messaging protocol designed to be open, simple, lightweight and easy to implement.

MQTT characteristics

1. Lightweight message queueing and transport protocol
2. Asynchronous communication model with messages (events)
3. Low overhead (2 bytes header) for low network bandwidth applications
4. Publish / Subscribe (PubSub) model
5. Decoupling of data producer (publisher) and data consumer (subscriber) through topics (message queues)
6. Simple protocol, aimed at low complexity , low power and low footprint implementations
7. Runs on connection-oriented transport (TCP).
8. MQTT caters for (wireless) network disruptions

- The MQTT protocol works by exchanging a series of MQTT control packets in a defined way. Each control packet has a specific purpose and every bit in the packet is carefully crafted to reduce the data transmitted over the network.
- A MQTT topology has a MQTT server and a MQTT client. MQTT control packet headers are kept as small as possible.
- Having a small header overhead makes this protocol appropriate for IoT by lowering the amount of data transmitted over constrained networks.

- MQTT is the protocol built for M2M and IoT which is used to provide new and revolutionary performance
- Fig. 3.1.3 shows MQTT publish/subscribe framework.

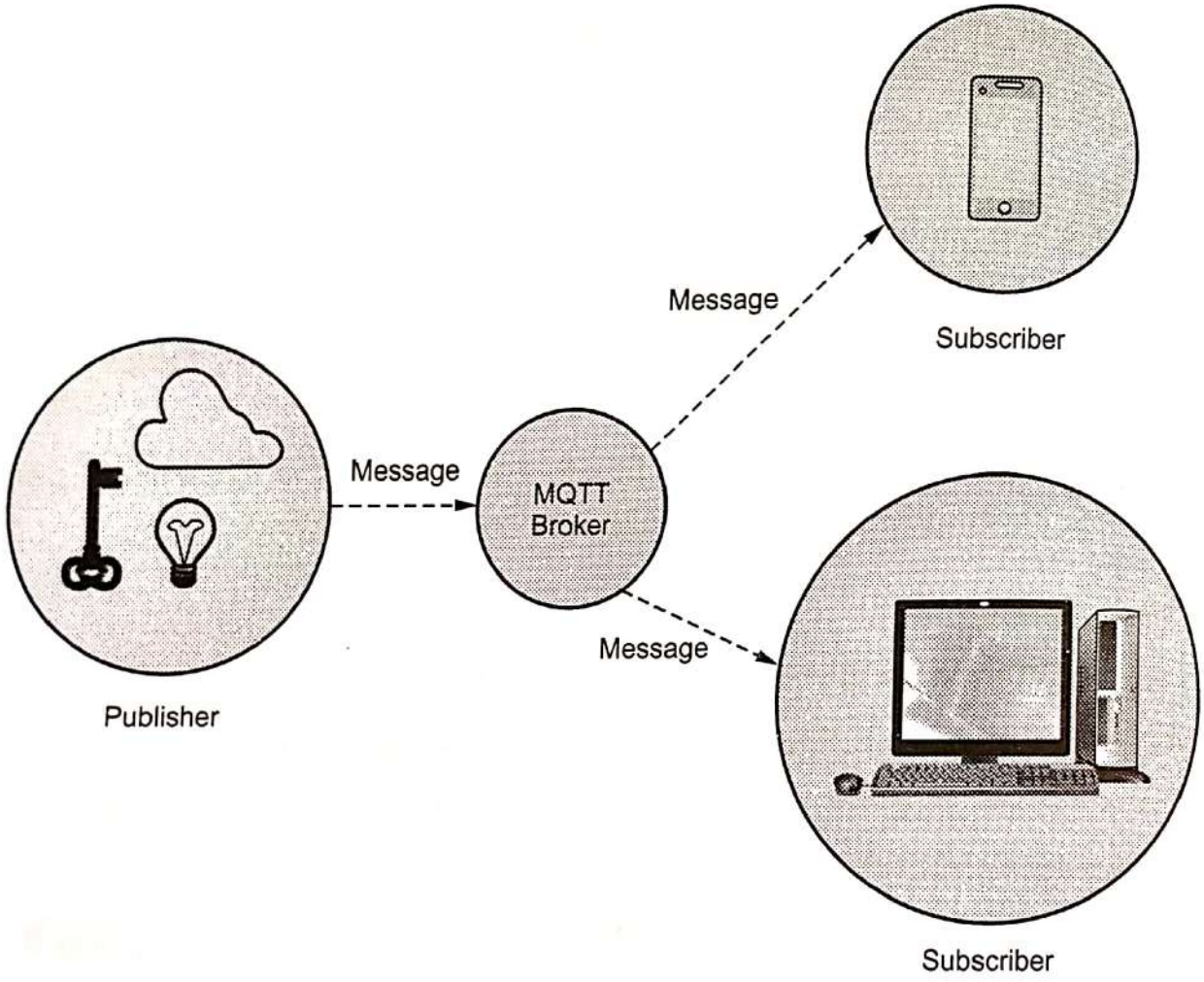


Fig. 3.1.3 MQTT publish/subscribe framework

- A producer publishes a message (publication) on a topic (subject). A consumer subscribes (makes a subscription) for messages on a topic (subject).
- A message server (called BROKER) matches publications to subscriptions.
- If none of them match the message is discarded after modifying the topic. If one or more matches the message is delivered to each matching consumer after modifying the topic
- Publish / Subscribe has three important characteristics:
  1. It decouples message senders and receivers, allowing for more flexible applications.
  2. It can take a single message and distribute it to many consumers.

3. This collection of consumers can change over time, and vary based on the nature of the message.

- The MQTT messages are delivered asynchronously ("push") through publish subscribe architecture.
- The MQTT protocol works by exchanging a series of MQTT control packets in a defined way.
- Each control packet has a specific purpose and every bit in the packet is carefully crafted to reduce the data transmitted over the network.
- A MQTT topology has a MQTT server and a MQTT client. MQTT client and server communicate through different control packets. Table below briefly describes each of these control packets
- MQTT control packet headers are kept as small as possible. Each MQTT control packet consist of three parts, a fixed header, variable header and payload.
- Each MQTT control packet has a 2 byte Fixed header. Not all the control packet have the variable headers and payload.
- A variable header contains the packet identifier if used by the control packet. A payload up to 256 MB could be attached in the packets.
- Having a small header overhead makes this protocol appropriate for IoT by lowering the amount of data transmitted over constrained networks.

MQTT Quality of Service :

- There are the three levels of MQTT QoS.

1. **QoS 0 : AT MOST ONCE**
   - Guarantees that a particular message is only ever received by the subscriber a maximum of one time. This does mean that the message may never arrive.
   - The sender and the receiver will attempt to deliver the message, but if something fails and the message does not reach its destination the message may be lost.
   - This QoS has the least network traffic overhead and the least burden on the client and the broker and is often useful for telemetry data where it doesn't matter if some of the data is lost.
2. **QoS 1 : AT LEAST ONCE**
   - Guarantees that a message will reach its intended recipient one or more times. The sender will continue to send the message until it receives an acknowledgment from the recipient, confirming it has received the message.

- The result of this QoS is that the recipient may receive the message multiple times, and also increases the network overhead than QoS 0.
- In addition more burden is placed on the sender as it needs to store the message and retry should it fail to receive an ack in a reasonable time.

3. **QoS 2 : EXACTLY ONCE**
   - The most costly of the QoS, this QoS will ensure that the message is received by a recipient exactly one time.
   - This ensures that the receiver never gets any duplicate copies of the message and will eventually get it, but at the extra cost of network overhead and complexity required on the sender and receiver.

3.1.3 Difference between CoAP and MQTT

CoAP	MQTT
CoAP uses UDP protocol.	MQTT uses TCP protocol.
It uses Request/Response messaging.	It uses publish/subscribe messaging.
Communication model is One-to-one.	Communication model is Many-to-many.
Advantages : • Lightweight and fast • Low overhead • Support for multicasting	Advantages : • Simple management • Scalability • Robust communication
Weakness : Not as reliable as TCP based MQTT	Weakness : Higher overhead, no multicasting support
Security type is DTLS.	Security type is SSL/TLS.
Effectiveness in LLN is excellent.	Effectiveness in LLN is low.

3.2 Transport Protocols

3.2.1 Bluetooth Low Energy

- Bluetooth Low Energy (BLE) is a low power wireless communication technology that can be used over a short distance to enable smart devices to communicate. It is also called Bluetooth Smart.

- The BLE technology provides an easy and a reliable interface, which is highly appreciated by consumer electronics manufacturers, mobile application developers and engineers.
- BLE uses 2.4 GHz ISM frequency band either in dual mode or single mode. Dual mode supports both bluetooth classic and low energy peripherals.
- All BLE devices use the GATT profile (Generic Attribute Profile). The GATT protocol provides series of commands for the client to discover information about BLE server.
- BLE device is acting in either a central or peripheral role and is sometimes also referred to as a client or server.
  1. Central (Client) : A device that initiates commands and requests, and accepts responses. Examples : Computer, Smartphone
  2. Peripheral (Server) : A device that receives commands and requests, and returns responses. Examples : A temperature sensor, Heart rate monitor
- Bluetooth device is identified by the Bluetooth device address. This address is 48-bit long. There are two types of device addresses :
  a. Public device address : This is a fixed, factory-programmed device address. It must be registered with the IEEE Registration Authority and will never get change during its entire lifetime.
  b. Random device address : This address can be pre-programmed or dynamically generated during the runtime. It has many practical uses in BLE.

Features

1. The lowest power consumption
2. Cost efficient and compatible
3. Ease of use and integration
4. Robustness, security, and reliability
5. License free
6. Support multibrand mobile

- Fig 3.2.1 shows BLE connection.

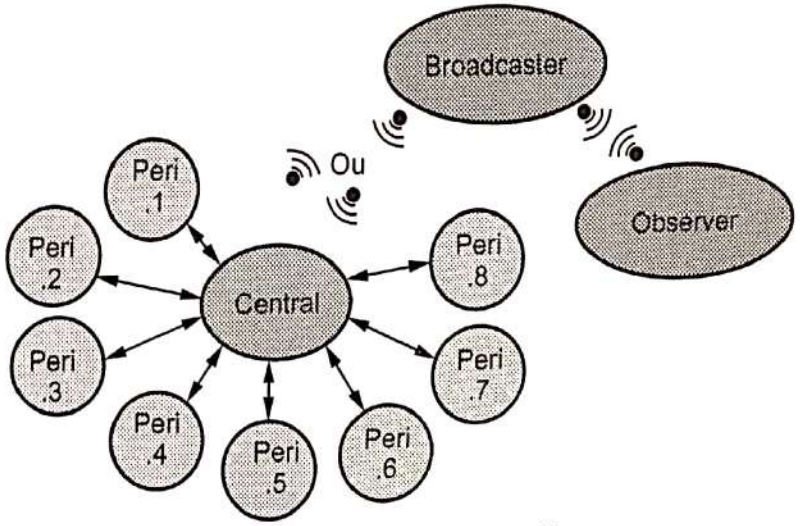


Fig. 3.2.1 : BLE connection

- A BLE connected item may have up to 4 different functions :
  1. The "Broadcaster" shall be used as a server. Thus, its purpose is to transfer data to a device on a regular basis, but it does not support any incoming connection.
  2. The "Observer" : In a second step, the device may only monitor and read the data sent by a "broadcaster". In such a case, the object is not able to send any connection to the server.
  3. The "Central" usually consists of a smartphone or tablet. This device provides two different types of connection : either in advertising mode or in connected mode. It is leading the overall process as it triggers data transfer.
  4. The "Peripheral" device allows connections and data transfer with the "Central" on a periodical basis. This system's goal is to ensure universal data transmission by using the standard process, so that other devices also may read and understand the data.

3.2.2 Components of BLE

- From architectural perspective, BLE has three components : application block, host and controller.
- Fig 3.2.2 shows BLE protocol stack.

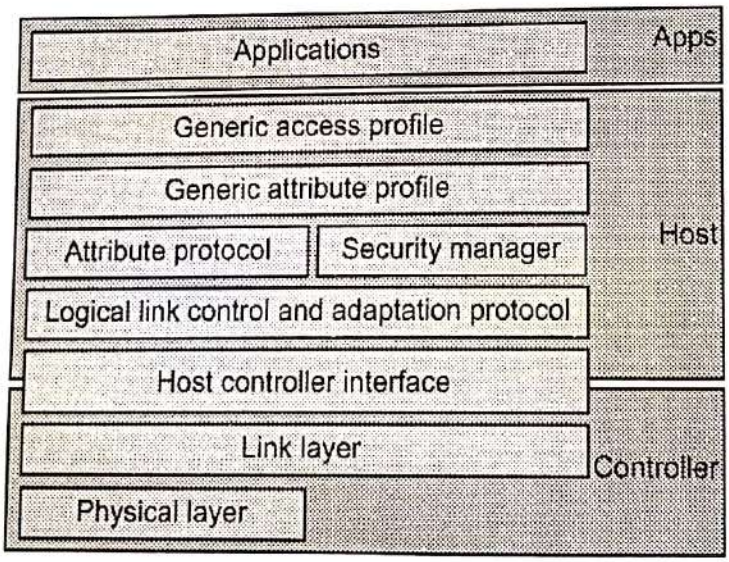


Fig. 3.2.2 : BLE protocol stack

1. Controller

- The controller includes the following layers :
  a) Host Controller Interface (HCI) - Controller side
  b) Link Layer
  c) Physical Layer (PHY)

a. **Host-Controller Interface (HCI) :** It provides communication between controller and host through standard interface types. This HCI layer can be implemented either using API or by interfaces such as UART/SPI/USB. Standard HCI commands and events are defined in the bluetooth specifications.

b. **Link Layer**

- Link Layer (LL) directly interfaces with the physical layer (PHY), and is usually implemented as a combination of custom hardware and software.
- It is responsible for advertising, scanning, and creating/maintaining connections. The role of BLE devices changes in peer to peer (i.e. Unicast) or broadcast modes. The common roles are Advertiser/Scanner (Initiator), Slave/Master or Broadcaster/Observer.

c. **Physical Layer**

- The physical (PHY) layer handles the operation involves analog communications. Specifically, it defines the modulation and demodulation of analog signals and applies source coding to transform the signals into digital symbols.
- The transmitter uses GFSK modulation and operates at unlicensed 2.4 GHz frequency band. Using this PHY layer, BLE offers data rates of 1 Mbps (Bluetooth v4.2)/2 Mbps (Bluetooth v5.0).
- It uses frequency hopping transceiver. Two modulation schemes are specified to deliver 1 Msym/s and 2 Msym/s.
- Two PHY layer variants are specified viz. uncoded and coded.
- A Time Division Duplex (TDD) topology is employed in both of the PHY modes.

2. Host

a. **The Generic Access Profile (GAP) :** It provides a framework that defines how BLE devices interact with each other. This includes : Roles of BLE devices, Advertisements, Connection establishment and Security. This layer directly interfaces with application layer and/or profiles on it. It handles device discovery and connection related services for BLE device. It also takes care of initiation of security features.

b. **The Generic Attribute Profile (GATT)** defines the format of the data exposed by a BLE device. It also defines the procedures needed to access the data exposed by a device. This layer is service framework which specifies sub-procedures to use ATT. Data communications between two BLE devices are handled through these sub-procedures. The applications and/or profiles will use GATT directly.

- There are two Roles within GATT : Server and Client. The Server is the device that exposes the data it controls or contains, and possibly some other aspects of its behaviour that other devices may be able to control. A Client, on the other hand, is the device that interfaces with the Server with the purpose of reading the Server's exposed data and/or controlling the Server's behaviour.

c. **Attribute Protocol (ATT) :** This layer allows BLE device to expose certain pieces of data or attributes.

d. **Security Manager :** This security Manager layer provides methods for device pairing and key distributions. It offers services to other protocol stack layers in order to securely connect and exchange data between BLE devices.

e. **Logical Link Control & Adaptation Protocol (L2CAP) :** This layer offers data encapsulation services to upper layers. This allows logical end to end data communication.

3. Application Layer :

- The BLE protocol stack layers interact with applications and profiles as desired. Application interoperability in the Bluetooth system is accomplished by Bluetooth profiles.
- The profile defines the vertical interactions between the layers as well as the peer-to-peer interactions of specific layers between devices.
- A profile composed of one or more services to address particular use case. A service consists of characteristics or references to other services.
- Any profiles/applications run on top of GAP/GATT layers of BLE protocol stack. It handles device discovery and connection related services for the BLE device.

3.2.3 BLE Topology

- Communication between BLE device and outside world is performed in two ways : broadcasting and connections.

1. **Broadcasting :** It consists of two devices i.e., broadcaster and observer. Broadcaster send data to all observer. (See Fig. 3.2.3 on next page.)
2. **Connections :** Connection is permanent and it send periodically data between two devices. It uses master-slave configuration. (See Fig. 3.2.4 on next page.)

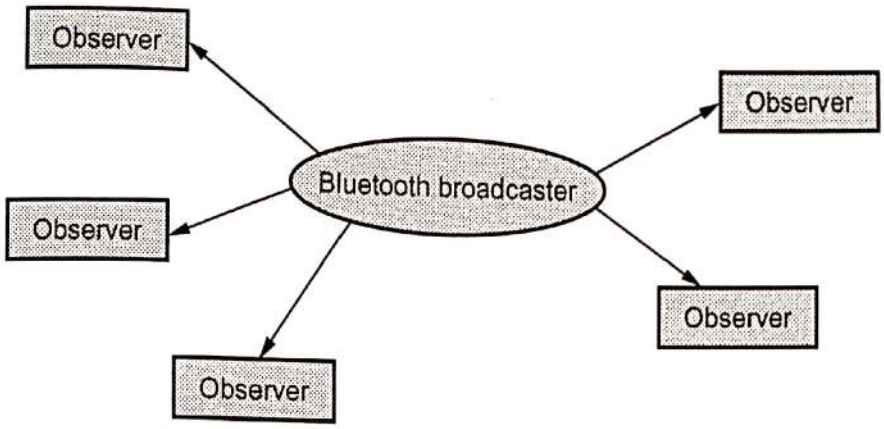


Fig. 3.2.3 : Broadcasting

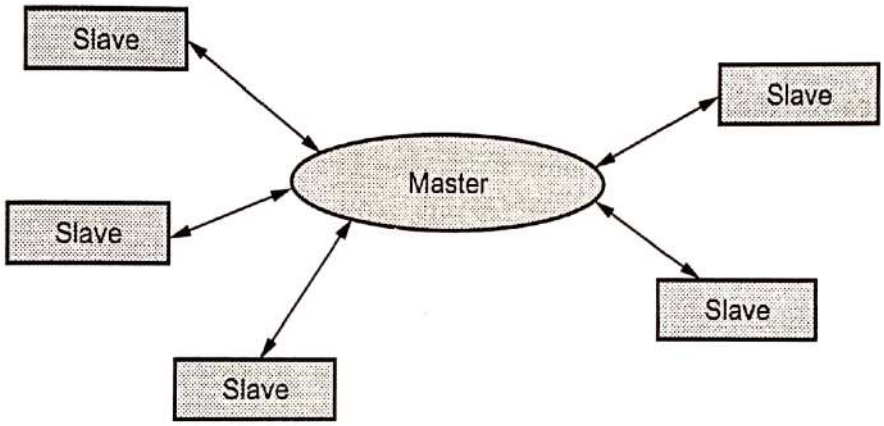


Fig. 3.2.4 : Connections

3.2.4 Comparison between Classic Bluetooth and Bluetooth Low Energy

Parameters	Classic Bluetooth	Bluetooth Low Energy
Distance/Range	100 m	50 m
Data Rate	1 - 3 Mbit/s	1 Mbit/s
Throughput	0.7 - 2.1 Mbit/s	0.27 Mbit/s
Number of Active Slaves	7	Not defined; implementation dependent
Security	56 / 128-bit and application layer user defined	128-bit AES with Counter Mode CBC-MAC and application layer user defined
Robustness	Adaptive fast frequency hopping, FEC, fast ACK	Adaptive frequency hopping, 24-bit CRC, Lazy Acknowledgement, 32-bit Message Integrity Check
Latency	100 ms	6 ms

Voice capable	Yes	No
Network topology	Scatternet	Star-bus
Peak current Consumption	Less than 30 mA	Less than 20 mA

3.2.5 Light Fidelity

- Light Fidelity (Li-Fi) is bidirectional wireless system that transmit data via LED o infrared light.
- Li-Fi potentially offers 10x the efficiency than traditional Wi-Fi, facilitatin high-speed data communication of up to 1 Gbps.
- The technology has become an international standard for wireless communicatio in its first version in November 2011 by the International Telecommunication Standardisation Committee.
- The LEDs can be used like lasers in optical telecommunication in order to transfe data. LED light sources present in our surroundings can therefore be used fo lighting but also used to transfer digital data.
- Li-Fi uses Light-Emitting Diodes (LEDs) bulbs, which flicker on and off at a ver high rate not noticeable to the human eye, as a medium to deliver high speed communication. Because LiFi uses light, it cannot penetrate walls and has limited working distance of typically a few meters.
- Fig. 3.2.5 shows working of Li-Fi.

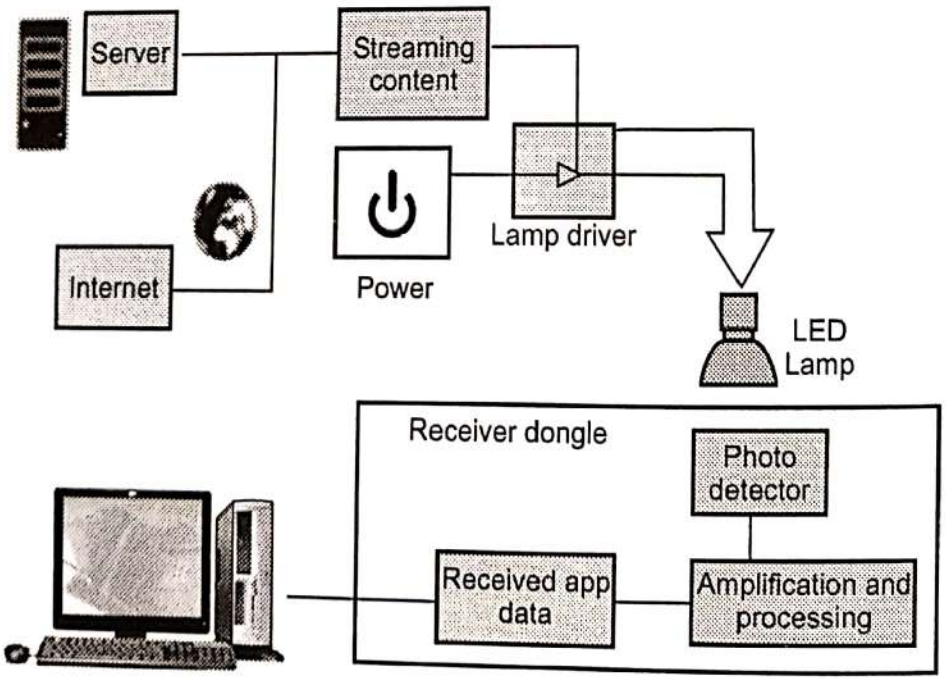


Fig. 3.2.5 : Working of Li-Fi

- Li-Fi is nothing but a Visible Light Communication (VLC) system which transmits data. It has two compsonents :
  a. At least one device with a photodiode which will be able to receive light signals and
  b. Light source equipped with a signal processing unit.
- Li-Fi technology requires LED a semiconductor light source, so that it can amplify light intensity and switch rapidly. Also, LED cells can modulate thousands of signals without the human eye ever noticing.
- The changes in light intensity from the LED light source are interpreted and converted as electrical current by the receiving photodiode device.
- Once the electronic signal is demodulated, it is converted into a continuous stream of binary data comprising of audio, video, web, and application information to be consumed by any internet-enabled device.
- Advantage
  1. No electromagnetic interference
  2. Fast and efficient
  3. Consumes less power
  4. Highly secure
  5. Low power consumption
  6. Unlimited bandwidth
- Disadvantages
  1. Cannot penetrate through walls
  2. Initial cost is high
  3. Shorter range
  4. Interferences from external light sources like sun, light, normal bulbs etc

3.2.6 Difference between Li-Fi and Wi-Fi

Li-Fi	Wi-Fi
Li-Fi transmits data using light with the help of LED bulbs.	Wi-Fi transmits data using radio waves with the help of WiFi router.
Cannot penetrate through walls	Wi-Fi can penetrate through walls
It covers distance of about 10 meters	Wi-Fi has a range of 30 meters
Power consumption is less	Power consumption is more

Speed of 1 Gbps in use commercially	Speed up to 2 Gbps can be achieved commercially
It uses standard IEEE 802.15.7	It uses standard IEEE 802.11
Uses visible light of electromagnetic spectrum	Uses radio waves of electromagnetic spectrum
Works in high dense environment	Works in less dense environment due to interference related issues

3.3 IPv4

- IPv4 is Internet Protocol version 4. It is the network layer protocol of the TCP/I protocol suite.
- IP is a connectionless, unreliable, best-effort delivery protocol. All the nodes a identified using an IP address. Packets are delivered from the source to t destination using IP address
- IPv4 addresses are encoded as a 32 bits field. IPv4 addresses are often represente in dotted-decimal format as a sequence of four integers separated by a dot.
- An IPv4 address is used to identify an interface on a router or a host. IP addresses are unique. Two devices on the internet can never have the sam address at the same time.
- The address structure was originally defined to have a two-level hierarchy **Network ID and host ID.**
- The network ID identifies the network the host is connected to. The host I identifies the network connection to the host rather than the actual host.

3.3.1 Packet Format

- The IP datagram consists of a header followed by a number of bytes of dat Fig 3.3.1 shows header format of IPv4. All IPv4 packets use the 20 bytes heade (See Fig. 3.3.1 on next page.)
- **Version field** indicates the version of IP used to build the header. Current versio is 4.
- **Header Length** indicates the length of the IP header in 32 bits words. This fie allows IPv4 to use options if required, but as it is encoded as a 4 bits field, th IPv4 header cannot be longer than 64 bytes.
- **Type of service :** The service type is an indication of the quality of servic requested for this IP datagram.

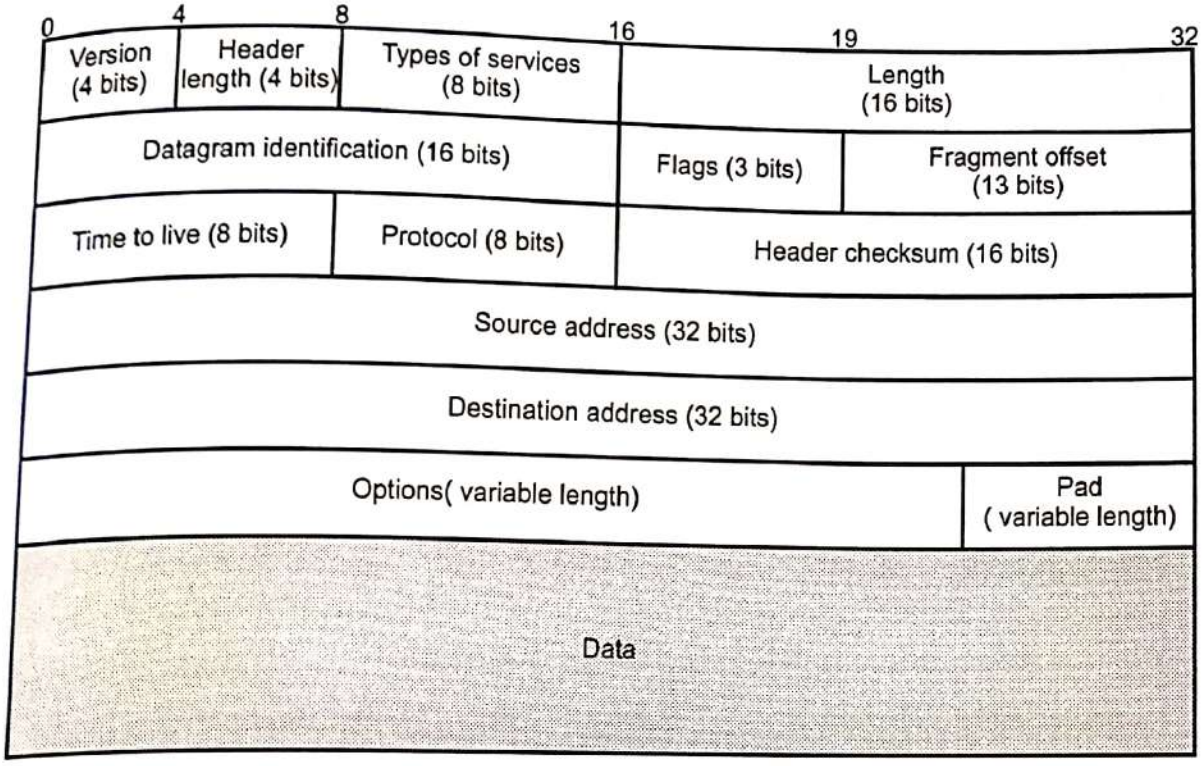


Fig. 3.3.1 : IPv4 header format

- **Protocol field** indicates the transport layer protocol that must process the packet's payload at the destination. Common values for this field are 6 for TCP and 17 for UDP
- **Length field** indicates the total length of the entire IPv4 packet (header and payload) in bytes. This implies that an IPv4 packet cannot be longer than 65535 bytes.
- **Fragment offset** is used to reassemble the full datagram. The value in this field contains the number of 64-bit segments (header bytes are not counted) contained in earlier fragments. If this is the first (or only) fragment, this field contains a value of zero.
- **Flags (3 bits) :** Only two of the bits are currently defined: MF(More Fragments) and DF(Don't Fragment).
- **More Fragments (MF) flag** is a single bit in the Flag field used with the Fragment Offset for the fragmentation and reconstruction of packets. The More Fragments flag bit is set, it means that it is not the last fragment of a packet. When a receiving host sees a packet arrive with the MF = 1, it examines the Fragment Offset to see where this fragment is to be placed in the reconstructed packet. When a receiving host receives a frame with the MF = 0 and a non-zero value in the Fragment offset, it places that fragment as the last part of the reconstructed packet.

- **Don't Fragment (DF) flag** is a single bit in the Flag field that indicates that fragmentation of the packet is not allowed. If the Don't Fragment flag bit is set, then fragmentation of this packet is NOT permitted. If a router needs to fragment a packet to allow it to be passed downward to the Data Link layer but the DF bit is set to 1, then the router will discard this packet.
- **Time-to-Live (TTL)** is an 8-bit binary value that indicates the remaining "life" of the packet. The TTL value is decreased by at least one each time the packet is processed by a router (that is, each hop).When the value becomes zero, the router discards or drops the packet and it is removed from the network data flow.
- **Source address** field that contains the IPv4 address of the source host
- **Destination address** field that contains the IPv4 address of the destination host
- **Checksum** that protects only the IPv4 header against transmission errors. Checksum is for the information contained in the header. If the header checksum does not match the contents, the datagram is discarded.
- **Options (variable) :** Encodes the options requested by the sending user.
- **Padding (variable) :** Used to ensure that the datagram header is a multiple of 32 bits.
- **Data (variable) :** The data field must be an integer multiple of 8 bits. The maximum length of the datagram (data field plus header) is 65,535 octets.

3.3.2 Classes of IP Address

- An IPv4 address is a 32-bit sequence of ones and zeros. To make the IP address easier to work with, it is usually written as four decimal numbers separated by periods.
- For example, an IP address of one computer is 192.168.1.2. This is called the dotted decimal format.
- Each part of the address is called an octet because it is made up of eight binary digits. For example, the IP address 192.168.1.8 would be 11000000.10101000.00000001.00001000 in binary notation.
- Address 0.0.0.0, 127.0.0.1 and 255.255.255.255 carries special meaning. IP address is divided into a network number and a host number. The network prefix identifies a network and the host number identifies a specific host.

- Example :

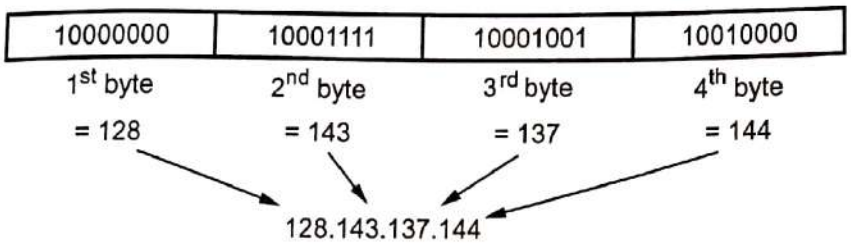


Fig. 3.3.2

- The IP address consists of a pair of numbers : **IP address = <network number><host number>**
- Example : Suppose IP address of technicalpublication.org is **129.144.136.146**, then network address is 129.144.0.0 and host number is 136.146.

Classes of IP Address

- IP address is divided into five classes: A, B, C, D, and E. Fig. 3.3.3 shows IP address classes with net ID and host ID.

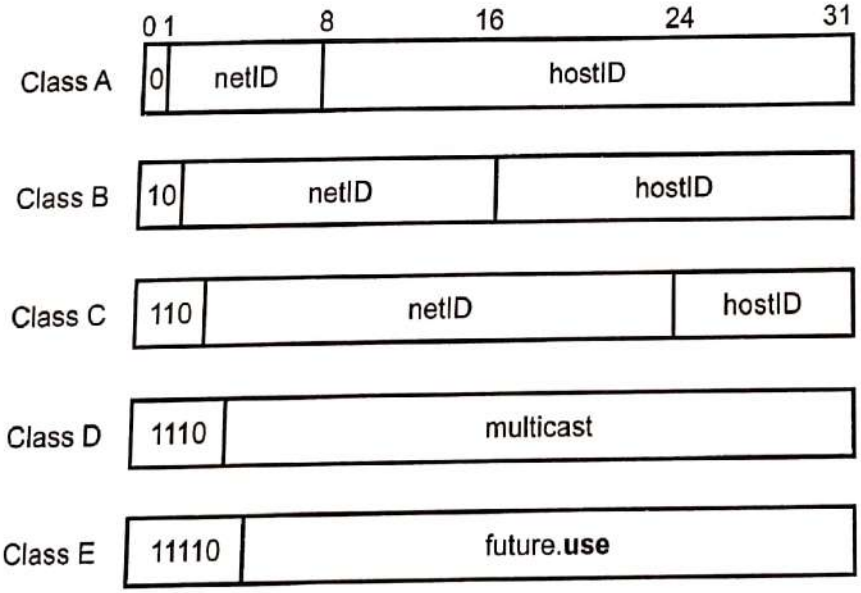


Fig. 3.3.3 : IP address classes with net ID and host ID

- The class of an IP address is identified in the most significant few bits. If the first bit is 0, it is a class A address.
- If the first bit is 1 and second is 0, it is a class B address. If the first two bits are 1 and third is 0, it is a class C address.
- Class D is used for what is known as multicasting, transmitting data to multiple computers or routers.
- Class E was reserved for future use, but this has given way to IPv6 instead

- **Range of IP address :**

Class	Starting address	Last Address
Class A	0.0.0.0	127.255.255.255
Class B	128.0.0.0	191.255.255.255
Class C	192.0.0.0	223.255.255.255
Class D	224.0.0.0	239.255.255.255
Class E	240.0.0.0	255.255.255.255

- **Loopback Testing :** The 127 network number isn't used by hosts as a *logical IP address*. Instead, this network is used for ***loopback IP addresses***, which allow for testing.
- Address 0.0.0.0, 127.0.0.1 and 255.255.255.255 carries special meaning.

3.3.3 Public and Private Addresses

- IPv4 addresses are further classified as either public or private. Public IP addresses are ones that are exposed to the Internet. Any other computers on the Internet can potentially communicate with them.
- Private IP addresses are hidden from the Internet and any other networks. They are usually behind an IP proxy or firewall device.
- Private Addresses are as follows :

Class	Starting Address	End of range
Class A	10.0.0.0	10.255.255.255
Class B	172.16.0.0	172.31.255.255
Class C	192.168.0.0	192.168.255.255

Subnetting

- Subnetting is the subdivision of logical IP network. By default, all computers are on one subnet or network with no divisions involved.
- Subnet mask of class A, B and C are as follows :

IP address Class	Subnet Mask
A	255.0.0.0
B	255.255.0.0
C	255.255.255.0

3.4 IPv6

- IPv4 provides the host to host communication between systems in the Internet.
- IPv4 has played a central role in the internetworking environment for many years. It has proved flexible enough to work on many different networking technologies.
- In the early 1990 the IETF began to work on the successor of IPv4 that would solve the address exhaustion problem and other scalability problems.
- IPv6 addresses are 128 bits in length. Addresses are assigned to individual interface on nodes, not to the node themselves.
- IPv6 addresses are assigned to interfaces, rather than to nodes, in recognition that a node can have more than one interface.
- A single interface may have multiple unique unicast addresses. The first field of any IPv6 address is the variable length format prefix, which identifies various categories of addresses.
- A new notation has been devised for writing 16-byte addresses. They are written as eight groups of four hexadecimal digits with colons between the groups, like this

  8000 : 0000 : 0000 : 0000 : 0123 : 4567 : 89AB : CDEF

Optimization

1. Leading zeros within a group can be omitted so 0123 can be written as 123.
2. One or more groups of 16 zero bits can be replaced by a pair of colons. The address new becomes

   8000 : : 123 : 4567 : 89AB : CDEF

Address Types

- IPv6 allows three types of addresses : Unicast, Anycast and Multicast.
- **Unicast :** An identifier for a single interface. A packet sent to a unicast address is delivered to the interface identified by that address.
- **Anycast :** An identifier for a set of interfaces. A packet sent to an anycast address is delivered to one of the interfaces identified by the address.
- **Multicast :** An identifier for a set of interfaces. A packet sent to a multicast address is delivered to all interfaces identified by that address.
- The first field of any IPv6 address is the variable-length format prefix, which identifies various categories of address.

3.4.1 Packet Format

- Fig. 3.4.1 the IPv6 datagram header format. Each packet is composed of a mandatory base header followed by the payload.
- The payload consists of two parts : Optional and data.

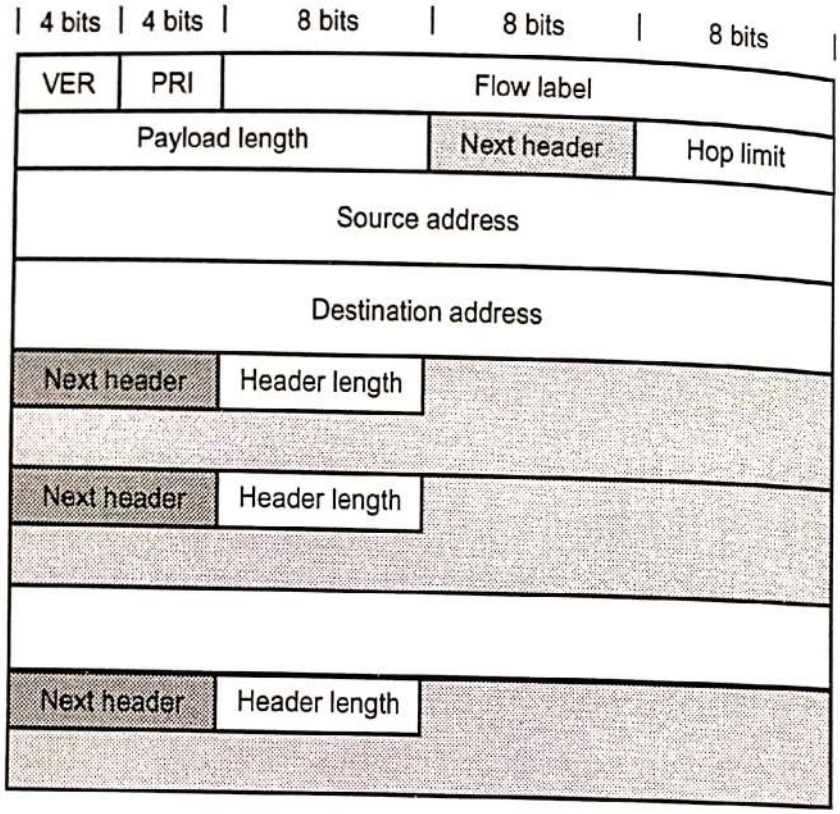


Fig. 3.4.1 : IPv6 header

1. Versions : This 4 bits field defines the version number of the IP. The value is 6 for IPv6.
2. Priority : The 4 bits priority field defines the priority of the packet with respect to traffic congestion.
3. Flow label : It is 24 bits field that is designed to provide special handling for a particular flow of data.
4. Payload length : The 16 bits payload length field defines the length of the IP datagram excluding the base header.
5. Next header : It is an 8 bits field defining the header that follows the base header in the datagram.
6. Hop limit : This 8 bits hop limit field serves the same purpose as the TTL field in IPv4.

7. Source address : The source address field is a 128 bits internet address that identifies the original.
8. Destination address : It is 128 bits Internet address that usually identifies the final destination of the datagram.

Priority :

- The priority field defines the priority of each packet with respect to other packets from the same source. IPv6 divides traffic into two broad categories
  1. Congestion controlled
  2. Noncongestion controlled
- If a source adapts itself to traffic slowdown when there is congestion, the traffic is referred to as congestion controlled traffic congestion controlled data are assigned priorities from 0 to 7.
- A priority of 0 is the lowest; a priority of 7 is the highest.
- Noncongestion controlled traffic refers to a type of traffic that excepts minimum delay. Discarding of packets is not desirable. Retransmission in most cases is impossible. Real time audio and video are examples of this type of traffic.
- Priority numbers from 8 to 15 are assigned to noncongestion controlled traffic.

3.4.2 Difference between IPv4 and IPv6

Sr. No.	IPv4	IPv6
1	Header size is 32 bits.	Header size is 128 bits.
2	It cannot support autoconfiguration.	Supports autoconfiguration.
3	Cannot support real time application.	Supports real time application.
4	No security at network layer.	Provides security at network layer.
5	Throughput and delay is more.	Throughput and delay is less.

3.4.3 Advantages of IPv6

1. Larger address space
2. Better header format
3. Security capabilities
4. Support for resource allocation
5. New options
6. Allowance for extension

3.5 URI

- Uniform Resource Identifier (URI) identifies the name and location of a file or resource in a uniform format. It includes a string of characters for the filename and may also contain the path to the directory of the file.
- A URI is a globally scoped character string that identifies an abstract or physical resource. This may be a digital resource on the Web Service, in which case the URI enables web clients, such as browsers and Web services, to interact with the resource. On the other hand, URIs may also be used more abstractly, to identify concepts or things, regardless of whether they have Web-accessible representations, as in formal ontologies.
- There are two subtypes of URI, URNs (Uniform Resource Name) and URLs (Uniform Resource Locator).
- The Uniform Resource Locator (URL) is a standard for specifying any kind of information on the Internet.
- URL has three parts :
  1. The protocol
  2. DNS name of the machine where the page is located.
  3. File name containing the page.

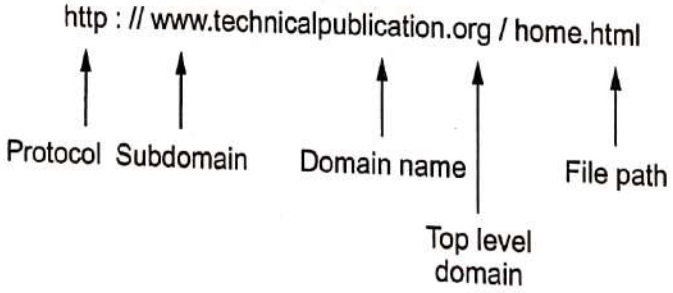


Fig. 3.5.1 URL

- The Uniform Resource Locator (URL) is a standard for specifying and kind of information on the Internet. The URL defines four things : protocol, host computer, port and path.
- The protocol is the client/server program used to retrieve the document. Many different protocols can retrieve a document; among them are FTP or HTTP. The most common today is HTTP.
- The host is the computer on which the information is located, although the name of the computer can be an alias.
- Web pages are usually stored in computers and computers are given alias names that usually begin with the characters "www".

- This is not mandatory, however, as the host can be any name given to the computer that hosts the Web page.
- The URL can optionally contain the port number of the server. If the port is included, it is inserted between the host and the path and it is separated from the host by a colon.
- Path is the pathname of the file where the information is located. Note that the path can itself contain slashes that, in the UNIX operating system, separate the directories from the subdirectories and files.
- A URI is an identifier of a specific resource. Like a page, or book, or a document.
- A URL is special type of identifier that also tells you how to access it, such as HTTPs, FTP, etc.-like https://www.google.com.

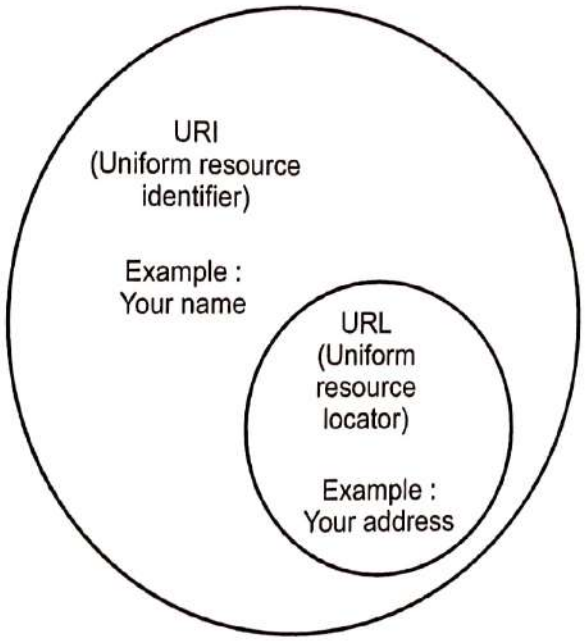


Fig. 3.5.2

3.6 Short Questions and Answers

Q.1 What is IP addressing ?

Ans. : An IP address is a numerical label assigned to each device in a computer network that uses Internet protocol for communication. Two important functions at IP address : 1) Host identification 2) Location addressing.

Q.2 What are the functions of the IP Protocol ?

Ans. :

1. Route IP data frames around an Internet. The IP protocol program running on each node knows the location of the gateway on the network. The gateway must then be able to locate the interconnected network. Data then passes from node to gateway through the Internet ;

4 Cloud for IoT

Syllabus

IoT and cloud, Fog computing, Security in cloud, Case study.

Contents

4.1 Introduction of Cloud Computing

- Cloud computing refer to a variety of services available over the Internet that deliver compute functionality on the service provider's infrastructure.
- Its environment (infrastructure) may actually be hosted on either a grid or utility computing environment, but that doesn't matter to a service user.
- Cloud computing is a general term used to describe a new class of network based computing that takes place over the Internet, basically a step on from Utility Computing.
- In other words, this is a collection/group of integrated and networked hardware, software and Internet infrastructure (called a platform).
- Cloud computing refers to applications and services that run on a distributed network using virtualized resources and accessed by common Internet protocols and networking standards.
- Fig. 4.1.1 shows cloud symbol. It denotes cloud boundary.

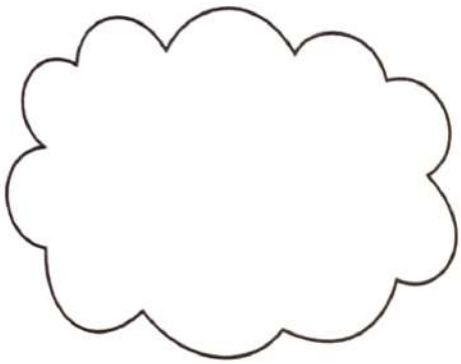


Fig. 4.1.1 : Cloud symbol

- IT resources include server, virtual server, storage device, networking device, services and software programs.
- An on-premise IT resource can access and interact with a cloud-based IT resource.
- An on-premise IT resource can be moved to a cloud, thereby changing it to a cloud-based IT resource.
- **Cloud Provider** : A person, organization, or entity responsible for making a service available to interested parties. When assuming the role of cloud provider, an organization is responsible for making cloud services available to cloud consumers, as per agreed upon Service Level Agreement (SLA) guarantees. Cloud provider have their own IT resources.

- **Cloud Consumer :** A person or organization that maintains a business relationship with, and uses service from, Cloud Providers. The cloud consumer uses a cloud service consumer to access a cloud service.
- **Cloud Service Owner :** The person or organization that legally owns a cloud service is called a cloud service owner. The cloud service owner can be the cloud consumer, or the cloud provider that owns the cloud within which the cloud service resides.
- **Resource Administrator :** Cloud resource administrator is the person or organization responsible for administering a cloud-based IT resource. The cloud consumer or cloud provider , or even third-party organization could be a cloud resource administrator

4.1.1 Cloud Components

- Cloud computing solutions are made up of several elements. Fig. 4.1.1 shows cloud components

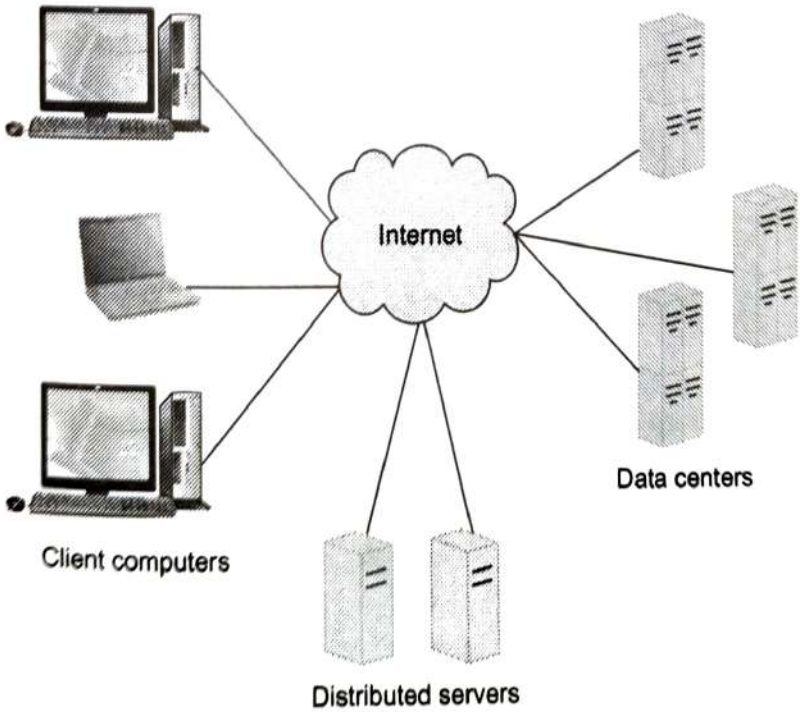


Fig. 4.1.2 : Cloud components

1. **Clients :** Mobile, terminals or regular computers.
2. **Benefits :** Lower hardware costs, lower IT costs, security, data security, less power consumption, ease of repair or replacement, less noise.
3. **Data centres :** Collection of servers where the application to subscribe is housed. It could be a large room in the basement of your building or a room full of servers on the other side of the world

4. **Virtualizing servers :** Software can be installed allowing multiple instances of virtual servers to be used and a dozen virtual servers can run on one physical server.

5. **Distributed servers :** Servers don't all have to be housed in the same location. It can be in geographically disparate locations. If something were to happen at one site, causing a failure, the service would still be accessed through another site. If the cloud needs more hardware, they can add them at another site.

4.1.2 Cloud Deployment models

- Cloud deployment models are refers to the location and management of the cloud's infrastructure.
- Deployment models are defined by the ownership and control of architectural design and the degree of available customization. Cloud deployment models are private, public and community clouds.
- Fig. 4.1.3 shows cloud deployment model.

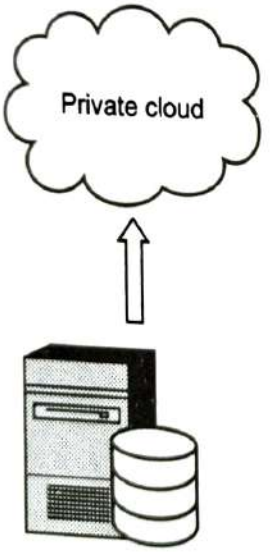


Fig. 4.1.3 (a) : Private cloud

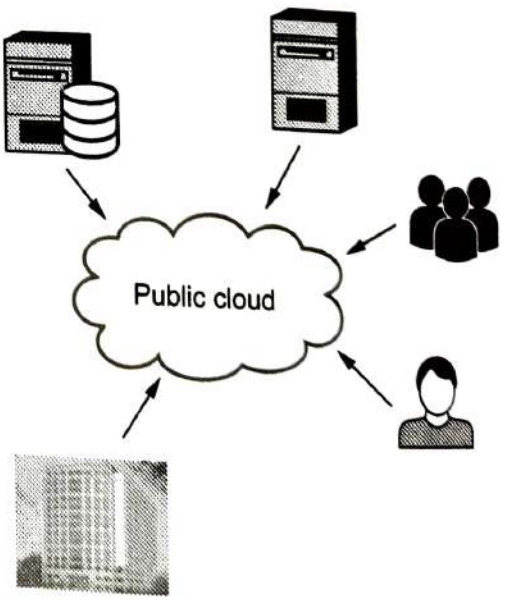


Fig. 4.1.3 (b) : Public cloud

1. Public Cloud :

- The cloud infrastructure is made available to the general public or a large industry group and is owned by an organization selling cloud services.
- Public Cloud is a huge data centre that offers the same services to all its users. The services are accessible for everyone and much used for the consumer segment.
- Examples of public services are Facebook, Google and LinkedIn
- Public cloud benefits :

  a) Low investment hurdle : Pay for what you use.

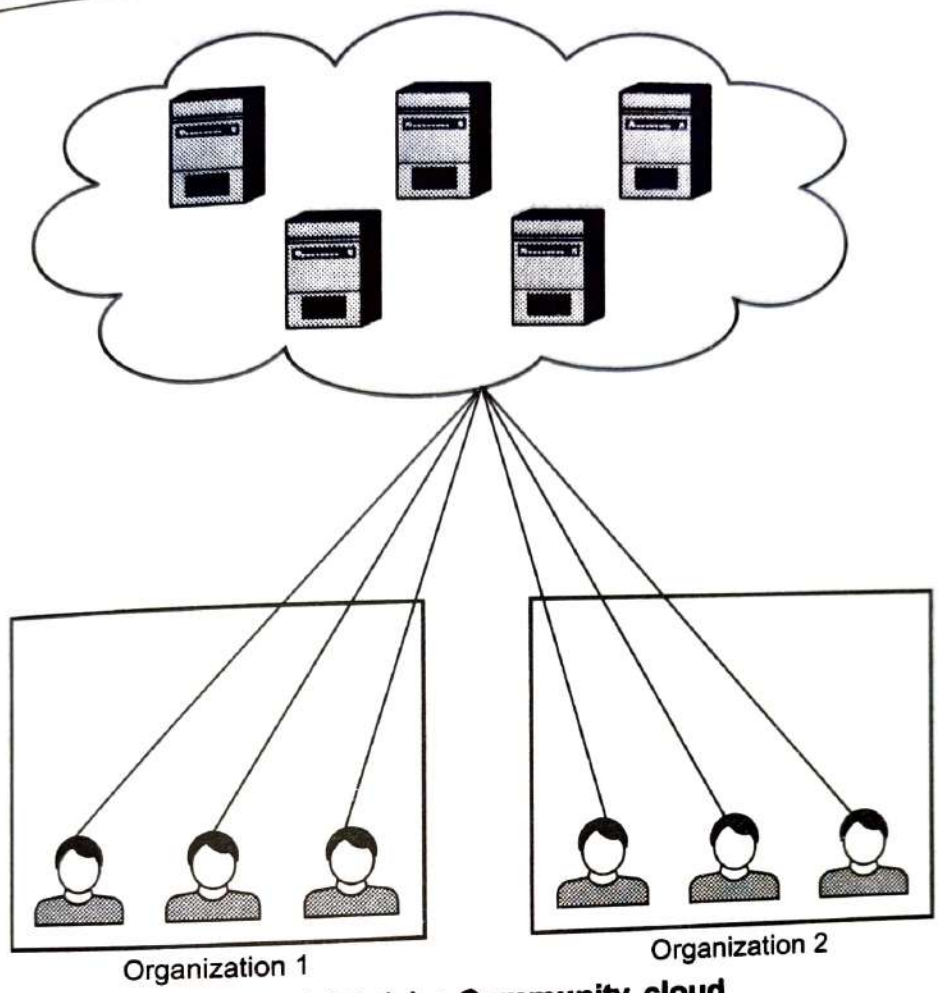


Fig. 4.1.3 (c) : Community cloud

b) Good test/development environment for applications that scale to many servers.

- **Public cloud risks :**
  a) Security concerns : Multi-tenancy and transfers over the Internet.
  b) IT organization may react negatively to loss of control over data center function.

2. **Private Cloud :**

- The cloud infrastructure is operated solely for a single organization. It may be managed by the organization or a third party and may exist on-premises or off-premises.
- **Private cloud benefits :**
  a) Fewer security concerns as existing data center security stays in place.
  b) IT organization retains control over data center.
- **Private cloud risks :**
  a) High investment hurdle in private cloud implementation, along with purchases of new hardware and software.

b) New operational processes are required; old processes not all suitable for private cloud.

3. Community Cloud :

- The cloud infrastructure is shared by several organizations and supports a specific community that has shared concerns (e.g. mission, security requirements, policy, or compliance considerations). It may be managed by the organizations or a third party and may exist on-premises or off-premises.

4. Hybrid Cloud :

- The cloud infrastructure is a composition of two or more clouds (private, community or public) that remain unique entities but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load-balancing between clouds).
- **Hybrid cloud benefits :**
  a) Operational flexibility : Run mission critical on private cloud, dev/test on public cloud
  b) Scalability : Run peak and bursty workloads on the public cloud
- **Hybrid cloud risks :**
  a) Hybrid clouds are still being developed; not many in real use.
  b) Control of security between private and public clouds, some of same concerns as in public cloud.

4.1.3 Difference between Public and Private Cloud

Public Cloud	Private Cloud
Public cloud infrastructure is offered via web applications and also as web services over Internet to the public.	Private cloud infrastructure is dedicated to a single organization.
Support multiple customer	Support dedicated customer
Full utilized of infrastructure.	Does not utilize shared infrastructure
Security is low as compared to private cloud	High level of security
Low cost	High cost
Azure, Amazon Web Services, Google App Engine and Force.com are a few examples of public clouds	An example of the Private Cloud is NIRIX's one Server with dedicated servers

4.1.4 Cloud Computing Challenges

1. Increased Security Vulnerabilities
2. Reduced Operational Governance Control
3. Limited Portability Between Cloud Providers
4. Multi-Regional Compliance and Legal Issues

- Use of cloud for business purpose means that the responsibility over data security becomes shared with the cloud provider. Organization extends their trust boundary to cloud consumer to external cloud.
- It is clear that the security issue has played the most important role in hindering cloud computing acceptance.
- Without doubt, putting your data, running your software on someone else's hard disk using someone else's CPU appears daunting to many.
- Well-known security issues such as data loss, phishing, pose serious threats to organization's data and software.
- Business concerns of cloud computing :

1. **Capacity planning** : Storage capacity is one of the main reasons for organization using cloud. Capacity planning is an unavoidable responsibility for most IT organizations.

- Future demands from business need to be planned for and accommodated. This can be very challenging because this involves estimating the usage and specially usage fluctuations over time.
- So there is constant need to balance peak usage requirements without unnecessarily over-spending on on-premise IT infrastructure.

2. **Cost reduction and operating overhead** : For any organization, initial investment of cloud is huge. The growth of IT environments often corresponds to the assessment of their maximum usage requirements. This can make the support of new and expanded business automations an ever-increasing investment
3. **Organizational agility** : From cloud perspective IT organizations, the IT resources needs to be more available and/or reliable than previously thought.

- The ability for an IT organization to be able to respond to these changes in capacity or availability helps to increase an organizational agility.

4.1.5 Cloud Applications

1. Through cloud cost flexibility, online marketplace gains access to more powerful analytics online. Cloud takes away the need to fund the building of hardware, installing software or paying dedicated software license fees.
2. Greater business scalability enables online video retailer to meet spikes in demand : Cloud enables businesses not just IT operations to add or provision computing resources just at the time they're needed.
3. Greater market adaptability provides online entertainment platform the ability to reach any type of customer device. A third of the executives we surveyed believe cloud can help them adapt to diverse user groups with a diverse assortment of devices.
4. Masked complexity enables access to services, no matter how intricate the technology they're built on.
5. With context-driven variability, "intelligent assistants" are possible. "Because of its expanded computing power and capacity, cloud can store information about user preferences, which can enable product or service customization," the report states.
6. Ecosystem connectivity enables information exchange across business partners.

4.2 Fog Computing

- **Edge computing :** it is also known as just "edge". It brings processing close to the data source, and it does not need to be sent to a remote cloud or other centralized systems for processing. Also called as "mist" computing.
- By eliminating the distance and time it takes to send data to centralized sources, we can improve the speed and performance of data transport, as well as devices and applications on the edge.
- **Fog computing** is a standard that defines how edge computing should work, and it facilitates the operation of compute, storage and networking services between end devices and cloud computing data centres.
- Examples : industrial controllers, switches, routers, embedded servers, and IoT gateways.

Fog nodes :

1. Receive feeds from IoT devices using any protocol, in real time.
2. Run IoT-enabled applications for real-time control and analytics, with millisecond response time.
3. Provide transient storage, often 1-2 hours.

4. Send periodic data summaries to the cloud.

- Additionally, many use fog as a jumping-off point for edge computing.
- With edge, compute and storage systems reside at the edge as well, as close as possible to the component, device, application or human that produces the data being processed.
- The purpose is to remove processing latency, because the data needn't be sent from the edge of the network to a central processing system, then back to the edge.
- The applications for edge make sense: Internet of Things-connected devices are a clear use for edge computing architecture.
- With remote sensors installed on a machine, component or device, they generate massive amounts of data.
- If that data is sent back across a long network link to be analyzed, logged and tracked, that takes much more time than if the data is processed at the edge, close to the source of the data.
- In essence, fog is the standard, and edge is the concept. Fog enables repeatable structure in the edge computing concept, so enterprises can push compute out of centralized systems or clouds for better and more scalable performance.

Characteristics of Fog computing :

1. Contextual location awareness and low latency.
2. Graphic distribution
3. Deployment near IoT endpoints.

Benefits of Fog Computing :

- **Greater business agility :** With the right tools, developers can quickly develop fog applications and deploy them where needed.
- **Better security :** Protect your fog nodes using the same policy, controls, and procedures you use in other parts of your IT environment. Use the same physical security and cyber security solutions.
- **Deeper insights, with privacy control :** Analyse sensitive data locally instead of sending it to the cloud for analysis.
- **Lower operating expense :** Conserve network bandwidth by processing selected data locally instead of sending it to the cloud for analysis.

4.3 Security in Cloud

- While cloud typically means outsourcing some or all of an organization's IT infrastructure, ultimately the organization is responsible for infrastructure security.
- Cloud security is more essential than ever as the number of attacks increases.
- Protecting the cloud starts at a secure architecture, which includes firewall placement and intrusion prevention systems.
- Organizations should practice compliance and due diligence to their countries' privacy standards.
- Monitoring and visibility into a cloud is key to detecting attacks in a timely manner.
- Authentication systems can act as the first line of defence to a potential attack.
- Identity Security : End-to-end identity management, third-party authentication services and identity must become a key element of cloud security. Identity security keeps the integrity and confidentiality of data and applications while making access readily available to appropriate users.
- Information Security : In the traditional data centre, controls on physical access, access to hardware and software and identity controls all combine to protect the data. In the cloud, that protective barrier that secures infrastructure is diffused.
- Infrastructure Security at the Network Level : When looking at the network level of infrastructure security, it is important to distinguish between public clouds and private clouds. With private clouds, there are no new attacks, vulnerabilities, or changes in risk specific to this topology that information security personnel need to consider.
- If public cloud services are chosen, changing security requirements will require changes to the network topology and the manner in which the existing network topology interacts with the cloud provider's network topology should be taken into account.
- Summary : in cloud computing, it is necessary to provide security to software, hardware, storage security and network security.

4.4 Case Study of Adafruit Cloud

- Adafruit Industries is an open-source hardware company based in New York City.
- The Adafruit cloud, Adafruit IO, is a cloud service primarily aimed at the maker market. Adafruit IO is an easy-to-use IoT platform that is useful for storing data, viewing data, and controlling devices.

- Adafruit.io is a web-based platform designed to help connect otherwise "dumb" devices to the rest of the internet.
- Adafruit IO is a cloud service built for IoT solutions. It provides two interfaces: MQTT and REST API.
- IO includes client libraries that wrap our REST and MQTT APIs.
- Feeds are the core of the Adafruit IO system. The feed holds metadata about the data you push to Adafruit IO. This includes settings for whether the data is public or private, what license the stored sensor data falls under, and a general description of the data. The feed also contains the sensor data values that get pushed to Adafruit IO from your device.
- You will need to create one feed for each unique source of data you send to the system. For example, if you have a project with one temperature sensor and two humidity sensors, you would need to create three feeds. One feed for the temperature sensor, and one feed for each humidity sensor.
- Dashboards allow you to visualize data and control Adafruit IO connected projects from any modern web browser. Widgets such as charts, sliders, and buttons are available to help you quickly get your IoT project up and running without the need for any custom code.

4.5 Short Questions and Answers

Q.1 What are the essential characteristics of cloud computing ?

Ans. : Five essential characteristics of Cloud Computing are on demand self-service, Broad network access, Resource pooling, Rapid Elasticity and Measured service

Q.2 Define NIST definition of cloud computing.

Ans. : NIST definition of cloud :Cloud computing is a pay-per-use model for enabling available, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, services) that can be rapidly provisioned and released with minimal management effort or service-provider interaction.

Q.3 What are the major characteristics of cloud computing as per NIST ?

Ans. : NIST five essential characteristics of Cloud Computing are as follows :

1. On demand self-service
2. Broad network access
3. Resource pooling
4. Rapid elasticity
5. Measured service

5 Application Building with IoT

Syllabus

Various application of IoT : Food, Healthcare, Lavatory maintenance, Water quality, Warehouse, Retail, Driver Assistance, Collision impact.

Contents

5.1 IoT Application in Food

- Many food and beverages are sensitive to temperature, and thus, reasonable control must be conducted to ensure the temperatures are maintained at the appropriate level. Failure to regulate the right temperature levels could cause food-borne diseases that might endanger the public.
- To eradicate the issue of unsafe food, companies are using smart thermostats to constantly monitor the temperature of the manufactured products in real-time.
- Real-time monitoring of the temperature means that if the temperature of the product falls below the set standard, that product is removed out of circulation to guarantee food safety.
- Integrated IoT systems are equipped with QR codes that customers can scan to confirm the safety of the product. This creates an assurance between the consumer and the manufacturing company that the food is safe for consumption.
- The food manufacturing industry requires speed and volume to be profitable. The powerful analysis and optimization tools IoT offers will allow machines to self-regulate and interact with other machines.
- Data no longer must be parsed and sorted by workers to create actionable insights, but can now be leveraged immediately in production.
- Major food sellers and distributers like walmart store their products in warehouses. As food demand increases, these companies stock their warehouses with these food to cater for the increasing demand.
- One challenge arises because it is difficult to monitor the movement of each product in real-time. Keeping an inventory of these fast-moving products becomes a challenge due to the large size of these warehouses.
- To enhance efficiency in inventory management, companies are using pressure-sensitive sensors to monitor the stock. The sensor sends alerts when the stock runs low.
- Companies can further integrate artificial intelligence with IoT to understand consumer purchasing habits that will facilitate in future planning.
- A fully integrated food manufacturing network will reduce downtime, alert workers of maintenance needs, and provide greater control than ever over quality control both in processing and during packaging and distributing.
- The food industry requires careful monitoring of all systems to work to be profitable. Balancing inventory with demand, ensuring consistent quality, and maintaining machine conditions are only a few of thousands if mini systems and processes that must work perfectly.

- IoT data from all these systems can be centralized and put where it's needed to allow for continuous improvement in food manufacturing.

5.2 Healthcare Application

- The World Health Organization (WHO) defines e-Health as : E-health is the transfer of health resources and health care by electronic means. It encompasses three main areas : The delivery of health information, for health professionals and health consumers, through the internet and telecommunications.
- E-health provides a new method for using health resources - such as information, money, and medicines - and in time should help to improve efficient use of these resources.
- E-Health brings special characteristics. The monitoring device's environment is a patient; a living and breathing human being. This changes some of the dynamics of the situation. Human interaction with the device means batteries could be changed, problems could be called in to technical support and possibly be resolved over the phone rather than some type of service call. In most cases, the devices on the patient are mobile not static with regard to location.
- Fig. 5.2.1 shows High Level e-Health ecosystem Architecture.

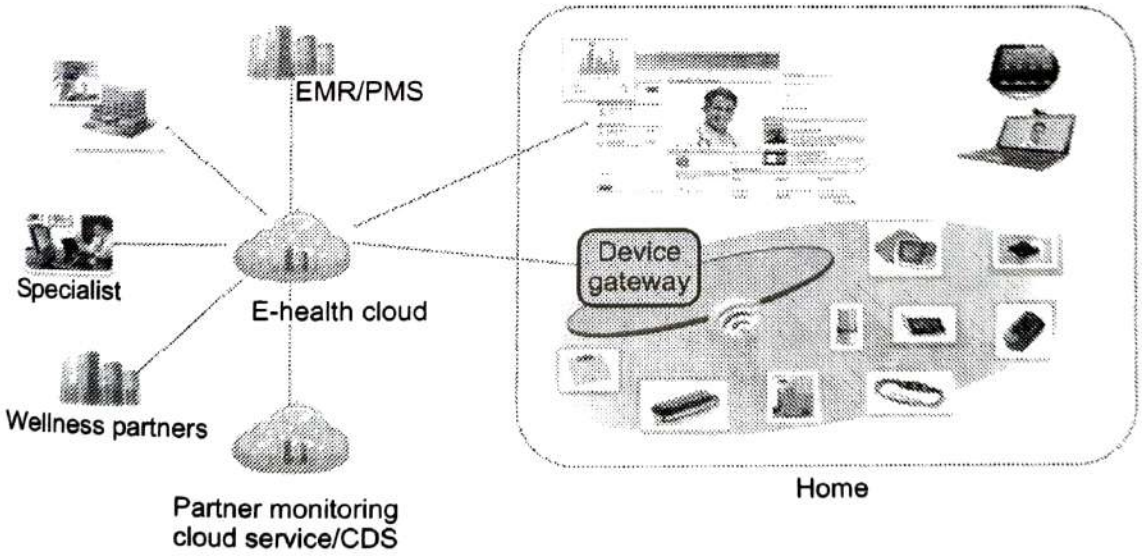
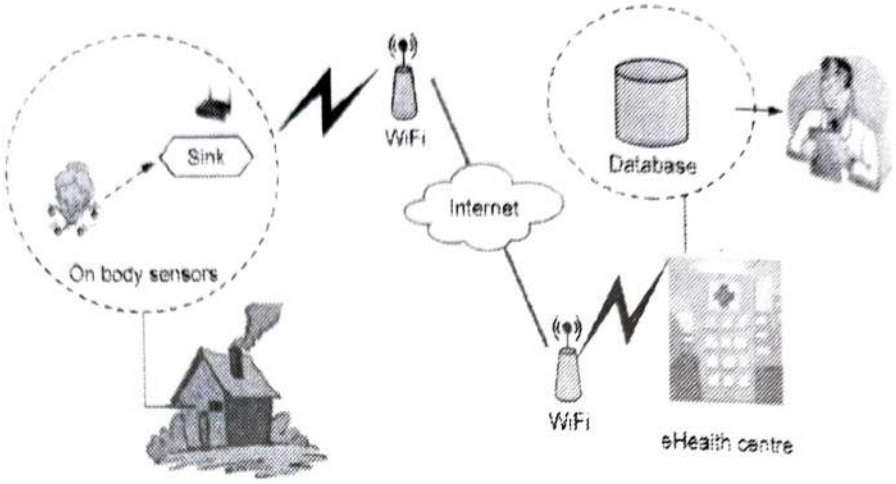


Fig. 5.2.1 : High Level e-Health ecosystem architecture

- The data flow architecture focuses on the source of the data, the destination the data and path the data. The source of the data is typically the sensor.
- The data can be either locally cached or is sent to the upstream systems without storing in the sensor. The path taken by the data includes a gateway, which can also cache some of the data and do distributed processing.

- Intermediate hubs can also store and process the data to filter out or make certain decisions. A distributed rules engine is used to make distributed decisions at the closest point of care. This enables data traffic to be filtered and processed efficiently without having every data being processed by the cloud service



- The development of wireless networks has led to the emergency of a new type of e-healthcare system, providing expert-based medical treatment remotely on time.
- With the e-healthcare system, wearable sensors and portable wireless devices can automatically monitor individuals' health status and forward them to the hospitals, doctors and related people.
- The system offers great conveniences to both patients and health care providers. For the patients, the foremost advantage is to reduce the waiting time of diagnosis and medical treatment, since they can deliver the emergent accident information to their doctors even if they are far away from the hospital or they don't notice their health condition.
- In addition, e-health system causes little interruption to patients' daily activities. For the health care providers, after receiving the abnormal signals from the patients, appropriate treatment can be made, which saves medical resources.
- Furthermore, without direct contact with medical facilities, medical personnel or other patients, the patients are unlikely to be infected with other diseases.
- However, to ensure the security and privacy of patients' medical records encounters a lot of challenges :
  1. How to achieve the confidentiality and integrity of patients' information,
  2. The security of wireless body area network,
  3. The privacy and unlink ability of patients' health status,
  4. The undeniability and unlinkability of doctors' treatment,

5. The location privacy of patients, the fine-grained access control of patients' medical record, the mutual authentication between patients and hospitals, etc.

- It would be useful to create an up-to-date bibliography on secure e-healthcare systems.

5.3 Lavatory Maintenance

- A main function of IoT is to collect data measured by sensors integrated with short range wireless networks such as bluetooth, zigbee, or Wi-Fi, which again transmit data to larger networks such as internet network gateways.
- IoT sensors provide low cost, scalable, efficient, low power, and integrated data through all sub-networks. As more sensors are incorporated and data collection period increases, the data becomes significantly large and hence the name "big data".
- Statistics show that a bathroom is one of the most hazardous places. Following sensors are available in market and we can use it in bathroom.

1. Leak Detection Sensor: It measures electronic resistance between two nodes. It can be installed on a bathroom floor to detect water leakage and flooding especially when an injured person is unable to move.
2. Digital Light / Lux Sensor: It measures intensity of light radiation. Light sensors detect most spectrum ranges including infrared ray, visible ray, and ultra-violet ray. The sensor can be used to detect human-presence and movement, total time spent in a bathroom, and identification and frequency of a person in a bathroom.
3. Voice Detection Sensor: It detects acoustic and noise signals. It is very useful as it can translate pre-programed messages (e.g., Help!) through a microcontroller in the case of medical emergency.
4. Pressure Sensor: It detects magnitude of contact-pressure. A simple pressure sensor is able to detect a person's status of using toilet and bath tub. A digital load cell sensor is same kind but provides more precise pressure readings that can be used for weight pattern analysis.
5. Positional Sensor (Gyro): It detects tilt, pitch, and inclination. These sensors can be used to detect falling, tilting, and locational changes for installed objects such as bath tub, shower room, and bathroom floor.
6. Motion Sensor: It detects movement of people in a bathroom. Mobility in a bathroom can provide pattern of bathroom usage and alarm for emergency situation.

7. Water Flow Sensor: Water leakage detection and measurement of water usage are important data for saving water and monitoring health conditions. Unusual water usage may indicate an emergency situation such as unconsciousness before finishing a shower or unnoticed internal plumbing leakage.
8. Energy Harvesting Sensor: One of key requirements for IoT sensors is low power consumption to minimize maintenance of those sensors. Energy harvesting sensors generate small amount of electronic energy enough for their operation from the sensor device or adjacent energy harvesting medium such as sunlight, pressure, and hydraulic flow.

Example : IoT based aircraft lavatory cleanliness monitoring system

- The aircraft lavatories are cleaned before take-off, after landing and during flight. For Aircraft Lavatory Cleanliness Monitoring System, we require two type of sensors: gas sensor (MQ-02) and infrared sensor.
- Fig. 5.3.1 shows lavatory cleanliness monitoring system

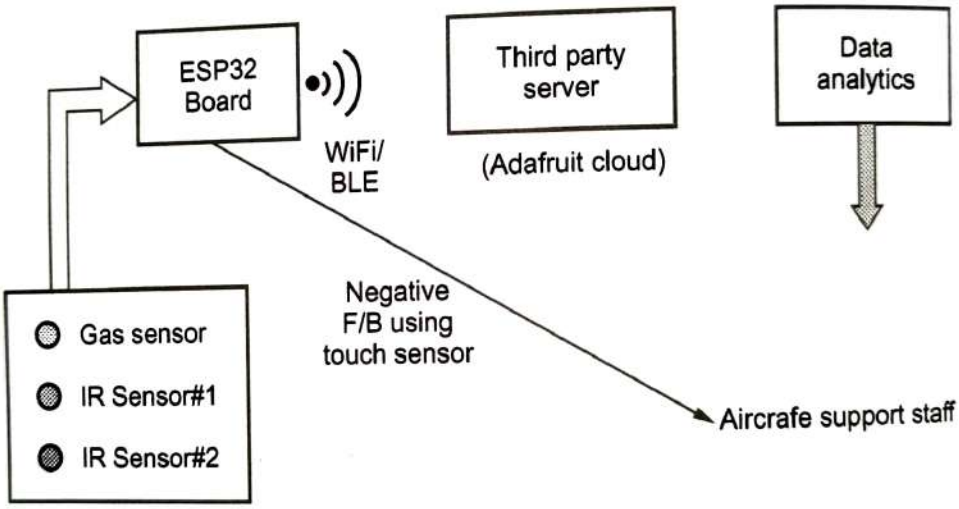


Fig. 5.3.1 Lavatory cleanliness monitoring system

- Gas sensor is used to check methane level of lavatory compartment. Threshold methane level for clean lavatory is identified and used as reference.
- The smell sensor is used to detect any unwanted gases present in the toilet. If any foul smell goes into the sensor, it creates a signal. All the signals are passed through the NodeMCU microcontroller where the constraints of foul smell and turbid water are checked.
- MQ2 gas sensor is an alcohol gas sensor which can detect the presence of gases which contain alcohol traces in them. It is made out of tin in the form of stannic oxide. It can detect alcohol, ethanol and smoke.
- IR sensor are used for passenger feedback: positive feedback and negative feedback.

- These sensors are connected to microcontroller board using wifi and BLE. The collected sensor data is quickly uploaded to the adafruit cloud storage before take-off and after landing at the airports.
- Based on comparison between collected real-time sensor data and threshold value of gas sensor aircraft support staff is alerted for cleaning the lavatories. The alert is sounded when real-time methane level exceeds the threshold value.

5.4 Water Quality

- In conventional systems, the monitoring process involves the manual collection of sample water from various regions, followed by laboratory testing and analysis. This process is ineffective, as this process is arduous and time-consuming and it does not provide real-time results.
- The quality of water should be monitored continuously, to ensure the safe supply of water from any water bodies and water resources. Hence, the design and development of a low-cost system for real-time monitoring of water quality using the Internet of Things (IoT) is essential.
- Monitoring water quality in water bodies using Internet of Things helps in combating environmental issues and improving the health and living standards of all living things.
- The proposed system monitors the quality of water relentlessly with the help of IoT devices, such as, NodeMCU. The in-built Wi-Fi module is attached in NodeMCU which enables internet connectivity transfers the measured data from sensors to the Cloud.
- Fig. 5.4.1 shows water monitoring system.

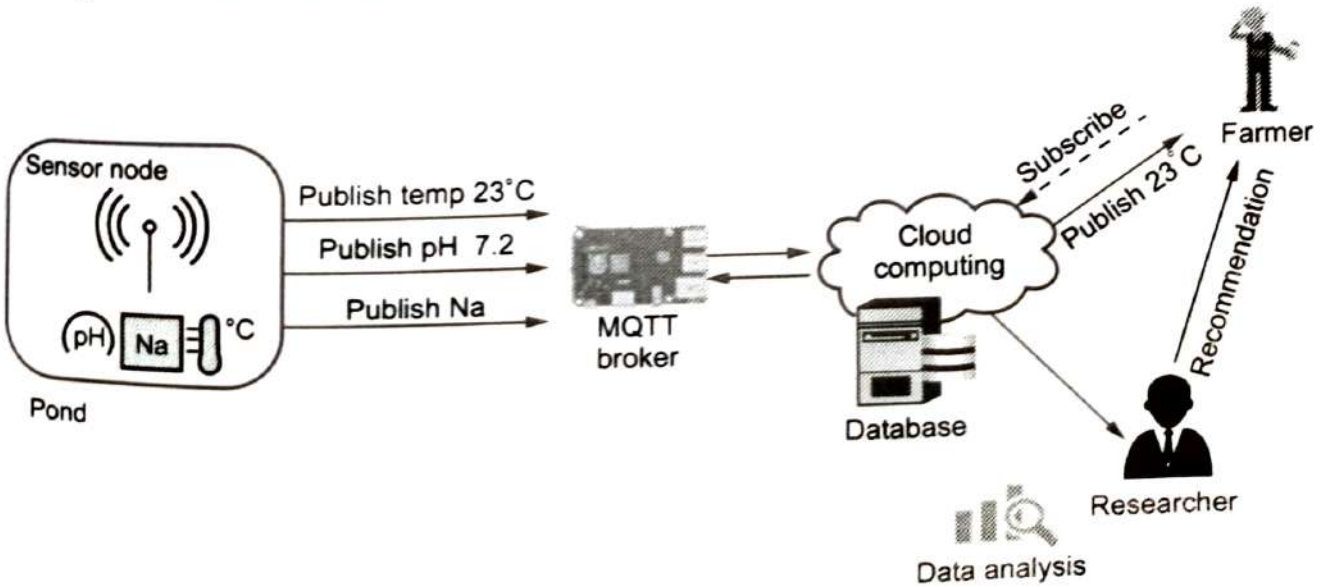


Fig. 5.4.1 water monitoring system

- The system mainly consists of sensor node as publishers, and Raspberry pi MQTT broker, and mobile client devices as subscribers. The sensor nodes are built with small embedded devices, LoRa wireless interface, and water quality sensors, i.e. water temperature sensor, pH sensor, and salinity sensor.
- Water quality sensor : The pH meter is used for the quality check if water is safe for use.
- Temperature sensor : A device which gives temperature measurement as an electrical signal is called as temperature sensor. This electrical signal will be in the form of electrical voltage and is proportional to the temperature measurement.
- Water level sensor : This sensor will help us decide if we have enough quantity of water to be supplied. An ultrasonic wave is triggered from the sensor and distance to target is determined by calculating the time required after the echo is returned. The sensor emits a high-frequency pulse, generally in the 20 kHz to 200 kHz range, and then listens for the echo.
- Sensor node technical specification is listed below :

Microcontroller	Arduino MEGA 2560
Wireless Interface	LoRa Shield with 915 MHz Antenna
Sensors	Water Temperature, Salinity, pH
Battery	12 V 18AH Rechargeable Sealed Lead Acid
Solar Cell	20 WP 12 V
Packet Size	17 bytes
Transmission Interval	60 seconds

5.5 Warehouse

- Today, warehouses are more than storage and inventory facilities. Many organizations are therefore investing in IoT-enabled warehouses for better Automated Control Systems (ACS) and Warehouse Management Systems (WMS) to improve their operational efficiency by reducing costs.
- IoT enabled warehouses gives businesses real-time data on product locations, transportation details, packaging, and routing. Due to these instant updates, store managers ensure no inventory is lost during transportation. Also, they ensure supply chain vendors manage deliveries responsibly.
- RFID technology enables businesses to track goods that enter and leave the warehouse in real-time.

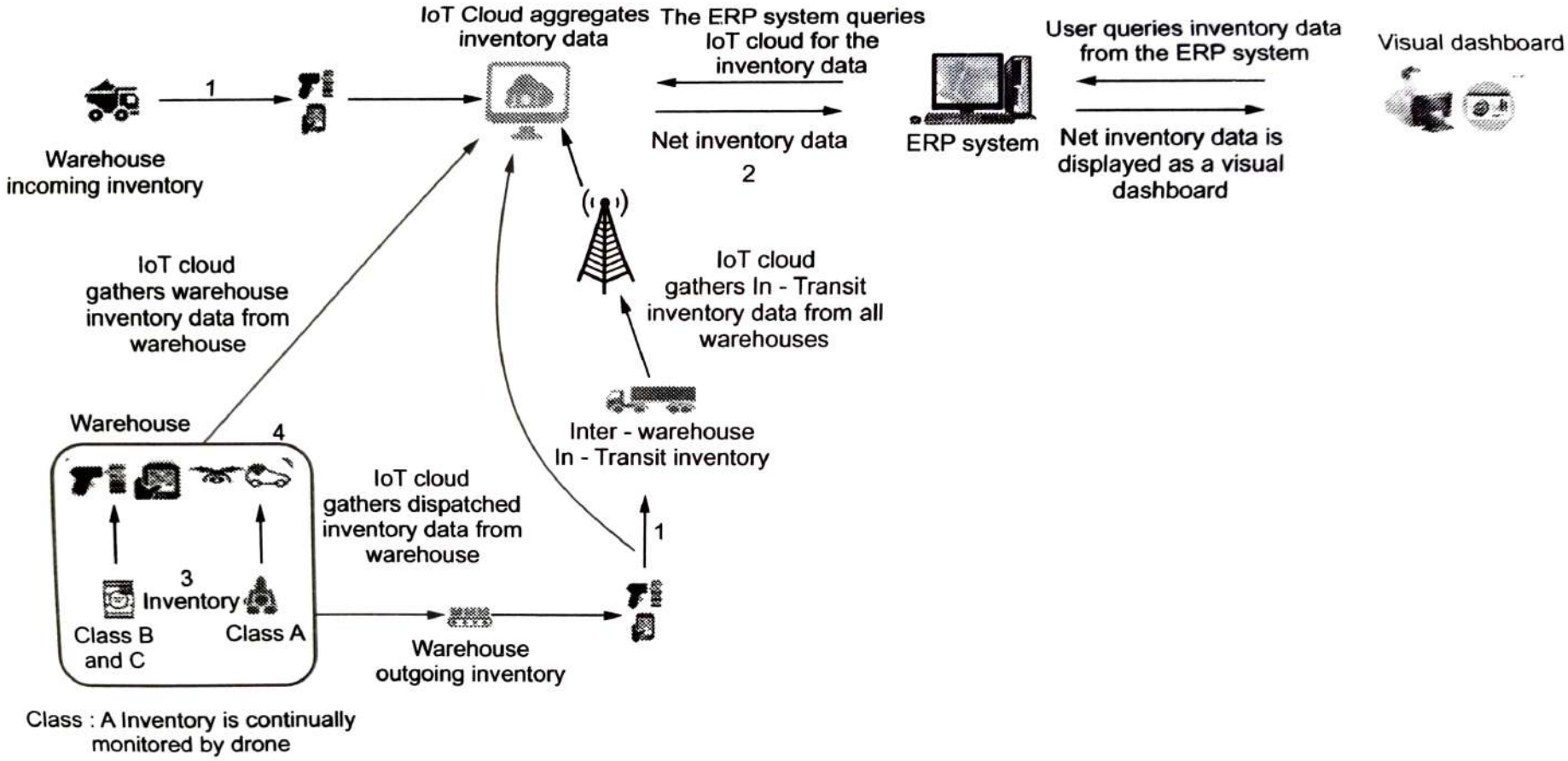

IoT Cloud aggregates inventory data
The ERP system queries IoT cloud for the inventory data
User queries inventory data from the ERP system
Visual dashboard
1
Warehouse incoming inventory
Net inventory data
2
ERP system
Net inventory data is displayed as a visual dashboard
IoT cloud gathers warehouse inventory data from warehouse
IoT cloud gathers In - Transit inventory data from all warehouses
Warehouse
4
Inter - warehouse In - Transit inventory
IoT cloud gathers dispatched inventory data from warehouse
1
3
Inventory
Class B and C
Class A
Warehouse outgoing inventory
Class : A Inventory is continually monitored by drone

Fig. 5.1.1

- Using IoT devices in warehouses enables businesses to detect risk and avoid accidents that can create disruptions in the supply chain. IoT sensors in warehouses can monitor temperature conditions. In addition, data gathered from vehicles, shipping conveyances and products can be combined to reduce counterfeiting, theft, and spoilage.
- Fig. 5.5.1 shows.
  1. Real time data received from sensors/beacons ensure that actual inventory levels are measured.
  2. Real-time updates on inventory count, alerts for mismatch between inventory data in ERP and actual inventory in warehouse.
  3. Restocking process becomes more efficient and stock out losses are avoided
  4. Automated navigation within warehouse
  5. End-to-End visibility on inventory

5.6 Retail

5.6.1 Inventory Management

- Retail involves the sale of goods from a single point (malls, markets, department stores etc) directly to the consumer in small quantities for his end use.
- Retail is a challenging business but the pressures of today's economic conditions are resulting in even more selective consumer shopping and spending.
- The effect of internet of things on inventory management is the next huge thing in progress when it comes to Business Process Management (BPM).
- In any typical business, the process of ordering, storing, tracking and managing good is a day to day requirement. As with all high investment top-tier businesses, this process becomes more complex with increasing amount of supply and demand.
- This process involves huge transaction of monetary resources and hence it is impervious that a high preference is given to this in a BPM. Inventories that are mismanaged can create significant financial problems for a business, leading to a inventory shortage.
- Existing technologies such as bar coding and Radio-Frequency Identification (RFID) already let retailers monitor their inventories.
- IoT will enable this to be taken to the next level with significantly more data coming in the monitoring systems and products moving through the supply chain.

- This can considerably improve supply chain efficiencies and enable leaner inventories. Large retailers such as **Walmart** are already using IoT for supply chain and inventory management.
- Tracking is done using RFID readers attached to the retail store shelves.

5.6.2 Smart Payments

- Smart payment system uses Near Field Communication (NFC) and bluetooth communication.
- Near Field Communication (NFC) technology is a standards-based wireless communication technology that allows data to be exchanged between devices that are a few centimeters apart.
- NFC operates at 13.56 MHz and transfers data at up to 424 Kbits/second.
- NFC is available as standard functionality in many mobile phones and allows consumers to perform safe contactless transactions, access digital content, and connect electronic devices simply.
- An NFC chip in a mobile device can act as a card or a reader or both, enabling consumer devices to share information and to make secure payments quickly.
- Using smart phone applications, payments can be made using a simple tap or waving the card within the proximity.
- Service providers can integrate payment option into smart phones using an NFC tag embedded inside the device. Apple pay, google wallet (android pay) and samsung pay are the most popular among smart phone payment systems.
- Data transfer using smart device are possible using NFC technology like android beam. Two users can share documents, photos, resumes and business cards by just waving their smart phone.

5.6.3 Smart Vending Machines

- Smart vending is about building remote management systems and telemetry tools, which integrate monitoring, transmission and delivery of operational data from each vending machine via the Internet.
- Smart vending Solution offers its customer's flexible payment options and monitors the machines remotely and in real time.
- Smart phone applications that communicate with smart vending machine allow user preferences to be remembered and learned with time.
- For instance, innovations like RFID based "smart" shelves continuously scan items on the shelf and notifies the appropriate systems. During low or out of stock

situations they create automatic replenishment alerts and send automatic orders directly to central warehouse and to manufacturers.

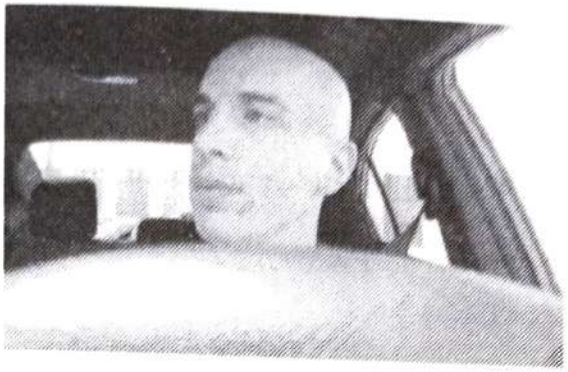
- Smart vending Machine provided following :
  1. Achieve high levels of efficiency in the management of their assets;
  2. Offers its customer's flexible payment options : RFID/NFC card; Mobile payments; Smartphone payments; Cash; Debit and Credit card;
  3. Monitor the machines remotely and in real time;
  4. Simplifies business since the vending machines contain multiple sensors that alert the owners about their location, the state inventory and eventual maintenance issues

5.7 Driver Assistance

- At present time, drowsy driving has become one of the major issues of the traffic collision. According to statistics, a large number of road accidents occur due to drowsy driving which results in severe injuries and deaths.
- Three techniques are used to detect the drowsiness of commercial drivers: recognizing the driver's eyes through cameras and using biosignals such as breathing, temperature, and heart rate to analyze operation patterns, such as the abnormal use of pedals and steering wheels.
- The term "drowsy" is synonymous with sleepy, which simply means an inclination to fall asleep. The stages of sleep can be categorized as awake, Non-Rapid Eye Movement sleep (NREM), and Rapid Eye Movement Sleep (REM).
- The crashes that occur due to driver drowsiness have a number of characteristics:
  a. Occur late at night (0:00 am-7:00 am) or during mid-afternoon (2:00 pm-4:00 pm)
  b. Involve a single vehicle running off the road
  c. Occur on high-speed roadways
  d. Driver is often alone
  e. Driver is often a young male, 16 to 25 years old
  f. No skid marks or indication of braking

Drowsiness detector algorithm :

1. **setup a camera that monitors a stream for faces :**



2. **If a face is found, we apply facial landmark detection and extract the eye regions :**

3. **Now that we have the eye regions, we can compute the eye aspect ratio to determine if the eyes are closed :**

- Each eye is represented by 6 (x, y)-coordinates, starting at the left-corner of the eye (as if you were looking at the person), and then working clockwise around the remainder of the region:

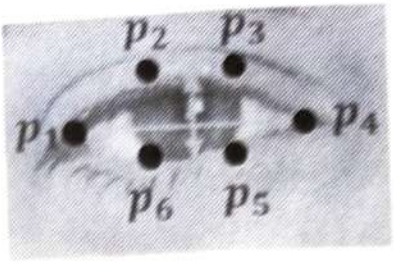
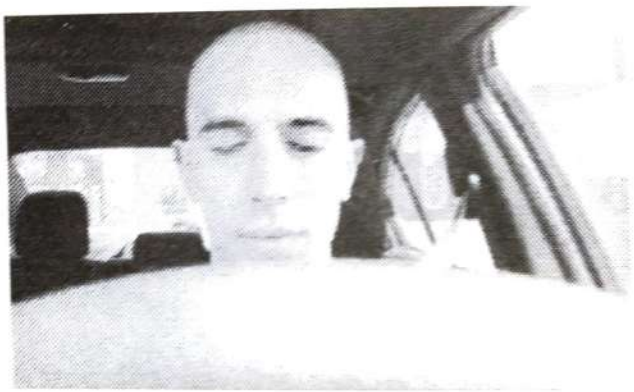


Fig. 5.7.1 Eye Aspect Ratio (EAR)

$$EAR = \frac{|p_2 - p_6| + |p_3 - p_5|}{2|p_1 - p_4|}$$

Where p1, ..., p6 are 2D facial landmark location

4. If the eye aspect ratio indicates that the eyes have been closed for a sufficiently long enough amount of time, we'll sound an alarm to wake up the driver.



- The system analyses the driver's consistency while driving in the form of various body movements, posture, steering input given by the driver on a certain time interval. Considering these scenarios, the system starts analyzing the other parameters like-speed, weather conditions, humidity, etc.
- A camera that is installed in the device is used to repeatedly records the facial behavioral landmark and movement of eyes and lips of the driver. Because of the eye closure period for sleepy drivers are longer than normal eye blinking.
- Through that live video streaming, a frame is extracted for image processing. Images are captured typically at a fix frame rate of 20fps.
- Using the image and annotation data set, the system understands the position the driver is feeling sleepy by measuring the coordinates of the right and left eye, nose, mouth, left and right ear brow. The human visual context somehow represents the feature of the scene with a few valuable information in it.

6 Arduino and Raspberry Pi

Syllabus

Arduino : Architecture, Programming and Application.
Raspberry Pi : Architecture, Programming and Application.

Contents

6.1 Arduino Architecture

- Arduino is an open-source electronics platform based on easy-to-use hardware and software.
- Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online.
- The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

Features :

- Support fast computations, ARM based MCU
- AVR micro-controller clock is ATSAMSX8I
- Operating input voltages is 3.3 Volt
- It uses EEPROM, SRAM and Flash memory
- It also support USB and UART
- Fig. 6.1.1 shows Arduino board.

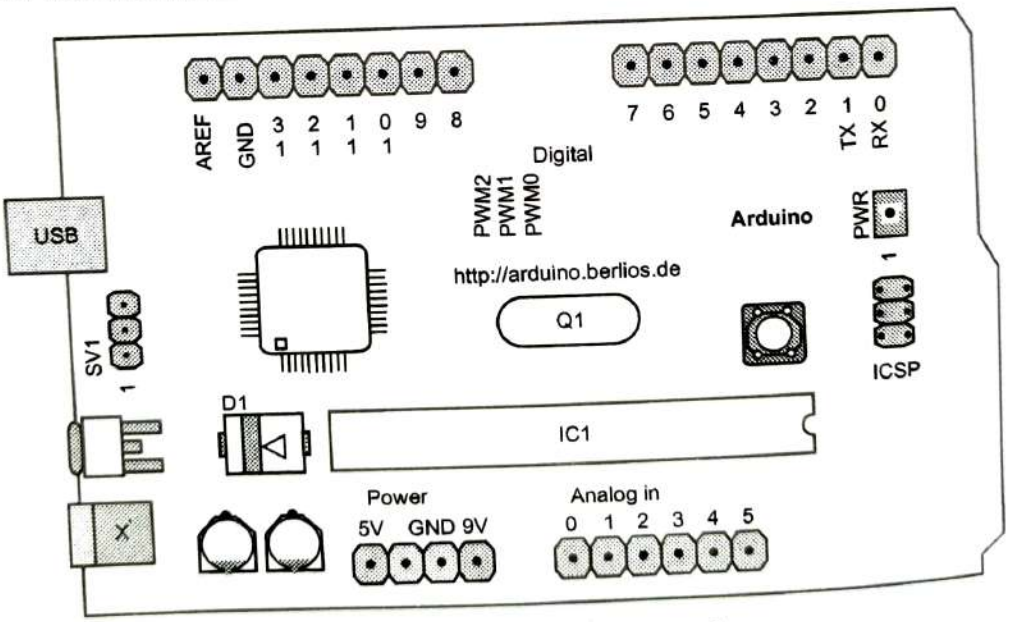


Fig. 6.1.1 : Arduino Board

Starting clockwise from the top center,

- Analog Reference pin (1st pin)
- Digital Ground
- Digital Pins 2 - 13 (green)
- Digital Pins 0-1/Serial In/Out - TX/RX : These pins cannot be used for digital I/O (digital Read and digital Write) if you are also using serial communication

- Reset Button - S1
- In-circuit Serial Programmer
- Analog In Pins 0 - 5
- Power and Ground Pins
- External Power Supply In (9-12VDC) - X1
- Toggles External Power and USB Power (place jumper on two pins closest to desired supply) - SV1 (purple)
- USB

Digital Pins

- In addition to the specific functions listed below, the digital pins on an Arduino board can be used for general purpose input and output via the pinMode( ), digitalRead( ), and digitalWrite( ) commands.
- Each pin has an internal pull-up resistor which can be turned on and off using digitalWrite( ), when the pin is configured as an input. The maximum current per pin is 40 mA.
- **Serial :** 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. On the Arduino Diecimila, these pins are connected to the corresponding pins of the FTDI USB-to-TTL Serial chip. On the Arduino BT, they are connected to the corresponding pins of the WT11 Bluetooth module. On the Arduino Mini and LilyPad Arduino, they are intended for use with an external TTL serial module (e.g. the Mini-USB Adapter).
- **External Interrupts :** 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
- **PWM :** 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function. On boards with an ATmega8, PWM output is available only on pins 9, 10, and 11.
- **BT Reset :** 7. (Arduino BT-only) Connected to the reset line of the bluetooth module.
- **SPI :** 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.
- **LED :** 13. On the Diecimila and LilyPad, there is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

Analog Pins

- In addition to the specific functions listed below, the analog input pins support 10-bit analog-to-digital conversion (ADC) using the analogRead() function.
- Most of the analog inputs can also be used as digital pins: analog input 0 as digital pin 14 through analog input 5 as digital pin 19. Analog inputs 6 and 7 (present on the Mini and BT) cannot be used as digital pins.
- **I2C : 4** (SDA) and 5 (SCL). Support I2C (TWI) communication.

Power Pins

- VIN (sometimes labelled "9 V"). The input voltage to the Arduino board when it's using an external power source.
- You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin. Note that different boards accept different input voltages ranges, please see the documentation for your board. Also note that the LilyPad has no VIN pin and accepts only a regulated input.
- **5 V :** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5 V supply.
- **3V3 :** (Diecimila-only) A 3.3 volt supply generated by the on-board FTDI chip.
- **GND :** Ground pins.

Other Pins

- **AREF :** Reference voltage for the analog inputs. Not currently supported by the Arduino software.
- **Reset :** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.
- It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button.
- The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7 V, however, the 5 V pin may supply less than five volts and the board may be unstable. If using more than 12 V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

Arduino Uno R3 Programming

- The programming of an Arduino Uno R3 can be done using IDE software. The microcontroller on the board will come with pre-burned by a boot loader that permits to upload fresh code without using an exterior hardware programmer.

- The communication of this can be done using a protocol like STK500.
- We can also upload the program in the microcontroller by avoiding the boot loader using the header like the In-Circuit Serial Programming.

6.2 Raspberry Pi

- A Raspberry Pi is a credit card-sized computer originally designed for education, inspired by the 1981 BBC Micro.
- Creator Eben Upton's goal was to create a low-cost device that would improve programming skills and hardware understanding at the pre-university level.
- The Raspberry Pi is slower than a modern laptop or desktop but is still a complete Linux computer and can provide all the expected abilities that implies, at a low-power consumption level.

Versions	Remarks
Raspberry Pi 1	• The original Raspberry Pi had 256 Mb of RAM, which increased to 512 MB in a later revision. • It has a 26-way GPIO connector
Pi Zero	• The Pi Zero includes the GPIO connector, but the header pins are not soldered
Raspberry Pi 2	• The Raspberry Pi 2 swapped the single-core processor for a much faster quad-core processor and increased the memory to 1GB RAM
Raspberry Pi 3	• The Raspberry Pi 3 changes the processor to an even more powerful 64-bit processor. • It also adds Wi-Fi and bluetooth which previously needed to be added as a USB device. • The Raspberry Pi 3 Model B was launched in February 2016.

- To get the Raspberry Pi working an SD card needs to be prepared with the Linux operating system installed.
- Raspberry Pi users have made many creative and impressive projects using this device. It can also be programmed to assist in 'housekeeping' your network by functioning as NAS, LDAP server, web server, media server, DNS server etc.
- The Raspberry Pi Foundation recommends Python. Any language which will compile for ARMv6 can be used. Installed by default on the Raspberry Pi : C, C++, Java, Scratch and Ruby.

6.2.1 About the Board

- Fig. 6.2.1 shows the Raspberry Pi board. The Raspberry Pi does not have a separate CPU, RAM or GPU. Instead they are all squeezed into one component called a system on Chip or SoC unit.

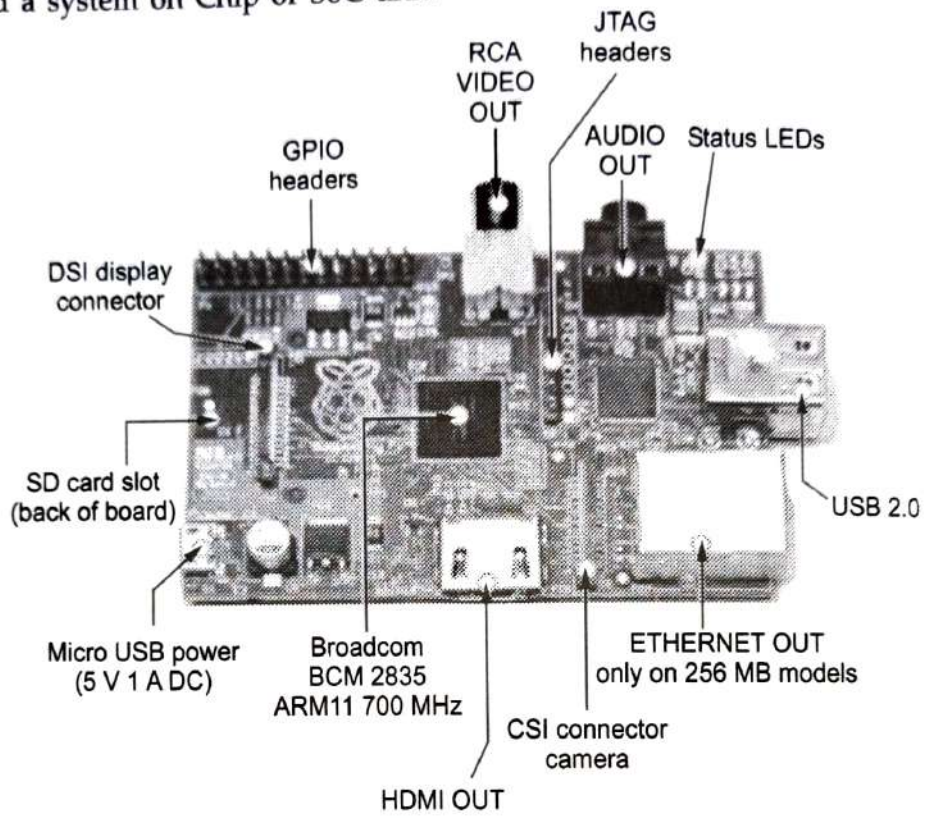


Fig. 6.2.1 (a) : Raspberry Pi circuit board

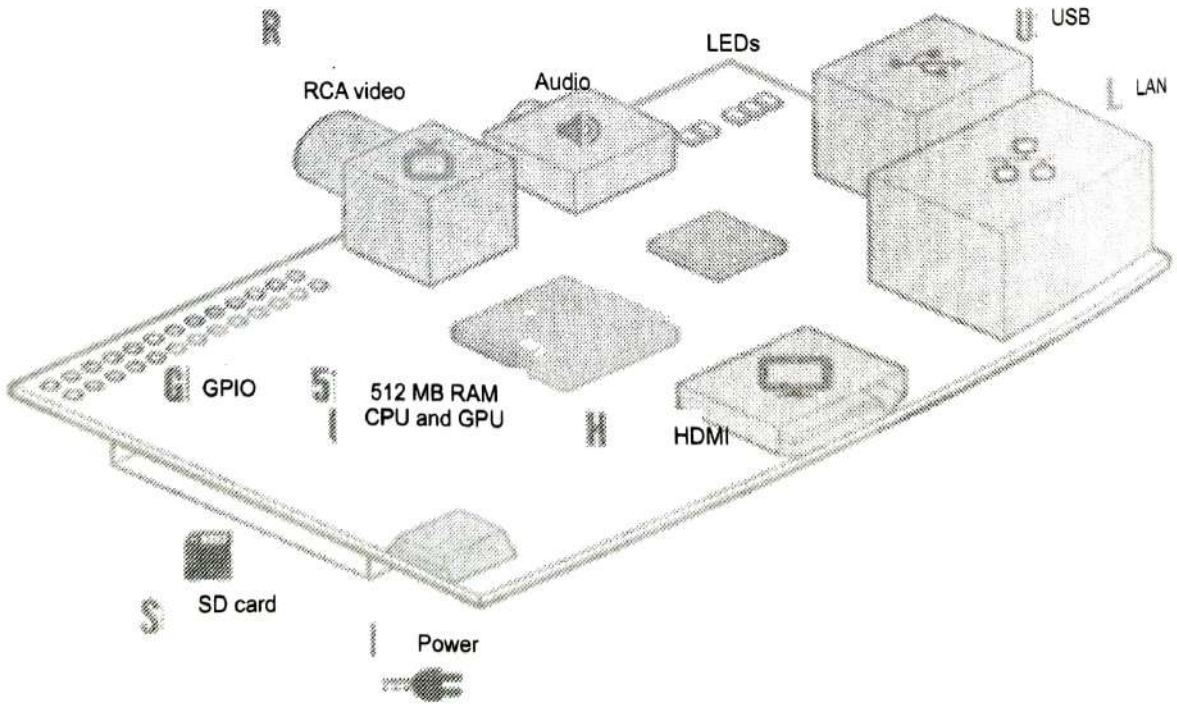


Fig. 6.2.1 (b) : Block diagram

- Raspberry Pi is open hardware with the exception of its primary chip, the Broadcomm SoC which runs the main components of the board - CPU, graphics, memory, USB controller etc.
- All of these Raspberry Pi Models share the following features :

1. Operating systems : Raspbian RaspBMC, Arch Linux, Rise OS, OpenELEC Pidora
2. Video output : HDMI Composite RCA
3. Supported resolutions : 640x350 to 1920x1200, including 1080p, PAL and NTSC standards
4. Power source : Micro USB

Components	Description
Processor	• Raspberry Pi uses an ARM processor which is also installed in a wide variety of mobile phones. • This CPU is single core, however it does have a co-processor to perform floating point calculations
Memory	• Model B Raspberry Pi has 512 MB SDRAM (Synchronous Dynamic RAM). • It store programs that are currently being run in the CPU
USB ports	• Board has two USB ports. USB port can provide a current upto 100 mA • Using powered hub, it is possible to connect more devices.
HDMI Output	• High Definition Multimedia Interface (HDMI) supports high-quality digital video & audio through a single cable. • It is also possible to connect a computer monitor with a DVI connection to HDMI using a converter.
Composite Video Output	• It support composite video output with RCA jack and also support PAL and NTSC. • The TVDAC pin can be used to output composite video.
Audio Output	• Audio output jack is 3.5 mm. • This jack is used for providing audio output to old television along with the RCA jack for video
GPIO Pins	• Both models have a total of 26 GPIO pins, organized into one pin header, named the P1 header • The newer Raspberry Pi (model B revision 2) adds 8 more GPIO pins in a new pin header called P5

	• Not all the GPIO pins are programmable. Some of them are 5.0 VDC or 3.3 VDC positive power pins, some of them are negative ground pins and a few of them are marked DNC (do not connect). • The P1 header has 17 programmable pins and the P5 header adds 4 more. • Fig 6.2.2 shows GPIO pin header. • Reading from various environmental sensors. Writing output to dc motors, LEDs for status.
Power Input	• Micro-USB connector is used for power input
Status LED	• It has five status LED.
CSI	• Camera Serial Interface (CSI) can be used to connect a camera module to Raspberry Pi
SD Card Slot	• This card is used for loading operating system

- The Raspberry Pi comes with a set of 26 exposed vertical pins on the board. These pins are a General Purpose Input /Output interface that is purposely not linked to any specific native function on the Raspberry Pi board.

Raspberry Pi P1 header

PIN #	Name			Name	PIN #
	3.3 VDC power	1	2	5.0 VDC power	
8	SDA0 (I2C)	3	4	DNC	
9	SCL0 (I2C)	5	6	0 V (Ground)	
7	GPIO 7	7	8	TxD	15
	DNC	9	10	RxD	16
0	GPIO 0	11	12	GPIO1	1
2	GPIO 2	13	14	DNC	
3	GPIO 3	15	16	GPIO4	4
	DNC	17	18	GPIO5	5
12	MOSI	19	20	DNC	
13	MISO	21	22	GPIO6	6
14	SCLK	23	24	CE0	10
	DNC	25	26	CE1	11

Fig. 6.2.2 : GPIO pin header

- Instead, the GPIO pins are there explicitly for the end user to have low-level hardware access directly to the board for the purposes of attaching other hardware boards, peripherals, LCD display screens and other hardware devices to the Pi.

The Status LEDs

Status LED	Color	Functions
ACT	Green	Lights when the SD card is accessed (marked OK on earlier boards)
PWR	Red	Hooked up to 3.3 V power
FDX	Green	On if network adapter is full duplex
LNK	Green	Network activity light
100	Yellow	On if the network connection is 100 Mbps

- The Raspberry Pi draws its power from a microUSB port and requires a microUSB-to-AC adapter. Because the Pi is a micro computer and not simply a cell phone getting a battery topped off, you need to use a high quality charger with stable power delivery that provides a consistent 5 V with at least 700 mA minimum output for older model units and 2.5 A for the Pi 3.

6.2.2 Linux on Raspberry Pi

- There are several unix like operating systems for the RPI and there is an operating system called RISC OS that has its origin at the developers of the first ARM chips.
- The Raspberry Pi Foundation recommends the use of the following Linux Distributions :

1. Debian 7
2. Raspbian
3. Arch Linux ARM
4. QtonPi

- Raspbian is a free operating system based on Debian optimized for the Raspberry Pi (RPI) hardware.
- The default command prompt on the Pi consists of four components shown in Fig. 6.2.3.
- Raspbian is the desired operating system for the Raspberry Pi. In order to download and install the operating system onto our Raspberry Pi ; you will need Raspbian, Win32DiskImager and USB memory card reader.

1. Download both Raspbian and Win32DiskImager and save somewhere easily accessible
2. Plug the USB memory card reader into your computer
3. Open Win32DiskImager
4. Find the location of the image file and the memory card
5. Click "Write"

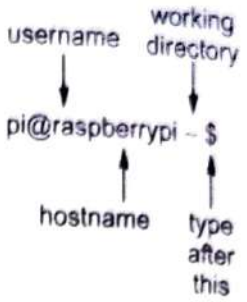


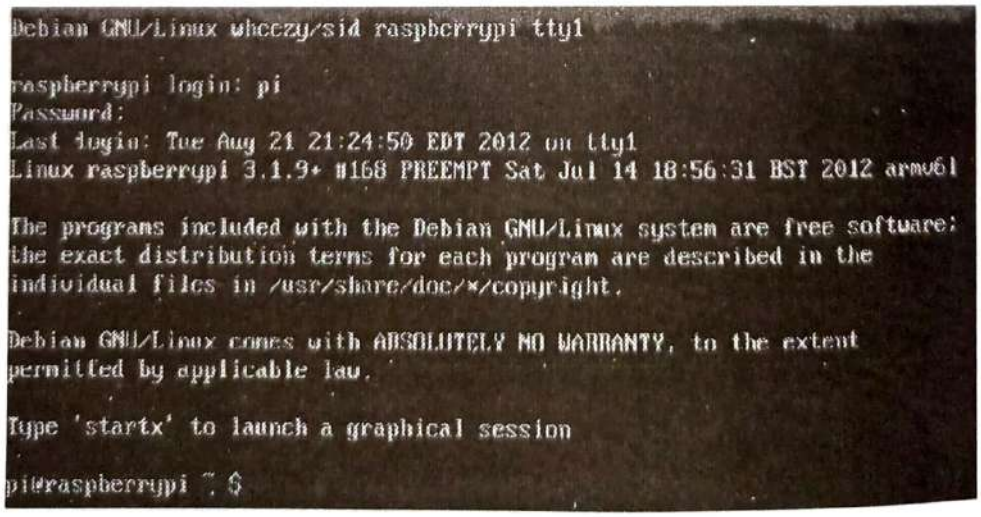
Fig. 6.2.3 : Command prompt

Logging In

- Now it is time to turn on our Raspberry Pi. When the memory card, HDMI lead, Ethernet cable, mouse and keyboard are plugged in, plug in the power lead.
- As soon as you do this. You screen should be black and filled with white text. This will be visible every time you turn on your raspberry pi.
- Wait until your screen reads "raspberrypi login :"

  Username = pi [ENTER]

  Password = raspberry [ENTER]



Starting the Raspbian GUI

- GUI stands for Graphical User Interface and is a type of operating system. It is the most common type of user interface as it is a very 'friendly' way for people to interact with the computer. It makes use of pictures, graphics, icons and pointers, hence the name 'Graphical' User Interface. Fig. 6.2.4 shows Rasbian Linux desktop

1. **Type the line : "startx"**

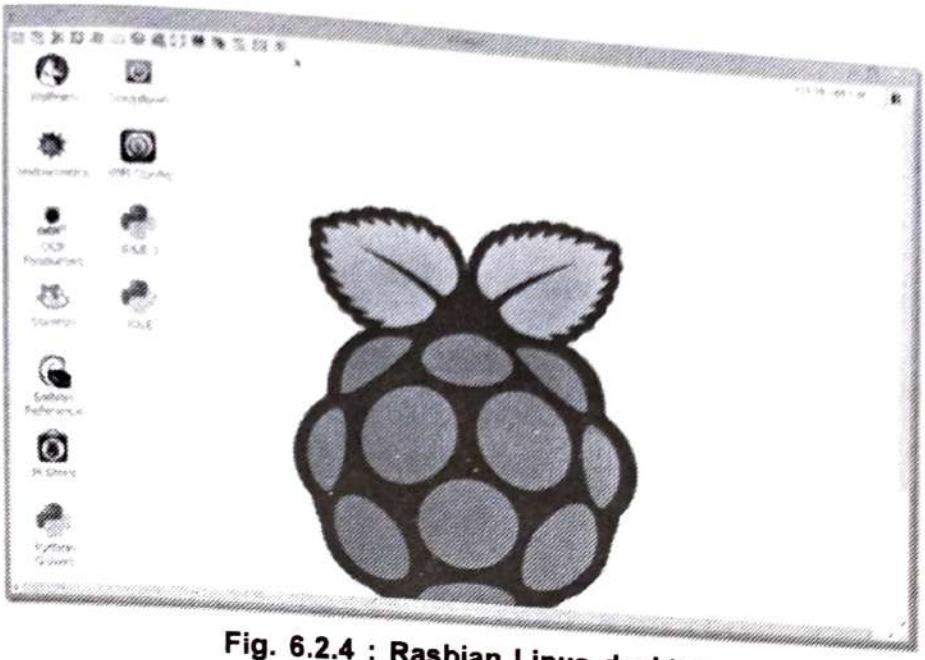


Fig. 6.2.4 : Rasbian Linus desktop

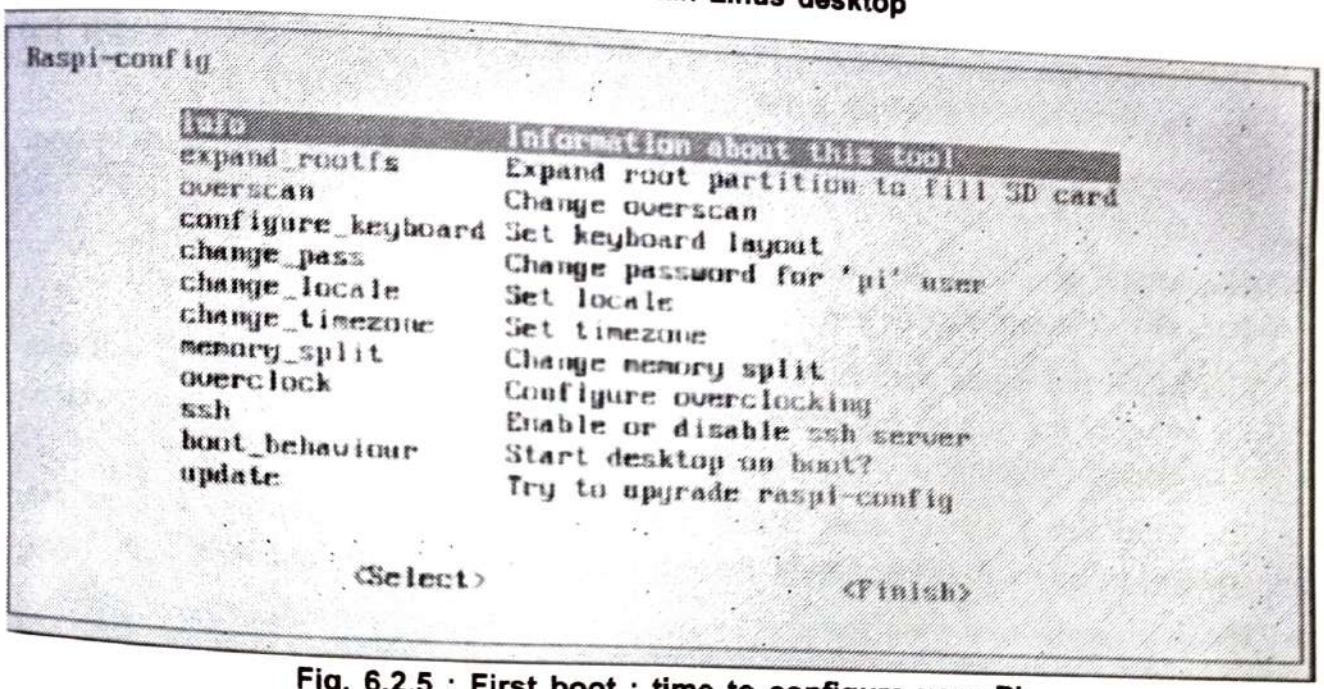


Fig. 6.2.5 : First boot : time to configure your Pi

6.2.3 Difference between Raspberry Pi is and Desktop Computers

- In Raspberry Pi, operating system is installed on SD card whereas in desktop computer, operating system is installed in hard disk.
- Raspberry Pi does not have their own CPU and RAM.
- Processing power of Raspberry Pi is less as compared to desktop computers.
- Raspberry Pi uses less power than desktop computers

6.3 Raspberry Pi Interface

- Three types of interface is supported by Raspberry Pi.

1. Serial

- It uses serial peripherals for serial communication.
- Transmit (Tx) and Receive (Rx) pin is used for serial communication.

2. Serial Peripheral Interface (SPI)

- SPI is a communication protocol used to transfer data between micro-computers like the Raspberry Pi and peripheral devices. These peripheral devices may be either sensors or actuators.
- SPI uses 4 separate connections to communicate with the target device. These connections are the serial clock (CLK), Master Input Slave Output (MISO), Master Output Slave Input (MOSI) and Chip Select (CS).
- The clock pin sense pulses at a regular frequency, the speed at which the Raspberry Pi and SPI device agree to transfer data to each other.
- For the ADC, clock pulses are sampled on their rising edge, on the transition from low to high.
- The MISO pin is a data pin used for the master to receive data from the ADC. Data is read from the bus after every clock pulse.
- The MOSI pin sends data from the Raspberry Pi to the ADC. The ADC will take the value of the bus on the rising edge of the clock. This means the value must be set before the clock is pulsed.
- The Chip Select line chooses which particular SPI device is in use. If there are multiple SPI devices, they can all share the same CLK, MOSI, and MISO.
- The SPI has the following features :

1. 16-bit shift register
2. 16-bit Receive buffer register (SPIBUF) and 16-bit Receive buffer emulation alias register (SPIEMU)
3. 16-bit Transmit data register (SPIDAT0) and 16-bit Transmit data and format selection register (SPIDAT1)
4. 8-bit baud clock generator
5. Serial clock (SPICLK) I/O pin
6. Slave in, master out (SPISIMO) I/O pin
7. Slave out, master in (SPISOMI) I/O pin

8. Multiple slave chip select (SPISCS[n]) I/O pins (4 pin mode only)
9. Programmable SPI clock frequency range
10. Programmable character length (2 to 16 bits)
11. Programmable clock phase (delay or no delay)
12. Programmable clock polarity (high or low)
13. Interrupt capability
14. DMA support (read/write synchronization events)
15. Up to 66 MHz operation

Master-slave configuration of SPI :

- Fig. 6.3.1 shows SPI system. SPI bus is composed by four signals, namely the Master Out Slave In (MOSI), Master In Slave Out (MISO), serial clock (SCK) and active low slave select (\SS).
- MOSI : This pin is used to transmit data out of the SPI module when it is configured as a Master and receive data when it is configured as Slave.
- MISO : This pin is used to transmit data out of the SPI module when it is configured as a Slave and receive data when it is configured as Master.
- /SS : This pin is used to output the select signal from the SPI module to another peripheral with which a data transfer is to take place when its configured as a Master and its used as an input to receive the slave select signal when the SPI is configured as Slave.
- SCLK : This pin is used to output the clock with respect to which the SPI transfers data or receive clock in case of Slave.

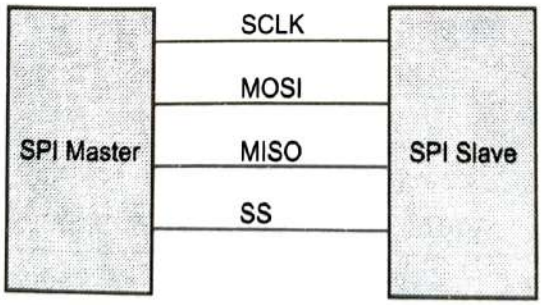


Fig. 6.3.1 : SPI

- SCK master device will generate a pulse and the data will be synchronized in both master and slave devices. There are four different clock types to define SIP protocol, depending on what the SCK polarity and phase may be. It must ensure these signals between the master and slave devices compatible with each other.

- SPI is a Synchronous protocol. The clock signal is provided by the master to provide synchronization. The clock signal controls when data can change and when it is valid for reading.
- SPI creates a data loop between two devices. Data leaving the master exits on the SDO (serial data output) line. Data entering the master enters on the serial data input, SDI line.
- A clock (SCK), is generated by the master device. It controls when and how quickly data is exchanged between the two devices.
- SS allows a master device to control when a particular slave is being addressed. This allows the possibility of having more than one slave and simplifies the communications. When the SS signal goes low at a slave device, only that slave is accessed by SPI
- For SPI, there are Serial Clocks (SCLK), Chip Select lines (CS), Serial Data In (SDI) and Serial Data Out(SDO). There is only one master, there number of slaves depends on the number of chip select lines of the master.
- Synchronous operation, latch on rising or falling edge of clock, SDI on rising edge, SDO on falling edge. It operates in 1 to 2 MHz range.
- Master sends out clocks and chip selects. Activates the slaves it wants to communicate with.
- Fig. 6.3.2 master with multiple slave interface.

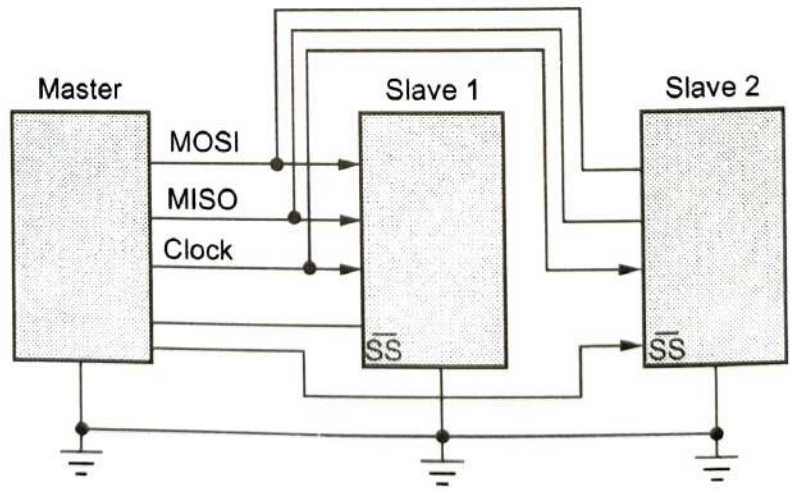


Fig. 6.3.2 : Multiple slave interface

- SPI data transmit and data receive register are the main elements of the SPI. When the communication takes place the data on the transmit register are transferred into the shift register.

- The shift register in the master of width (8,16,32) and the shift register in the slave are linked by MOSI and MISO pins to form a distributed 16,32,64 bit register respectively.
- When the data transfer operation needs to be performed these 16,32,64- bit registers are serially shifted eight, sixteen, thirty-two bit positions by the serial clock generated by the master so that the data can be exchanged between the master and the selected slave.
- Data on the master SPI data transmit register becomes the input data for the slave read from the MOSI and the data read from the master SPI data receive register was the data send from the slave from MISO.
- Data on the shift registers are transferred into data receive register when the transfer completes and this data may be read from the data receive register any time before next transfer has completed.

3. I^2C

- I^2C is a communication protocol that the Raspberry Pi can use to speak to other embedded devices (temperature sensors, displays, accelerometers, etc).
- I^2C is a useful bus that allows data exchange between microcontrollers and peripherals with a minimum of wiring.
- I^2C is a two wire bus, the connections are called SDA (Serial Data) and SCL (Serial Clock). Each I^2C bus has one or more masters (Raspberry Pi) and one or more slave devices, like the I/O Expander.
- As the same data and clock lines are shared between multiple slaves, we need some way to choose which device to communicate with.

6.4 Raspberry Pi with Python Programming

- General Purpose Input/Output (GPIO) is a generic pin on a chip whose behavior can be controlled by the user at run time. The GPIO connector has a number of different types of connection :

1. True GPIO pins that you can use to turn LEDs on and off etc.
2. I2C interface pins that allow you to connect hardware modules with just two control pins.
3. SPI interface with SPI devices, a similar concept to I2C but uses a different standard.
4. Serial Rx and Tx pins for communication with serial peripherals.

6.4.1 Controlling LED with Raspberry Pi

- Fig 6.4.1 shows diagram of connecting LED to Raspberry Pi. The LED will initially be off because the GPIO pins are initialized as inputs at power-on.
- Install Python 2 library Rpi.GPIO. A library that will let us control the GPIO pins.

  Install commands :

  sudo apt?get update

  sudo apt?get install python?dev

  sudo apt?get install python?rpi.gpio

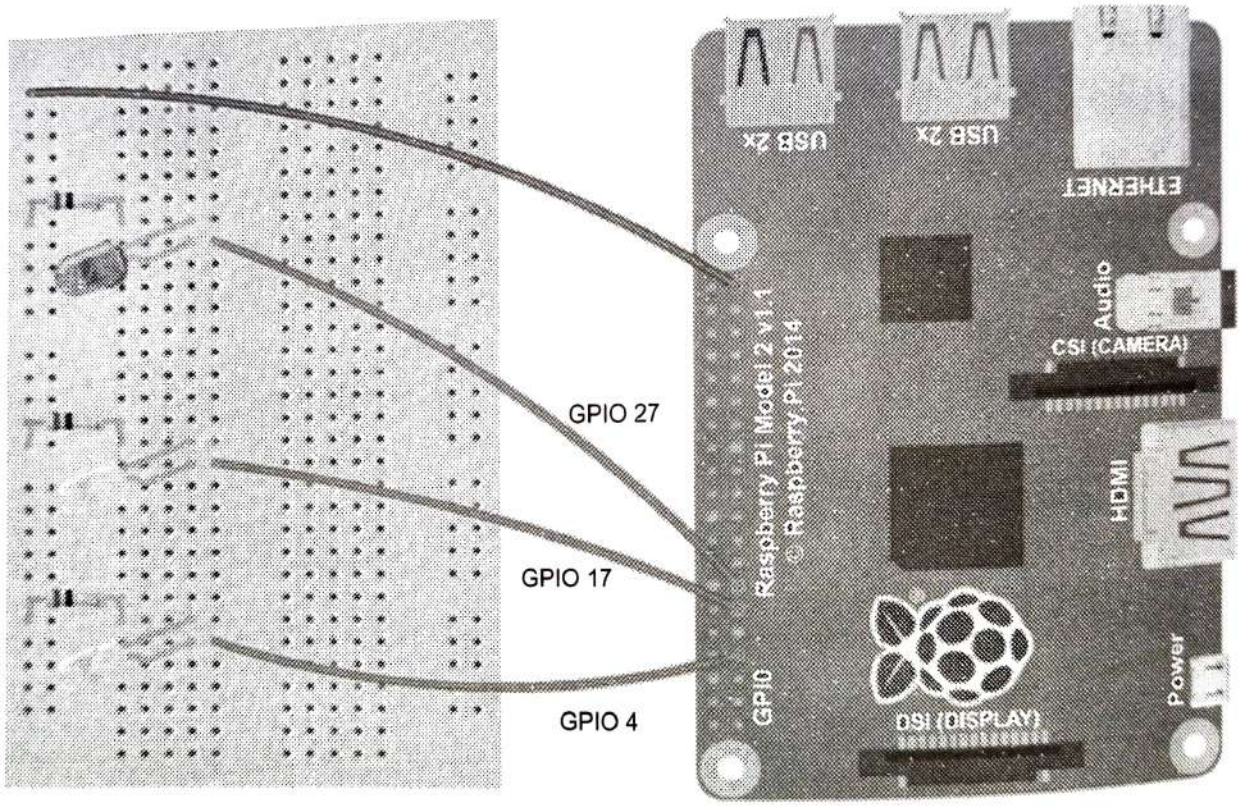


Fig. 6.4.1 : Diagram of connecting LED to Raspberry Pi

- Simple LED Circuit is shows below :

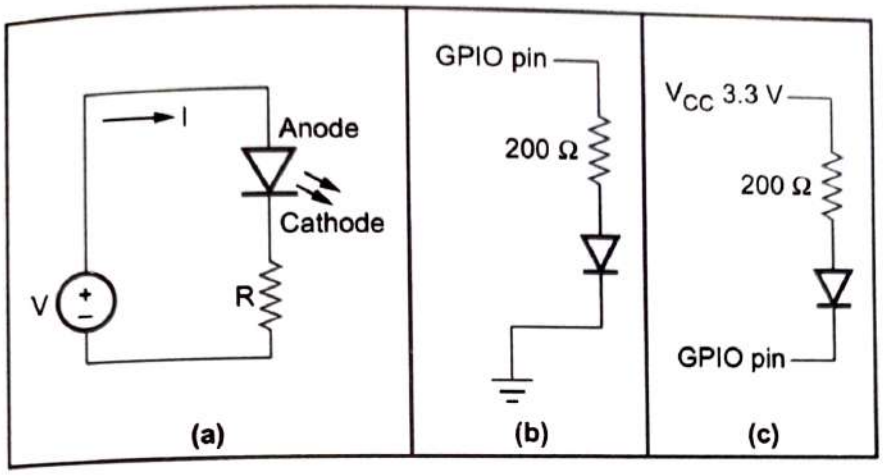


Fig. 6.4.2

- Current flows from the anode (+) to cathode (–). Anode is longer pin and cathode is shorter pin.
- Open up IDLE, the Python programming software and create a New file. Save it as led.py and input the code from the code listing. What the code does is first tell Python to use the GPIO module so we can connect to the GPIO pins, by importing the module.
- We then import the time module so we can create a delay between commands. We then tell the code to treat the GPIO pins as the number they are on the board and to turn the seventh pin into an output.
- We alternate between True and False so that it turns the pin on and off. Once it's cycled a few times, it will print the message 'Done' into IDLE and finally turn off the GPIO pins.

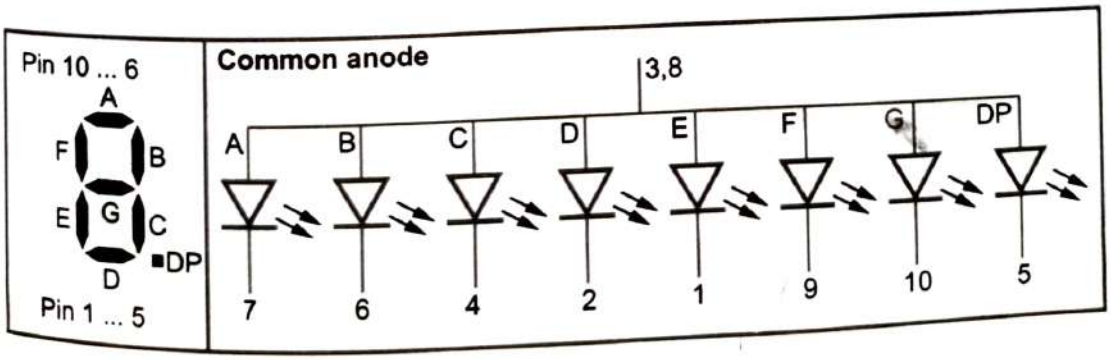


Fig. 6.4.3

```
Import RPi.GPIO as GPIO
Import time
GPIO.setmode(GPIO.BOARD)
GPIO.setup(7, GPIO.OUT)
GPIO.output(7,True)
time.sleep(1)
GPIO.output(7,False)
time.sleep(1)
GPIO.output(7,True)
time.sleep(1)
GPIO.output(7,False)
print"Done"
GPIO.cleanup()
```

Task 1 : Turn LED on for 2 seconds and off for 1 second, loop forever. Code is given below :

(In this example, we use diagram (b), i.e. controlling the LED by controlling the voltage at the anode (+)).

```
import RPi.GPIO as GPIO
import time
def main( ) :
GPIO.cleanup( )
GPIO.setmode(GPIO.BOARD)  # to use Raspberry Pi board pin numbers
GPIO.setup(11, GPIO.OUT)  # set up GPIO output channel
while True :
GPIO.output(11, GPIO.LOW)  # set RPi board pin 11 low. Turn off LED.
time.sleep(1)
GPIO.output(11, GPIO.HIGH)  # set RPi board pin 11 high. Turn on LED.
time.sleep(2)
main( )
```

- Example : Display digit on 7-segment LED. It is most direct way to control display :

1. Connect pin 3/8 of 7-seg-LED to Vcc

2. Connect the other 8 pins to 8 GPIO pins
3. Configure the 8 GPIO pins as out

- For example : display "2". Turn on segments A, B, D, E, G and turn off segments C, F, DP. Set A, B, D, E, G to LOW and set C, F, DP to HIGH. Set Pin 7, 6, 2, 1, 10 LOW and Set pin 4, 9, 5 HIGH

6.4.2 Interfacing an LED and Switch with Raspberry Pi

- When the switch is not pushed : GPIO detects Vcc (HIGH)
- When the switch is pushed : GPIO detects GND (LOW

GPIO Input Sample Code

```
import RPi.GPIO as GPIO
# Use the pin numbers from the ribbon cable board
GPIO.setmode (GPIO.BCM)
# Set up this pin as input.
GPIO.setup (17, GPIO.IN)
# Check the value of the input pin
GPIO.input (17)
# Hold down the button, run the command again. The output should be "true".
GPIO.input(17)
```

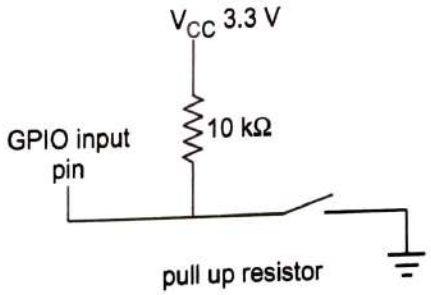


Fig. 6.4.4

6.4.3 Interfacing Light Sensor

- Unlike some other devices the Raspberry Pi does not have any analogue inputs. All 17 of its GPIO pins are digital. They can output high and low levels or read high and low levels.
- For sensors that act as a variable resistor such as LDRs (Light Dependent Resistors) or thermistors (temperature sensors) there is a simple solution. It allows

you to measure a number of levels using a single GPIO pin. In the case of a light sensor this allows you to measure different light levels.

- Fig 6.4.5 shows diagram of connecting an LDR to Raspberry Pi.

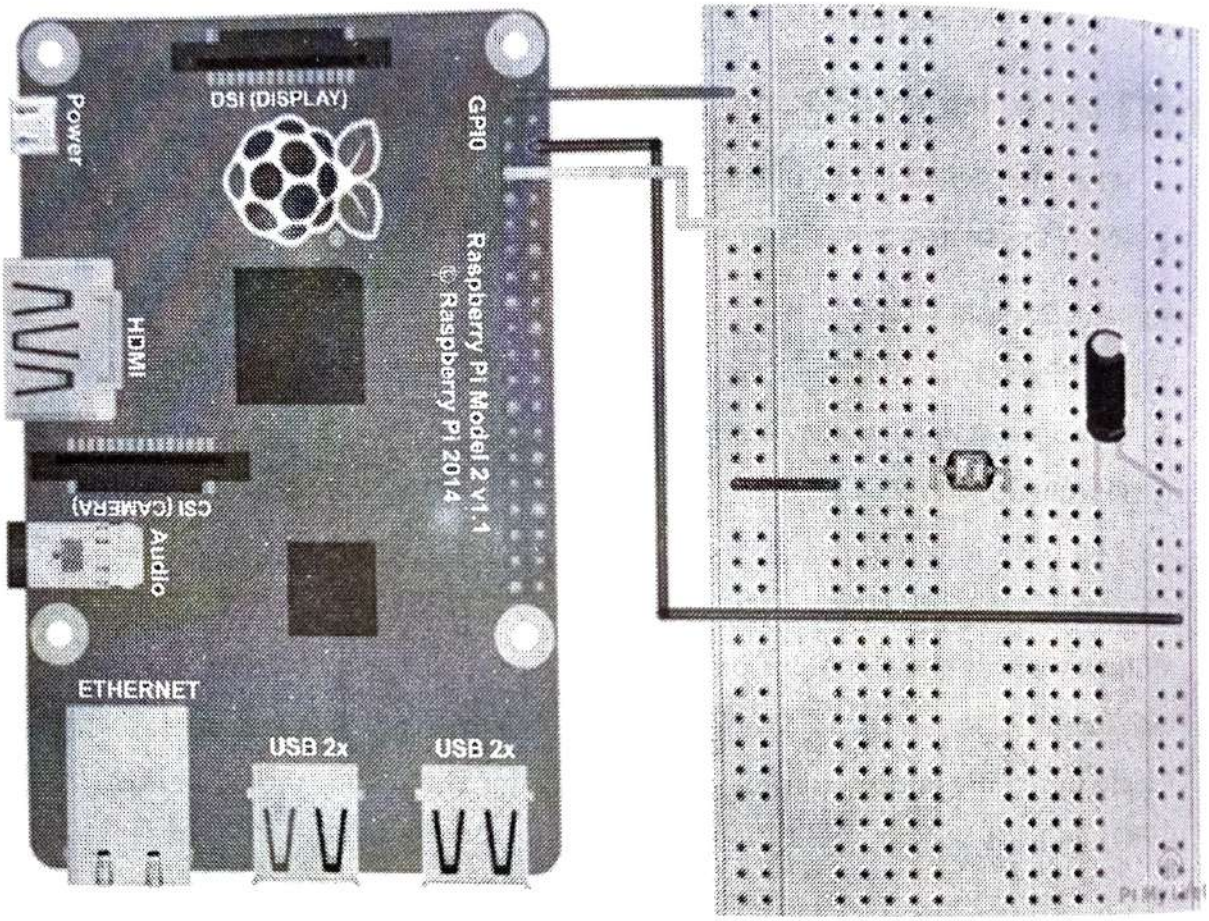


Fig. 6.4.5 : Diagram of connecting an LDR to Raspberry Pi

- Following are steps :

1. First connect pin number 1 (3v3) to the positive rail on the breadboard.
2. Next connect pin number 6 (ground) to the ground rail on the breadboard.
3. Now place the LDR sensor onto the board and have a wire go from one end to the positive rail.
4. On the other side of the LDR sensor place a wire leading back to the Raspberry Pi. Hook this to pin number 7.
5. Finally place the capacitor from the wire to the negative rail on the breadboard. Make sure you have the negative pin of the capacitor in the negative rail.

- Fig. 6.4.6 shows circuit diagram for above configuration.

The sequence of events :

1. Set the GPIO pin as an output and set it Low. This discharges any charge in the capacitor and ensures that both sides of the capacitor are 0 V.
2. Set the GPIO pin as an input. This starts a flow of current through the resistors and through the capacitor to ground. The voltage across the capacitor starts to rise. The time it takes is proportional to the resistance of the LDR.

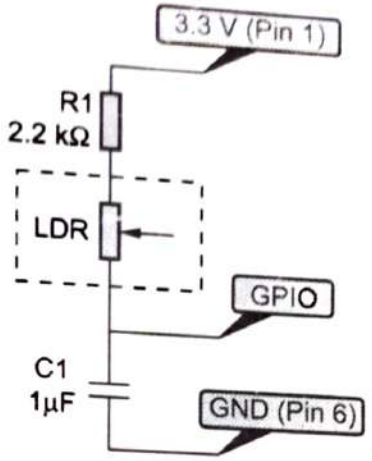


Fig. 6.4.6 : Circuit diagram for LDR

3. Monitor the GPIO pin and read its value. Increment a counter while we wait.
4. At some point the capacitor voltage will increase enough to be considered as a High by the GPIO pin (approx 2v). The time taken is proportional to the light level seen by the LDR.
5. Set the GPIO pin as an output and repeat the process as required.

Python Code :

```
#!/usr/local/bin/python

# Reading an analogue sensor with a single GPIO pin

import RPi.GPIO as GPIO, time

# Tell the GPIO library to use
# Broadcom GPIO references
GPIO.setmode(GPIO.BCM)

# Define function to measure charge time
def RCtime (PiPin) :
 measurement = 0

 # Discharge capacitor
 GPIO.setup(PiPin, GPIO.OUT)
 GPIO.output(PiPin, GPIO.LOW)
```

```
time.sleep(0.1)
GPIO.setup(PiPin, GPIO.IN)
# Count loops until voltage across
# capacitor reads high on GPIO
while (GPIO.input(PiPin) == GPIO.LOW) :
 measurement += 1
return measurement
# Main program loop
while True :
 print RCtime(4) # Measure timing using GPIO4
```

6.5 Short Questions and Answers

Q.1 What is Raspberry Pi ?

Ans. : Raspberry Pi is a low-cost, credit card-sized computer that connects to a computer monitor or TV using HDMI and uses a standard keyboard and mouse. It can run a host of operating systems, such as Raspbian, Android, Windows 10, IoT Core, etc.

Q.2 What are the different raspberry Pi model types ?

Ans. : The Raspberry Pi models are of two types :

1. Model A (introduced later as a hardware-reduced model)
2. Model B (introduced first and is the full hardware model)

Q.3 What is Arduino ?

Ans. : Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button or a twisster message and turn it into an output - activating a motor, turning on an LED, publishing something online.

Q.4 Explain difference between Model A and Model B of Raspberry Pi.

Ans. :

Parameters	Model A	Model B
GPU type	VideoCore IV	VideoCore IV
USB port	1	2
Memory	256 MB	512 MB

Ethernet port	No Ethernet port	10 / 100 Ethernet
SoC Type	Broadcom BCM2837B0	Broadcom BCM2837B0
Number of Cores	4	4
Type	It is hardware-reduced model.	It is full hardware model.

Q.5 What is Python ?

Ans. : Python is an object-oriented, high-level programming language with integrated dynamic semantics primarily for web and app development.

Q.6 List and explain features of Python.

Ans. : Features :

1. Python is a simple and minimalistic language
2. Easy to Learn
3. Free and Open Source
4. Python supports procedure-oriented programming as well as object-oriented programming
5. Extensive Libraries : The Python Standard Library is huge indeed
6. Embeddable : You can embed Python within your C/C++ programs to give scripting capabilities for your program's users.

Q.7 List the benefits of Python.

Ans. :

- Python can be used to develop prototypes.
- Python allows for a more productive coding environment than massive languages like C# and Java.
- Python powers Django, a complete and open source web application framework.
- Most automation, data mining and big data platforms rely on Python.
- Python supports modules and packages, which encourages program modularity and code reuse.

Q.8 What is GND in GPIO ?

Ans. : GND means ground pins. Ground GPIO pins are physical numbers 6, 9, 14, 20, 25, 30, 34 and 39.

Q.9 Define Raspberry Pi hardware.

Ans. : Raspberry Pi hardware includes ARM processor, GPU, RAM and USB port.

7 IoT Security

Syllabus

Various security issues and need, architecture, requirement, challenges and algorithms

Contents

7.1 Various Security Issues and Need

- The Internet of Things (IoT) refers to a concept of connected objects and devices of all types over the Internet wired or wireless. The popularity of IoT or the Internet of Things has increased rapidly, as these technologies are used for various purposes, including communication, transportation, education, and business development.
- The unconscious use, not changing passwords and the lack of device updates have increased cybersecurity risks and access to malicious applications to the IoT systems sensitive data.
- Most of the security professionals consider IoT as the vulnerable point for cyber-attacks due to weak security protocols and policies. Even though several security mechanisms were developed to protect IoT devices from cyber-attacks, security guidelines are not appropriately documented.
- IoT enabled devices have been used in industrial applications and for multiple business purposes. The apps help these businesses to attain a competitive edge over their competitors.
- However, due to the excessive adoption of various smart devices with data sharing and integration, the privacy and data breach becomes a significant concern to most businesses, as it interrupts the flow of work, activities, and network services.
- IoT system functionalities :
  1. Security patch must be upload time to time in microprocessor firmware.
  2. Monitor the access and usage of public network.
  3. User authentication is necessary.
  4. Only after authentication can the controller direct commands for things control that are present in the system.
- The Internet of Things (IoT) has become a ubiquitous term to describe the tens of billions of devices that have sensing or actuation capabilities and are connected to each other via the Internet.

Risks :

- The IoT includes everything from wearable fitness bands and smart home appliances to factory control devices, medical devices and even automobiles. Security has not been a high priority for these devices until now.
- The security of the Internet of Things, the following principles can be established.

a) **Identity** : Trust is always tied to an identity. Therefore every device needs a unique identity that can't be changed. The device must also be able to prove its identity at all times.

b) **Positive intention** : The device and linked service have positive intentions.

c) **Predictability and transparency** : The functional scope of the service provided by devices is known to its full extent. There are no undocumented (secret) functions. The behaviour of the system can be checked at any time by independent third parties.

d) **Reputation** : An increasing number of positive interactions between the things gradually form a reputation based intelligent network.

7.2 Security Architecture

- Fig. 7.2.1 shows IoT security architecture. (See Fig. 7.2.1 on next page.)
- IoT systems are often highly complex, requiring end-to-end security solutions that span cloud and connectivity layers, and support resource-constrained IoT devices that often aren't powerful enough to support traditional security solutions.
- Application layer support user services. This layer helps users access IoT through the interface using PC, mobile equipment etc. This layer also support secure communication protocol and authentication protocols.
- Network layer support wired and wireless communication protocol and technology. This layer is responsible for dependable broadcast of data and information from the below layer.
- Sensors are the monitors that pick up data and relay it for further analysis. Actuators are devices that act as robotic controls. Many IoT attacks have used actuators, such as printers, as launch points into a business's network.
- An IoT security architecture is a blueprint that illustrates all components of the IoT infrastructure for all IoT projects and details how to secure each component.
- In both cases, it is imperative to ensure device access is controlled via settable passwords, encrypt any data stored locally and monitor and contain any executable code run by the device.
- Physical layer gathers all types of information with the help of physical equipment. IoT devices face many threats, including malicious data that can be sent over authenticated connections, exploiting vulnerabilities and/or misconfigurations.
- Such attacks frequently exploit many weaknesses, including but not limited to
  a) Failure to use code signature verification and secure boot,
  b) Poorly implemented verification models which can be bypassed.

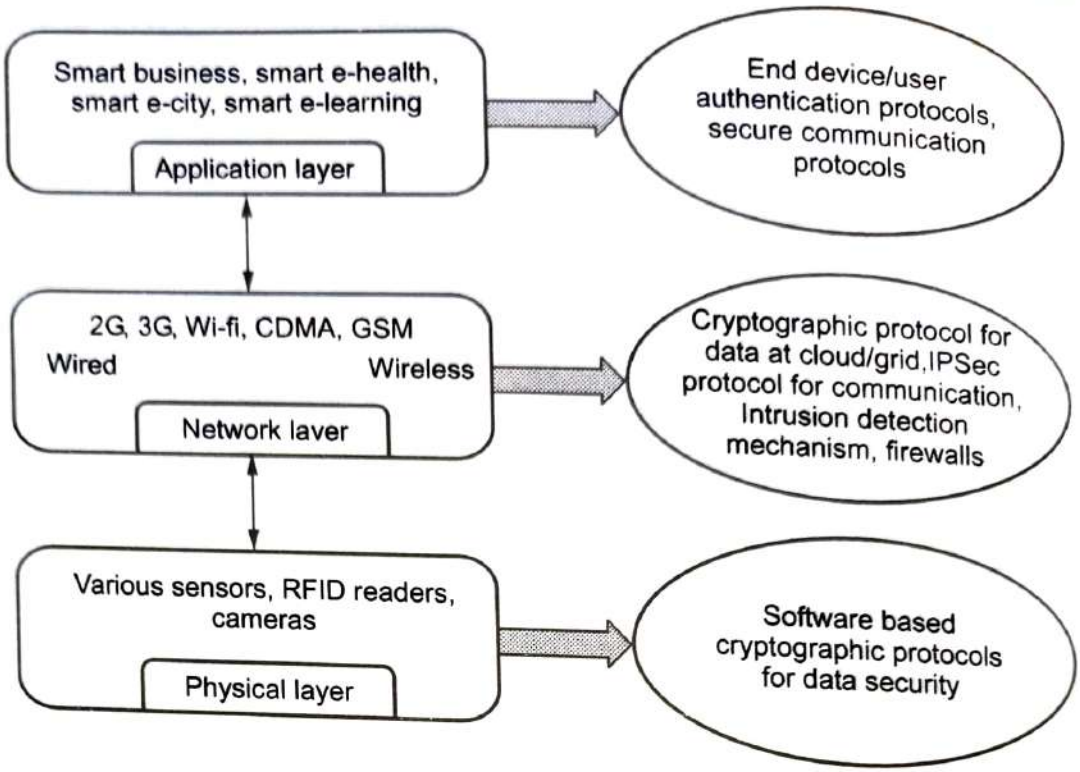


Fig. 7.2.1

- Attackers often use those weaknesses to install backdoors, sniffers, data collection software, file transfer capabilities to extract sensitive information from the system, and sometimes even Command & Control (C&C) infrastructure to manipulate system behaviour.

7.3 Security Requirement

- The key requirements for any IoT security solution are :
  1. Device and data security, including authentication of devices and confidentiality and integrity of data.
  2. Implementing and running security operations at IoT scale.
  3. Meeting compliance requirements and requests.
  4. Meeting performance requirements as per the use case.
- Application layer : Verification and user's confidentiality
- Support layer : Various encryption algorithms
- Network layer : Distributed denial of service attack
- Physical layer : Authentication.

7.4 Challenges

- The security challenges are as follows :
  a. **Devices are not reachable :** Most of the time a device is not connected.
  b. **Devices can be lost and stolen :** Makes security difficult when the device is not connected.
  c. **Devices are not crypto-engines :** Strong security difficult without processing power.
  d. **Devices have finite life :** Credentials need to be tied to lifetime.
  e. **Devices are transportable :** Will cross borders.
- IoT system has a cloud database that is connected to all your devices. These devices are connected to the internet and it could be accessed by the cybercriminals and hackers. As the number of connected devices increases, chances for hackers to breach the security system gets increased.

7.5 Short Questions and Answers

Q.1 What is senseIoT ?

Ans. : SenseIoT is a great sensor data storage platform. With senseIoT you can easily store the data from your sensors and devices safely and securely.

Q.2 What do you mean by risk in IoT ?

Ans. : The IoT includes everything from wearable fitness bands and smart home appliances to factory control devices, medical devices and even automobiles. IoT devices allow hackers to produce physical effects.

Q.3 What risks do insecure IoT devices bring to privacy and security ?

Ans. :

- Using insecure IoT devices increases the risks of personal data being exposed/stolen and privacy compromised :
- A smart camera using default username and password combination can be used to spy on you or be compromised to send junk information to the Internet.
- A wearable smart device that sends health information over un-encrypted channels can expose personal data.
- A smart home device like a television that lacks sufficient updates can be vulnerable to new attacks and be used to share private data.