Unit-2 Internet Of Things

2.1 Embedded Systems:

Definition:

An embedded system is a microcontroller or microprocessor based system which is designed to perform a specific task.

OR

An embedded system is a combination of computer hardware and software, either fixed in capability or programmable, designed for a specific function or functions within a larger system.

2.1.1 Embedded system concepts

An embedded system can be defined as a microprocessor or microcontroller-based, software-driven, Realible,real-time control system deigned to perform a specific task. An Embedded System may be either an independent system or a part of a large system. An Embedded System consists of Input Device, Microcontroller (The Brain) and Output Device. There is a main difference between the embedded system and general purpose system is the computing device like a microprocessor has external peripherals i.e. Real-time Clock, USB, Ethernet, WiFi, Bluetooth, ports etc.) connected to it and are visible outside. But an embedded device contains few or all the peripherals inside the module which is called as SOC (System On Chip).

2.1.2 Purpose of embedded systems:

The embedded system is used in many domain areas such as consumer electronics, home automation, telecommunication, automotive industries, healthcare, control and instrumentation, banking application, military application etc. According to application usage, the embedded system may have the different functionalities. Every embedded system is designed to accomplished the purpose of any one or a combination of following task.

- **Data collection/storage/Representation:** Data is collected from the outside world using various sensors for storage, analysis, manipulation and transmission. The data may be information such as voice, text, image, graphics, video, electrical signals or other measurable quantities. The Collected data may be stored or transmitted to other device or processed by the embedded system for meaningful representation.
- **Data communication in embedded system:**The data can be transmitted either through wireless media or wired media. The data can be an analog or digital, The data transmission can be done through wireless mcdia such as Bluetooth, ZigBee, WiFi, GPRS, Edge etc or wired media such as RS232C, USB, TCP/IP, PC2, Firewire port, SPI, CAN, I^2C etc.
- **Data processing:**The data which may in the form of Voice, Image, Video, electrical signal or any other measurable quantities is collected by an embedded system and used for various kind of processing depending on the application
- **Monitoring the performance/operation of embedded system:**The embedded systems mostly used for monitoring purpose. For example, ECG (Electro cardiogram) machine is used to monitor the heartbeat of the patient.
- **Control the embedded system:**The embedded system having control functionalities executes control over some variables as per the input variable. The embedded system having control functionalities contains both sensor and actuator. Sensors are

connected as input to the ports of the system to capture the change in measuring variable and actuator are connected to output port as a final control element to control the system as per change in input variables within the specified range. For example, air conditioning system at home is used to control the room temperature as per the specified limit.

- **Application specific user's interface:** Most of the embedded system comes with Application specific user's interface such as switches, buttons, display, light, bell, keypad etc. For example, mobile phone comes with user interface such as Keyboard, LCD or LED display, Speaker, vibration alert etc.

2.1.3 Architecture of Embedded System:

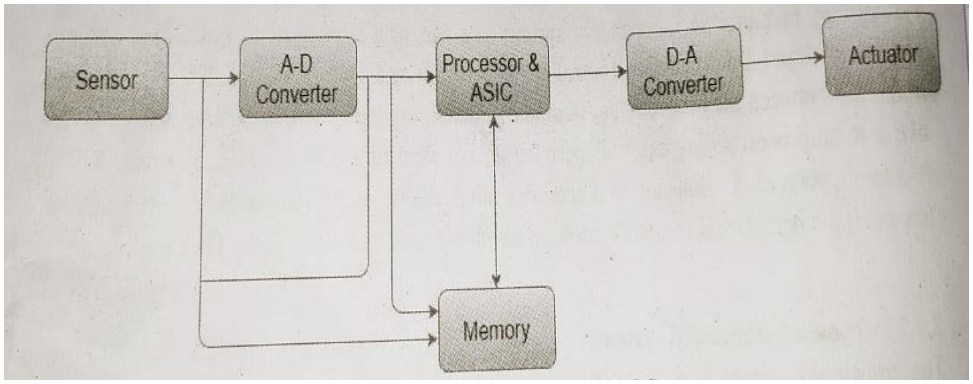

Fig.2.1: Basic Structure of an Embedded System

- **Sensor** - Sensor is used to measure the physical quantity and converts it to an electrical signal which can be read by any electronic device like an A-D converter.
- **A-D Converter** - An analog-to-digital converter converts the analog signal given by the sensor into a digital signal.
- **Processor & ASICS** - Processors process the data to measure the output and store it to the memory.
- **D-A Converter**- A digital-to-analog converter converts the digital data given by the Processor to analog data**.**
- **Actuator** - An actuator compares the output given by the D-A Converter to generates the actual or expected output.

An embedded system has three main components:

- **Embedded system hardware:** An embedded system uses a hardware platform to execute the operation. Hardware of the embedded system consist of Power Supply, Reset, Oscillator Circuit, Memory i.e. Program and data, Processor (Microcontroller, ARM, PIC, ASIC), Timers, Input/Output circuits, Serial communication ports, SASC (System application specific circuits), Interrupt Controller, Parallel ports. Normally, an embedded system includes the following hardware as shown in Fig. 2.2.

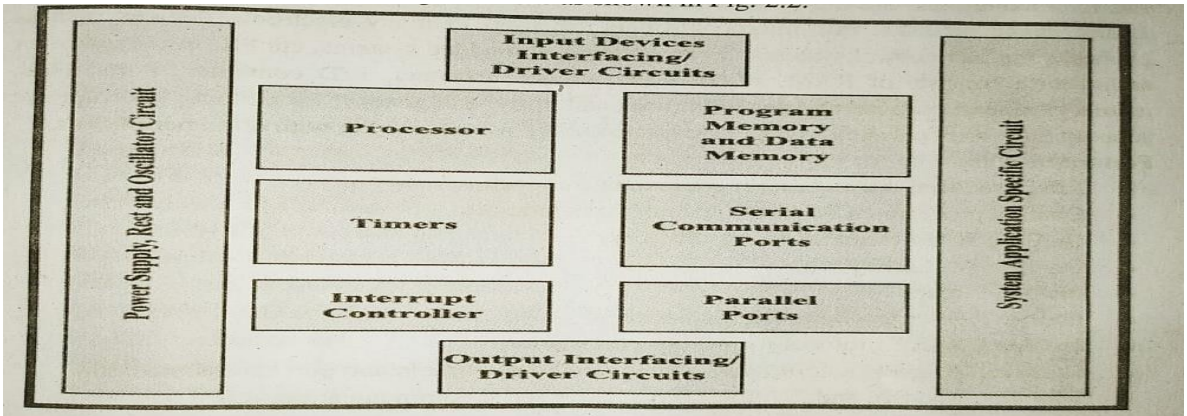

Fig. 2.2: Embedded System Hardware

- **Embedded system software:** The software of an embedded system is written to execute a particular function. The software used in the embedded system is set of instructions i.e. program. The microprocessors or microcontrollers used in the hardware circuits of embedded systems are programmed to perform specific tasks by following the set of instructions. These programs are mainly written using any programming software like Proteus or Lab-view using any programming languages such as C or C++ or embedded C. Then, the program is stored into the microprocessors or microcontrollers memory that are used in the embedded system circuits.
- **Embedded Operating System:** An Embedded Operating System (OS) is a dedicated operating system designed to perform a specific task for a device. The main job of an embedded operating system is to run the code that allows the device to perform its job. The Embedded operating system also allows the device hardware accessible to the software this is running on top of the OS. Embedded operating system are also known as real time operating system (RTOS). The most common examples of embedded operating system around us include Windows Mobile/CE(handheld Personal Data Assistant) Symbian (Cell Phones) and Linux, Palm OS, iOS- Subset of Mac OS X, used in Apples mobile devices.

2.1.4 Embedded processors PIC, ARM, AVR, ASIC

Embedded Processor consists of Control Unit (CU), Execution unit (EU), inbuilt Program and Data Memory, Timer, Interrupts, Serial communication port, Parallel ports, Input and Output Driver Circuits, Power supply, Reset and Oscillator Circuits, System Application Specific Circuits such as ADC, DAC etc.

(a) PIC (Programmable/Peripheral Interface Controllers)

PIC microcontrollers are the smallest microcontrollers which can be programmed to perform a large range of tasks. PIC microcontrollers are used in many electronic devices such as phones, computer control systems, alarm systems, embedded systems, etc PIC microcontroller architecture consists of RAM, ROM, CPU, timers, counters, A/D converter, Ports, Flash memory, general purpose register (GPR), special purpose register (SPR), Stack, Interrupt and supports the protocols such as SPI, CAN, and UART for interfacing with other peripherals.

Features of PIC

- RISC (reduced instruction set computer) architecture.
- On chip program ROM in the form of flash memory.

- On Chip RAM (random access memory)
- On Chip Data EEPROM
- Include Timers.
- Include ADC (Analog to Digital converter).
- Include USART protocol for PC communication.
- Contains I/O ports and I/O port register are bit accessible and port accessible both.
- Include CAN, SPI and 12C PROTOCOL for serial communication.
- Support n-stage pipelining
- Provide interrupts

Application of PIC:

1. Motor Control, Digital Power & Lighting

- Motor Control
- Digital Power
- Lighting
- Automotive
- Home Appliance
- High Temperature for 150C

2. Human Interface

- Graphics Solutions
- Segmented LCD
- Touch Sensing Solutions
- Audio and Specch

3. Connectivity

- Wireless
- USB
- Ethernet
- CAN

(b) AVR (Alf-EgilBogenVegardWollan RISC microcontroller or Advanced Virtual RISC)

AVR was developed in the year 1996 by Atmel Corporation and the architecture of AVR was designerd By Alf-EgilBogen and VegardWollan. AVR and Alf-EgilBogenVegardWollanRISC microcontroller, also known as Advanced Virtual RISC, AVR microcontroller executes most of the instructions in single execution cycle. AVRS are about four times faster than PICS and consumes less power. AVRS can be operated in different power saving modes.

Features of AVR

AVRS provides a wide range of features:

- Internal, self-programmable instruction flash memory up to 256 KB
- In-system programmable (ISP) using serial/parallel low-voltage proprietary interfaces andOn-chip debugging support through JTAG
- Internal data EEPROM up to 4 KB and SRAM up to 16 KB
- External 64 KB little endian data space in some models of AVR
- 8-bit and 16-bit timers
- PWM output, Analog comparator
- 10 or 12-bit A/D converters, with multiplex of up to 16 channels
- 12-bit D/A converters
- Synchronous/asynchronous serial peripherals (UART/USART), Scrial Peripherul Interface Bus (SPI), I'C
- Multiple power-saving sleep modes
- Lighting and motor control (PWM) .controller models
- CAN, USB. Ethernet, LCD, DMA controller support
- Low-operating voltage devices i.e.1.8 V

Applications of AVR

- Signal sensing and Data acquisition
- Motion control and Interface motors
- Displays on LCD
- Interface any type of sensors and transducers
- Interface GSM and GPS
- Control and automation of industrial plants, mechanical & electrical systems
- Automation of heavy machineries
- Developments for UAVS (Unmanned Aerial Vehicles)
- Light sensing, Temperature sensing & controlling devices
- Fire detection & safety devices
- Industrial instrumentation devices
- Process control devices

(c) ARM microcontroller

The ARM (Advanced RISC machine) is a 32-bit Reduced Instructions Set Computer (RISC) microcontroller and introduced by the Acron computers' organization in 1987.The ARM architecture uses a 'Harvard architecture' which support separate data and instruction buses for communicating with the ROM and RAM memories.The ARM microcontrollers support for both low-level and high-level programming languages.

Features of ARM mierocontroller

- Load/store RISC architecture.
- An ARM and Thumb instruction sets i.e. 32-bit instructions can be freely intermixed with16-bit instructions in a program.
- Efficient multi-core processing and easier coding for developers.
- Support multi-processing

- Enhanced power-saving design.
- 64 and 32-bit execution states for scalable high performance.
- Supports Memory Management Unit (MMU) and the Memory Protection Unit (MPU).
- Support for Digital Signal Processing (DSP) algorithms.
- Smaller size, reduced complexity and lower power consumption.
- Floating point support.

Applications of ARM microcontroller

- Smartphones
- Multimedia players
- 3dshandheld game consoles
- Digital cameras
- Tablet computers
- Industrial instrument control systems
- Wireless networking and sensors
- Automotive body system
- Robotics
- Consumer electronics
- Set-top boxes
- Digital television
- Smart watches
- Wireless lan, 802.11, Bluetooth

(d)ASIC (Application-specifie integrated circuit)

An ASIC (application-specific integrated circuit) is a microchip designed for a special anplication, such as a particular kind of transmission protocol or a hand-held computer, You might contrast it with general integrated circuits, such as the microprocessor and the random access memory chips in your PC. ASICS are used in a wide-range of applications, including Auto emission control, environmental monitoring, and personal digital assistants (PDAS). An ASIC can be pre-manufactured for a special application or it can be custom manufactured (typically using components from a "building block" library of components) for a particular customer application.

The advantages of ASIC include the following.

- The small size of ASIC makes it a high choice for sophisticated larger systems.
- As a large number of circuits built over a single chip, this causes high-speed applications.
- ASIC has low power consumption.
- As they are the system on the chip, circuits are present side by side. So, very minimal routing is needed to connect various circuits.
- ASIC has no timing issues and post-production configuration.

The disadvantages of ASIC include the following.

- As these are customized chips they provide low flexibility for programming
- As these chips have to be designed from the root level they are of high cost per unit.
- ASIC have larger time to market margin.

2.2 IoT Definition:

- The internet of things (IoT) is a computing concept that describes the idea of everyday physical objects being connected to the internet and being able to identify themselves to other devices.
- Internet of Things (loT) refers to physical and virtual objects that have unique identities and are connected to the internet to facilitate intelligent applications that make energy, logistics, industrial control, retail, agriculture and many other domains "smarter".
- Internet of things (loT) is a new revolution in which endpoints connected to the internet and driven by the advancements in sensor networks, mobile devices, wireless communications, networking and cloud technologies.

Characteristics of IoT:

- **Dynamic &Self-Adapting**:IoT devices and systems may have the capability to dynamically adapt with the changing contexts and take actions based on their operating conditions, user's context, or sensed environment.For example, the surveillance cameras can adapt their modes (to normal or infra-red night modes) base on whether it is day or night.
- **Self-Configuring:**IoT devices may have self-configuring capability, allowing a large number of devices to work together to provide certain functionality (such as weathe monitoring).
- **Interoperable Communication Protocols:**loT devices may support a number of interoperable communication protocols and can communicate with other devices and also with the infrastructure.
- **Unique Identity:** Each loT device has a unique identity and a unique identifier (such as an IP address or a URI). loT device interfaces allow users to query the devices, monitor their status, and control them remotely, in association with the control. configuration and management infrastructure.
- **Integrated into Information Network:** loT devices are usually integrated into the information network that allows them to communicate and exchange data with other devices and systems.
- **Enormous scale:** The number of devices that need to be managed and that communicate with each other will be at least an order of magnitude larger than the devices connected to the current Internet.

Features of IoT:

- **Connectivity:** Connectivity refers to establish a proper connection between all the things of IoT to IoT platform it may be server or cloud.
- **Analyzing:** After connecting all the relevant things, it comes to real-time analyzing the data collected and use them to build effective business intelligence.
- **Integrating:** loT integrating the various models to improve the user experience as well.
- **Artificial Intelligence:** IoT makes things smart and enhances life through the use of data.
- **Sensing:** The sensor devices used in IoT technologies detect and measure any change in the environment and report on their status.
- **Active Engagement:** IoT makes the connected technology, product, or services to active engagement between each other.

- **Endpoint Management:** It is important to be the endpoint management of all the loT system otherwise; it makes the complete failure of the system.

Advantages and Disadvantages of loT:

Advantages of IoT:

- **Efficient resource utilization:** If we know the functionalíty and the way that how each device work we definitely increase the efficient resource utilization as well as monitor natural resources.
- **Minimize human effort:** As the devices of IoT interact and communicate with each other and do lot of task for us, then they minimize the human effort.
- **Save time:** As it reduces the human effort then it definitely saves out time. Time is the primary factor which can save through IoT platform.
- **Improve security:** Now, if we have a system that all these things are interconnected then we can make the system more secure and efficient.
- **Reduced Waste:** loT makes areas of improvement clear. Current analytics give us superficial insight, but loT provides real-world information leading to more effective management of resources.
- **Enhanced Data Collection:** Modern data collection suffers from its limitations and its design for passive use. loT breaks it out of those spaces, and places it exactly where humans really Want to go to analyze our world, It allows an accurate picture of everything.

Disadvantages of loT

- **Security:** As the loT systems are interconnected and communicate over networks. The system offers little control despite any security measures, and it can be lead the various kinds of network attacks.
- **Privacy:** Even without the active participation on the user, the loT system provides substantial personal data in maximum detail.
- **Complexity:** The designing, developing, and maintaining and enabling the large technology to loT system is quite complicated.
- **Flexibility:** Many are concermed about the flexibility of an loT system to integrate casily with another. They worry about finding themselves with several conflicting or locked systems.
- **Compliance:** loT, like any other technology in the realm of business, must comply with regulations. Its complexity makes the issue of compliance seem incredibly challenging when many consider standard software compliance a battle.

2.2.1 Physical design of IoT:

- **Things of IoT:**
  - The "Things" in IoT usually refers to loT devices which have unique identities and can perform remote sensing, actuating and monitoring capabilities.
  - IoT devices can exchange data with other connected devices and applications (directly or indirectly), or collect data from other devices and process the data either locally or send the data to centralized servers or cloud-based application back-ends for processing the data, or perform some tasks locally and other tasks within the loT infrastructure, based on temporal and space constraints (i.c., memory, processing capabilities, communication latencies and speeds, and deadlines).

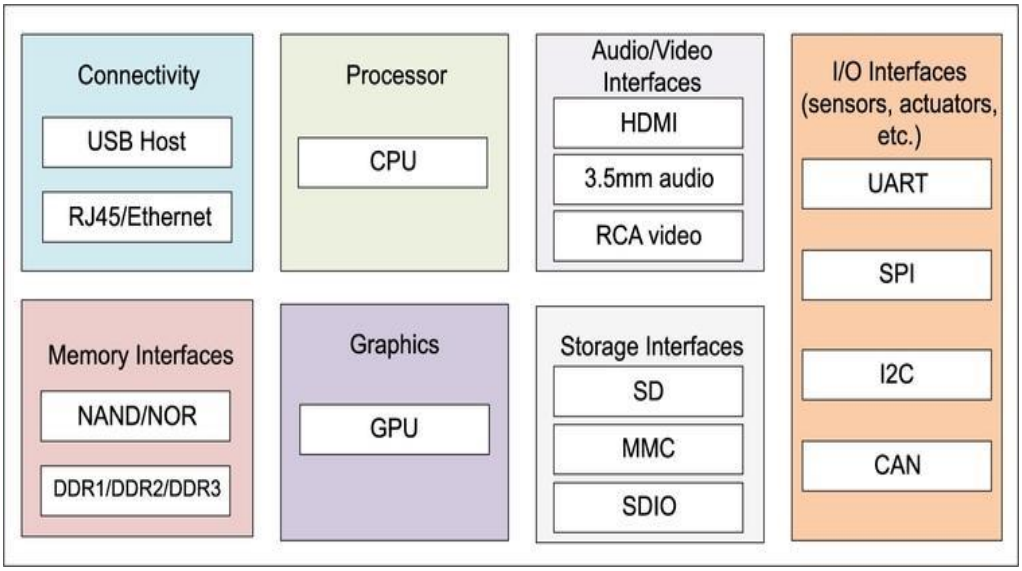

Fig2.3 Generic Block Diagram of an IoT Device

- An IoT device may consist of several interfaces for connections to other devices, both wired and wireless. These include (1) I/O interfaces for sensors, (ii) interfaces for Internet connectivity, (iii) memory and storage interfaces and (iv) audio/video interfaces.
- An loT device can collect various types of data from the on-board or attached sensors. such as temperature, humidity, light intensity. The sensed data can be communicated either to other devices or cloud-based servers/storage.
- loT devices can be connected to actuators that allow them to interact with other physical entities (including non-IoT devices and systems) in the vicinity of the device. For example, a relay switch connected to an IoT device can turn an appliance on/off based on the commands sent to the IoT device over the Internet.
- IoT devices can also be of varied types, for instance, wearable sensors, smart watches, LED lights, automobiles and industrial machines.
- Almost all IoT devices generate data in some form or the other which when processed by data analytics systems leads to useful information to guide further actions locally or remotely.
- For instance, sensor data generated by a soil moisture monitoring device in a garden, when processed can help in determining the optimum watering schedules. Following Figure shows different types of IoT devices.

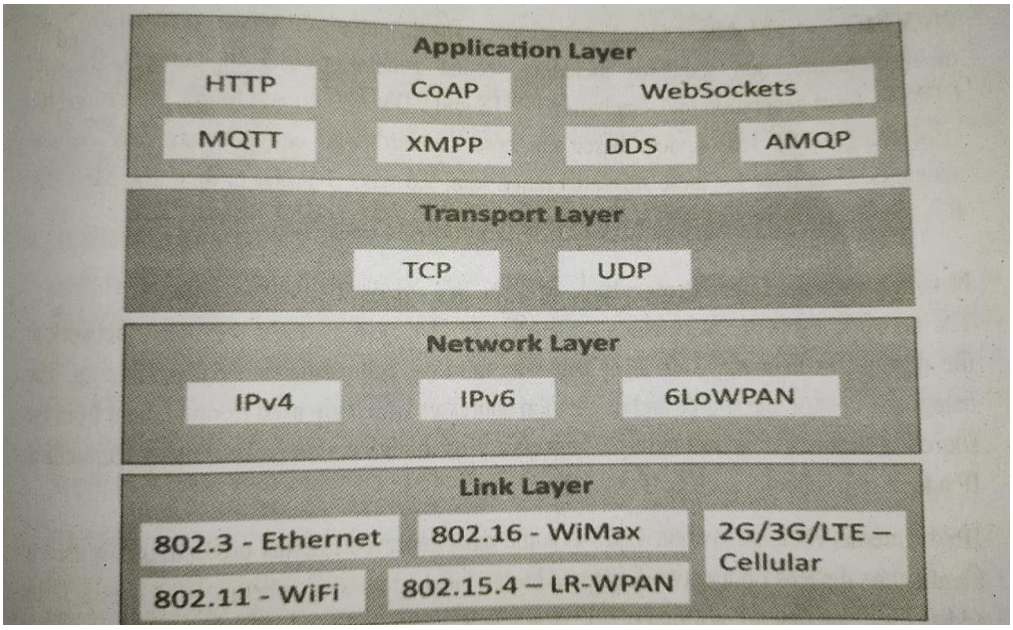

Fig 2.4 IoT Protocols

Link Layer Protocols:

- Link layer protocols determine how the data is physically sent over the network's physical layer or medium (e.g., copper wire, coaxial cable, or a radio wave). Link layer determines how the packets are coded and signaled by the hardware device over the medium to which the host is attached (such as a coaxial cable).
- **802.3-Ethernet:** IEEE 802.3 is a collection of wired Ethernet standards for the link layer. For example, 802.3 is the standard for 10BASE5 Ethernet that uses coaxial cable as a shared medium, 802.3.i is the standard for 10BASE-T Ethernet over copper twisted-pair connections, 802.3.j is the standard for 10BASE-F Ethernet over fiber optic connections, 802.3ae is the standard for 10 Gbit/s Ethernet over fiber, and so on.
- **802.11- WiFi:** IEEE 802.11 is a collection of wireless local area network (WLAN) communication standards, including extensive description of the link layer. 802.11a operates in the 5 GHz band, 802.11b and 802.11g operate in the 2.4 GHz band, 802.11n operates in the 2.4/5 GHz bands, 802.1lac operates in the 5 GHz band and 802.11ad operates in the 60 GHzband. These standards provide data rates from 1 Mb/s to upto 6.75 Gb/s.
- **802.16-WiMax**: IEEE 802.16 is a collection of wireless broadband standards, including extensive descriptions for the link layer (also called WiMax). WIMaxstandards provide data rates from 1.5 Mb/s to 1 Gb/s. The recent update (802.16m) provides data rates of 100 Mbit/s for mobile stations and 1 Gbit/s for fixed stations.
- **802.15.4-LR-WPAN**: IEEE 802.15.4 is a collection of standards for low-rate wireless personal area networks (LR-WPANS). These standards form the basis of specifications for high level communication protocols such as ZigBee. LR-WPAN standards provide data rates from 40 Kb/s 250 Kb/s. These standards provide low-cost and low-speed communication for power constrained devices.

2G/3G/4G- Mobile Communication: There are different generations Mobile communication standards including second generation (2G including GSM and CDMA), third generation (3G - including UMTS and CDMA2000) and fourth generation (4G - including LTE). IOT devices based on these standards can communicate over cellular

networks. Data rates for these standards range from 9.6 Kb/s (for 2G) to upto 100 Mb/s (for4G) and are available from the 3GPP websites.

Network/Internet Layer Protocols: The network layers are responsible for sending of IP datagrams from the source network to the destination network. This layer performs the host addressing and packet routing. The datagrams contain the source and destination addresses which are used to route them from the source to destination across multiple networks. Host identification is done using hierarchical IP addressing schemes such as IPV4 or IPV6.

IPV4: Internet Protocol version 4 (IPV4) is the most deployed Internet protocol that is used to identify the devices on a network using a hierarchical addressing scheme. IPV4 uses a 32-bit address scheme that allows total of 232 or 4,294,967,296 addresses. IPV4 has been succeeded by IPV6. The IP protocols establish connections on packet networks, but do not guarantee delivery of packets. Guaranteed delivery and data integrity are handled by the upper layer protocols (such as TCP).

IPV6: Internet Protocol version 6 (IPV6) is the newest version of Internet protocol and successor to IPv4, IPV6 uses 128-bit address scheme that allows total of 2128 or 3.4 x 1038 addresses.

6LOWPAN: 6LOWPAN (IPV6 over Low power Wireless Personal Area Networks) brings IP protocol to the low-power devices which have limited processing capability, 6LOWPAN operates in the 2.4 GHz frequency range and provides data transfer rates of 250 Kb/s. 6LOWPAN works with the 802.15.4 link layer protocol and defines compression mechanisms for IPV6 datagrams over IEEE 802.15.4-based networks.

Transport Layer Protocols:

The Transport layer protocols provide end-to-end message transfer capability independent of the underlying network. The message transfer capability can be set up on connections, either using handshakes (as in TCP) or without handshakes/acknowledgements (as in UDP). The transport layer provides functions such as error control, segmentation, flow control and congestion control.

TCP: Transmission Control Protocol (TCP) is the most widely used transport layer protocol, that is used by web browsers (along with HTTP, HTTPS application layer protocols), email programs (SMTP application layer protocol) and file transfer (FTP), TCP is a connection oriented and stateful protocol. TCP ensures reliable transmission of packets in-order and also provides error detection capability so that duplicate packets can be discarded and lost packets are retransmitted,

UDP: UDP is a connectionless protocol. UDP is useful for time-sensitive applications that have very small data units to exchange and do not want the overhead of connection setup. UDP is a transaction oriented and stateless protocol. UDP does not provide guaranteed delivery, ordering of messages and duplicate elimination. Higher levels of protocols can ensure reliable delivery or ensuring connections created are reliable.

Application Layer Protocols: Application layer protocols define how the applications interface with the lower layer protocols to send the data over the network. The application data, typically in files, is encoded by the application layer protocol and encapsulated in the transport layer protocol which provides connection or transaction oriented communication over the network. Port numbers are used for application addressing (for example port 80 for

HTTP, port 22 for SSH, etc.). Application layer protocols enable process-to-process connections using ports.

HTTP: Hypertext Transfer Protocol (HTTP) is the application layer protocol that forms the foundation of the World Wide Web (WWW)., HTTP includes commands such as GET, PUT, POST, DELETE, HEAD, TRACE, OPTIONS, etc. The protocol follows a request-response model where a client sends requests to a server using the HTTP commands. HTTP is a stateless protocol and each HTTP request is independent of the cther requests. An HTTP client can be a browser or an application running on the client (e.g., an application running onan IoT device, a mobile application or other software). HTTP protocol uses Universal Resource Identifiers (URIS) to identify HTTP resources.

COAP: Constrained Application Protocol (COAP) is an application layer protocol for machine-to-machine (M2M) applications, meant for constrained environments with constrained devices and constrained networks. Like HTTP, COAP is a web transfer protocol and uses a request-response model, however it runs on top of UDP instead of TCP. COAP uses a client-server architecture where clients communicate with servers using connectionless datagrams. COAP is designed to easily interface with HTTP. Like HTTP, COAP supports methods such as GET, PUT, POST, and DELETE. COAP draft specifications are available on IEFT Constrained environments (CORE) Working Group website.

WebSocket: WebSocket protocol allows full-duplex communication over a single socket connection for sending messages between client and server, WebSocket is based on TCP and allows streams of messages to be sent back and forth between the client and server while keeping the TCP connection open. The client can be a browser, a mobile application or an IoT device.

MQTT: Message Queue Telemetry Transport (MQTT) is a light-weight messaging protocol based on the publish-subscribe model. MQTT uses a client-server architecture where the client (such as an IoT device) connects to the server (also called MQTT Broker) and publishes messages to topics on the server. The broker forwards the messages to the clients subscribed to topics. MQTT is well suited for constrained environments where the devices have limited processing and memory resources and the network bandwidth is low.

XMPP: Extensible Messaging and Presence Protocol (XMPP) is a protocol for mal communication and streaming XML data between network entities. XMPP powers wide range of applications including messaging, presence, data syndication, gaming, multi-party chat vice /video calls. XMPP allows sending small chunks of XML data from one network cu to another in near real-time. XMPP is a decentralized protocol and uses a client. architecture XMPP supports both client-to-server and server-to-server communication part la the context of IoT, XMPP allows real-time communication between IoT devices,

DDS: Data Distribution Service (DDS) is a data-centric middleware standard for device -to-device or machine-to-machine communication. DDS uses a publish-subscribe model where publishers (e.g. devices that generate data) create topics to which subscribers (e.g., devices that want to consume data) can subscribe. Publisher is an object responsible for data distribution and the subscriber is responsible for receiving published data. DDS provides quality-of-service (QoS) control and configurable reliability.

AMOP: Advanced Message Queuing Protocol (AMQP) is an open application layer protocol for business messaging. AMQP supports both point-to-point and publisher/subscriber models, routing and queuing. AMQP brokers receive messages from publishers (c.g., devices or applications that generate data) and route them over connections to consumers (applications that process data). Publishers publish the messages to exchanges which then distribute message copies to queues. Messages are either delivered by the broker to the consumers which have subscribed to the qucues or the consumers can pull the messages from the queues.

2.2.2 Logical design of IoT:

Logical design of an IoT system refers to an abstract representation of the entities and processes without going into the low-level specifics of the implementation.

IoT functional blocks: An IoT system comprises of a number of functional blocks that provide tha system the capabilities for identification, sensing, actuation, communication, and management

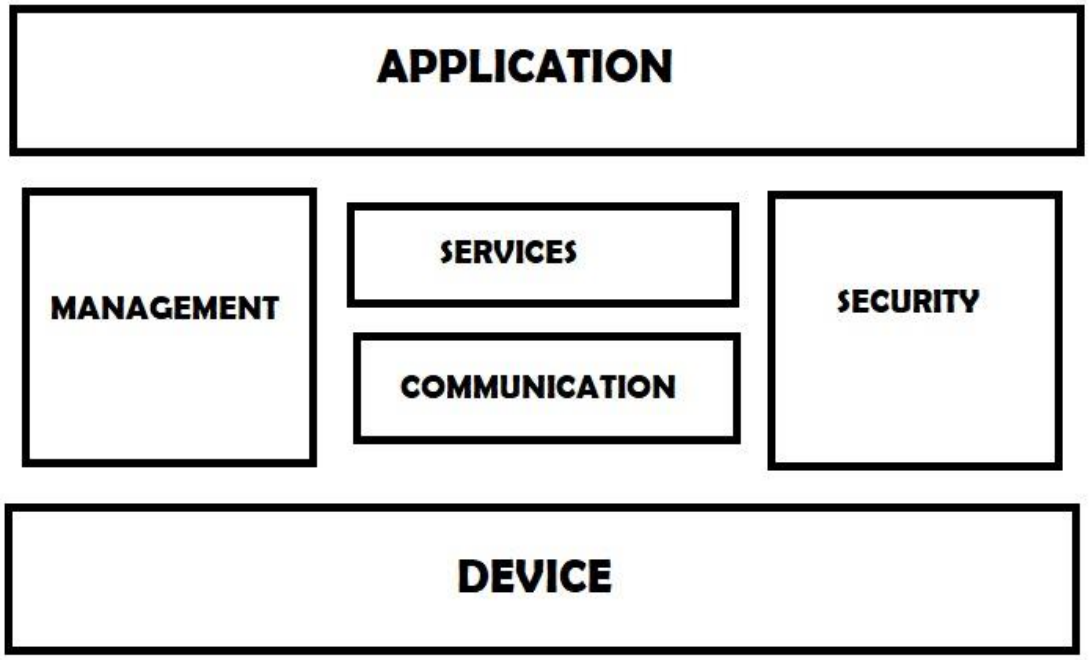

Fig. 2.5 Fundamental block of IoT

- **Device:** An IoT system comprises of devices that provide sensing, actuation monitoring and control functions
- **Communication**: The communication block handles the communication for the IoT system.
- **Services:** An IoT system uses various types of IoT services such as services for device
  monitoring, device control services, data publishing services and services for device discovery.

- **Management:** Management functional block provides various functions to govern the IoT system.
- **Security:** Security functional block secures the IoT system and by providing functions
  such as authentication, authorization, message and content integrity, and data security
- **Application:** IoT applications provide an interface that the users can use to control and monitor various aspects of the IoT system Applications also allow users to view the system status and view or analyze the processed data

IoT Communication models:

Request-Response: Request-Response is a communication model in which the client sends requests to the server and the server responds to the requests. When the server receives a request, it decides how to respond, fetches the data, reternieves resource representations, prepares the response, and then sends the response to the client. Request-Response model is a stateless communication model and each request-response pair is independent of others.

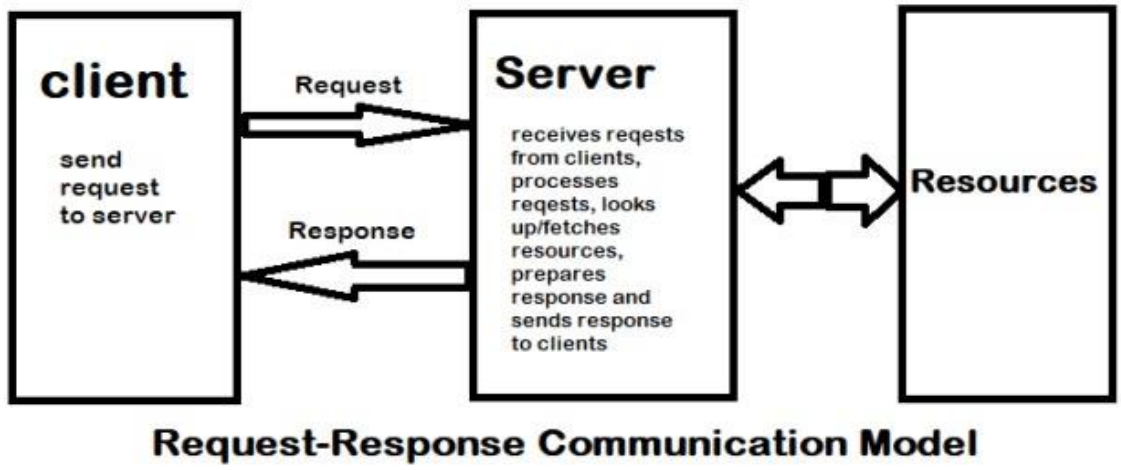

Fig.2.6 Request-Response communication model

Publish-Subscribe: Publish-Subscribe is a communication model that involves publishers brokers and consumers. Publishers are the source of data. Publishers send the data to the topics which are managed by the broker. Publishers are not aware of the consumers. Consumers subscribe to the topics which are managed by the broker. When the broker receives data for a topic from the publisher, it sends the data to all the subscribed consumers.

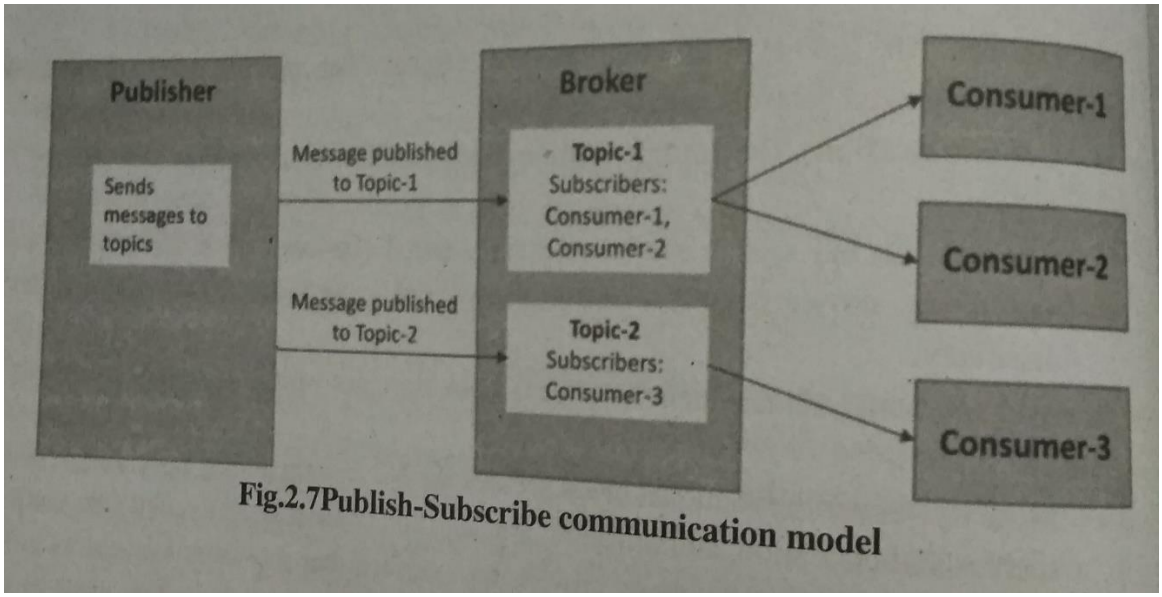

Fig.1.7 Publish-Subscribe communication model

Push-Poll: Push-Pull is a communication model in which the data producers push the data to queues and the consumers pull the data from the queues. Producers do not need to be aware of the consumers. Queues help in decoupling the messaging between the producers and consumers. Queues also act as a buffer which helps in situations when there is a mismatch between the rate at which the producers push data and the rate rate at which the consumers pull data.

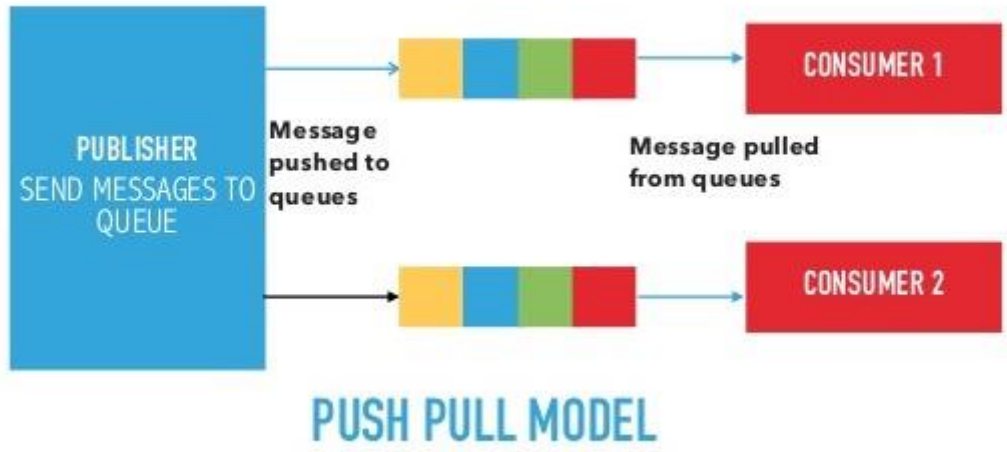

Fig. 2.8Push-Pull communication model

Exclusive Pair: Exclusive Pair is a bi-directional, fully duplex communication model that uses a persistent connection between the client and server. Once the connection is setup it remains open until the client sends a request to close the connection. Client and server can send messages to each other after connection setup. Exclusive pair is a stateful communication model and the server is aware of all the open connections.

Fig.2.9 Exclusive Pair communication model

IoT Communication APIS:

REST-based Communication APIS REST is acronym for REpresentational State Transfer. It is architectural style for distributed

hypermedia systems. It is a set of architectural principles by which you can design web services and web APIS that focus on a system's resources and how resource states are addressed and transferred The REST architectural constraints are as follows:

Client-server By separating the user interface concerns from the data storage concerns, we improve the portability of the user interface across multiple platforms and improve scalability by simplifying the server components.

Stateless - Each request from client to server must contain all of the information necessary to understand the request, and cannot take advantage of any stored context on the server. Session state is therefore kept entirely on the client.

Cacheable- Cache constraints require that the data within a response to a request be implicitly or explicitly labeled as cacheable or non-cacheable. If a response is cacheable, then a client cache is given the right to reuse that response data for later, equivalent requests.

Uniform interface- By applying the software engineering principle of generality to the component interface, the overall system architecture is simplified and the visibility of interactions is improved. In order to obtain a uniform interface, multiple architectural constraints are needed to guide the behavior of components. REST is defined by four interface constraints: identification of resources; manipulation of resources through representations; self-descriptive messages; and, hypermedia as the engine of application state,

Layered system - The layered system style allows an architecture to be composed of hierarchical layers by constraining component behaviour such that each component cannot "sec" beyond the immediate layer with which they are interacting.

Code on demand (optional)- REST allows client functionality to be extended by downloading and executing code in the form of applets or scripts. This simplifies clients by reducing of features required to be pre-implemented.

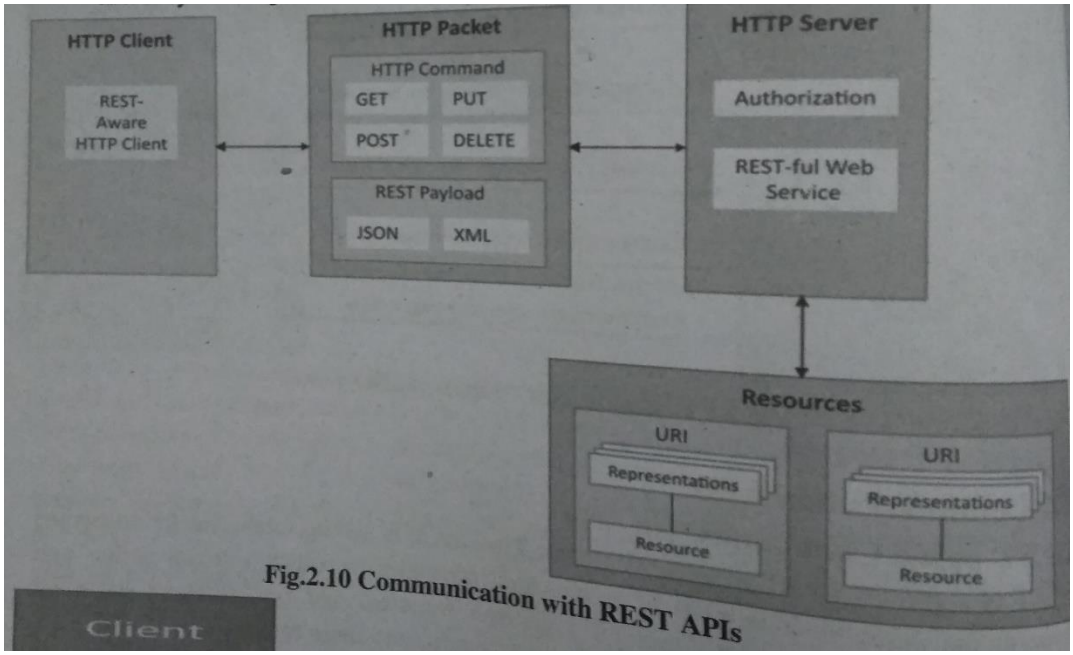

Fig.2.10 communication with REST API's

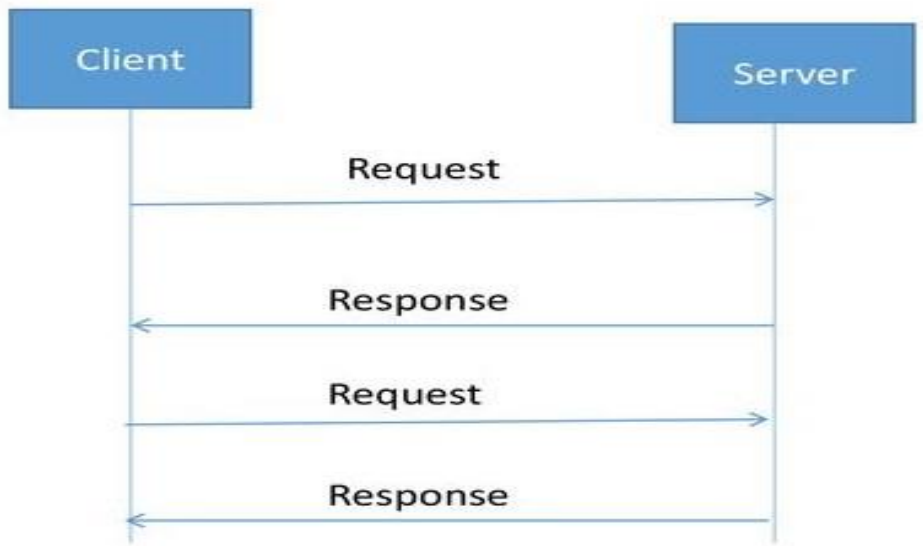

Fig.2.11 Request-Response Model used by REST

A RESTful web service is a "web API" implemented using HTTP and REST principles.

HTTP Methods	Resource Type	Action	Example
GET	Collection URL	List all the resource in a collection	http://example.com/api/tasks/(list all task)
GET	Element URL	Get information	http://example.com/api/tasks/1/(get

		about a resources	info on task-1)
POST	Collection URL	Create a new resource	http://example.com/api/tasks/(create a new task from data provided in the request)
POST	Element URL	Generally not used	
PUT	Collection URL	Replace the entire collection with another collection	http://example.com/api/tasks/(replace the entire collection data provided in the request)
PUT	Element URL	Update resource	http://example.com/api/tasks/1/(update task-1 with data provided in the request)
DELETE	Collection URL	Delete the entire colection	http://example.com/api/tasks/(delete all tasks)
DELETE	Element URL	Delete a resource	http://example.com/api/tasks/1/(delete task-1)

Table 2.1:HTTP request methods and actions

WebSocket-based Communication APIS: WebSocket APIS allow bi-directional, full duplex communication between clients and servers. WebSocket APIs follow the exclusive pair communication model described in previous section and as shown in Figure.

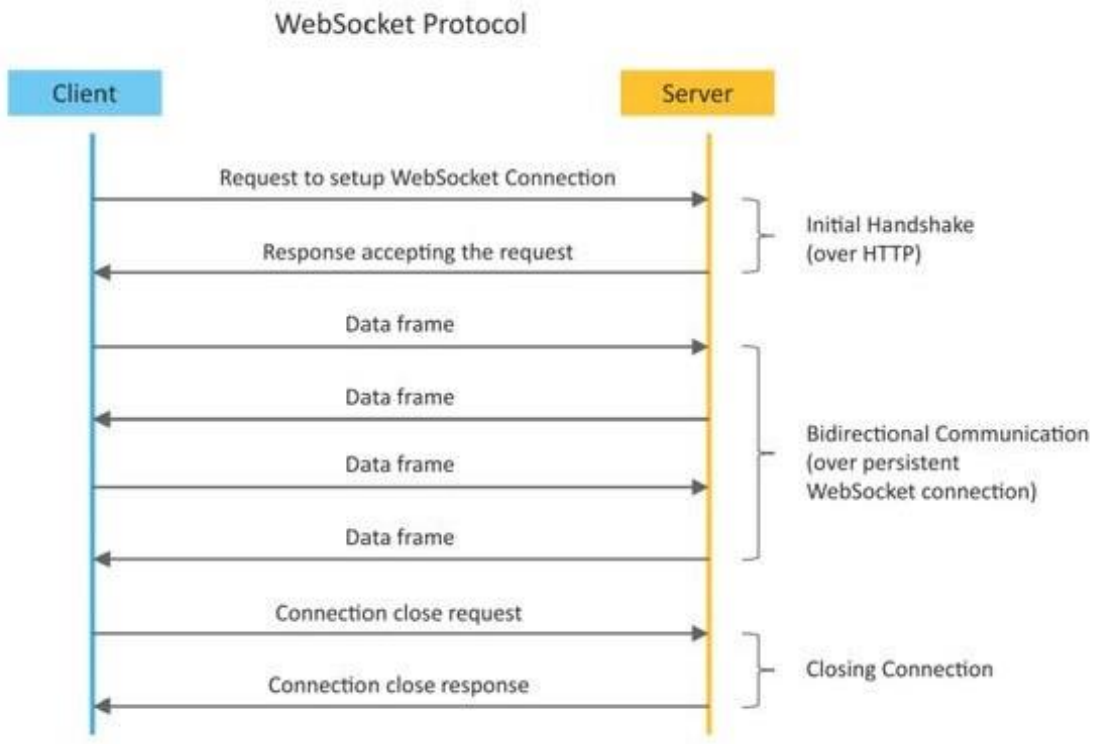

Fig.2.12Exclusive pair model used by WebSocket APIS

Unlike request-response APIS such as REST, the WebSocket APIS allow full duplex communication and do not require a new connection to be setup for each message to be sent. WebSocket communication begins with a connection setup request sent by the client to the server. This request (called a WebSocket handshake) is sent over HTTP and the server interprets it as an upgrade request. If the server supports WebSocket protocol, the serve responds to the WebSocket handshake response. After the connection is setup, the client and server can send data/messages to each other in full-duplex mode. WebSocket APIS reduce the network traffic and latency as there is no overhead tor connection setup and termination requests for cach message. WeBSocket is suitable for IoT applications that have low latency high throughput requirements.

2.2.3 IoT 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 Following are some technologies which play a key role in IoT.

Wireless Sensor Networks: A Wireless Sensor Network (WSN) comprises of distributed devices with sensors which are used to monitor the environmental and physical conditions AWSN consist of a number of end-nodes and routers and a coordinator, End nodes have several sensors attached them. End nodes can also act as routers. Routers are responsible for routing the data packets from end-nodes to the coordinator The coordinator collects the data from all the nodes. Coordinator also acts as a gateway that connects the WSN to the Intemet Some examples of WSNS used in IoT systems are described as follows:

- Weather monitoring systems
- Indoor air quality monitoring systems,
- Soil moisture monitoring systems
- Surveillance systems
- smart grids
- Structural health monitoring systems

ZigBee is one of the most popular wireless technologies used by WSNS. ZigBee specifications are based on IEEE 802.15.4. ZigBee operates at 2.4 GHz frequency and offers data rates upto 250 KB/s and range from 10 to 100 meters depending on the power output and environmental conditions.

Cloud Computing: Cloud computing is a transformative computing paradigm that involves delivering applications and services over the internet. Cloud computing services are offered to user in different forms:

Infrastructure-as-a-Service (IaaS): IaaS provides the users the ability to provision computing and storage resources, These resources are provided to the users as virtual machine instances and virtual storage. Users can start, stop, configure and manage he virtual machine instances and virtual storage. Users can deploy operating systems a applications of their choice on the virtual resources provisioned in the cloud. The cloud service provider manages the underlying infrastructure. Virtual resource provisioned by the users are billed based on a pay-per-use paradigm. Some example of the wide usage of IaaS are automated, policy-driven operations such as backup recovery, monitoring, clustering, internal

networking, website hosting, etc. The service provider is responsible for building the serves and storage, networking firewalls/

security, and the physical data center. Some key players offering IaaS are amazon EC2, Microsoft Azure, Google Cloud Platform, GoGrid, Rackspace, DigitalOcean among others.

Platform-as-a-Service (PaaS): PaaS provides the users the ability to develop and deploy application in the cloud using the development tools, application programming interfaces (APIS), software libraries and services provided by the cloud service provider. The cloud service provider manages the underlying cloud infrastructure including servers, network, operating systems and storage. The users, themselves, are responsible for developing, deploying, configuring and managing applications on the cloud infrastructure. The PaaS environment enables cloud users (accessing them via a webpage) to install and host data sets, development tools and business analytics applications, apart from building and maintaining necessary hardware. Some key players offering PaaS are Bluemix, CloudBees, Salesforce.com, Google App Engine, Heroku, AWS, Microsoft Azure, OpenShift, Oracle Cloud, SAP and OpenShift.

Software-as-a-Service (SaaS): SaaS provides the users

application or the user interface to the application itself. The cloud service provider manages the underlying cloud infrastructure including servers, network, operating systems, storage and application software, and the user is unaware of the underlying architecture of the cloud. Applications are provided to the user through a thin client interface (e.g., a browser). SaaS applications are platform independent and can be accessed from various client devices such as workstations, laptop, tablets and smart- a complete software phones, running different operating systems. Since the cloud service provider manages both the application and data, the users are able to access the applications from anywhere. SaaS lets users easily access software applications - such as emails- over the internet. Most common examples of SaaS are Microsoft Office 360, AppDynamics, Adobe Creative Cloud, Google G Suite, Zoho, Salesforce, Marketo, Oracle CRM, Pardot Marketing Automation, and SAP Business ByDesign.

Benefits of cloud computing services

- Faster implementation and time to value
- Anywhere access to applications and content
- Rapid scalability to meet demand
- Higher utilization of infrastructure investments
- Lower infrastructure, encrgy, and facility costs
- Greater IT staff productivity and across organization
- Enhanced security and protection of information assets

Big Data Analytics:

Big Data analytics is the process of collecting, organizing and analyzing large scts of data (called Big Data) to discover patterns and other useful information. Big Data 'analytics can help organizations to better understand the information contained within the data and will also help identify the data that is most important to the business and future business decisions Analysts working with Big Data typically want the knowledge that comes from analyzing the

data. Big Data Analytics involved several steps starting from data cleansing, data munging (or wrangling), data processing and visualization.

2.2.4 IoT levels and deployment templates:

IoT Level1: System has a single node that performs sensing and/or actuation, stores de performs analysis and host the application as shown in fig. Suitable for modeling low cost low complexity solutions where the data involved is not big and analysis requirement are computationally intensive. An e.g., of IoT Level1 is Home automation.

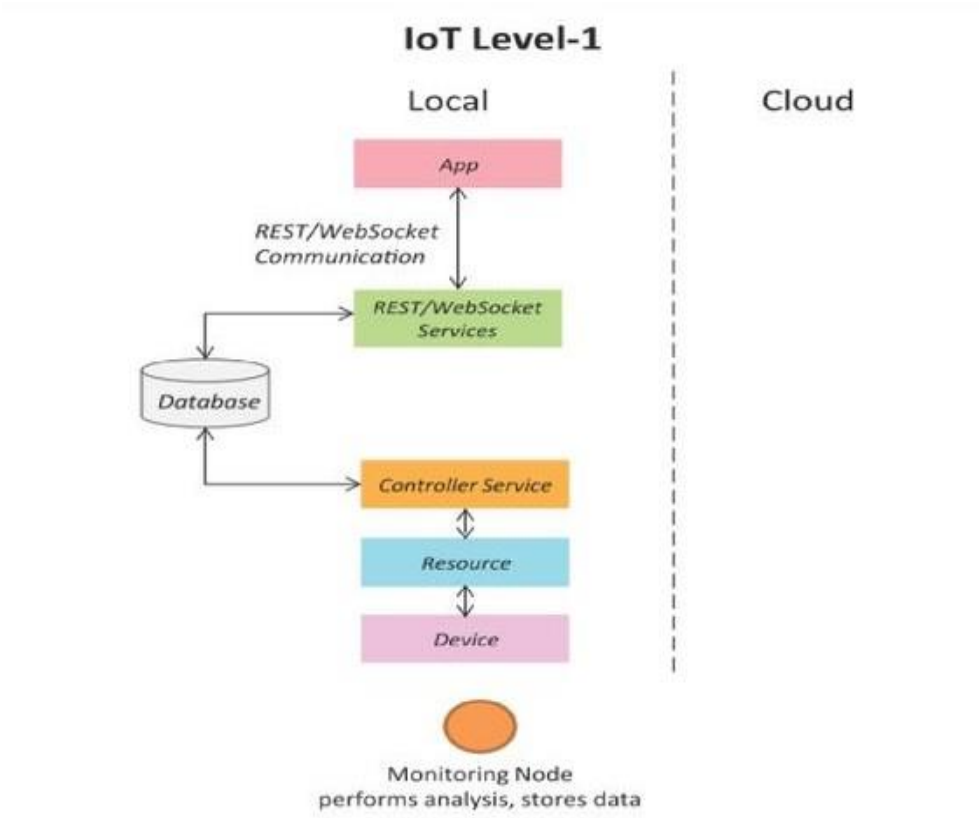

Fig.2.1310T Level-1

IoT Level2: has a single node that performs sensing and/or actuating and local analysis a shown in fig. Data is stored in cloud and application is usually cloud based. Level2 101 systems are suitable for solutions where data are involved is big, however, the primaly analysis requirement is not computationally intensive and can be done locally itself. An e.g of Level2 IoT system for Smart Irrigation.

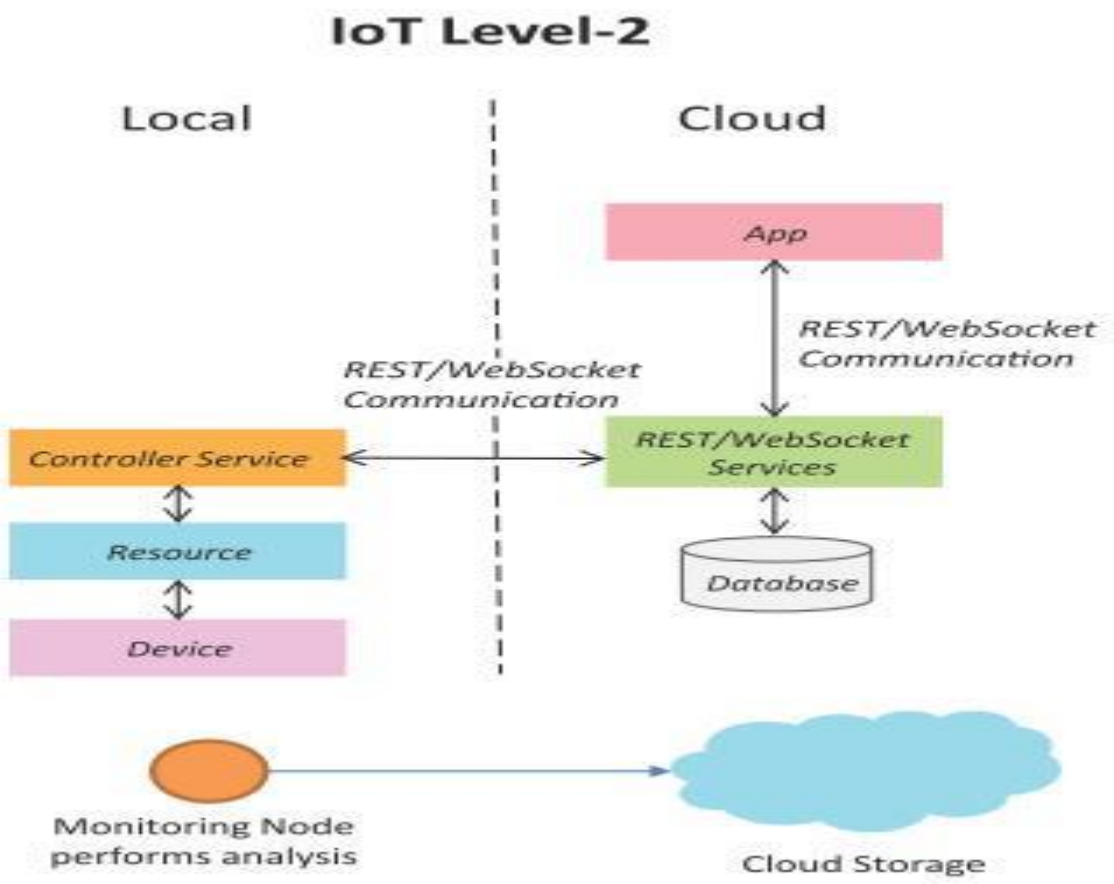

Fig. 2.14IoT Level-2

IoT Leve3: system has a single node, Data is stored and analyzed in the cloud application is cloud based as shown in fig. Level3 IoT systems are suitable for solutions where the data involved is big and analysis requirements are computationally intensive. An example of IoT level3 system for tracking package handling.

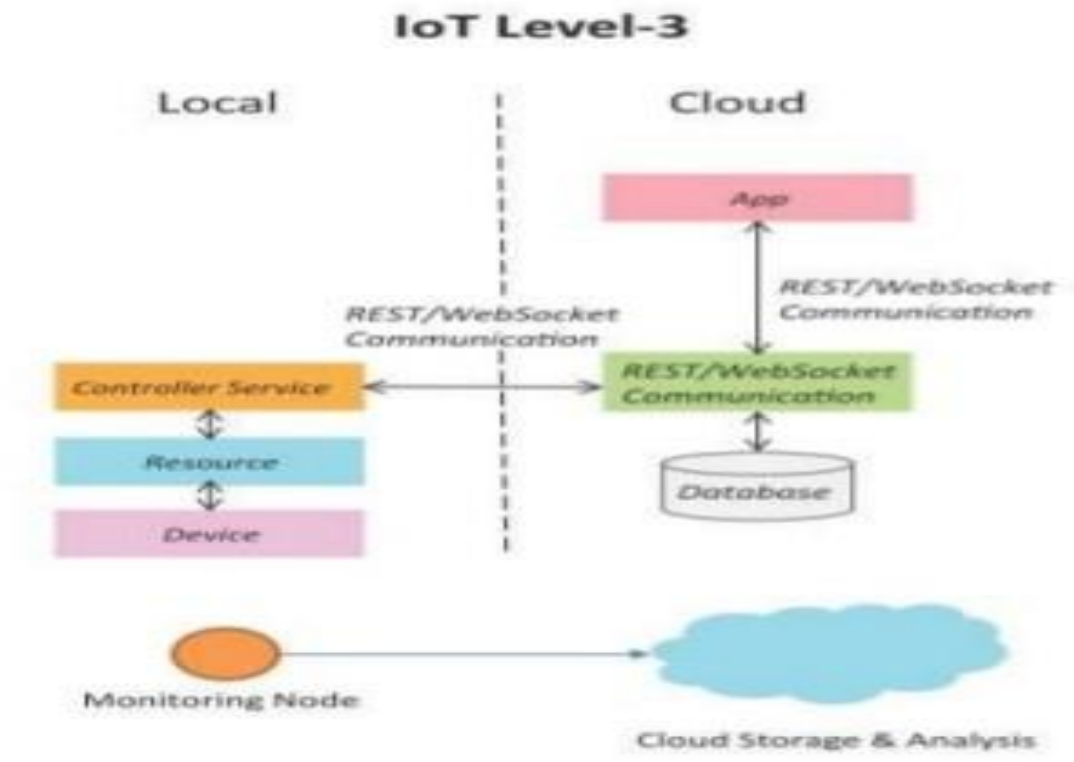

Fig. 2.1510T Level-3

IoT Level4: System has multiple nodes that perform local analysis. Data is stored in the cloud and application is cloud based as shown in fig. Level4 contains local and cloud based observer nodes which can subscribe to and receive information collected in the cloud from IoT devices example of a Level4 IoT system for Noise Monitoring.

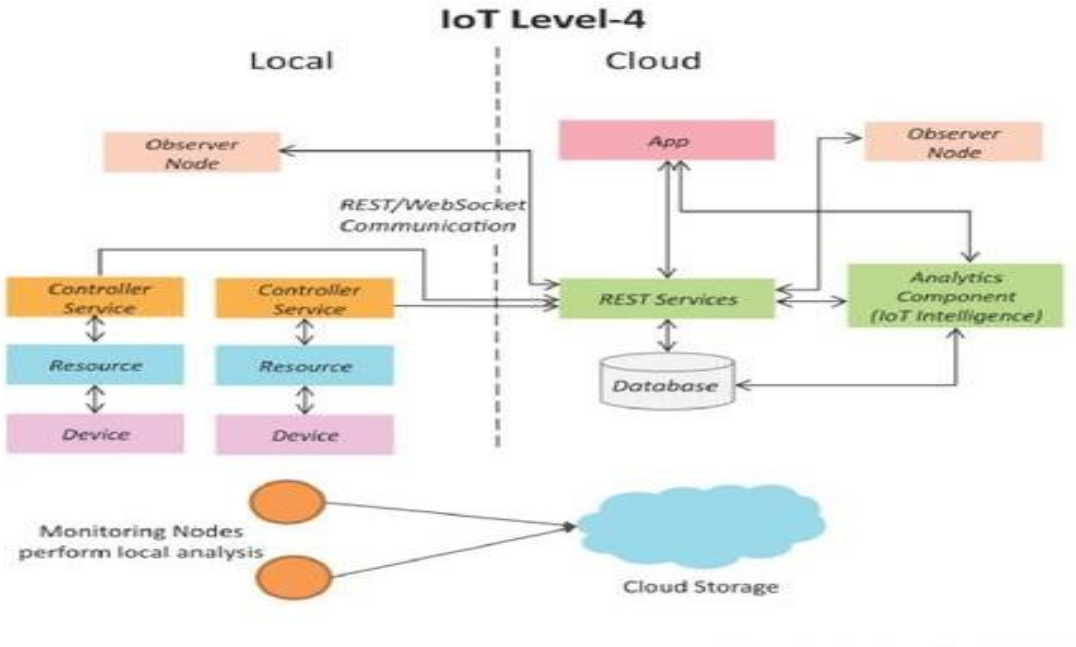

Fig.2.16IoT Level-4

IoT Level5: System has multiple end nodes and one coordinator node as shown in fig. The end nodes that perform sensing and/or actuation. Coordinator node collects data from the end nodes and sends to the cloud. Data is stored and analyzed in the cloud and application is cloud based, Level5 IIOT systems are suitable for solution based on wireless sensor network, in which data involved is big and analysis requirements are computationally intensive. An example of Level5 system for Forest Fire Detection.

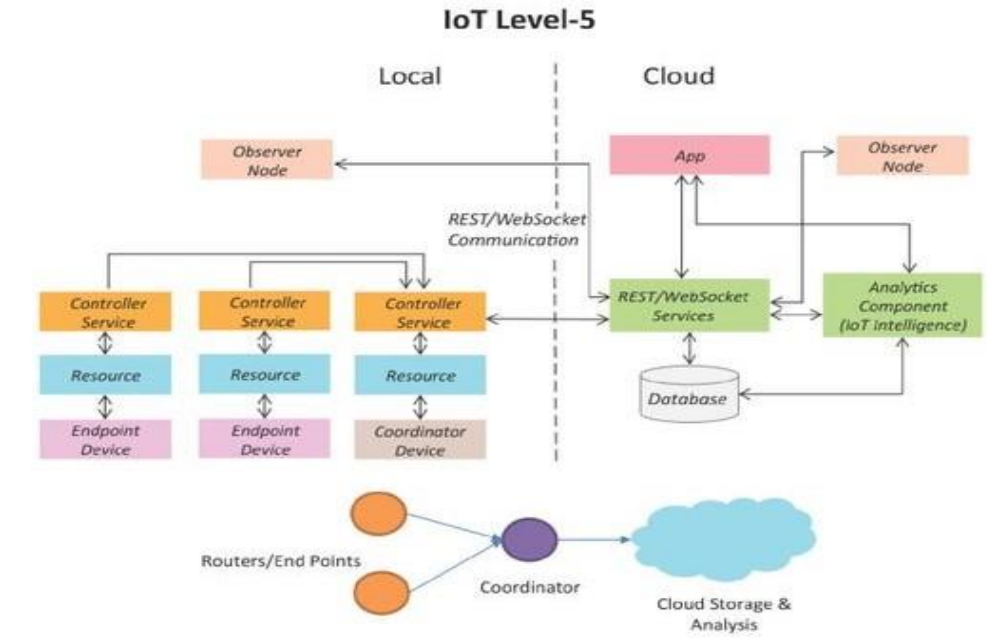

Fig.2.16IoT Level-5

IoT Level6: System has multiple independent end nodes that perform sensing and/or actuation and sensed data to the cloud. Data is stored in the cloud and application is cloud based as shown in fig The analytics component analyses the data and stores the result in the cloud data base, The results are visualized with cloud based application. The centralize controller is aware of the status of all the end nodes and sends control commands to nodes. An example of a Level6 IoT system for Weather Monitoring System.

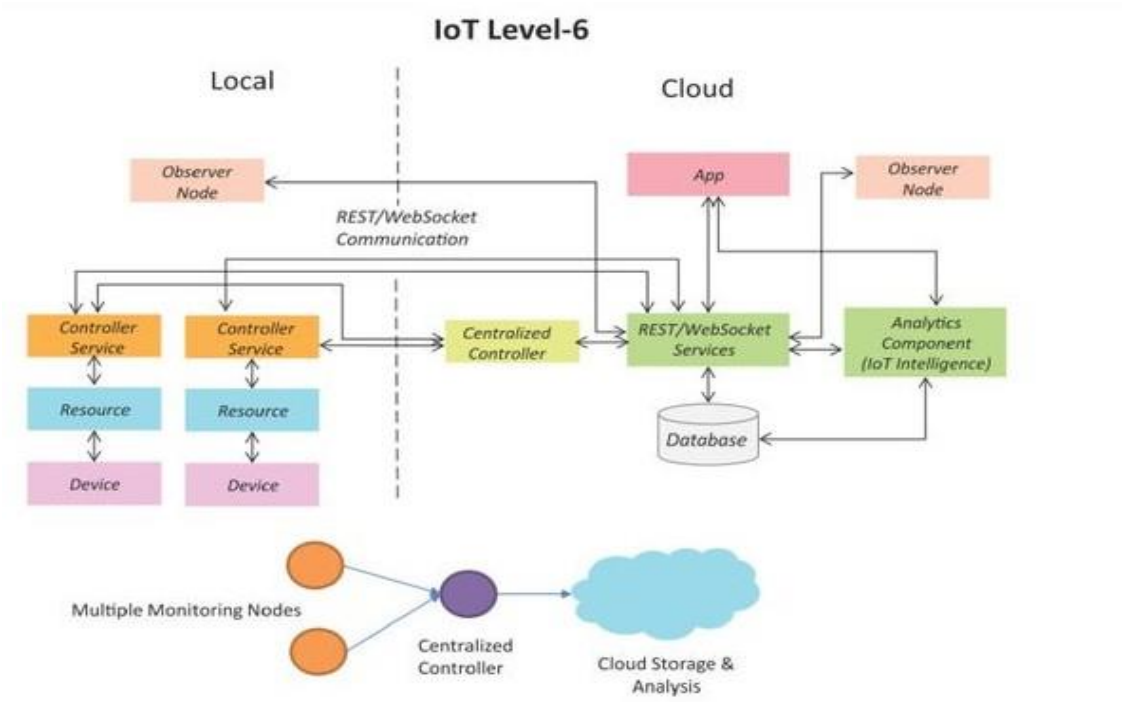

Fig. 2.18IoT Level-6

2.2.5 IoT Issues and Challenges, Applications

Most of Issues and Challenges relevant to IoTare:

- **Data Privacy:** Some manufacturers of smart TVs collect data about their customers to analyze their viewing habits so the data collected by the smart TVs may have a challenge for data privacy during transmission.
- **Data Security:** Data security is also a great challenge. While transmitting data seamlessly, it is important to hide from observing devices on the internet.
- **Insurance Concerns:** The insurance companies installing IoT devices on vehicles collect data about health and driving status in brder to take decisions about insurance,
- **Lack of Common Standard:** Since there are many standards for IoT devices and IoT manufacturing industries. Therefore, it is a big challenge to distinguish between permitted and non-permitted devices connected to the internet.
- **Technical Concerns:** Due to the increased usage of IoT devices, the traffic generated by these devices is also increasing. Hence there is a need to increase network capacity, therefore, it is also a challenge to store the huge amount of data for analysis and further final storage.
- **Security Attacks and System Vulnerabilities:** There has been a IoT of work done in the scenario of IoT security up till now. The related work can be divided into system security, application security, and network security.
    - **System Security:** System security mainly focuses on overall IoT system - identify different security challenges, to design different security framework and to provide proper security guidelines in order to maintain the security o

- **Application security:** Application Security works for IoT application to handle security issues according to scenario requirements.
- **Network security:** Network security communication network for communication of different IoT devices, a network deals with securing the IoT

Applications-Domain Specific IoTs

Home Automation:

- **Smart Lighting:** helps in saving energy by adapting the lighting to the ambient

  conditions and switching on/off or diming the light when needed.

- **Smart Appliances:** make the management easier and also provide status information to the users remotely.
- **Intrusion Detection:** use security cameras and sensors (PIR sensors and door sensors to detect intrusion and raise alerts. Alerts can be in the form of SMS or email sent to the user.

- **Smoke/Gas Detectors:** Smoke detectors are installed in homes and buildings to detect smoke that is typically an early sign of fire. Alerts raised by smoke detectors can be in the form of signals to a fire alarm system. Gas detectors can detect the presence of harmful gases such as CO, LPG etc.,

Cities:

- **Smart Parking:** make the search for parking space easier and convenient for drivers. Smart parking are powered by IoT systems that detect the no, of empty parking sIoTs and send information over internet to smart application back ends.
- **Smart Lighting:** for roads, parks and buildings can help in saving energy.

  Smart Roads: Equipped with sensors can provide information on driving condition, travel time estimating and alert in case of poor driving conditions, traffic condition and accidents.

- **Structural Health Monitoring:** uses a network of sensors to monitor the vibration levels in the structures such as bridges and buildings.
- **Surveillance:** The video feeds from surveillance cameras can be aggregated in cloud based scalable storage solution.
- **Emergency Response:** IoT systems for fire detection, gas and water leakage detection can help in generating alerts and minimizing their effects on the critical infrastructures.

Environment:

- **Weather Monitoring:** Systems collect data from a no. of sensors attached and send the data to cloud based applications and storage back ends. The data

collected in cloud can then be analyzed and visualized by cloud based applications.

- **Air Pollution Monitoring:** System can monitor emission of harmful gases (CO2, CO,NO, NO2 etc) by factories and automobiles using gaseous and meteorological
  sensors. The collected data can be analyzed to make informed decisions on pollutions control approaches.
- **Noise Pollution Monitoring**: Due to growing urban development, noise levels in
  cities have increased and even become alarmingly high in some cities. IoT based noise pollution monitoring systems use a no, of noise monitoring systems that are deployed at different places in a city, The data on noise levels from the station is collected on servers or in the cloud. The collected data is then aggregated to generate noise maps.
- **Forest Fire Detection:** Forest fire can cause damage to natural resources, property and human life, Early detection of forest fire can help in minimizing damage.
- **River Flood Detection:** River floods can cause damage to natural and human resources and human life. Early warnings of floods can be given by monitoring the water level and flow rate. IoT based river flood monitoring system uses a no. of sensor nodes that monitor the water level and flow rate sensors.

Retail:

- **Inventory Management:** IoT systems enable remote monitoring of inventory using data collected by RFID readers.
- **Smart Payments:** Solutions such as contact-less payments powered by technologies such as Near Field Communication(NFC) and Bluetooth.
- **Smart Vending Machines:** Sensors in a smart vending machines monitors its operations and send the data to cloud which can be used for predictive maintenance.

Logistics:

- **Route generation & scheduling:** IoT based system backed by cloud can provide first
  response to the route generation queries and can be scaled up to serve a large transportation network.
- **Fleet Tracking:** Use GPS to track locations of vehicles in real-time.
- **Shipment Monitoring:** IoT based shipment monitoring systems use sensors such as temp, humidity, to monitor the conditions and send data to cloud, where it can be analyzed to detect food spoilage.
- **Remote Vehicle Diagnostics:** Systems use on-board IoT devices for collecting data on Vehicle operation's (speed, RPM etc.,) and status of various vehicle sub systems.

Agriculture:

- **Smart Irrigation:** to determine moisture amount in soil.
- **Green House Control:** to improve productivity.

Industry:

- Machine diagnosis and prognosis
- Indoor Air Quality Monitoring

Health and Life Style:

- Health & Fitness Monitoring
- Wearable Electronics

2.2.6 IoT Devices and its features: Arduino, Uno, Raspberry Pi, Node u

IoT Devices:

- Internet of Things Devices is non-standard devices that connect wirelessly to network with each other and able to transfer the data. IoT devices are enlarging the internet connectivity beyond standard devices such as smartphones, laptops, tablets and desktops.
- There are large varieties of IoT devices available based on IEEE 802.15.4 standard These devices range from wireless motes, attachable sensor-boards to interface-board which are useful for researchers and developers.
- IoT devices include computer devices, software, wireless sensors, and actuators. These IoT devices are connected over the internet and enabling the data transfer among objects or people automatically without human intervention.
- Some of the common and popular IoT devices are given below

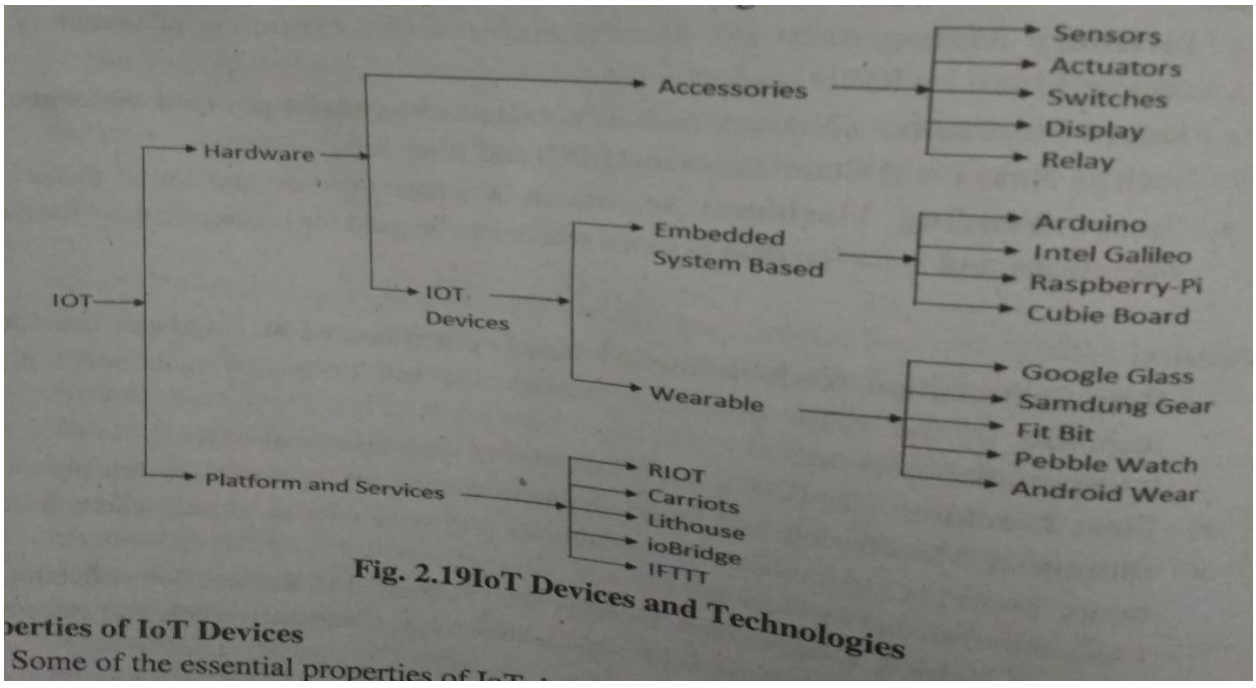

Fig. 2.19IoT Devices and Technologies

perties of IoT Devices

Some of the essential properties of

Fig. 2.1910T Devices and Technologies

Properties of IoT Device:

Some of the essential properties of IoT devices are mention below:

- **Sense:** The devices that sense its surrounding environment in the form of temperature. movement, and appearance of things, etc.
- **Send and receive data:** IoT devices are able to send and receive the data over the network connection.
- **Analyze:** The devices can able to analyze the data that received from the other device over the internet networks.
- **Controlled:** IoT devices may control from some endpoint also. Otherwise, the IoT devices are themselves communicate with each other endlessly leads to the system failure.

Arduino Uno:

- Arduino devices are the microcontrollers and microcontroller kit for building digital devices that can be sense and control objects in the physical and digital world.
- Arduino boards are furnished with a set of digital and analog input/output pins that may be interfaced to various other circuits.
- Some Arduino boards include USB (Universal Serial Bus) used for loading programs from the personal computer.
- Arduino is an open-source electronics platform based on easy-to-use hardware and software.

Properties of Arduino:

- **Inexpensive:** Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than $50,
- **Cross-platform:** The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows. Simple, clear programming environment: The Arduino Software (IDE) is easy-to- use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.
- **Open source and extensible software:** The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.
- **Open source and extensible hardware:** The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money.

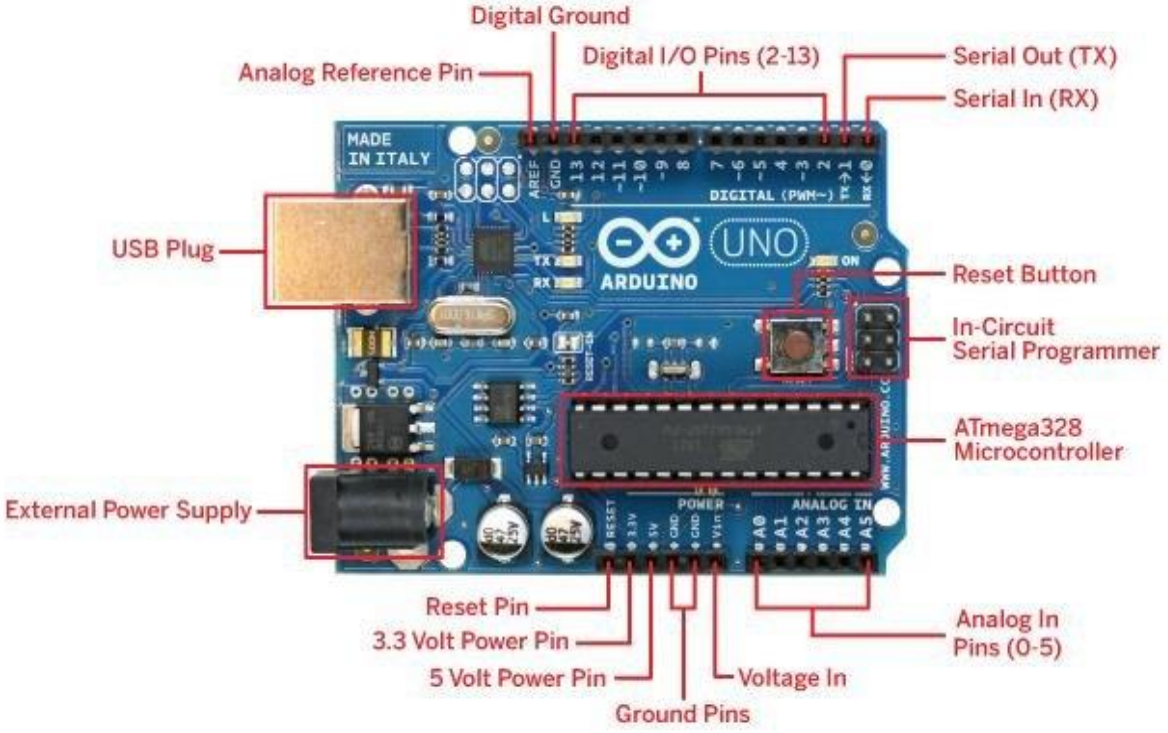

Fig:2.20 Arduino Uno

Raspberry Pi:

The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. The Raspberry Pi is a very cheap computer that runs Linux, but it also provides a set of GPIO (general purpose input/output) pins that you to control electronic components for physical computing and explore the Internet of Things (IoT). Raspberry Pi has an ARMV6 700 MHz single-core processor, a VideoCore IV GPU and 512MB of RAM. it uses an SD card for its operating system and data storage. The Raspberry Pi officially supports Raspbian, a lightweight linux OS based on Debian. Back in 2006, while Eben Upton, his colleagues at University of Cambridge, in conjunction with Pete Lomas and David Braben, formed the Raspberry Pi Foundation.

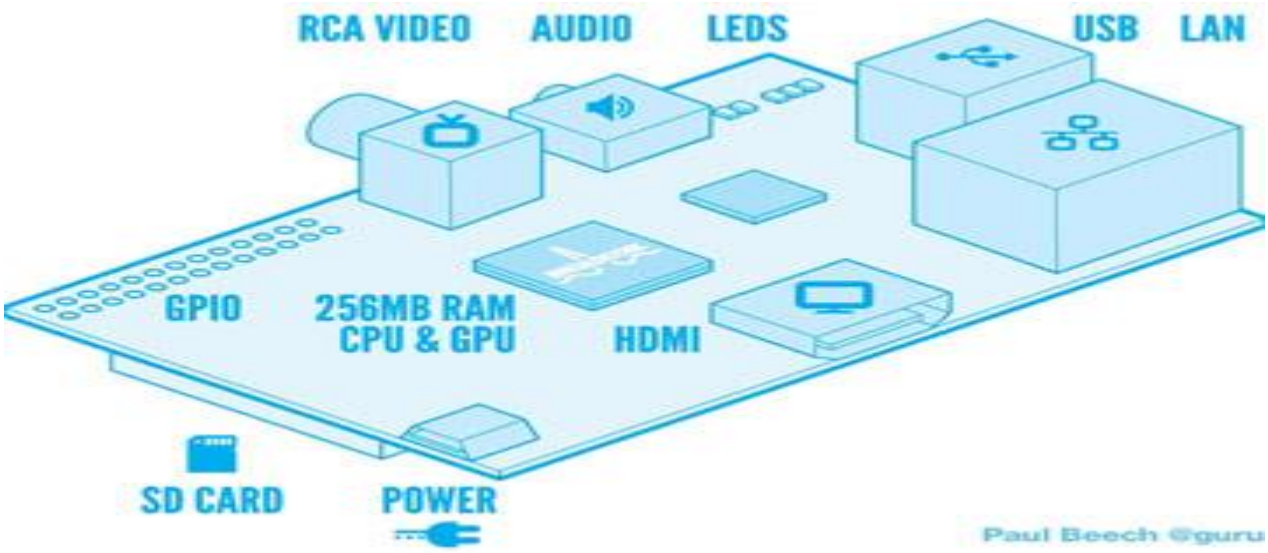

Fig.2.21 Raspberry Pi Model

Components of Raspberry Pi Board

- **ARM CPU/GPU** -- This is a Broadcom BCM2835 System on a Chip (SoC) that's made up of an ARM central processing unit (CPU) and a Videocore 4 graphics processing unit (GPU). The CPU handles all the computations that make a computer work (taking input, doing calculations and producing output), and the GPU handles graphics output.
- **GPIO** - These are exposed general-purpose input/output connection points that will allow the real hardware hobbyists the opportunity to tinker.
- **RCA -** An RCA jack allows connection of analog TVs and other similar output devices.
- **Audio out -** This is a standard 3.55-millimeter jack for connection of audio output devices such as headphones or speakers, There is no audio in.
- **LEDS --** Light-emitting diodes, for all of your indicator light needs.
- **USB--** This is a common connection port for peripheral devices of all types (including your mouse and keyboard), Model A has one, and Model B has two. You can use a USB hub to expand the number of ports or plug your mouse into your keyboard if it has its own USB port.
- **HDMI --** This connector allows you to hook up a high-definition television or other compatible device using an HDMI cable.
- **Power**-- This is a 5v Micro USB power connector into which you can plug your compatible power supply.
- **SD cardslot --** This is a full-sized SD card slot. An SD card with an operating system (OS) installed is required for booting the device. They are available for purchase from the manufacturers, but you can also download an OS and save it to the card yourself If you have a Linux machine and the wherewithal,
- **Ethernet--** This connector allows for wired network access and is only available on the Model B.

Advantages of Different Raspberry Pi Models

- The size of the raspberry pi is in small of credit card
- The price of the raspberry pi is low
- Gathering a set of raspberry pi to work as a server is more effective than the normal server.

Applications of Raspberry pi

The different applications of the raspberry pi model are

- Media steamer
- Tablet computer
- Home automation
- Internet radio
- Controlling robots
- Cosmic Computer
- Arcade machines
- Raspberry pi based projects

Nodeμ

- NodeMCU is an open source loT platiorm.
- The NodeMCU (Node MicroController Unit) 1s an open source software and hardware development environment that is built around a very inexpensive System- 0-2-Chip (SoC) called the ESP8266.
- The ESP8266 can be controlled from your Jocal Wi-Fi network or from the internet (afier port forwarding). The ESP-01 module has GPIO pins that can be programmed to turn an LED or a relay ON/OFF through the internet.
- The module can be programmed using an Arduino/USB-to-TTL converter through the serial pins (RX,TX).
- It uses the Lua scripting and C language with arduinosoftware(using arduino library).
- It has 10 GPIO, every GPIO can be PWM, 12C, 1-wire. It is Wi-Fi enabled device.
- NodeMCU Development boerd is featured with wifi capability, analog pin, digital pins and serial communication protocols.
- NodeMCUDev Kit has Arduino like Analog (i.e. A0) and Digital (D0-D8) pins on its board. It supports serial communication protocols Le. UART, SPI, 12C etc. Using such serial protocols we can connect it with serial devices like 12C enabled LCD display, Magnetometer HMC5883, MPU-6050 Gyro meter + Accelerometer, RTC chips, GPS modules, touch screen displays, SD cards etc.

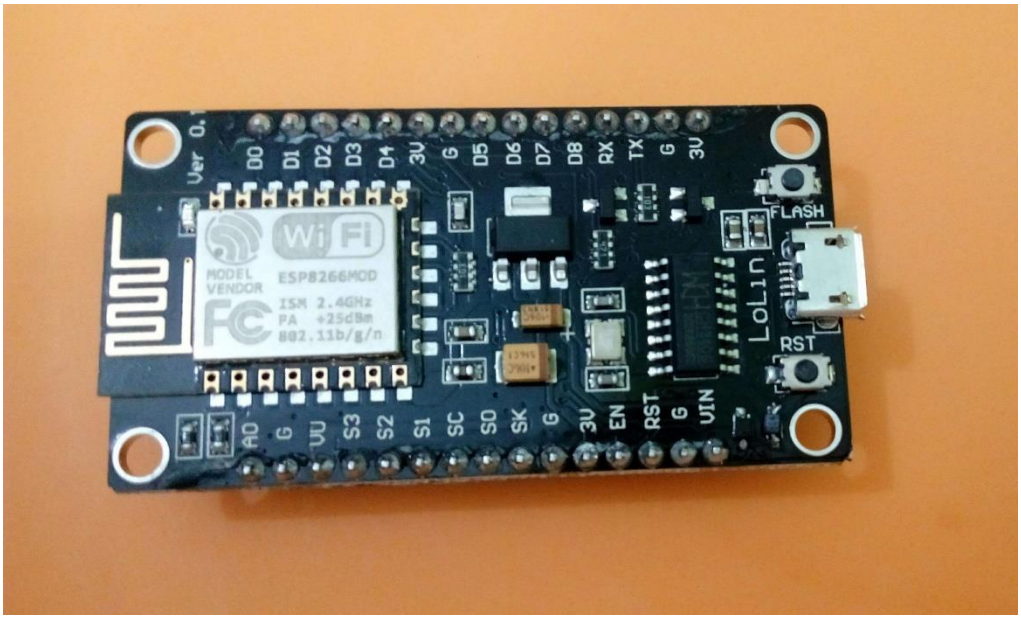

Fig. 2.22 NodeMcuESP8266

2.2.7 Case study on IoT Applications using various Sensors and actuators

Sensors: A sensor is an electronic instrument that is able to measure the physical quantity and generate a considerate output. These output of the sensors are usually in the form of electrical signals. Sensors are placed as such they can directly interact with the environment to sense the input energy with the help of sensing element. This sensed energy is converted into a more suitable form by a transduction element. There are various types of sensors such as position, temperature, pressure, speed sensors, but fundamentally there are two types – analog and digital. The different types come under these two basic types. A digital sensor is incorporated with an Analog-to-digital converter while analog sensor does not have any ADC.

Actuators: An actuator is a device that alters the physical quantity as it can cause a mechanical component to move after getting some input from the sensor. In other words, it receives control input (generally in the form of the electrical signal) and generates a change in the physical system through producing force, heat, motion, etcetera. An actuator can be interpreted with the example of the stepper motor, where an electrical pulse drives the motor. Each time a pulse given in the input accordingly motor rotates in a predefincd amount. A stepper motor is suitable for the applications where the position of the object has to be controlled precisely, for example, robotic arm.

Types of IoT Sensors

Temperature sensors: These devices measure the amount of heat encrgy generated from an object or surrounding area. They find application in air-conditioners, refrigerators and similar devices used for environmental control. They are also used in manufacturing processes, agriculture and health industry. Temperature sensors include thermocouples, thermistors, resistor temperature detcctors (RTDS) and integrated circuits (ICs).

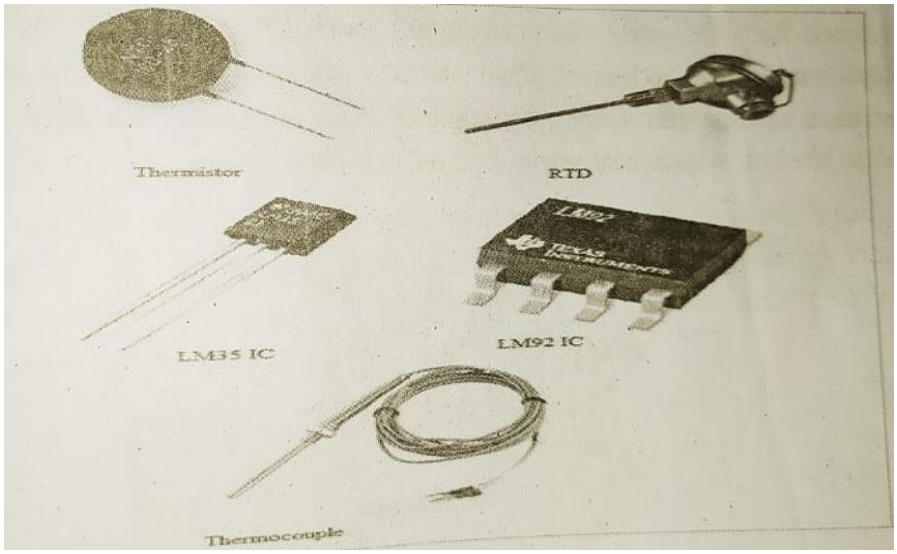

Fig. 2.23 Temperature Sensors

Humidity sensors: The amount of water vapour in air, or humidity, can affect human comfort as well as many manufacturing processes in industries. So monitoring humidity level is important. Most commonly used units for humidity measurement are relative humidity (RH), dew/frost point (D/F PT) and parts per million (PPM).

Fig. 2.24 Humidity Sensor

Motion sensors: Motion sensors are not only used for security purposes but also in automatic door controls, automatic parking systems, automated sinks, automated toilet flushers, hand uryers, energy management systems, etc. You use these sensors in the loT and monitor them from your smartphone or computer. HC-SR501 passive infrared (PIK) sensor is popular motion sensor for hobby projects.

Fig. 2.25 Motion Sensor

Gas sensors: These sensors are used to detect toxic gases. The sensing tehnologies most commonly used are electrochemical, photo-ionisation and semiconductor, With technical advancements and new specifications, there are a multitude of gas sensors availavle to help extend the wired and wireless connectivity deployed in loT applications.

Fig. 2.26 Gas Sensor

Smoke sensors: Smoke detectors have been in use in hormes and industries for quite a long time. With the advent of the IoT, their application has become more convenient and use friendly. Furthermore, adding a wireless connection to smoke detectors enables additional features that increase safety and convenience.

Fig. 2.27 Smoke Sensor

Pressure sensors: These sensors are used in loT systems to monitor systems and devices that are driven by pressure signals. When the pressure range is beyond the threshold level, the device alerts the user about the problems that should be fixed, For exarmple, BMP180 is a popular digital pressure sensor for use in mobile phones, PDAS, GPS navigation devices and outdoor equipment. Pressure sensors are also used in smart vehicles and aircrafts to determine force and altitude, respectively. In vehicle, tyre pressure monitoring system (TPMS) is used to alert the driver when tyre pressure is too low and could create unsafe driving conditions.

Image sensors: These sensors are found in digital cameras, medical imaging systems, night-vision equipment, thermal imaging devices, radars, sonars, media house and biometric stems. In the retail industry, these sensors are used to monitor customers visiting the store through loT network. In offices and corporate buildings, they are used to monitor employees nd various activities through IoT networks.

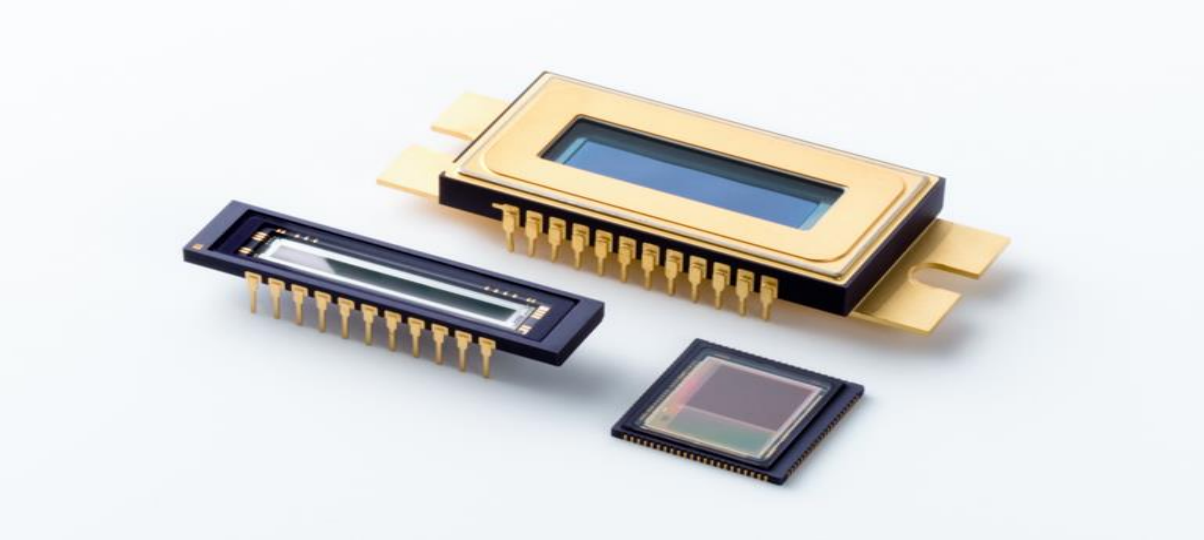

Fig. 2.28 Image Sensor

Accelerometer sensors: These sensors are used in smartphones, vehicles, aircrafts and other applications to detect orientation of an object, shake, tap, tilt, motion, positioning, shock or vibration. Different types of accelerometers include Hall-effect accelerometers, capacitive accelerometers and piezoelectric accelerometers.

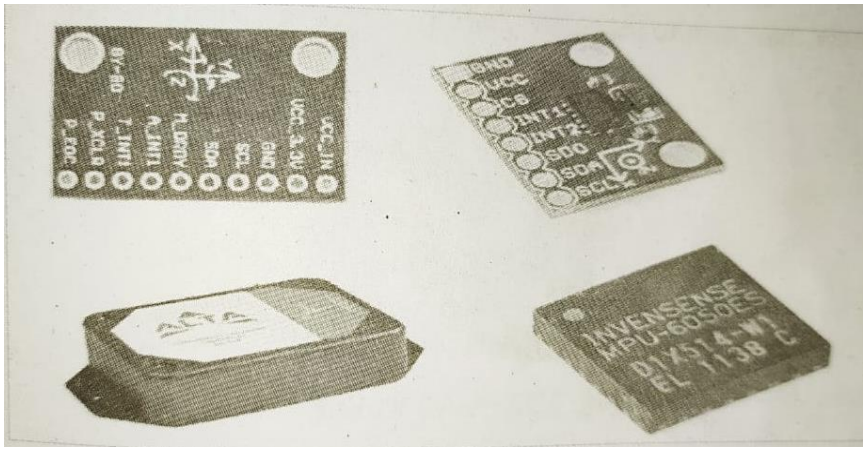

Fig. 2.29 Accelerator Sensors

IR sensors: These sensors can measure the heat emitted by objects. They are used in various loT projects including healthcare to monitor blood flow and blood pressure, smartphones to use as remote control and other functions, wearable devices to detect amount of light, thermometers to monitor temperature and blind-spot detection in vehicles.

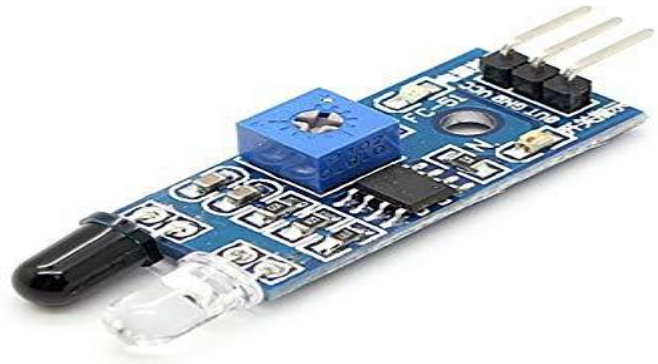

Fig. 2.30 IR Sensor

Proximity sensors: These sensors detect the presence or absence of a nearby object without any physical contact. Different types of proximity sensors are inductive, capacitive photoelectric, ultrasonic and magnetic. These are mostly used in object counters, process monitoring and control.

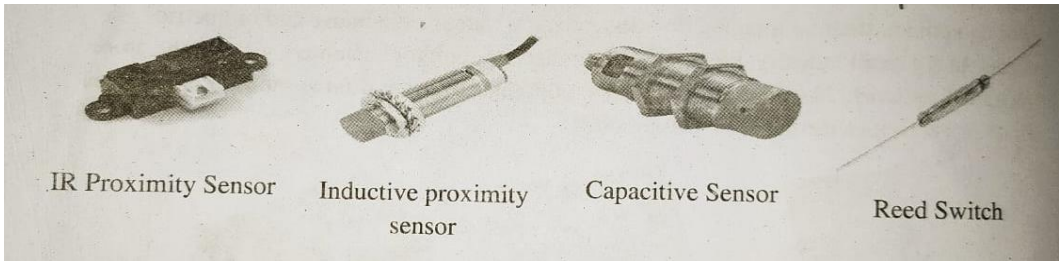

Fig. 2.31 Proximity sensors

Basic actuators you may use in your loT projects

Servo motors: A Servo is a small device that incorporates a two wire DC motor, a gear train, a potentiometer, an integrated circuit, and a shaft (output spine). The shaft can be positioned to specific angular positions by sending the servo a coded signal. Of the three wires that stick out from the servo casing, one is for power, one is for ground, and one is a control input line.It uses the position-sensing device to determine the rotational position of the shaft, so it knows which way the motor must tum to move the shaft to the commanded position.

Fig. 2.32 Servo Motor

Stepper Motor: Stepper motors are DC motors that move in discrete steps. They have multiple coils that are organized in groups called "phases". By energizing cach phase in sequence, the motor will rotate, one step at a time. With a computer controlled stepping, you can achieve very precise positioning and/or speed control.

Fig. 2.33 Stepper Motor

DC motors: Direct Current (DC) motor is the most common actuator used in electronics projects. 1hey are Simple, cheap, and easy to use, DC motors convert electrical motor mechanical energy. They consist of permanent magnets and loops of wire Inside. When current Is applied, the wire loops generate a magnetic field, which reacts against the outside field of the static magnets.

Fig. 2.34 DC Motor

Linear Actuator: A linear actuator is an actuator that creates motion in a straight line, in contrast to the circular motion of a conventional electric motor. Linear actuators are used in machine tools and industrial machinery, in computer peripherals such as disk drives and printers, in valves and dampers, and in many other places where linear motion is required,

Fig. 2.35 Linear Actuator

Relay: A relay is an electrically operated switch. Many relays use an electromagnet to mechanically operate a switch, but other operating principles are also used, such as solid-state relays. The advantage of relays is that it takes a relatively small amount of power to operate the relay coil, but the relay itself can be used to control motors, heaters, lamps or AC circuits which themselves can draw a lot more electrical power.

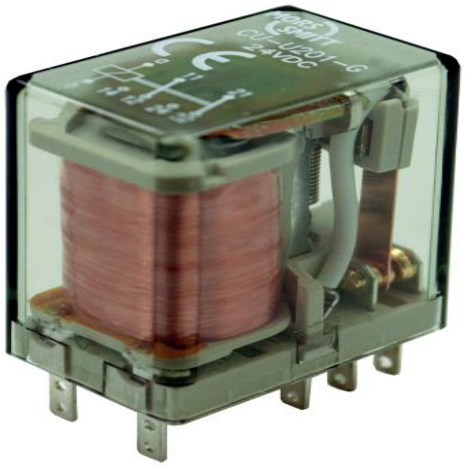

Fig. 2.36 Relay

Solenoid: A solenoid is simply a specially designed electromagnet. Solenoids are inexpensive, and their use is primarily limited to on-off applications such as latching, locking, and triggering. They are frequently used in home appliances (e.g. washing machine valves), office equipment (e.g. copy machines), automobiles (e.g. door latches and the starter solenoid), pinball machines (e.g., plungers and bumpers), and factory automation.

Fig. 2.37Solenoid

References:

• https://data-flair.training/blogs/iot-applications/

• https://books.google.co.in/books?id=JPKGBAAAQBAJ&pg=PA45&source-gbs_toc_r&cad=2#v=onepage&q&f=false

• https://www.tutorialspoint.com/arduino/arduino_board_description.htm#

ttps://kainjan1.files.wordpress.com/2018/01/chapter-1_iot.pdf

• https://www.tutorialspoint.com/internet_of_things/internet_of_things_tutorial.pdf

https://www.iare.ac.in/sites/default/files/lecture_notes/IOT%20LECTURE%20NOTES_IT.pdf

https://components 101.com/microcontrollers/arduino-uno

https://computer.howstuffworks.com/raspberry-pi2.htm

https://www2.deloitte.com/content/dam/insights/us/articles/iot-primer-iot-technologies-applications/DUP_1102_InsideTheInternetOfThings.pdf

https://techdifferences.com/difference-between-sensors-and-actuators.html

https://electronicsforu.com/technology-trends/tech-focus/iot-sensors

https://iotbytes.wordpress.com/basic-iot-actuators/

• https://en.wikipedia.org/wiki/NodeMCU

https://www.electronicwings.com/nodemcu/introduction-to-nodemcu

https://www.instructables.com/id/Programming-ESP8266-ESP-12E-NodeMCU-Using-Arduino-/

• https://datafloq.com/read/3-major-challenges-facing-future-iot/2729

Sample Multiple Choice Questions:

1. IoT stands for

    a. Internet of Technology

    b. Intranet of Things

    c. Internet of Things

    d. Information of Things

2. Which is not the Feature of IoT

    a. Connectivity

    b.Self-Configuring

    c. Endpoint Management

    d. Artificial Intelligence

3. Which not anloT Communication model

    a. Request-Response

    b. Publish-Subscribe

    c. Push-Producer

    d. Exclusive Pair

4. WSN Stands for

    a. Wide Sensor Network

    b. Wireless Sensor Network

    Wired Sensor Network

    d. None of these

5. Devices that transforms electrical signals into physical movements.

    a. Sensors

    b. Actuators

    c. Switches

    d. display