# Unit 1: Introduction to Internet of Things

## Introduction to Internet of Things (IoT)

**Internet of Things** refers to the network of physical devices, vehicles, home appliances, and other items embedded with electronics, software, sensors, and network connectivity, allowing them to collect and exchange data. The IoT enables these devices to interact with each other and with the environment and enables the creation of smart systems and services.

Some examples of IoT devices include:

- Smart home devices such as thermostats, lighting systems, and security systems.
- Wearables such as fitness trackers and smart watches.
- Healthcare devices such as patient monitoring systems and wearable medical devices.
- Industrial systems such as predictive maintenance systems and supply chain management systems.
- Transportation systems such as connected cars and autonomous vehicles.

There are four main components used in IoT:

1. **Low-power embedded systems:** Less battery consumption, high performance are the inverse factors that play a significant role during the design of electronic systems.

2. **Cloud computing:** Data collected through IoT devices is massive and this data has to be stored on a reliable storage server. This is where cloud computing comes into play. The data is processed and learned, giving more room for us to discover where things like electrical faults/errors are within the system.

3. **Availability of big data:** We know that IoT relies heavily on sensors, especially in real-time. As these electronic devices spread throughout every field, their usage is going to trigger a massive flux of big data.

4. **Networking connection:** In order to communicate, internet connectivity is a must where each physical object is represented by an IP address. However, there are only a limited number of addresses available according to the IP naming. Due to the growing number of devices, this naming system will not be feasible anymore. Therefore, researchers are looking for another alternative naming system to represent each physical object.

**IoT Enablers:**

- RFIDs: uses radio waves in order to electronically track the tags attached to each physical object.

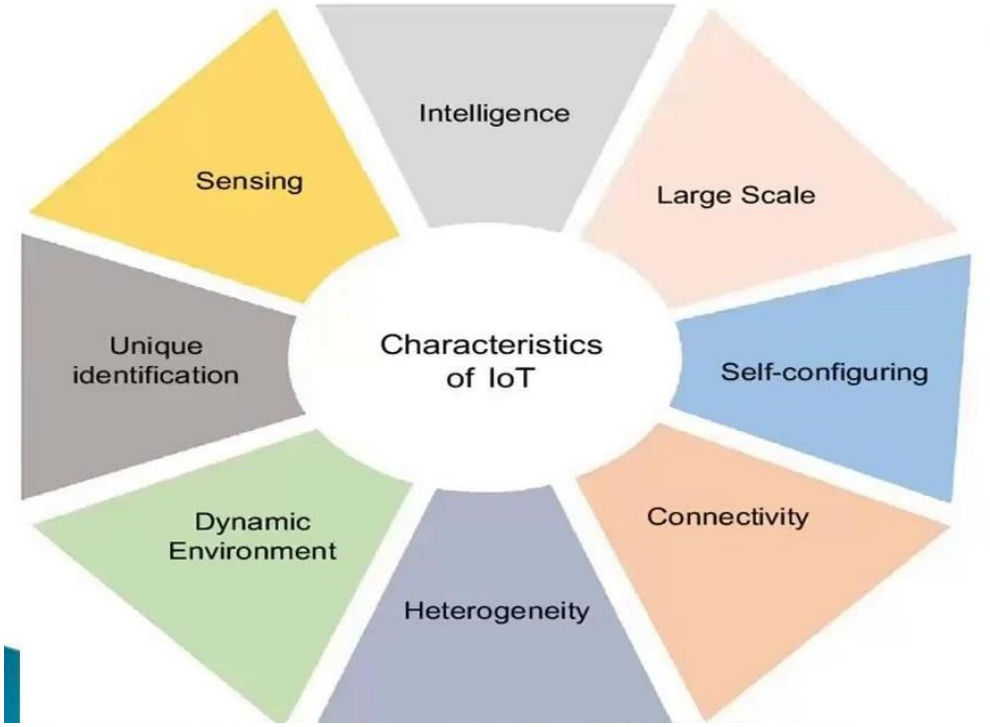
- Sensors: devices that are able to detect changes in an environment (ex: motion detectors).

- Nanotechnology: as the name suggests, these are extremely small devices with dimensions

usually less than a hundred nanometers.

- Smart networks: Various smart networks like mesh topology.

**Characteristics of IoT:**




- **Massively scalable and efficient:**

  The number of things connected in IoT is increasing day by day. Hence, an IoT setup should be capable of handling the massive growth. The data generated as an outcome is huge, and it should be handled properly.

- **Connectivity**

  Connectivity is an important requirement of the IoT infrastructure. Things of IoT should be connected to the IoT infrastructure. Anyone, anywhere, anytime can connect, this should be guaranteed at all times. For example, the connection between people through Internet devices like mobile phones, and other gadgets, also a connection between Internet devices such as routers, gateways, sensors, etc.

- **Intelligence and Identity**

  The extraction of knowledge from the generated data is very important. For example, a sensor generates data, but that data will only be useful if it is analysed properly. Each IoT device has a unique identity. This identification is helpful in tracking the equipment and at times for querying its status.

- **Dynamic and Self-Adapting (Complexity)**

  IoT devices should be dynamically adaptive to changing settings and situations. Assume a camera meant for surveillance. It should be adaptable to work in different conditions and different light intensity.

- **Architecture**

  IoT Architecture shall be hybrid in nature. It should be, supporting different manufacturers products to function in the IoT network. IoT is possible when multiple domains come together.

- **Safety**

  Data security and safety is the major challenge as the devices are connected to the internet. There is a danger of the sensitive personal details of the users getting stolen or compromised. Also, the equipment involved is huge. Therefore, equipment safety is also critical.

- **Self-Configuring**

  This is one of the most important characteristics of IoT. IoT devices are able to upgrade their software in agreement with requirements with a minimum of user participation. Additionally, they can set up the network, allowing for the addition of new devices to an already-existing network.

- **Interoperability and protocols**

  IoT devices use standardized protocols and technologies to ensure they can communicate with each other and other systems. Interoperability is one of the key characteristics of the Internet of Things (IoT). It refers to the ability of different IoT devices and systems to communicate and exchange data with each other, regardless of the underlying technology or manufacturer.

  Interoperability is critical for the success of IoT, as it enables different devices and systems to work together seamlessly and provides a seamless user experience. Without interoperability, IoT systems would be limited to individual silos of data and devices, making it difficult to share information and create new services and applications.

  To achieve interoperability, IoT devices, and systems use standardized communication protocols and data formats. These standards allow different devices to understand and process data in a consistent and reliable manner, enabling data to be exchanged between devices and systems regardless of the technology used.

**Applications of IoT:**

1. Smart Grids and energy saving
2. Smart cities
3. Smart homes
4. Healthcare
5. Earthquake detection
6. Radiation detection/hazardous gas detection
7. Smartphone detection
8. Water flow monitoring
9. Traffic monitoring
10. Wearables

**Advantages and Disadvantages of IoT**

Internet of things facilitates the several advantages in day-to-day life in the business sector. Some of its benefits are given below:

1. **Efficient resource utilization**: In IoT all the devices are connected to achieve efficiency If the functionality and the way that how each device work are known, it will increase the efficient resource utilization as well as monitor natural resources.
2. **Minimize human effort:** As the devices of IoT interact and communicate with each other and do lot of tasks hence they minimize the human effort.
3. **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.
4. **Enhance Data Collection:** As the data is collected through sensors, a huge amount of data is collected.
5. **Improve security:** As all these things are interconnected, the system can be made more secure and efficient.

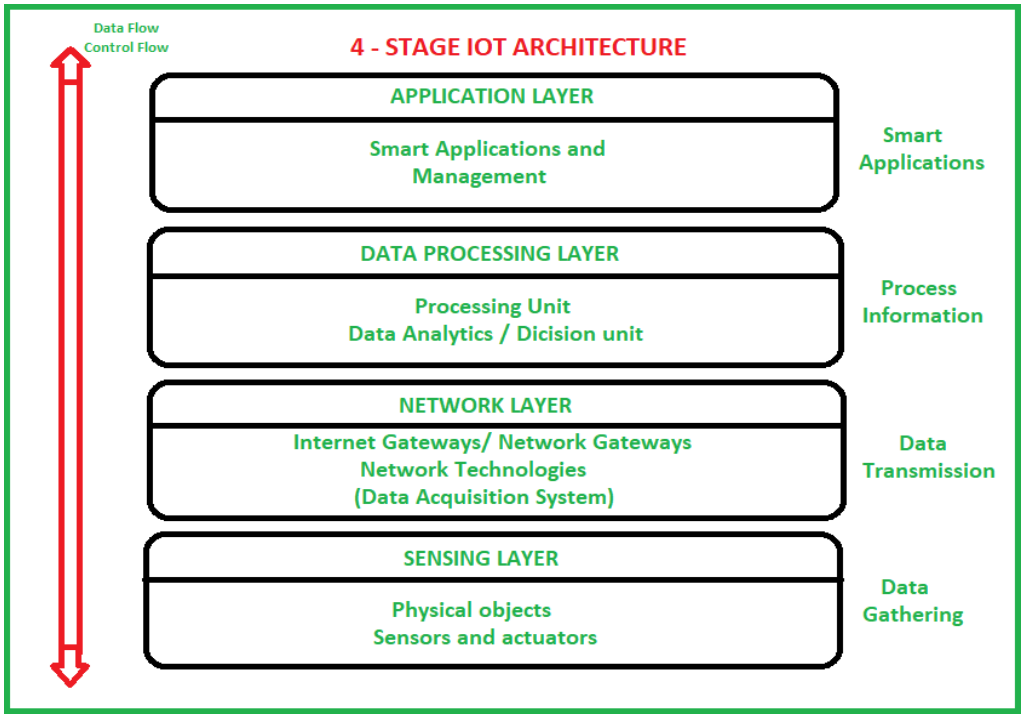
**Disadvantages of IoT**

As the Internet of things facilitates a set of benefits, it also creates a significant set of challenges. Some of the IoT challenges are given below:

1. **Security:** As the IoT 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.
2. **Privacy:** Even without the active participation on the user, the IoT system provides substantial personal data in maximum detail.
3. **Complexity:** The designing, developing, and maintaining and enabling the large technology to IoT system is quite complicated.

## Architecture of Internet of Things (IoT)

Internet of Things (IoT) technology has a wide variety of applications and use of Internet of Things is growing so faster. Depending upon different application areas of Internet of Things, it works accordingly as per it has been designed/developed. But it has not a standard defined architecture of working which is strictly followed universally. The architecture of IoT depends upon its functionality and implementation in different sectors. Still, there is a basic process flow based on which IoT is built.




There is 4 layers are present that can be divided as follows:

Sensing Layer, Network Layer, Data processing Layer, and Application Layer. These are explained as following below.

1. Sensing Layer –
   Sensors, actuators, devices are present in this Sensing layer. These Sensors or Actuators accepts data (physical/environmental parameters), processes data and emits data over network.

2. Network Layer –

   Internet/Network gateways, Data Acquisition System (DAS) are present in this layer. DAS

performs data aggregation and conversion function (Collecting data and aggregating data then converting analog data of sensors to digital data etc). Advanced gateways which mainly opens up connection between Sensor networks and Internet also performs many basic gateway functionalities like malware protection, and filtering also sometimes decision making based on inputted data and data management services, etc.

3. Data processing Layer –

   This is processing unit of IoT ecosystem. Here data is analyzed and pre-processed before sending it to data center from where data is accessed by software applications often termed as business applications where data is monitored and managed and further actions are also prepared. So here Edge IT or edge analytics comes into picture.

4. Application Layer –

   This is last layer of 4 stages of IoT architecture. Data centers or cloud is management stage of data where data is managed and is used by end-user applications like agriculture, health care, aerospace, farming, defense, etc.

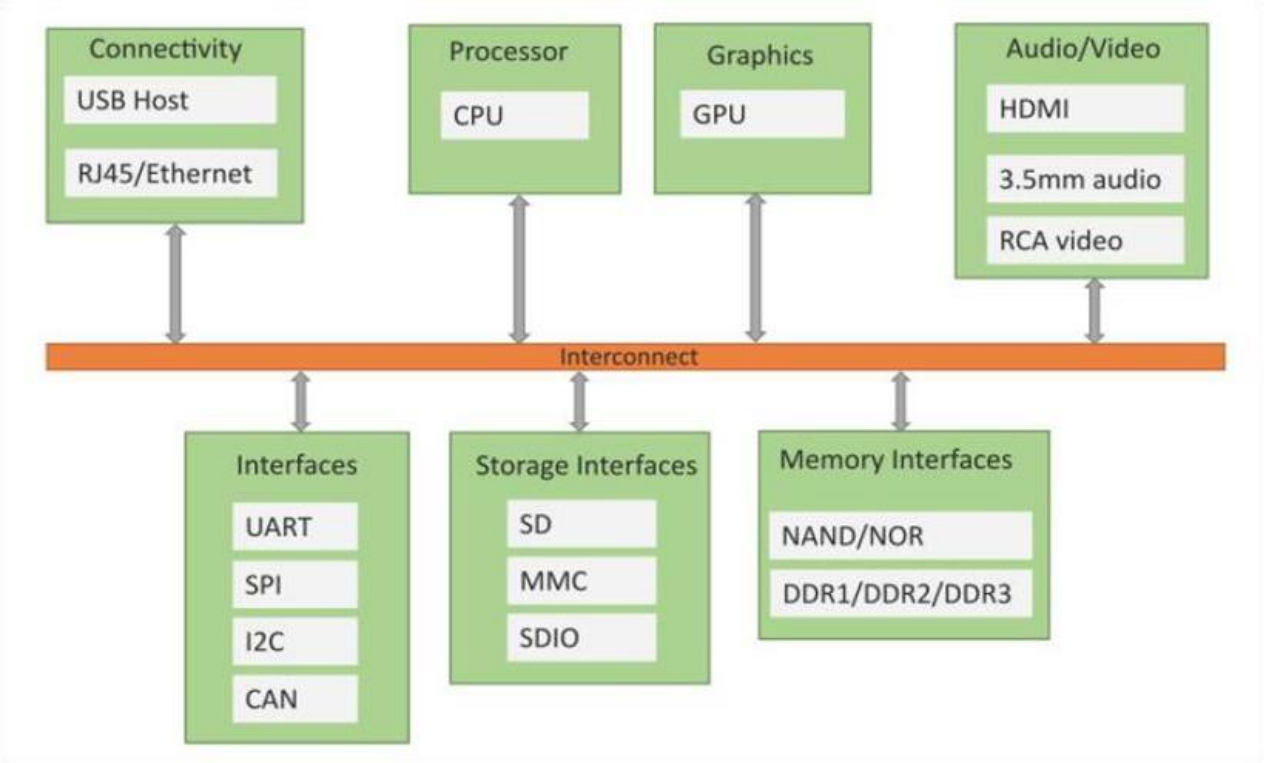
## Physical Design of Internet of Things (IOT)

- The physical design of an IoT system is referred to the Things/Devices and protocols that used to build an IoT system. all these things/Devices are called Node Devices and every device has a unique identity that performs remote sensing, actuating, and monitoring work. and the protocols that used to established communication between the Node devices and server over the internet.


**Things/Devices**

- Things/Devices are used to build a connection, process data, provide interfaces, provide storage, and provide graphics interfaces in an IoT system. all these generate data in a form that can be analyzed by an analytical system and program to perform operations and used to improve the system.

- For example, temperature sensor that is used to analyze the temperature generates the data from a location and then determined by algorithms.




**Connectivity**

- Devices like USB host and ETHERNET are used for connectivity between the devices and server.

**Processor**

- A processor like a CPU and other units are used to process the data. these data are further used to improve the decision quality of an IoT system.

**Audio/Video Interfaces**

- An interface like HDMI and RCA devices is used to record audio and videos in a system.

**Input/Output interface**

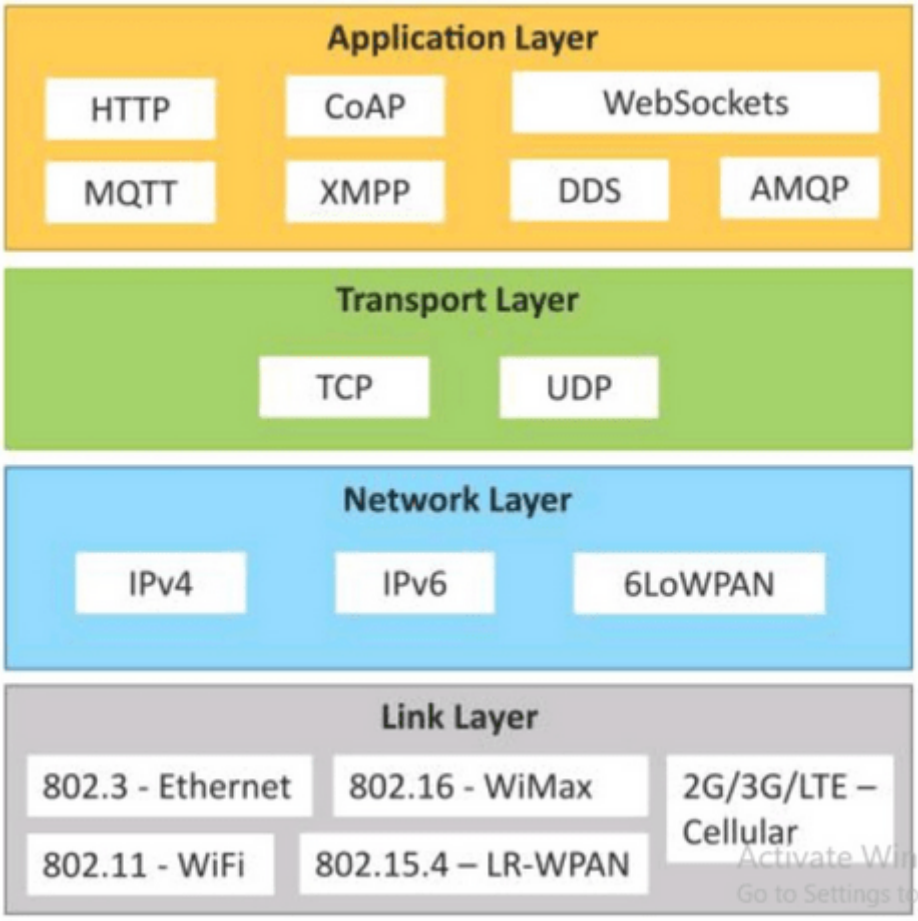
- To giving input and output signals to sensors, and actuators we use things like UART, SPI, CAN, etc.

**Storage Interfaces**

- Things like SD, MMC, SDIO are used to store the data generated from an IoT device.
- Other things like DDR, GPU are used to control the activity of an IoT system.

**IoT Protocols**

These protocols are used to establish communication between a node device and server over the internet. it helps to send commands to an IoT device and receive data from an IoT device over the internet. we use different types of protocols that present on both the server and client-side and these protocols are managed by network layers like application, transport, network, and link layer.




**Application Layer protocol**

In this layer, protocols define how the data can be sent over the network with the lower layer protocols using the application interface. these protocols including HTTP, WebSocket, XMPP, MQTT, DDS, and AMQP protocols.

**HTTP**

Hypertext transfer protocol is a protocol that presents in an application layer for transmitting media documents. it is used to communicate between web browsers and servers. it makes a request to a server and then waits till it receives a response and in between the request server does not keep any data between two requests.

**WebSocket**

This protocol enables two-way communication between a client and a host that can be run on an untrusted code in a controlled environment. this protocol is commonly used by web browsers.

**MQTT**

Message Queue Telemetry Transport (MQTT) was introduced by IBM in 1999 and standardized by OASIS in 2013. It is a machine-to-machine connectivity protocol that was designed as a publish/subscribe messaging transport. and it is used for remote locations where a small code footprint is required.

It is designed to provide embedded connectivity between applications and middleware's on one side and networks and communications on the other side. It follows a publish/subscribe architecture, as shown in figure

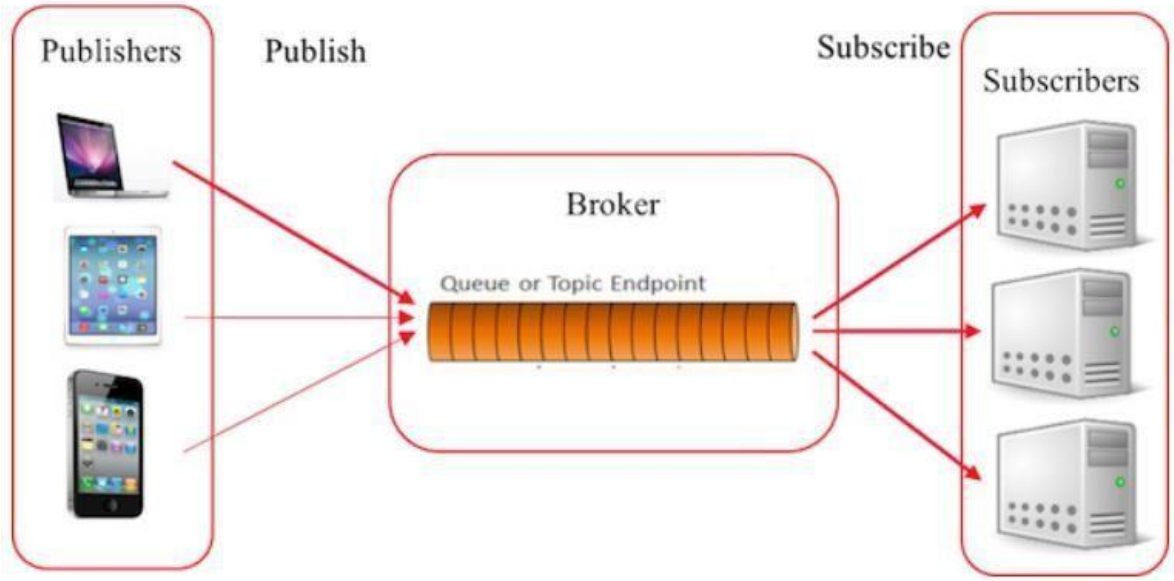



**Figure 5: MQTT Architecture**

The system consists of three main components: publishers, subscribers, and a broker. From IoT point of view, publishers are basically the lightweight sensors that connect to the broker to send their data and go back to sleep whenever possible. Subscribers are applications that are interested in a certain topic, or sensory data, so they connect to brokers to be informed whenever new data are received. The brokers classify sensory data in topics and send them to subscribers interested in the topics.

**CoAP:**

CoAP or Constrained Application Protocol, is an application layer protocol that was introduced by the Internet Engineering Task Force in the year 2014. CoAP is designed for the constrained environment. It is a web-based protocol that resembles HTTP. It is also based on the request-response model. Based on the REST-style architecture, this protocol considers the various objects in the network as resources. These resources are uniquely assigned a URI or Uniform Resource Identifier. The data from one resource to another resource is transferred in the form of CoAP message packets whose format is briefly described later. The Client requests for some

resources and in response to that, the server sends some response over which the client sends an acknowledgement. Although, some types of CoAP do not involve the receiver sending acknowledgments for the information received.

**XMPP:**

It is an XML-driven protocol utilized typically in open standard communication. To say it concisely, it is a chat protocol that permits the seamless sending of essential XML components such as data. Besides making IM and real-time 'talks' possible, XMPP also finds its applications in contact list maintenance and presence details. XMPP is most generally used in forthright message interaction between two points, checking the user status, sharing the status details with the server. It also takes care of server status record keeping, subscription management, contact list updates, and blockading certain users.

**DDS:**

DDS (Data Distribution Service) is a data-centric middleware protocol and API standard published by the OMG organization. It integrates system components to provide low-latency data connectivity, high reliability, and a highly scalable architecture, making it suitable for various commercial-grade IoT applications**.** DDS operates as a publish-subscribe messaging protocol, enabling real-time, reliable, and scalable data distribution.

**AMQP:**

The Advanced Message Queuing Protocol (AMQP) is an open standard application layer protocol for message-oriented middleware. The defining features of AMQP are message orientation, queuing, routing (including point-to-point and publish-and-subscribe), reliability and security. Advanced Message Queuing Protocol. In IoT applications, AMQP is used to collect data from various sensors and devices and transmit it to processing systems or cloud services. Its ability to operate over constrained networks and its secure, reliable messaging make it an excellent choice for IoT

## Transport Layer

This layer is used to control the flow of data segments and handle the error control. also, these layer protocols provide end-to-end message transfer capability independent of the underlying network.

**TCP**

The transmission control protocol is a protocol that defines how to establish and maintain a network that can exchange data in a proper manner using the internet protocol.

**UDP**

a user datagram protocol is a part of internet protocol called the connectionless protocol. this

protocol not required to establish the connection to transfer data.

**Network Layer**

This layer is used to send datagrams from the source network to the destination network. we use IPv4 and IPv6 protocols as a host identification that transfers data in packets.

**IPv4**

This is a protocol address that is a unique and numerical label assigned to each device connected with the network. an IP address performs two main functions host and location addressing. IPv4 is an IP address that is 32 bit long.

**IPv6**

It is a successor of IPv4 that uses 128 bits for an IP address. it is developed by the IETF task force to deal with the long-anticipated problems.

**6LoWPAN:**

6LoWPAN stands for IPv6 over Low-power Wireless Personal Area Networks. It is a standard protocol for realizing IPv6 communication on wireless networks composed of low-power wireless modules. 6LoWPAN specification contains packet compression and other optimization mechanisms to enable the efficient transmission of IPv6 packets on a network with limited power resources and reliability, which makes efficient IPv6 communication over low-power wireless networks possible.

## Link Layer

Link-layer protocols are used to send data over the network's physical layer. it also determines how the packets are coded and signaled by the devices.

**802.3-Ethernet**

It is a set of technologies and protocols that are used primarily in LANs. it defines the physical layer and the medium access control for wired ethernet networks.

**802.11-WiFi**

It is a set of LAN protocols and specifies the set of media access control and physical layer protocols for implementing wireless local area networks.

**802.16-WiMax**

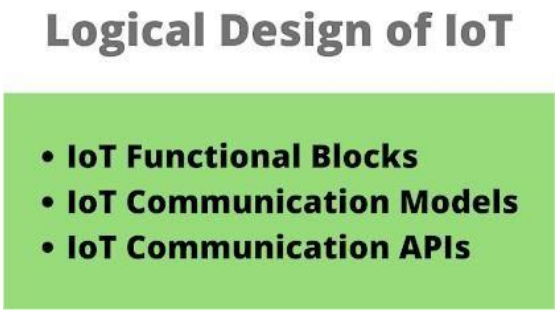
The IEEE 802.16/WiMAX protocol is a wireless broadband standard that provides internet access in remote areas. It's used to connect to infrastructure networks like the internet.

**802.15.4 LR-WPAN**

IEEE 802.15.4 is a technical standard that defines the physical layer and media access control (MAC) for low-rate wireless personal area networks (LR-WPANs). It was defined in 2003 by the IEEE 802.15 working group.

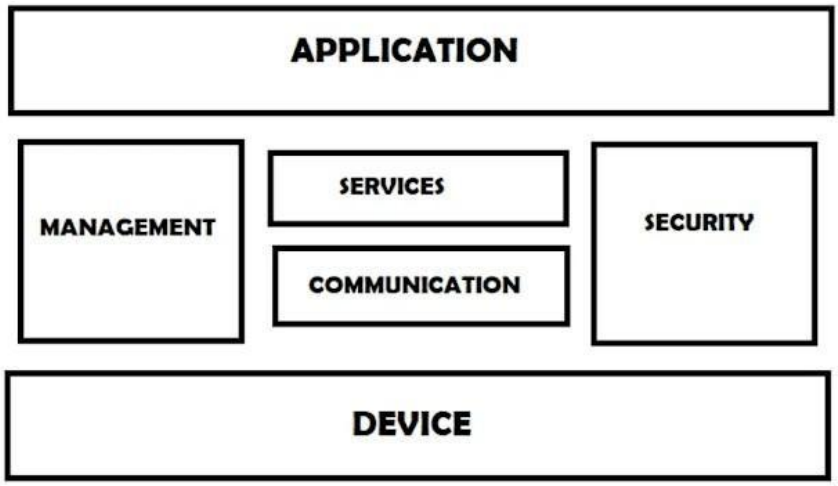
## Logical Design of IoT

The logical design of an IoT system refers to an abstract representation of entities and processes without going into the low-level specifies of implementation. it uses Functional Blocks, Communication Models, and Communication APIs to implement a system.




IoT Functional blocks

An IoT system consist number of functional blocks like Devices, services, communication, security, and application that provides the capability for sensing, actuation, identification, communication, and management.



These functional blocks consist of devices that provide monitoring control functions, handle communication between host and server, manage the transfer of data, secure the system using authentication and other functions, and interface to control and monitor various terms.

**Application**

It is an interface that provides a control system that use by users to view the status and analyze of system.

**Management**

This functional block provides various functions that are used to manage an IoT system.

**Services**

This functional block provides some services like monitoring and controlling a device and publishing and deleting the data and restore the system.

**Communication**

This block handles the communication between the client and cloud-based server and sends/receives the data using protocols.

**Security**

This block is used to secure an IoT system using some functions like authorization, data security, authentication, 2 step verification, etc.

**Device**

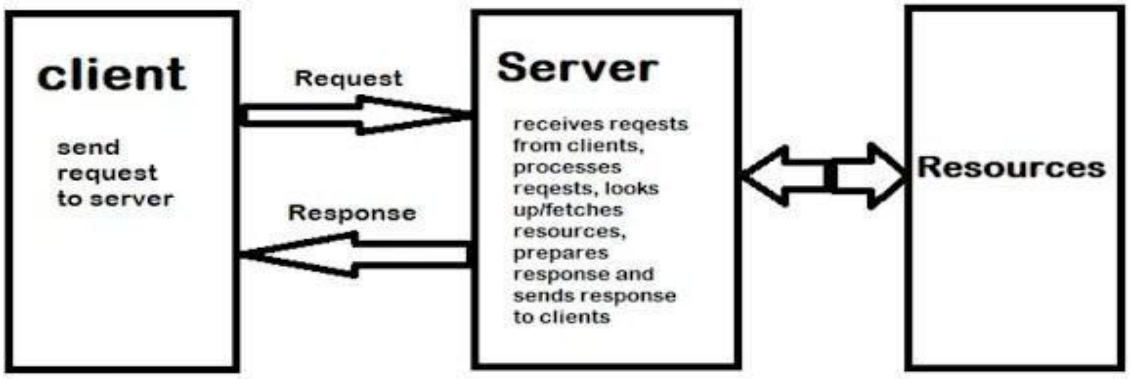
These devices are used to provide sensing and monitoring control functions that collect the data from the outer environment.

**IoT Communication Models**

There are several different types of models available in an IoT system that used to communicate between the system and server like the request-response model, publish-subscribe model, push-pull model, and exclusive pair model, etc.

**Request-Response Communication Model**

This model is a communication model in which a client sends the request for data to the server and the server responds according to the request. when a server receives a request it fetches the data, retrieves the resources and prepares the response, and then sends the data back to the client.

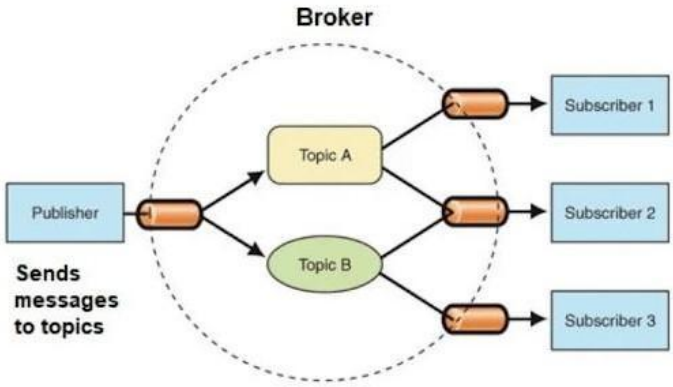



**Request-Response Communication Model**

In simple terms, we can say that in the request-response model server send the response of equivalent on the request of the client. in this model, HTTP works as a request-response protocol between a client and server. Example, When we search a query on a browser then the browser submits an HTTP request to the server and then the server returns a response to the browser(client).

**Publish-Subscribe Communication Model**

In this communication model, we have a broker between publisher and consumer. here publishers are the source of data but they are not aware of consumers. they send the data managed by the brokers and when a consumer subscribes to a topic that managed by the broker and when the broker receives data from the publisher it sends the data to all the subscribed consumers.

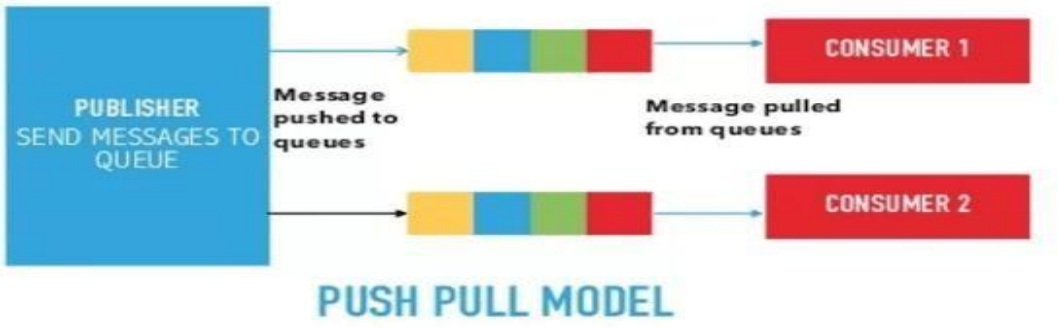



Example, On the website many times we subscribed to their newsletters using our email address. these email addresses managed by some third-party services and when a new article published on the website it directly sends to the broker and then the broker send these new

data or post to all the subscribers.

**Push-Pull Communication Model**

It is a communication model in which the data push by the producers in a queue and the consumers pull the data from the queues. here also producers are not aware of the consumers.




Example

When we visit a website, we saw a number of posts that published in a queue and according to our requirements, we click on a post and start reading it.

**Exclusive Pair Communication Model**

It is a bidirectional fully duplex communication model that uses a persistent connection between the client and server. here first set up a connection between the client and the server and remains open until the client sends a close connection request to the server.


**IoT communication APIs**

These APIs like REST and WebSocket are used to communicate between the server and system in IoT.

**REST-based communication APIs**

Representational state transfer (REST) API uses a set of architectural principles that used to design web services. these APIs focus on the systems' resources that how resource states are transferred using the request-response communication model. this API uses some architectural constraints.

**Client-server**

Here the client is not aware of the storage of data because it is concerned about the server and similarly the server should not be concerned about the user interface because it is a concern of the client. and this separation is needed for independent development and updating of server and client. no matter how the client is using the response of the server and no matter how the server is using the request of the client.

**Stateless**

It means each request from the client to the server must contain all the necessary information to understand by the server. because if the server can't understand the request of the client then it can't fetch the request data in a proper manner.

**Cacheable**

In response, if the cache constraints are given then a client can reuse that response in a later request. it improves the efficiency and scalability of the system without loading the extra data.

A RESTful web APIs is implemented using HTTP and REST principles.

## WebSocket based communication API

This type of API allows bi-directional full-duplex communication between server and client using the exclusive pair communication model. this API uses full-duplex communication so it does not require a new connection setup every time when it requests new data. WebSocket API begins with a connection setup between the server and client and if the WebSocket is supported by the server then it responds back to the client with the successful response and after setup of a connection server and client can send data to each other in full-duplex mode.

This type of API reduces the traffic and latency of data and makes sure that each time when we request new data it cannot terminate the request.

# Unit 2. IoT and M2M

## Machine-to-Machine (M2M)

Machine-to-machine, or M2M, is a broad label that can be used to describe any technology that enables networked devices to exchange information and perform actions without the manual assistance of humans. Artificial intelligence (AI) and machine learning (ML) facilitate the communication between systems, allowing them to make their own autonomous choices.

M2M technology was first adopted in manufacturing and industrial settings, where other technologies, such as SCADA and remote monitoring, helped remotely manage and control data from equipment. M2M has since found applications in other sectors, such as healthcare, business and insurance. M2M is also the foundation for the internet of things (IoT).

**How M2M works**

The main purpose of machine-to-machine technology is to tap into sensor data and transmit it to a network. Unlike SCADA or other remote monitoring tools, M2M systems often use public networks and access methods -- for example, cellular or Ethernet -- to make it more cost-effective.

The main components of an M2M system include sensors, RFID, a Wi-Fi or cellular communications link, and autonomic computing software programmed to help a network device interpret data and make decisions. These M2M applications translate the data, which can trigger preprogrammed, automated actions.

One of the most well-known types of machine-to-machine communication is telemetry, which has been used since the early part of the last century to transmit operational data. Pioneers in telemetric first used telephone lines, and later, radio waves, to transmit performance measurements gathered from monitoring instruments in remote locations.

The Internet and improved standards for wireless technology have expanded the role of telemetry from pure science, engineering and manufacturing to everyday use in products such as heating units, electric meters and internet-connected devices, such as appliances.

Beyond being able to remotely monitor equipment and systems, the top benefits of M2M include:

- reduced costs by minimizing equipment maintenance and downtime;
- boosted revenue by revealing new business opportunities for servicing products in the field; and
- improved customer service by proactively monitoring and servicing equipment before it fails or only when it is needed.

**M2M applications and examples**

Machine-to-machine communication is often used for remote monitoring. In product restocking, for example, a vending machine can message the distributor's network, or machine, when a particular item is running low to send a refill. An enabler of asset tracking and monitoring, M2M is vital in warehouse management systems (WMS) and supply chain management (SCM).

Utilities companies often rely on M2M devices and applications to not only harvest energy, such as oil and gas, but also to bill customers -- through the use of smart meters -- and to detect worksite factors, such as pressure, temperature and equipment status.


In telemedicine, M2M devices can enable the real time monitoring of patients' vital statistics, dispensing medicine when required or tracking healthcare assets.

The combination of the IoT, AI and ML is transforming and improving mobile payment processes and creating new opportunities for different purchasing behaviors. Digital wallets, such as Google Wallet and Apple Pay, will most likely contribute to the widespread

adoption of M2M financial activities.

Smart home systems have also incorporated M2M technology. The use of M2M in this embedded system enables home appliances and other technologies to have real time control of operations as well as the ability to remotely communicate.

M2M is also an important aspect of remote-control software, robotics, traffic control, security, logistics and fleet management and automotive.

**Key features of M2M**

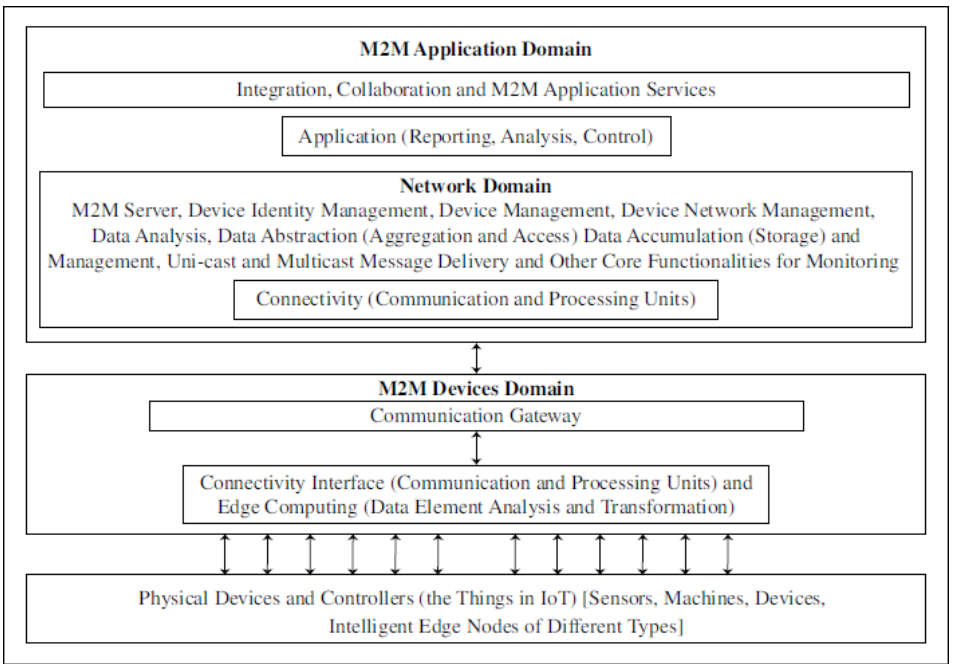
Key features of M2M technology include:

- Low power consumption, in an effort to improve the system's ability to effectively service M2M applications.
- A Network operator that provides packet-switched service
- Monitoring abilities that provide functionality to detect events.
- Time tolerance, meaning data transfers can be delayed.
- Time control, meaning data can only be sent or received at specific predetermined periods.
- Location specific triggers that alert or wake up devices when they enter particular areas.
- The ability to continually send and receive small amounts of data.

## M2M Architecture

M2M architecture consists of three domains:

1. M2M device domain
2. M2M network domain
3. M2M application domain



**M2M device communication domain** consists of three entities: physical devices, communication interface and gateway. Communication interface is a port or a subsystem, which receives the input from one end and sends the data received to another.

**M2M network domain** consists of M2M server, device identity management, data analytics and data and device management similar to IoT architecture (connect + collect + assemble + analyse) level.

**M2M application domain** consists of application for services, monitoring, analysis and controlling of devices networks.

**M2M requirements**

According to the European Telecommunications Standards Institute (ETSI), requirements of an M2M system include:

- **Scalability** - The M2M system should be able to continue to function efficiently as more connected objects are added.
- **Anonymity** - The M2M system must be able to hide the identity of an M2M device when requested, subject to regulatory requirements.
- **Logging** - M2M systems must support the recording of important events, such as failed installation attempts, service not operating or the occurrence of faulty information. The logs should be available by request.
- **M2M application communication principles** - M2M systems should enable communication between M2M applications in the network and the M2M device or gateway using communication techniques, such as short message service (SMS) and IP Connected devices should also be able to communicate with each other in a peer-to-peer (P2P) manner.
- **Delivery methods** - The M2M system should support Unicast, anycast, multicast and broadcast communication modes, with broadcast being replaced by multicast or anycast whenever possible to minimize the load on the communication network.
- **Message transmission scheduling** - M2M systems must be able to control network access and messaging schedules and should be conscious of M2M applications' scheduling delay tolerance.
- **Message communication path selection** - Optimization of the message communication paths within an M2M system must be possible and based on policies like transmission failures, delays when other paths exist and network costs.

**M2M vs. IoT**

While many use the terms interchangeably, M2M and IoT are not the same. IoT needs M2M,

but M2M does not need IoT.

Both terms relate to the communication of connected devices, but M2M systems are often isolated, stand-alone networked equipment. IoT systems take M2M to the next level, bringing together disparate systems into one large, connected ecosystem.

M2M systems use point-to-point communications between machines, sensors and hardware over cellular or wired networks, while IoT systems rely on IP-based networks to send data collected from IoT-connected devices to gateways, the cloud or middleware platforms.

Data collected from M2M devices is used by service management applications, whereas IoT data is often integrated with enterprise systems to improve business performance across multiple groups. Another way to look at it is that M2M affects how businesses operate, while IoT does this and affects end users.

For example, in the product restocking example above, M2M involves the vending machine communicating to the distributor's machines that a refill is needed. Incorporate IoT and an additional layer of analytics is performed; the vending machine can predict when particular products will need refilling based on purchase behaviors, offering users a more personalized experience.

# M2M vs. IoT: What's the difference?

| M2M | IoT |
|---|---|
| Machines | Sensors |
| Hardware-based | Software-based |
| Vertical applications | Horizontal applications |
| Deployed in a closed system | Connects to a larger network |
| Machines communicating with machines | Machines communicating with machines, humans with machines, machines with humans |
| Uses non-IP protocol | Uses IP protocols |
| Can use the cloud, but not required to | Uses the cloud |
| Machines use point-to-point communication, usually embedded in hardware | Devices use IP networks to communicate |
| Often one-way communication | Back and forth communication |
| Main purpose is to monitor and control | Multiple applications; multilevel communications |
| Operates via triggered responses based on an action | Can, but does not have to, operate on triggered responses |
| Limited integration options, devices must have complementary communication standards | Unlimited integration options, but requires software that manages communications/protocols |
| Structured data | Structured and unstructured data |

**M2M security**

Machine-to-machine systems face a number of security issues, from unauthorized access to wireless intrusion to device hacking. Physical security, privacy, fraud and the exposure of mission-critical applications must also be considered.

Typical M2M security measures include making devices and machines tamper-resistant, embedding security into the machines, ensuring communication security through encryption and securing back-end servers, among others. Segmenting M2M devices onto their own network and managing device identity, data confidentiality and device availability can also help combat M2M security risks.

### Introduction to Sensors Technology

Sensing technology and its various applications are constantly evolving in line with advancements in technology and business needs. Sensors are available to detect a wide variety of real-world properties– from distance to heat to pressure. Sensors have the capacity to be extremely accurate, consume less power, and are inexpensive to install and maintain. Sensors are proving to be vital components in creating new value for their process and respective businesses.

Sensors have a very wide range, and there are many types, but fundamentally, sensors are devices that detect the feature quantity of a measurement object and convert this quantity into a readable signal, which is displayed on an instrument. And sensing technology, simply put, is a technology that uses sensors to acquire information by detecting the physical, chemical, or biological property quantities and convert them into readable signal.

### Criteria to Choose a Sensor

The following are certain features that are considered when choosing a sensor.

1. **Type of Sensing**: The parameter that is being sensed like temperature or pressure.
2. **Operating Principle:** The principle of operation of the sensor.
3. **Power Consumption:** The power consumed by the sensor will play an important role in defining the total power of the system.
4. **Accuracy:** The accuracy of the sensor is a key factor in selecting a sensor.
5. **Environmental Conditions:** The conditions in which the sensor is being used will be a factor in choosing the quality of a sensor.
6. **Cost:** Depending on the cost of application, a low-cost sensor or high-cost sensor can be used.
7. **Resolution and Range:** The smallest value that can be sensed and the limit of measurement are important.
8. **Calibration and Repeatability:** Change of values with time and ability to repeat measurements under similar conditions.

## Basic Requirements of a Sensor or Transducer

The basic requirements of a sensor are:

1. **Range**: It indicates the limits of the input in which it can vary. In case of temperature measurement, a thermocouple can have a range of 25 – 250 0C.

2. **Accuracy**: It is the degree of exactness between actual measurement and true value. Accuracy is expressed as percentage of full range output.

3. **Sensitivity**: Sensitivity is a relationship between input physical signal and output electrical signal. It is the ratio of change in output of the sensor to unit change in input value that causes change in output.

4. **Stability**: It is the ability of the sensor to produce the same output for constant input over a period of time.

5. **Repeatability**: It is the ability of the sensor to produce same output for different applications with same input value.

6. **Response Time:** It is the speed of change in output on a stepwise change in input.

7. **Linearity**: It is specified in terms of percentage of nonlinearity. Nonlinearity is an indication of deviation of curve of actual measurement from the curve of ideal measurement.

8. **Ruggedness**: It is a measure of the durability when the sensor is used under extreme operating conditions.

9. **Hysteresis**: The hysteresis is defined as the maximum difference in output at any measurable value within the sensor's specified range when approaching the point first with increasing and then with decreasing the input parameter. Hysteresis is a characteristic that a transducer has in being unable to repeat its functionality faithfully when used in the opposite direction of operation.

## Classification of Sensors

The scheme of classifying sensors can range from very simple to very complex. The stimulus that is being sensed is an important factor in this classification.

1. **Acoustic**: Wave, spectrum and wave velocity.

2. **Electric**: Current, charge, potential, electric field, permittivity and conductivity.

3. **Magnetic**: Magnetic field, magnetic flux and permeability.

4. **Thermal**: Temperature, specific heat and thermal conductivity.

5. **Mechanical**: Position, acceleration, force, pressure, stress, strain, mass, density, momentum, torque, shape, orientation, roughness, stiffness, compliance, crystallinity and structural.
6. **Optical**: Wave, wave velocity, refractive index, reflectivity, absorption and emissivity.

All the sensors can be classified into two types based on the power or signal requirement.

- Active sensors:

  Active sensors, require power signal from an external source. This signal is called an excitation signal, and based on this excitation signal the sensor produces output. Strain gauge is an example of active sensor. It is a pressure sensitive resistive bridge network and doesn't produce the output electrical signal on its own. The amount of force applied can be measured by relating it to the resistance of the network. The resistance can be measured by passing current through it. Current acts as the excitation signal.

- Passive sensors.

  Passive sensors directly produce the output electrical signal in response to the input stimulus. All the power required by a passive sensor is obtained from the measurand. A thermocouple is a passive sensor.

**Commonly used Sensors and Transducers**

Some of the most commonly used sensors and transducers for different stimuli (the quantity to be measured) are

1. For sensing light, the input devices or sensors are photo diode, photo transistor, light dependent resistor and solar cells. The output devices or actuators are LEDs, displays, lamps and fiber optics.

2. For sensing temperature, the sensors are thermistor, thermocouple, resistance temperature detectors and thermostat. The actuators are heaters.

3. For sensing position, the input devices are potentiometer, proximity sensor, and differential transformer. The output devices are motor and panel meter.

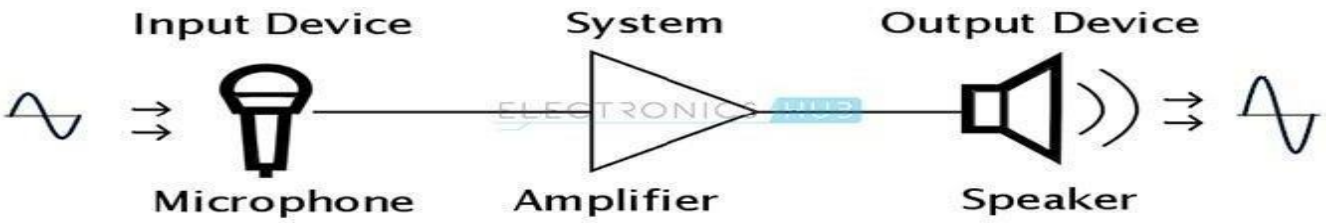
4. For sensing pressure, the sensors are strain gauge and load cell. The actuators are lifts and jacks and electromagnetic vibrations.

5. For sensing sound, the input devices are microphones and output devices are loudspeakers and buzzers.

6. For sensing speed, the sensors used are tachogenerator and Doppler Effect sensors. The actuators are motors and brakes.

**A Simple System using Transducers**

A public addressing system is an example of a system using sensors and actuators.



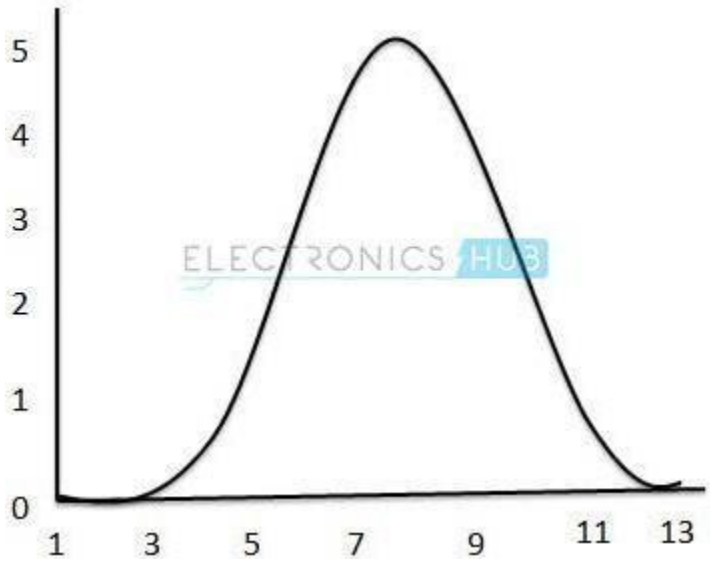

It consists of a microphone, an amplifier and a loudspeaker. The sensor or the device with input function is a microphone. It senses the sound signals and transforms them into electrical signals. The amplifier receives these electrical signals and amplifies their strength.

The actuator or the device with output function is loudspeaker. It receives the amplified electrical signals from the amplifier and converts them back into sound signals but with more reach.

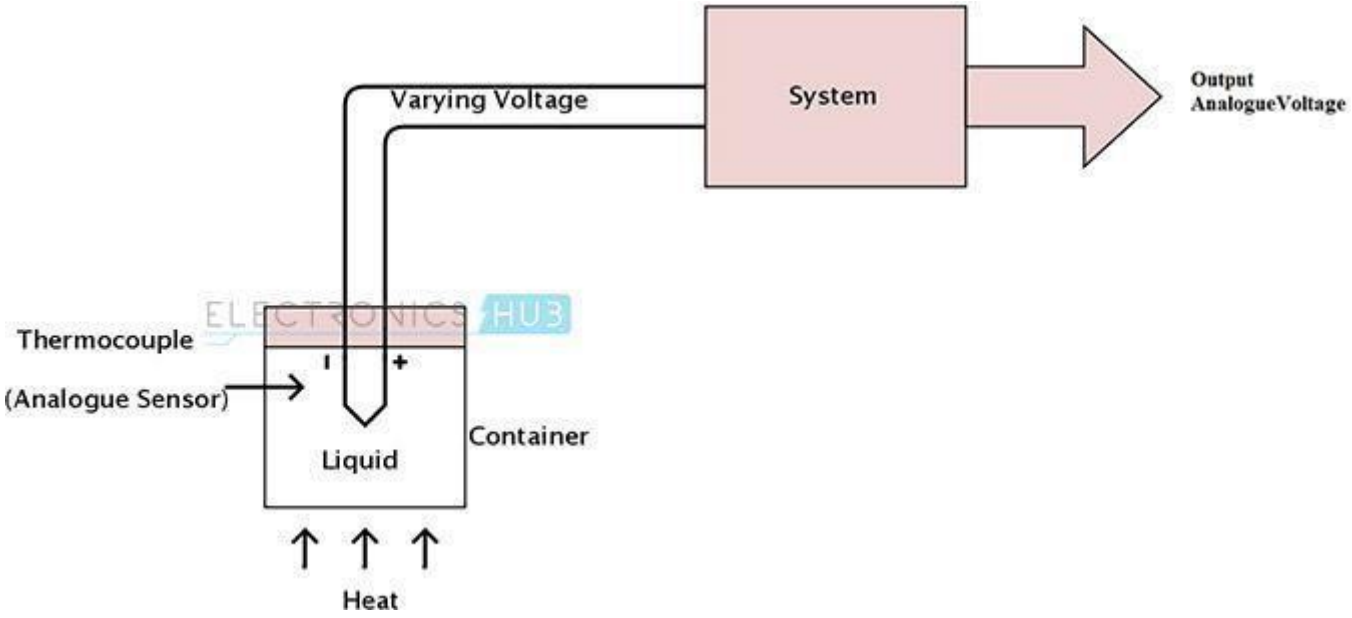
**Analog Sensors**

An analogue sensor produces continuously varying output signals over a range of values. Usually, the output signal is voltage and this output signal is proportional to the measurand. The quantity that is being measured like speed, temperature, pressure, strain, etc. are all continuous in nature and hence they are analogue quantities.

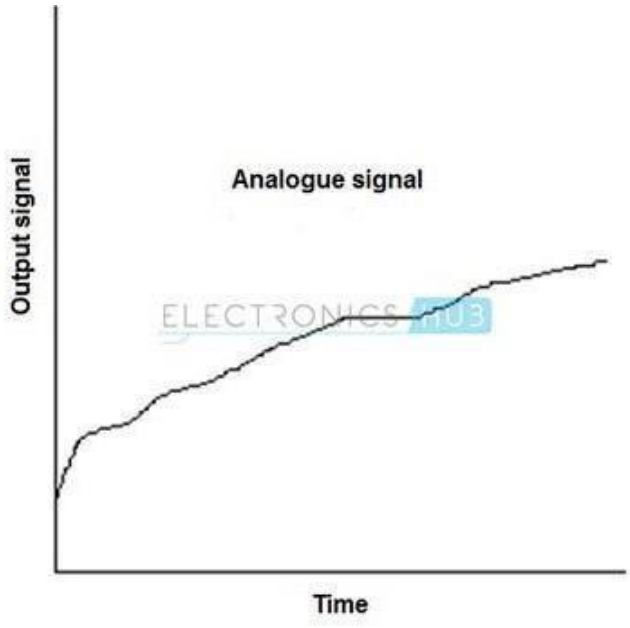
A Cadmium Sulfide Cell (CdS Cell) which is used to measure the intensity of light is an analogue sensor. The resistance of a CdS cell varies according to the intensity of the light incident on it. When connected to a voltage divider network, the change in resistance can be observed through varying output voltage. In this circuit, the output can vary from anywhere between 0 V to 5 V.



A thermocouple or a thermometer is an analog sensor. The following setup is used to measure




The output signal of the above setup can be depicted as follows:



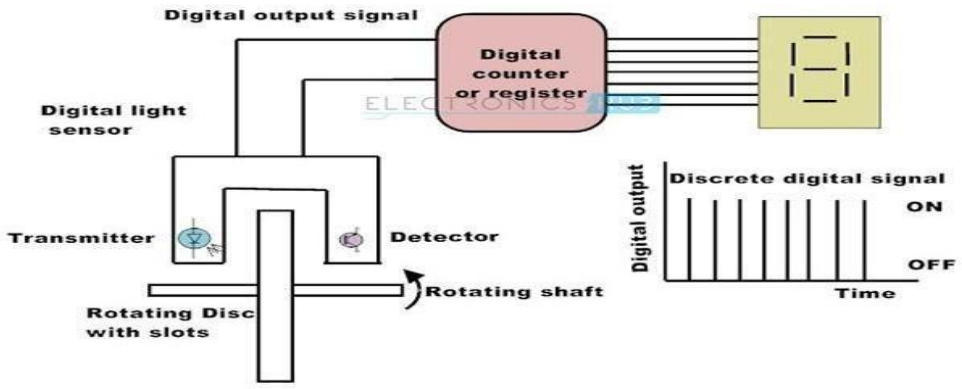

The output of an analogue sensor tends to change smoothly and continuously over time. Hence the response time and accuracy of circuits employing analogue sensors is slow and less. In order to use these signals in a micro-controller-based system, Analog to Digital converters can be used.

Analogue sensors generally require an external power supply and amplification of some form to produce appropriate output signals. Op Amps are very useful in providing amplification and filtering.

**Digital Sensors**

A digital sensor produces discrete digital signals. The output of a digital sensor has only two states, namely 'ON' and 'OFF'. ON is logic 1 and OFF is logic 0. A push button switch is the best example of a digital sensor. In this case, the switch has only two possible states: either it is ON when pushed or it is OFF when released or not pushed.

The following setup uses a light sensor to measure the speed and produces a digital signal.




In the above setup, the rotating disc is connected to the shaft of a motor and has number of transparent slots. The light sensor captures the presence or absence of the light and sends logic 1 or logic 0 signal accordingly to the counter. The counter displays the speed of the disc. The accuracy can be increased by increasing the transparent slots on the disc as it allows more counts over the same amount of time.

In general, the accuracy of a digital sensor is high when compared to an analogue sensor. The accuracy depends on the number of bits that are used to represent the measurand. Higher the number of bits, the greater is the accuracy.

**IoT security (internet of things security)**

Security in IoT is the act of securing Internet devices and the networks they're connected to from threats and breaches by protecting, identifying, and monitoring risks all while helping fix vulnerabilities from a range of devices that can pose security risks to your business.

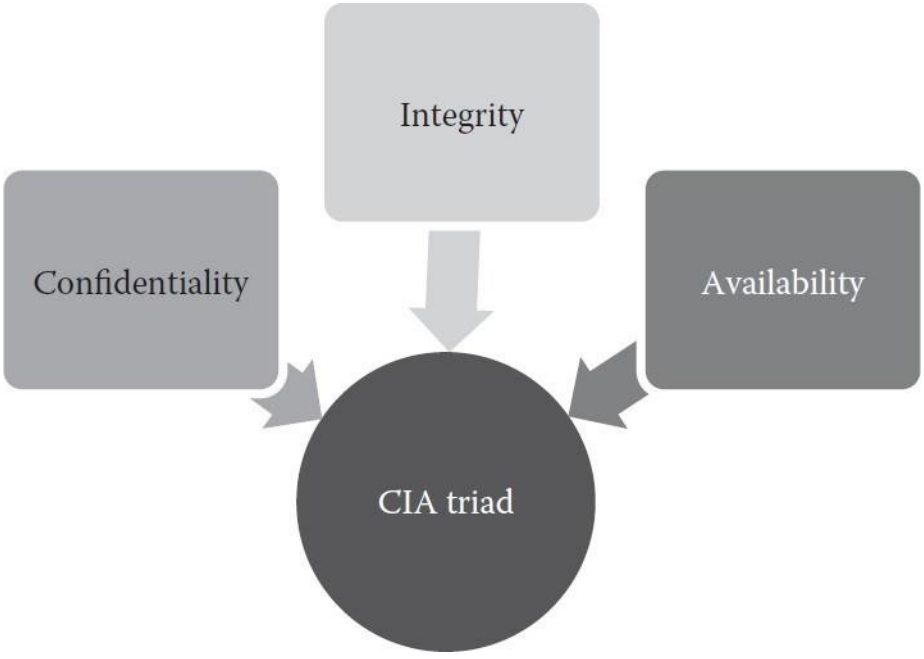
Internet of Things (IoT) security is the practice of protecting devices and networks that are connected to the internet. The goal of IoT security is to keep user data safe and prevent cyber attacks.

Types of IoT security

- Network security: Protects devices from unauthorized access and exploitation
- Embedded security: Uses nano agents to provide on-device security
- Firmware assessment: Assesses the security of a device's firmware

Improving IoT security

- Update devices: Keep devices up to date with the latest patches and operating system updates
- Use strong passwords: Use unique, strong passwords for all devices
- Enable multi-factor authentication: Enable multi-factor authentication whenever possible
- Limit access: Limit the number of devices that can access your network
- Secure your network: Use technologies, policies, and procedures to protect your network from cyberattacks
- Disable unused devices: Regularly take inventory of your connected devices and disable any that you don't use often.



**Confidentiality, Integrity and Availability Triad(CIA)**

The CIA triad is a model for information security that stands for confidentiality, integrity, and availability. It's a widely accepted model that helps security teams analyze risks and protect data.

Confidentiality

- Protects information from unauthorized access
- Preserves restrictions on who can access and disclose information
- Protects personal privacy and proprietary

information Integrity

- Ensures data is trustworthy and free from tampering
- Protects against improper information modification or destruction
- Ensures information is authentic and non-

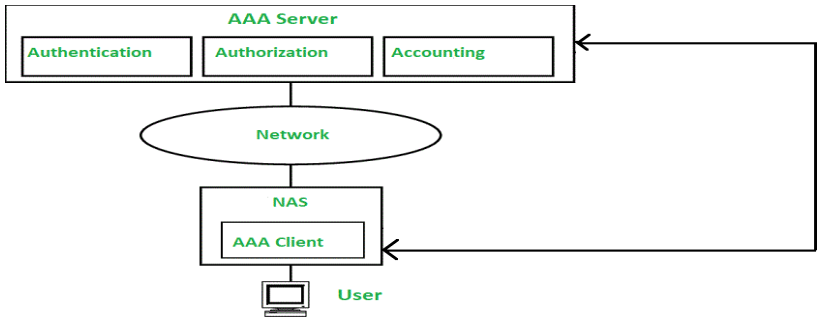
repudiable Availability

- Ensures timely and reliable access to information
- Ensures systems, networks, and applications are functioning properly
- Ensures users can access information when needed, even under duress

The CIA triad is a guiding model for security measures, controls, and overall strategy. It helps organizations build resilience and confidence in the face of cybersecurity challenges.

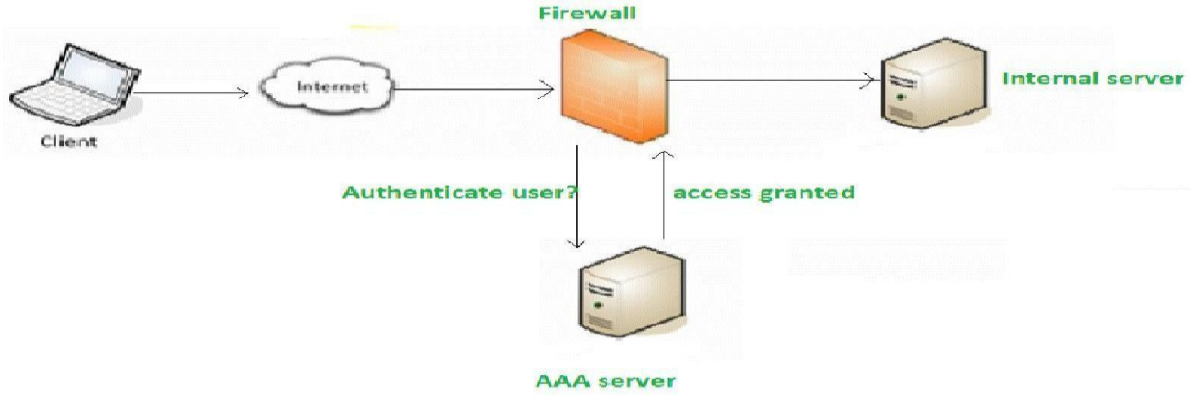
**Authentication, Authorization and Audit Trial (AAA) Framework**

Authentication, Authorization, and Accounting (AAA) is an architectural framework to gain access to computer resources, enforcing policies, auditing usage, to provide essential information required for billing of services and other processes essential for network management and security. This process is mainly used so that network and software application resources are accessible to some specific and legitimate users. The AAA concept is widely used in reference to the network protocol RADIUS.



The first step: Authentication

Authentication is the method of identifying the user. With the help of the user's authentication credentials, it checks if the user is legitimate or not or if the user has access to the network, by checking if the user's credentials match with credentials stored in the network database. After the authentication is approved the user gains access to the internal resources of the network.




Authorization

For the user to perform certain tasks or to issue commands to the network, he must gain authorization. It determines the extent of access to the network and what type of services and resources are accessible by the authenticated user. Authorization is the method of enforcing policies**.**

Audit trail

Audit trail is a detailed record that tracks all changes and activities within a system, helping ensure transparency and accountability. It logs who did what and when, making it easier to detect and resolve issues. This is crucial for security, compliance, and troubleshooting. In this article, we'll explore the importance and basics of audit trails in a straightforward way.

## Internet of Things (IoT) Enabling Technologies

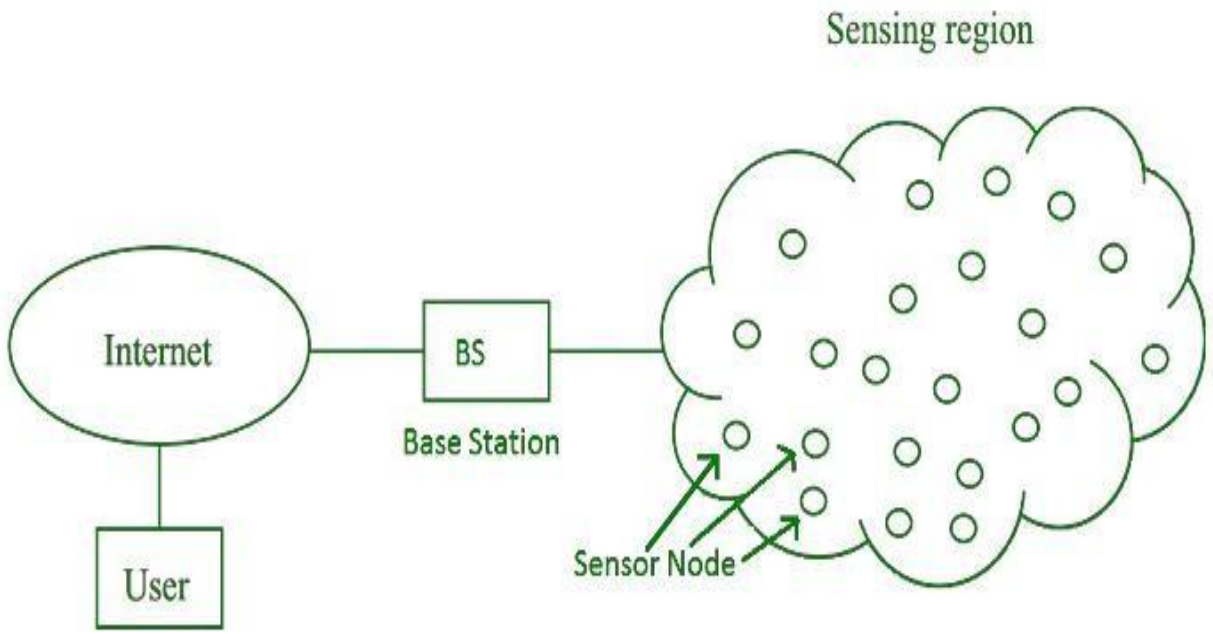
IoT(internet of things) enabling technologies are

1. Wireless Sensor Network
2. Cloud Computing
3. Big Data Analytics
4. Communications Protocols
5. Embedded System

### 1. Wireless Sensor Network (WSN) :

A WSN comprises distributed devices with sensors which are used to monitor the environmental and physical conditions. A wireless sensor network consists of end nodes, routers and coordinators. End nodes have several sensors attached to them where the data is passed to a coordinator with the help of routers. The coordinator also acts as the gateway that connects WSN to the internet. Example:

- Weather monitoring system
- Indoor air quality monitoring system
- Soil moisture monitoring system
- Surveillance system
- Health monitoring system




- Sensor nodes are used in WSN with the on-board processor that manages and monitors the environment at particular area.
- They are connected to the base station which acts as a processing unit in WSN system.
- Base stations are connected through the internet to share data used for processing, analysis, storage and mining of the data.
- Base station or sink acts like an interface between the users and the network.
- Sensor nodes can communicate amongst themselves using radio signals.
- A WSN contains hundreds of thousands of sensor nodes.

- A wireless sensor node is equipped with sensing and computing devices, radio transceivers and power components.

**Applications of WSN:**

- Military applications
- Communications, computing, intelligence, battlefield surveillance, reconnaissance and targeting system.
- Surveillance and monitoring security, threat detection
- Environmental temperature, humidity
- Pollution monitoring, forest fire detection, greenhouse monitoring, etc.
- Transportation
- Real-time traffic information to alert drivers of traffic problems.
- Medical applications
- Diagnosis, drug administration in hospitals, tracking and monitoring doctors and patients in the hospital.
- Agriculture
- Automated irrigation, cultivation according to the weather.
- Industrial monitoring
- Significant cost savings and enabling new functionalities.
- Infrastructural monitoring
- Monitoring the movement within infrastructure such as bridges, flyover, embankments, tunnels, etc. enabling engineering practices to monitor assets remotely without the need for costly site visit.

**Challenges of WSN:**

- Quality of service
- Security issue
- Energy efficiency
- Network throughput
- Performance
- Ability to cope with network failure
- Scalability to large scale of deployment
- Cross layer optimization

## Components of WSN:

- **Sensors:** They capture the environmental variables and use it for data acquisition. Sensor signals are converted in to electrical signals.
- **Radio Nodes:** It receives the data produced by the sensors and sends it to access points.
- It consists of microcontroller, transceiver, external memory and power source.
- **WLAN Access Points**: It receives the data which is sent by the radio nodes wirelessly, generally through internet

- **Evaluation Software**: The data received by the access point is processed by this software for presenting it to the users for further processing of the data which is used for processing, analysis, storage and mining of the data.

## 2. Cloud Computing :

It provides us the means by which we can access applications as utilities over the internet. Cloud means something which is present in remote locations. With Cloud computing, users can access any resources from anywhere like databases, webservers, storage, any device, and any software over the internet. Characteristics –

1. Broad network access
2. On demand self-services
3. Rapid scalability
4. Measured service
5. Pay-per-use

**Provides different services, such as –**

- **IaaS (**Infrastructure as a service**)**

Infrastructure as a service provides online services such as physical machines, virtual machines, servers, networking, storage and data center space on a pay per use basis. Major IaaS providers are Google Compute Engine, Amazon Web Services and Microsoft Azure etc. Ex : Web Hosting, Virtual Machine etc.

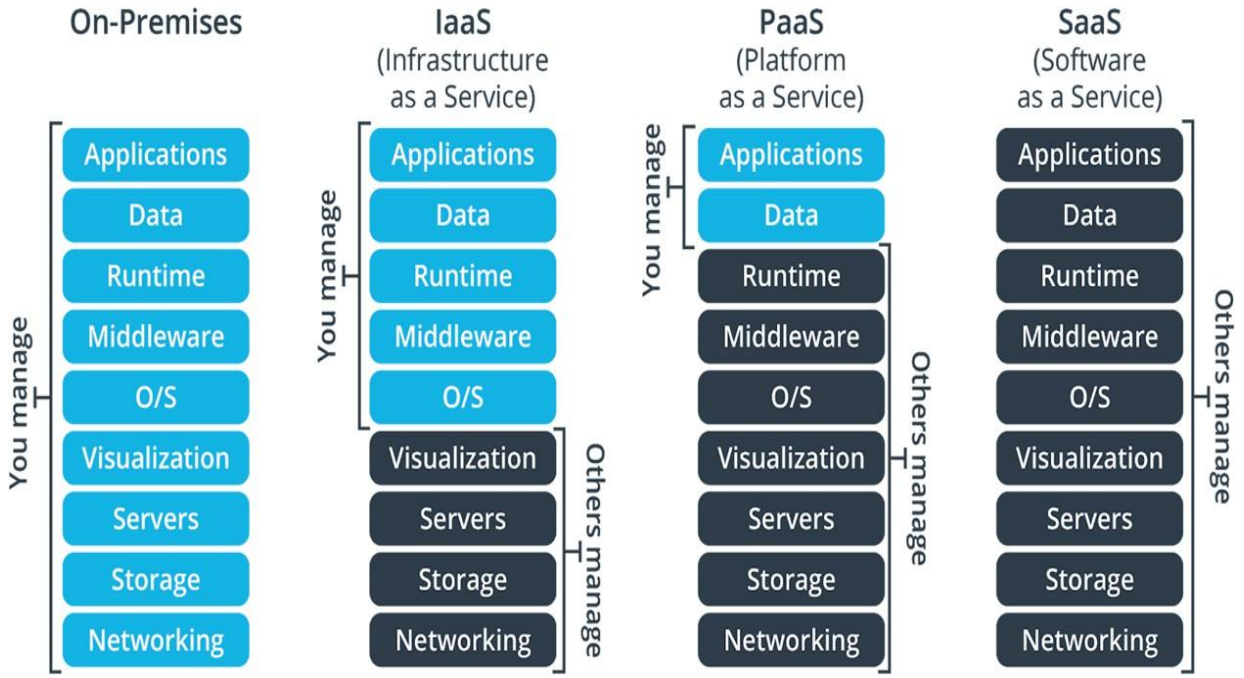
- **PaaS (**Platform as a service**)**

Provides a cloud-based environment with a very thing required to support the complete life cycle of building and delivering West web based (cloud) applications – without the cost and complexity of buying and managing underlying hardware, software provisioning and hosting. Computing platforms such as hardware, operating systems and libraries etc. Basically, it provides a platform to develop applications. Example: App Cloud, Google app engine

- **SaaS (**Software as a service**)**

It is a way of delivering applications over the internet as a service. Instead of installing and maintaining software, you simply access it via the internet, freeing yourself from complex software and hardware management. SaaS Applications are sometimes called web-based software on demand software or hosted software. SaaS applications run on a SaaS provider's service and they manage security availability and performance. Example: Google Docs, Gmail,

office etc.




**3. Big Data Analytics :**

It refers to the method of studying massive volumes of data or big data. Collection of data whose volume, velocity or variety is simply too massive and tough to store, control, process and examine the data using traditional databases. Big data is gathered from a variety of sources including social network videos, digital images, sensors and sales transaction records.

It is a process used to extract meaningful insights, such as hidden patterns, unknown correlation, market trends and customer preferences.

Ex: Spotify, youtube, etc. In such platforms, they have millions of users that generate a tremendous amount of data every day. Through this information, the cloud-based platform automatically generates the suggested songs through a smart recommendation engine based on likes, shares, search history, etc

Several steps involved in analyzing big data –

1. Data cleaning
2. Munging
3. Processing
4. Visualization

**Life cycle or Phases of Big Data Analytics:**

o Business Case Evaluation – Define the reason and goal behind the analysis.

o Identification of data – Broad variety of data sources are identified.

o Data filtering – Variety of data sources are filtered here to remove corrupt data.

o Data Extraction – Incompatible data are transformed into the compatible data.

o Data aggregation – Data with the same fields across different datasets are integrated.

o Data analysis – Data is evaluated using analytical and statistical tools to discover useful information.

o Visualization of data – Graphic visualization (i.e. generating graphs, charts, pictures, etc.) of the analysis is prepared using the tools like Tableau, QlikView, etc.

o Final analysis result – The final results of the analytics is made available to the stakeholders who will take the action.

**Types of Big Data Analytics:**

**(a) Descriptive Analytics:**

- This summarizes past data into a form that people can easily read.
- This helps in creating company's reports. Like a company's revenue, sales, profit/loss, etc.
- Scenario: A company uses data to analyze facility utilization across its office and lab space. Using the analytics, company can identify the underutilized space and thus can use it effectively saving a lot of money and space.

**(b) Diagnostic Analytics:**

- This helps to understand what caused a problem in the first place.
- It provides an in-depth insight into a particular problem.
- Ex: data mining, data recovery, etc.
- Scenario: sales of an e-commerce company going down. Many customers are adding their products to the cart but not buying them due to many different reasons like delivery charges, taxes, delivery duration and location, less payment options available, etc. The company diagnosis the reason behind the issue.

**(c) Predictive Analytics:**

- This analytics looks into past and present data to make predictions of the future.
- It uses artificial intelligence, data mining and machine learning to analyze current data and make predictions about the future.
- Ex: studying customer trends, market trends, etc.
- Scenario: precautions required to protect clients of PayPal. The company uses predictive analytics to study the historical payment data of a user and determine the user's behavior to generate an algorithm that predicts the future fraudulent activities.

**(d) Prescriptive Analytics:**

- It prescribes the solution to a particular problem and works with both descriptive and predictive analytics.
- It mostly relies on artificial intelligence and machine learning.
- Scenario: to maximize airlines' profit. Using prescriptive analytics the airlines company

will generate an algorithm which will automatically adjust the flight fares based on various factors like customer demand, weather conditions, destination, holiday season and oil prices.

**Big Data Analytics Tools:**

- o Hadoop – It helps in storing and analyzing data
- o MongoDB – Used on datasets that change frequently
- o Talend – Used for data integration and management.
- o Cassandra – A distributed database used to handle chunks of data.
- o Spark – Used for real-time processing and analyzing large amount of data
- o Storm – An open-source real-time computational system
- o Kafka – A distributed streaming platform that is used for fault- tolerant storage**.**

Examples –

- Bank transactions
- Data generated by IoT systems for location and tracking of vehicles
- E-commerce and in Big-Basket
- Health and fitness data generated by IoT system such as a fitness band

**4. Communications Protocols:**

They are the backbone of IoT systems and enable network connectivity and linking to applications. Communication protocols allow devices to exchange data over the network. Multiple protocols often describe different aspects of a single communication. A group of protocols designed to work together is known as a protocol suite; when implemented in software they are a protocol stack. They are used in

- Data encoding
- Addressing schemes

**5. Embedded Systems:**

It is a combination of hardware and software used to perform special tasks. It includes microcontroller and microprocessor memory, networking units (Ethernet Wi-Fi adapters), input output units (display keyword etc.) and storage devices (flash memory). It collects the data and sends it to the internet.

**Characteristics of Embedded Systems:**

o **Single functioned:**

- Embedded systems usually perform a specialized operation and odes the same task repeatedly as they are designed for a specific task only.

o **Tightly constrained:**

- Design metrics is a measure of implementation's features such as cost, size, power and performance.
- It must be of a size to fit on a single chip, and must perform fast enough to process the data in real-time and consume minimum power to extend the battery life.

o **Reactive and Real-time:**

- Many embedded systems (like fire-alarm system, radar- system in army, etc.) must continuously react to the changes in the system's environment and generate some results based on real-time without any delays.

o **Microprocessor based:**

- It must be microprocessor or microcontroller based.

o **Memory:**

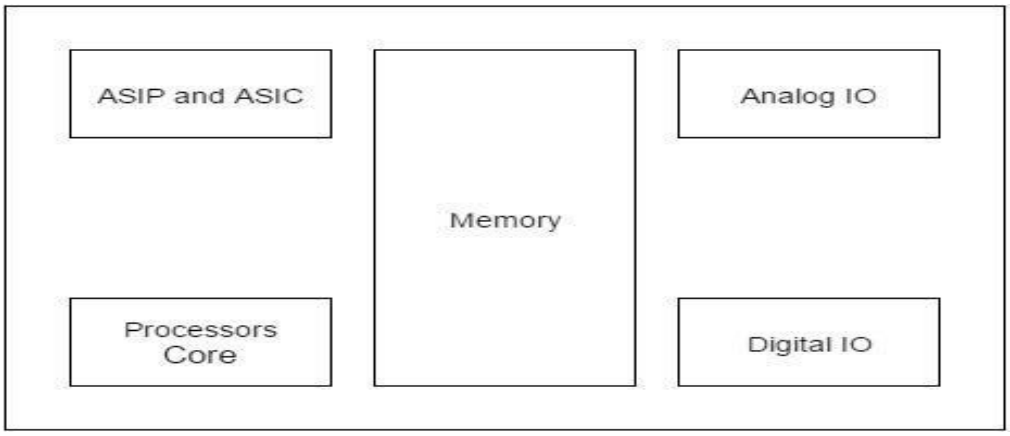
- It must have a memory, as its software is usually embedded into ROM.
- It does not need any secondary memory.

o **Connected:**

- It must have connected peripheral for the input and output devices.

o **HW-SW systems:**

- Software is used for more features and flexibility.
- Hardware is used for performance and security.




**Advantages of Embedded Systems:**

- o Easily customizable
- o Low cost

o Lower power consumption

o Enhanced performance

**Disadvantages of Embedded Systems:**

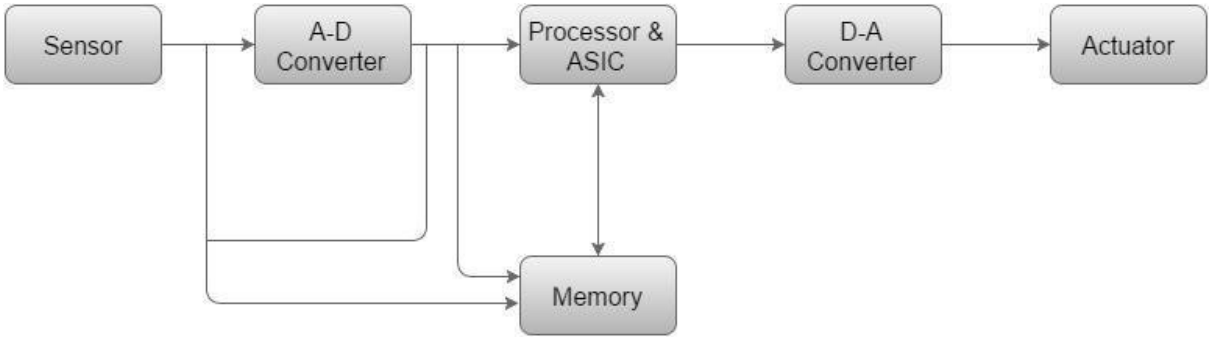
- o High development efforts o Larger time to market

Embedded systems used in Examples –

- Digital camera
- DVD player, music player
- Industrial robots
- Wireless Routers etc.

## Basic Structure of an Embedded System

The following illustration shows the basic structure of an embedded system –

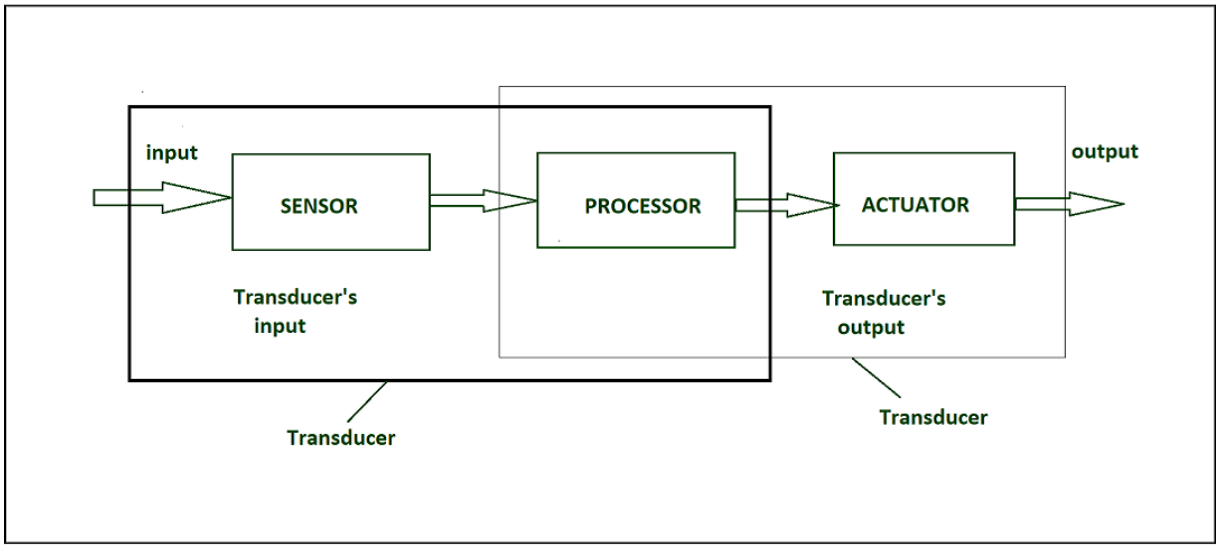



- **Sensor** − It measures the physical quantity and converts it to an electrical signal which can be read by an observer or by any electronic instrument like an A2D converter. A sensor stores the measured quantity to the memory.
- **A-D Converter** − An analog-to-digital converter converts the analog signal sent 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 fed by the processor to analog data
- **Actuator** − An actuator compares the output given by the D-A Converter to the actual (expected) output stored in it and stores the approved output.

# UNIT 3 Sensors and Actuators in IoT

## Sensors in Internet of Things(IoT)

Generally, sensors are used in the architecture of IOT devices. Sensors are used for sensing things and devices etc. A device that provides a usable output in response to a specified measurement. The sensor attains a physical parameter and converts it into a signal suitable for processing (e.g. electrical, mechanical, optical) the characteristics of any device or material to detect the presence of a particular physical quantity. The output of the sensor is a signal which is converted to a human-readable form like changes in characteristics, changes in resistance, capacitance, impedance etc.




IOT HARDWARE

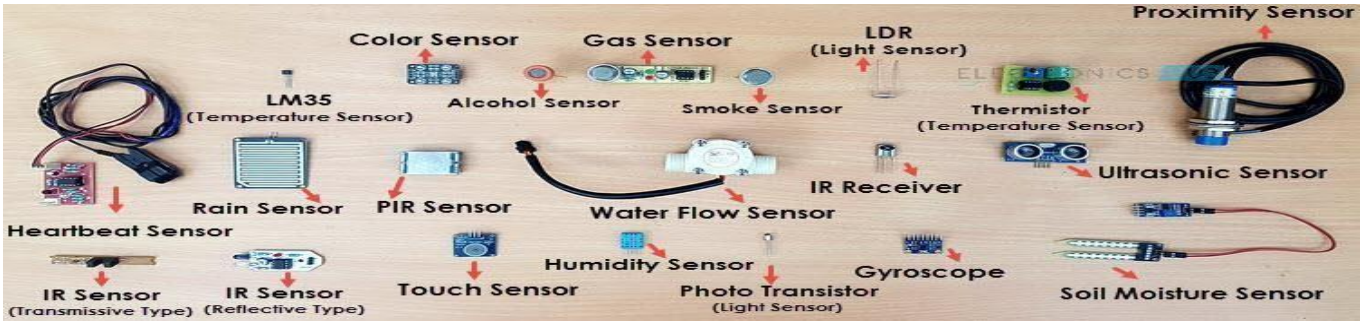
**What is a Sensor?**

There are numerous definitions as to what a sensor is but I would like to define a Sensor as an input device which provides an output (signal) with respect to a specific physical quantity (input).

The term “input device” in the definition of a Sensor means that it is part of a bigger system which provides input to a main control system (like a Processor or a Microcontroller).

Another unique definition of a Sensor is as follows: It is a device that converts signals from one energy domain to electrical domain. The definition of the Sensor can be better understood if

we take an example in to consideration.




The simplest example of a sensor is an LDR or a Light Dependent Resistor. It is a device, whose resistance varies according to intensity of light it is subjected to. When the light falling on an LDR is more, its resistance becomes very less and when the light is less, well, the resistance of the LDR becomes very high.

We can connect this LDR in a voltage divider (along with other resistor) and check the voltage drop across the LDR. This voltage can be calibrated to the amount of light falling on the LDR. Hence, a Light Sensor.

Now that we have seen what a sensor is, we will proceed further with the classification of Sensors.

**Classification of Sensors**

There are several classifications of sensors made by different authors and experts. Some are very simple and some are very complex. The following classification of sensors may already be used by an expert in the subject but this is a very simple classification of sensors.

In the first classification of the sensors, they are divided in to Active and Passive. Active Sensors are those which require an external excitation signal or a power signal.

Passive Sensors, on the other hand, do not require any external power signal and directly generates output response.

The other type of classification is based on the means of detection used in the sensor. Some of the means of detection are Electric, Biological, Chemical, Radioactive etc.

The next classification is based on conversion phenomenon i.e., the input and the output. Some of the common conversion phenomena are Photoelectric, Thermoelectric, Electrochemical, Electromagnetic, Thermo-optic, etc.

The final classification of the sensors is Analog and Digital Sensors. Analog Sensors produce an analog output i.e., a continuous output signal (usually voltage but sometimes other quantities like Resistance etc.) with respect to the quantity being measured.

Digital Sensors, in contrast to Analog Sensors, work with discrete or digital data. The data in digital sensors, which is used for conversion and transmission, is digital in nature.

## Different Types of Sensors

The following is a list of different types of sensors that are commonly used in various applications. All these sensors are used for measuring one of the physical properties like Temperature, Resistance, Capacitance, Conduction, Heat Transfer etc.

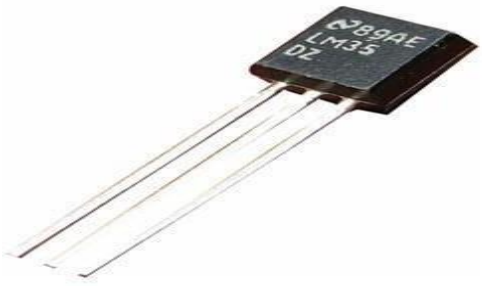
1. Temperature Sensor
2. Proximity Sensor
3. Accelerometer
4. IR Sensor (Infrared Sensor)
5. Pressure Sensor
6. Light Sensor
7. Ultrasonic Sensor
8. Smoke, Gas and Alcohol Sensor
9. Touch Sensor
10. Color Sensor
11. Humidity Sensor
12. Position Sensor
13. Magnetic Sensor (Hall Effect Sensor)
14. Microphone (Sound Sensor)
15. Tilt Sensor
16. Flow and Level Sensor
17. PIR Sensor
18. Touch Sensor
19. Strain and Weight Sensor

More information about the sensors will be added subsequently. A list of projects using the

above sensors is given at the end of the page.

**Temperature Sensor**

One of the most common and most popular sensors is the Temperature Sensor. A Temperature Sensor, as the name suggests, senses the temperature i.e., it measures the changes in the temperature.




LM35 - Temperature Sensor IC

10KΩ NTC Thermistor

There are different types of Temperature Sensors like Temperature Sensor ICs (like LM35, DS18B20), Thermistors, Thermocouples, RTD (Resistive Temperature Devices), etc.

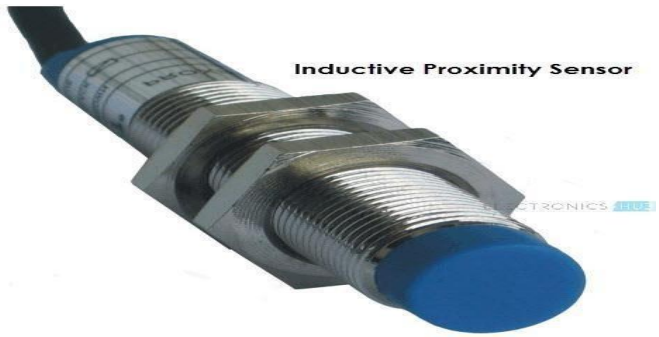
Temperature Sensors can be analog or digital. In an Analog Temperature Sensor, the changes in the Temperature correspond to change in its physical property like resistance or voltage. LM35 is a classic Analog Temperature Sensor.

Coming to the Digital Temperature Sensor, the output is a discrete digital value (usually, some numerical data after converting analog value to digital value). DS18B20 is a simple Digital Temperature Sensor.

Temperature Sensors are used everywhere like computers, mobile phones, automobiles, air conditioning systems, industries etc.

**Proximity Sensors**

A Proximity Sensor is a non-contact type sensor that detects the presence of an object. Proximity Sensors can be implemented using different techniques like Optical (like Infrared or Laser), Sound (Ultrasonic), Magnetic (Hall Effect), Capacitive, etc.

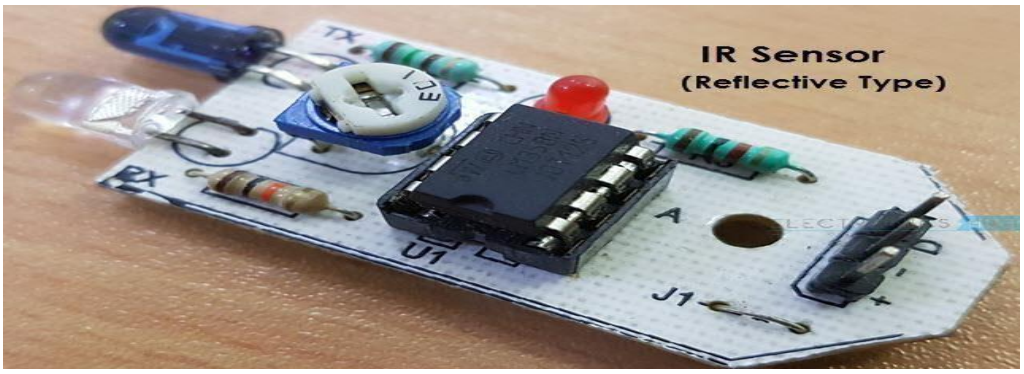


Inductive Proximity Sensor

Some of the applications of Proximity Sensors are Mobile Phones, Cars (Parking Sensors), industries (object alignment), Ground Proximity in Aircrafts, etc.

**Infrared Sensor (IR Sensor)**

IR Sensors or Infrared Sensor are light based sensor that are used in various applications like Proximity and Object Detection. IR Sensors are used as proximity sensors in almost all mobile phones.




There are two types of Infrared or IR Sensors: Transmissive Type and Reflective Type. In Transmissive Type IR Sensor, the IR Transmitter (usually an IR LED) and the IR Detector (usually a Photo Diode) are positioned facing each other so that when an object passes between them, the sensor detects the object.

The other type of IR Sensor is a Reflective Type IR Sensor. In this, the transmitter and the detector are positioned adjacent to each other facing the object. When an object comes in front of the sensor, the infrared light from the IR Transmitter is reflected from the object and is detected by the IR Receiver and thus the sensor detects the object.

Different applications where IR Sensor is implemented are Mobile Phones, Robots, Industrial assembly, automobiles etc.

**Ultrasonic Sensor**

An Ultrasonic Sensor is a non-contact type device that can be used to measure distance as well as velocity of an object. An Ultrasonic Sensor works based on the properties of the sound waves with frequency greater than that of the human audible range.

Using the time of flight of the sound wave, an Ultrasonic Sensor can measure the distance of the object (similar to SONAR). The Doppler Shift property of the sound wave is used to measure the velocity of an object.

**Light Sensor**

Sometimes also known as Photo Sensors, Light Sensors are one of the important sensors. A simple Light Sensor available today is the Light Dependent Resistor or LDR. The property of LDR is that its resistance is inversely proportional to the intensity of the ambient light i.e., when the intensity of light increases, its resistance decreases and vise-versa.

By using LDR is a circuit, we can calibrate the changes in its resistance to measure the intensity of Light. There are two other Light Sensors (or Photo Sensors) which are often used in complex electronic system design. They are Photo Diode and Photo Transistor. All these are Analog Sensors.

There are also Digital Light Sensors like BH1750, TSL2561, etc., which can calculate intensity of light and provide a digital equivalent value.

**Smoke and Gas Sensors**

One of the very useful sensors in safety related applications are Smoke and Gas Sensors. Almost all offices and industries are equipped with several smoke detectors, which detect any smoke (due to fire) and sound an alarm.

Gas Sensors are more common in laboratories, large scale kitchens and industries. They can detect different gases like LPG, Propane, Butane, Methane (CH4), etc.

Now-a-days, smoke sensors (which often can detect smoke as well gas) are also installed in most homes as a safety measure.

The “MQ” series of sensors are a bunch of cheap sensors for detecting CO, CO2, CH4, Alcohol, Propane, Butane, LPG etc. You can use these sensors to build your own Smoke Sensor Application.

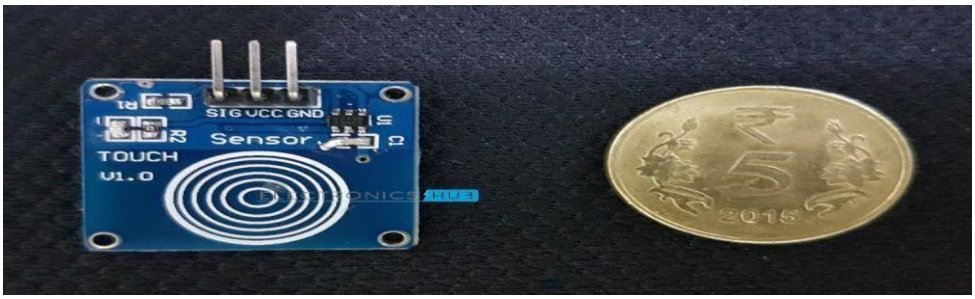
**Alcohol Sensor**

As the name suggests, an Alcohol Sensor detects alcohol. Usually, alcohol sensors are used in breathalyzer devices, which determine whether a person is drunk or not. Law enforcement personnel uses breathalyzers to catch drunk-and-drive culprits.


**Touch Sensor**

We do not give much importance to touch sensors but they became an integral part of our life. Whether you know or not, all touch screen devices (Mobile Phones, Tablets, Laptops, etc.) have touch sensors in them. Another common application of touch sensor is trackpads in our laptops.

Touch Sensors, as the name suggests, detect touch of a finger or a stylus. Often touch sensors are classified into Resistive and Capacitive type. Almost all modern touch sensors are of Capacitive Types as they are more accurate and have better signal to noise ratio.




**Color Sensor**

A Color Sensor is an useful device in building color sensing applications in the field of image processing, color identification, industrial object tracking etc. The TCS3200 is a simple Color Sensor, which can detect any color and output a square wave proportional to the wavelength of the detected color.


**Humidity Sensor**

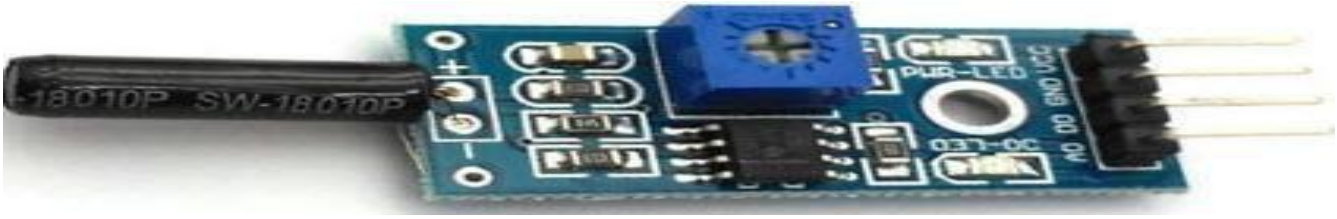
If you see Weather Monitoring Systems, they often provide temperature as well as humidity data. So, measuring humidity is an important task in many applications and Humidity Sensors help us in achieving this.

Often all humidity sensors measure relative humidity (a ratio of water content in air to maximum potential of air to hold water). Since relative humidity is dependent on temperature of air, almost all Humidity Sensors can also measure Temperature.


Humidity Sensors are classified into Capacitive Type, Resistive Type and Thermal Conductive Type. DHT11 and DHT22 are two of the frequently used Humidity Sensors in DIY Community (the former is a resistive type while the latter is capacitive type).

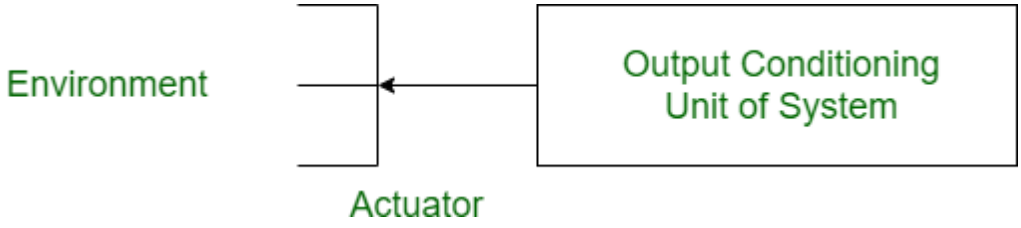
**Tilt Sensor**

Often used to detect inclination or orientation, Tilt Sensors are one of the simplest and inexpensive sensors out there. Previously, tilt sensors are made up of Mercury (and hence they are sometimes called as Mercury Switches) but most modern tilt sensors contain a roller ball.

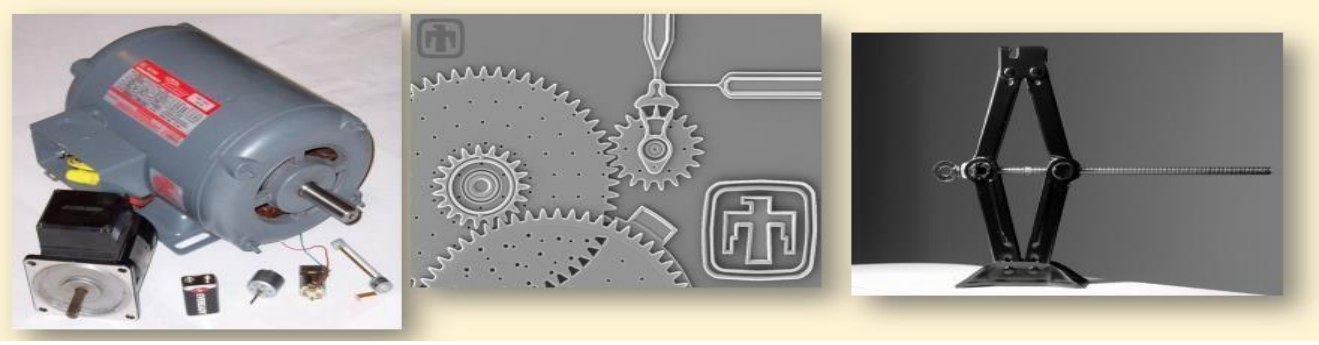


## What are Actuators?

Actuator is a device that converts the electrical signals into the physical events or characteristics. It takes the input from the system and gives output to the environment. For example, motors and heaters are some of the commonly used actuators.




An actuator is a motor that transfers energy from whatever is powering it into motion. This explanation may sound simplistic, but at heart, that's what it does. Within that simplicity lies a great deal of utility; the majority of modern devices that involve motion of any kind could not exist without these components.

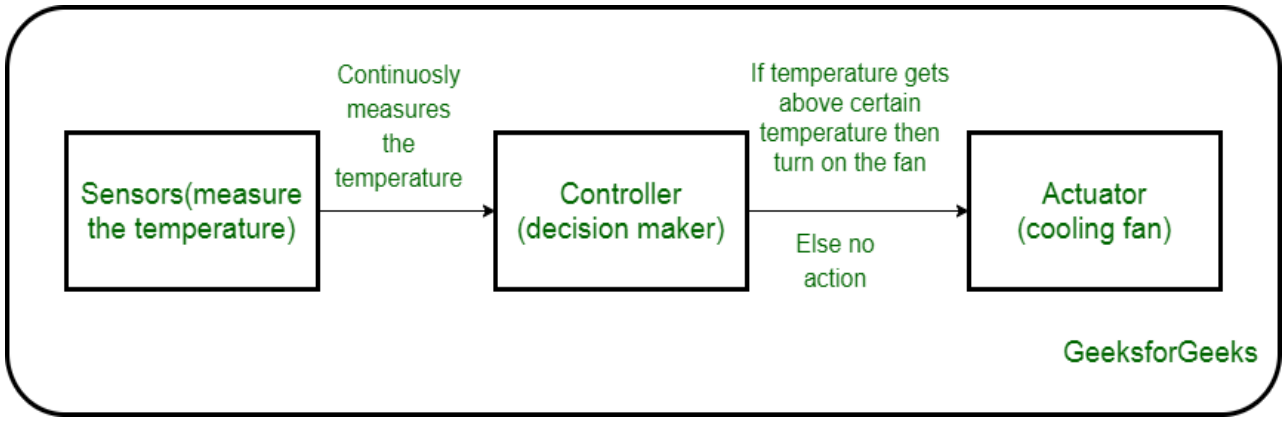


## Actuators in IoT

An IoT device is made up of a Physical object ("thing") + Controller ("brain") + Sensors + Actuators + Networks (Internet). An actuator is a machine component or system that moves or controls the mechanism or the system. Sensors in the device sense the environment, then control signals are generated for the actuators according to the actions needed to perform.

A servo motor is an example of an actuator. They are linear or rotatory actuators, can move to a given specified angular or linear position. We can use servo motors for IoT applications and make the motor rotate to 90 degrees, 180 degrees, etc., as per our need.

The following diagram shows what actuators do, the controller directs the actuator based on the sensor data to do the work.




**Working of IoT devices and use of Actuators**

The control system acts upon an environment through the actuator. It requires a source of energy and a control signal. When it receives a control signal, it converts the source of energy to a mechanical operation. On this basis, on which form of energy it uses, it has different types given below.

## Types of Actuators:

**1. Hydraulic Actuators –**

A hydraulic actuator uses hydraulic power to perform a mechanical operation. They are actuated by a cylinder or fluid motor. The mechanical motion is converted to rotary, linear, or oscillatory motion, according to the need of the IoT device. Ex- construction equipment uses hydraulic actuators because hydraulic actuators can generate a large amount of force.

**Advantages**:

- Hydraulic actuators can produce a large magnitude of force and high speed.
- Used in welding, clamping, etc.
- Used for lowering or raising the vehicles in car transport carriers.

**Disadvantages**:

- Hydraulic fluid leaks can cause efficiency loss and issues of cleaning.
- It is expensive.
- It requires noise reduction equipment, heat exchangers, and high maintenance systems.

**2. Pneumatic Actuators –**

A pneumatic actuator uses energy formed by vacuum or compressed air at high pressure to convert into either linear or rotary motion. Example- Used in robotics, use sensors that work like human fingers by using compressed air.

**Advantages** :

- They are a low-cost option and are used at extreme temperatures where using air is a safer option than chemicals.
- They need low maintenance, are durable, and have a long operational life.
- It is very quick in starting and stopping the motion.

**Disadvantages** :

- Loss of pressure can make it less efficient.
- The air compressor should be running continuously.
- Air can be polluted, and it needs maintenance.

**3. Electrical Actuators –**

An electric actuator uses electrical energy, is usually actuated by a motor that converts electrical energy into mechanical torque. An example of an electric actuator is a solenoid based electric bell.

**Advantages** :

- It has many applications in various industries as it can automate industrial valves.
- It produces less noise and is safe to use since there are no fluid leakages.
- It can be re-programmed and it provides the highest control precision positioning.

**Disadvantages :**

- It is expensive.
- It depends a lot on environmental conditions.

**4. Thermal/Magnetic Actuators –**

These are actuated by thermal or mechanical energy. Shape Memory Alloys (SMAs) or Magnetic Shape-Memory Alloys (MSMAs) are used by these actuators. An example of a thermal/magnetic actuator can be a piezo motor using SMA.

**5. Mechanical Actuators –**

A mechanical actuator executes movement by converting rotary motion into linear motion. It involves pulleys, chains, gears, rails, and other devices to operate. Example – A crankshaft.

- Soft Actuators
- Shape Memory Polymers
- Light Activated Polymers
- With the expanding world of IoT, sensors and actuators will find more usage in commercial and domestic applications along with the pre-existing use in industry.

**Difference between Sensor and Actuator:**

| SENSOR | ACTUATOR |
|---|---|
| It converts physical characteristics into electrical signals. | It converts electrical signals into physical characteristics. |
| It takes input from environment. | It takes input from output conditioning unit of system. |
| It gives output to input conditioning unit of system. | It gives output to environment. |
| Sensor generated electrical signals. | Actuator generates heat or motion. |
| It is placed at input port of the system. | It is placed at output port of the system. |
| It is used to measure the physical quantity. | It is used to measure the continuous and discrete process parameters. |
| It gives information to the system about environment. | It accepts command to perform a function. |
| Example: Photo-voltaic cell which converts light energy into electrical energy. | Example: Stepper motor where electrical energy drives the motor. |

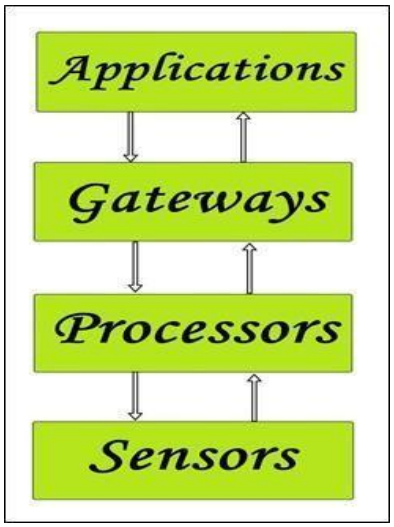
# UNIT4 Introduction to Raspberry pi and Arduino

## Introduction to IoT Devices.

IoT devices are pieces of hardware, such as sensors, actuators, gadgets, appliances, or machines, that are programmed for certain applications and can transmit data over the internet or other networks. They can be embedded into other mobile devices, industrial equipment, environmental sensors, medical devices, and more.

**BUILDING BLOCKS of IoT**

Four things form basic building blocks of the IoT system –sensors, processors, gateways, applications. Each of these nodes has to have its own characteristics in order to form an useful IoT system.




Simplified block diagram of the basic building blocks of the IoT

**Sensors**:

- These form the front end of the IoT devices. These are the so-called "Things" of the system. Their main purpose is to collect data from its surroundings (sensors) or give out data to its surrounding (actuators).
- These have to be uniquely identifiable devices with a unique IP address so that they can be easily identifiable over a large network.
- These have to be active in nature which means that they should be able to collect real-time data. These can either work on their own (autonomous in nature) or can be made to work by the user depending on their needs (user-controlled).

- Examples of sensors are gas sensor, water quality sensor, moisture sensor, etc.

**Processors**:

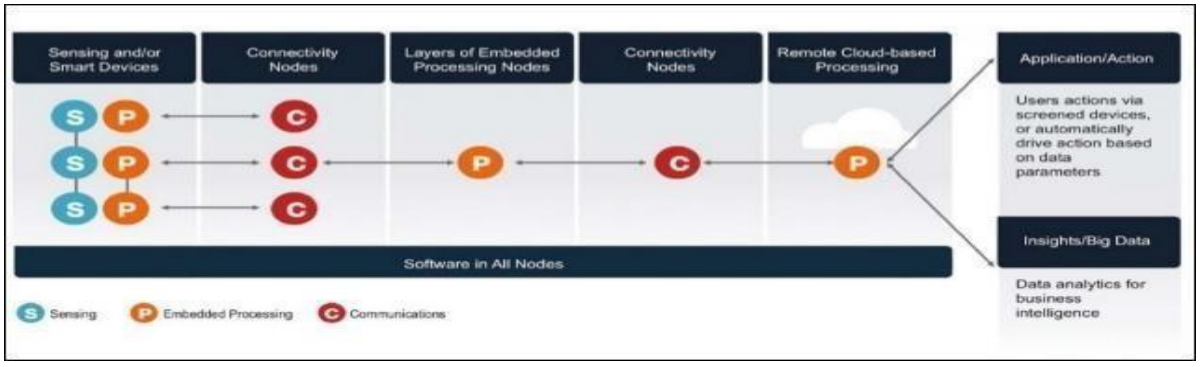
- Processors are the brain of the IoT system. Their main function is to process the data captured by the sensors and process them so as to extract the valuable data from the enormous amount of raw data collected. In a word, we can say that it gives intelligence to the data.
- Processors mostly work on real-time basis and can be easily controlled by applications. These are also responsible for securing the data – that is performing encryption and decryption of data.
- Embedded hardware devices, microcontroller, etc are the ones that process the data because they have processors attached to it.

**Gateways**:

- Gateways are responsible for routing the processed data and send it to proper locations for its (data) proper utilization.
- In other words, we can say that gateway helps in to and fro communication of the data. It provides network connectivity to the data. Network connectivity is essential for any IoT system to communicate.
- LAN, WAN, PAN, etc are examples of network gateways.

**Applications**:

- Applications form another end of an IoT system. Applications are essential for proper utilization of all the data collected.
- These cloud-based applications which are responsible for rendering the effective meaning to the data collected. Applications are controlled by users and are a delivery point of particular services.
- Examples of applications are home automation apps, security systems, industrial control hub, etc.



Basic building blocks of IoT

## Introduction to Raspberry Pi

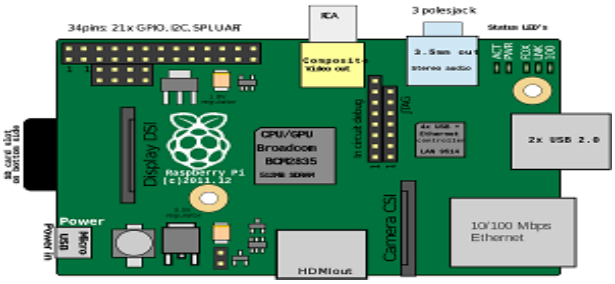
The Raspberry pi is a single computer board with credit card size, that can be used for many tasks that your computer does, like games, word processing, spreadsheets and also to play HD video. It was established by the Raspberry pi foundation from the UK.

It has been ready for public consumption since 2012 with the idea of making a low-cost educational microcomputer for students and children.

The main purpose of designing the raspberry pi board is, to encourage learning, experimentation and innovation for school level students.

The raspberry pi board is a portable and low cost. Maximum of the raspberry pi computers is used in mobile phones.

## Raspberry Pi Working




The Raspberry Pi Foundation looks to bridge the gap by providing an economically friendly way to learn beginning programming languages.

In 2012, the Foundation launched Raspberry Pi, a single-board computer designed to teach programming skills, build hardware projects, do home automation, and explore industrial applications of computer technology.

An economical computer that runs on Linux (a computer operating system) and provides general-purpose input and output (GPIO) pins that allow the user to control electronic components for physical computing and exploring the Internet of Things (IoT).

The Raspberry Pi Foundation promotes the study of basic computer science in schools with a single-board computer called Raspberry Pi. Raspberry Pi is available throughout the world through the Foundation's website and in select countries through approved Pi Resellers.

The operating system for all Raspberry Pi products is Linux. Linux is an open-source operating system that interfaces between the computer's hardware and software programs. The language used with Raspberry Pi is Python – a general-purpose and high-level programming language used to develop graphical user interface (GUI) applications, websites, and web applications. One of the benefits of Raspberry Pi is that it is not necessary to have an intimate knowledge of Linux or Python before beginning a project with Raspberry Pi. In fact, the purpose of the product is to teach the system and language through engaging projects.

The most basic model is the Raspberry Pi Zero or Raspberry Pi Zero W – the Zero W comes with WiFi and Bluetooth capabilities while the Zero does not. The basic model provides the user the opportunity to learn the computer language and explore the Internet of Things (IoT) with projects designed to keep the learner engaged. The IoT is a system that consists of interrelated computing devices and mechanical and digital machines providing the ability to transfer data over a network. Using Raspberry Pi Zero, you can undertake projects such as word clocks, environmental monitors (temperature, humidity, etc.), airplay speakers, informational displays, drones, retro games, and selfiebots.

With each project that you complete, you gain knowledge and skills that increase your ability and confidence. When you outgrow the Zero, move up to the next model. Each version of the Raspberry Pi integrates new user functions such as USB ports, a media center, and a smart home hub. Use the Raspberry Pi's later versions to build your home security system, run dual HD monitors, and build a home theater PC. The projects and applications for Raspberry Pi are limited only by your curiosity and desire to learn.

## Applications of Raspberry Pi

The raspberry pi boards are used in many applications like Media streamer, Arcade machine, Tablet computer, Home automation, Carputer, Internet radio, Controlling robots, Cosmic Computer, Hunting for meteorites, Coffee and also in raspberry pi based projects.

## Uses Of Raspberry Pi

Raspberry Pi is a series of small, single-board computers developed to teach computer science basics to school students and other people in low-income countries. It became a popular and easy to experiment tool to develop school projects, hardware programming, robotics, basic automated machines, circuits, etc. The Uses of Raspberry Pi is a small, quite affordable, and very much capable hardware device called a credit card size computer.

There are several benefits of using a Raspberry Pi. Please find the below sections, where Raspberry Pi has been used widely and effectively. Below is the list of the top 10 uses of Raspberry Pi.

### 1. Desktop PC

Using Raspberry Pi, the microSD card, and a power supply, a simple desktop can be made. We would also need an HDMI cable and a suitable display, maybe an old monitor. A USB keyboard and mouse are also needed.

The new version, which is Raspberry Pi 3, has built-in Wi-Fi and Bluetooth too. If a different model is used, compatible USB dongles would be required.

Once everything is set up, and preferred operating system installed (the latest version of Raspbian), your desktop computer is ready to be used.

### 2. Wireless print server

This requires installing Samba file-sharing software and CUPS (Common Unix Printing System). CUPS provide drivers for the printer and administration console.

After this, Pi configuration is needed to ensure a Windows or Mac computers can access the printer via a network. The printer must have a USB cable.

### 3. Media Usage

Many estimates suggest one of the main uses of Raspberry Pi is a Kodi media center. Several Kodi builds have been released as disk images. OSMC and OpenElec are among the most popular.Installing Kodi comes with some caveats. It is recommended that we install only safe and legal add-ons from the official Kodi repositories. Also, a Raspberry Pi running Kodi is vulnerable to a few security issues. Hence, setting up a VPN to encrypt data is recommended.

### 4. Game Servers

Raspbian, the default OS of pi comes with a special version of Minecraft game pre-installed. But, the applications of Raspberry Pi can be used as a game server as well. It is an excellent

game server for Minecraft. If multiple Raspberry Pis are used, making one as a dedicated server, a great gaming experience can be achieved.

Other multiplayer network games can be set up on the Raspberry Pi.

**5. Retro Gaming Machine**

Raspberry Pi is ideal as a retro gaming machine. it fits as one of the lightest components of a machine. Particularly, it's a version, The Raspberry Pi Zero, that can fit into small spaces for gaming projects. There are two main options, Recalbox and RetroPie. Other platforms can be emulated too. Classic MS-DOS PC gaming and Commodore 64 can also be set-up and also many other popular 16-bit games consoles.

**6. Robot Controller**

There are many robot-controller Raspberry Pi projects. There is a dedicated robotics package for Pi, duly powered with the device battery and used to communicate and control robots.for robots, Pi Zero W can only be used. Zero, a slim line version of the Raspberry Pi, has features of onboard wireless connectivity suitable for lightweight robots.

It's quite lighter than the Model B+ boards of version 2 and 3 of pi, and the low profile ensures it can be placed in an efficient position without having a concern about USB ports.

**7. Stop Motion Camera**

Using Python and a suitable mount (standard tripod for clay- or toy-based) and the availability of a well-lit area Stop motion camera can be built. But, this is a time-consuming process. One needs a good amount of practice to get good results.

**8. Time-lapse Camera Combining**

The Raspberry Pi camera module and different script creates another use that captures movies. This can be achieved by taking single frames with a time delay. Also needed is, perhaps a portable battery solution, and a tripod can be used. A smartphone tripod is most preferred to ensure the device remains sturdy.

**9. FM Radio Station**

Raspberry Pi can also be used to broadcast on FM radio. Pi can broadcast only over a short-range. A portable battery and soldering skills may be required here. Any audio which needs to broadcast will need to be loaded beforehand to the microSD card.

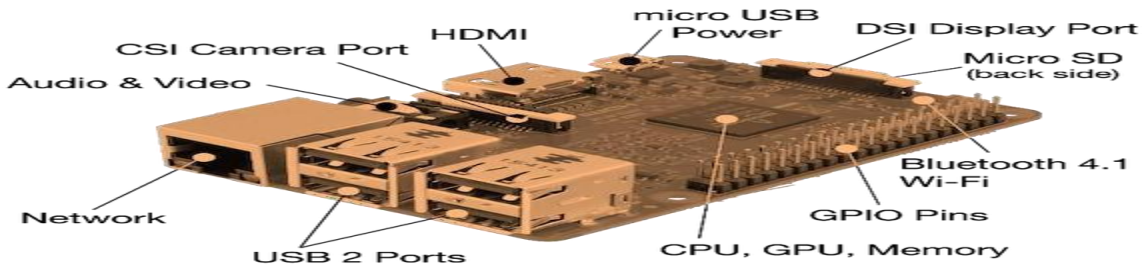
**10. Web Servers**

Another great application of Raspberry Pi is to create a web server out of it. What this means is that it can be configured to host a website much like any other server. It can host blogs too.

First of all, the right software needs to be installed and that is Apache and its dependent libraries. A full LAMP stack can also be installed with PHP, MySQL, and Apache too. Setting up FTP is also helpful.

Once all these steps as mentioned are completed, HTML files can be saved into the /www/ directory, and the webserver is ready to be used. Specific web software like WordPress can also be used once the server setup is complete.

## Parts/Components of the Raspberry Pi




**GPIO**

GPIO is arguably the most important feature of the Raspberry Pi and is the equivalent of GPIO pins on the Arduino. These pins can be used in programs to read electrical signals from circuits as well as provide electrical signals for controlling circuits. Be very careful when using GPIO as they are easily damaged and use 3.3V logic. If you intend to control external devices that draw more than 20mA current, you should use a driver circuit (see 3.3 connecting I/O). This includes devices such as relays, inductors, and high brightness LEDs.

**DSI Display Port**

The DSI display port allows the Raspberry Pi to connect to a serial display similar to those used in tablets. Such display modules are available with touch controls and in common sizes such as 7 inches.

**CSI Camera Port**

The CSI camera port is a connector that allows the Raspberry Pi to connect to a Raspberry Pi camera module. Generic web cameras will not work as they commonly have only a USB connector.

**MicroSD Slot**

This slot is used to house the microSD card that holds the Raspberry Pi operating system. The microSD card does not come with the Pi. This SD card also holds all files, folders, documents, and pictures created by the user. It is essentially the hard drive of the computer.

**HDMI / USB / Network**

These slots are used to connect the Pi to an HDMI screen, USB devices such as mice and keyboards, and to an ethernet connection for internet access. However, the Raspberry Pi 3 comes with integrated Wi-Fi so there is often no need for the ethernet connector.

**Micro USB Power**

Power to the Raspberry Pi can be provided using either a micro USB lead to the micro USB connector (recommended) or 5V can be directly fed into the 5V GPIO pin.

## What is Arduino?

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. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike.

Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open-source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

## Importance of Arduino?

Arduino is used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. It is used to build low-cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments.

There are many other microcontrollers and microcontroller platforms available for physical

computing. Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantages like;

- 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.
- 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.

## Basics of Arduino UNO

Let's have a look at the basic details of Arduino UNO:

- Arduino UNO is a Microcontroller board designed by Arduino.cc in Italy.
- It used an Atmega328 Microcontroller which acts as the brain of this board.
- Arduino Bootloader is installed on Atmega328 which makes it capable to work with Arduino Programming.
- Arduino is an open-source platform so it has a lot of support from third-party developers.
- Anyone can design its Libraries for different sensors and modules.

## Arduino Applications

Below are the Arduino Applications:

**Home Automation**

This application makes use of the Arduino Uno board, Bluetooth interface for connectivity, and smartphones. Software loaded boards are connected to the home devices like lamps, A/C, TV, Refrigerator, and Bluetooth software is interfaced with the board. The app loaded in the smartphone talk to the processor through Bluetooth connectivity and inputs from the phone are used to control the operation of the devices.

Operations like switch on, switch off, increasing or decreasing the intensity, volume, and other operating of parameters of these devices. Remote monitoring and operation is also enabled. These applications simplify the operation of household gadgets and enables better control.

**Public Utility Automation**

Applications to manage public utilities like street lighting, Dynamic traffic management systems are being implemented.

Street lighting Street lights are fitted with Arduino boards and sensors. The microcontroller is programmed to read the inputs from the signal sent by the sensor on the light and temperature change and dynamically change the voltage supplied to the lights and control the intensity of the light. This system can be used to switch on and switch off the light also. Dynamic traffic Management Arduino controller along with infra-red sensors helps in managing the traffic dynamically. Input from the sensor helps the controller to measure the volume of traffic and accordingly control the timing of signals as per the traffic flow and its direction.

**IoT**

Poka-yoke This system suggests the right component be fitted at any stage in the assembly line. This system senses the product that is being assembled and refers ERP system and finds out the component to be fitted at that stage and accordingly illuminates the light of the compartment of that component. The operator picks up that component where the light glows and thus picking the right component is ensured and mistake-proofing is ensured. Arduino board based on the input from the central server send a signal to right bulb in the circuit and illuminates it.

Production counting Sensor placed in the conveyor is activated when the product being assembled moved from one stage to the next stage. Arduino board takes the signal from the sensor and adds to the count and sends the data back to the central server.

## Difference between Arduino and Raspberry Pi

There are a wide variety of controller boards that we can use for our hardware projects. The

two most popular among them are: Arduino and Raspberry Pi.

Arduino is based on the ATmega family and has a relatively simple design and software structure. Raspberry Pi, basically is a single-board computer. Both of them have a CPU which executes the instructions, timers, memory and I/O pins. The key distinction between the two is that Arduino tends to have a strong I/O capability which drives external hardware directly. Whereas Raspberry Pi has a weak I/O which requires transistors to drive the hardware.



| Sr. No. | Arduino | Raspberry Pi |
|---|---|---|
| 1. | Control unit of Arduino is from Atmega family. | While control unit of Raspberry Pi is from ARM family. |
| 2. | Arduino is based on a microcontroller. | While Raspberry Pi is based on a microprocessor. |
| 3. | It is designed to control the electrical components connected to the circuit board in a system. | While Raspberry Pi computes data and produces valuable outputs, and controls components in a system based on the outcome of its computation. |
| 4. | Arduino boards have a simple hardware and software structure. | While Raspberry Pi boards have a complex architecture of hardware and software. |
| 5. | CPU architecture: 8 bit. | CPU architecture: 64 bit. |
| 6. | It uses very less RAM, 2 kB. | While Raspberry Pi requires more RAM, 1 GB. |
| 7. | It clocks a processing speed of 16 MHz. | While Raspberry Pi clocks a processing speed of 1.4 GHz. |

| | | |
|---|---|---|
| 8. | It is cheaper in cost. | While Raspberry Pi is expensive. |
| 9. | It has a higher I/O current drive strength. | While Raspberry Pi has a lower I/O current drive strength. |
| 10. | It consumes about 200 MW of power. | While it consumes about 700 MW of power. |

# UNIT 5 Case Study

**IoT for Smart Cities: Use Cases and Implementation Strategies**

The UN predicts that by 2050, the world's urban population is likely to double and reach the point of nearly 6.7 million people. As the number of urban residents grows, cities face new opportunities… And challenges. To prevent environmental deterioration, avoid sanitation problems, mitigate traffic congestion, and thwart urban crime, municipalities turn to the Internet of Things (IoT).

IoT has the potential to tame the pressure of urbanization, create new experience for city residents, and make day-to-day living more comfortable and secure.

In this article, we will share our IoT consulting experience and shed light on the smart city applications, present an optimal approach to the implementation of smart city solutions, as well as explore the peculiarities of rolling out IoT solutions in cities of different sizes.

**IoT use cases for smart cities**

IoT-enabled smart city use cases span multiple areas: from contributing to a healthier environment and improving traffic to enhancing public safety and optimizing street lighting. Below, we provide an overview of the most popular use cases that are already implemented in smart cities across the globe.

**Road traffic**

Smart cities ensure that their citizens get from point A to point B as safely and efficiently as possible. To achieve this, municipalities turn to IoT development and implement smart traffic solutions.

Smart traffic solutions use different types of sensors, as well as fetch GPS data from drivers' smart phones to determine the number, location and the speed of vehicles. At the same time, smart traffic lights connected to a cloud management platform allow monitoring green light timings and automatically alter the lights based on current traffic situation to prevent congestion. Additionally, using historical data, smart solutions for traffic management can predict where the traffic could go and take measures to prevent potential congestion.

For example, being one of the most traffic-affected cities in the world, Los Angeles has implemented a smart traffic solution to control traffic flow. Road-surface sensors and closed-circuit television cameras send real-time updates about the traffic flow to a central traffic management platform. The platform analyzes the data and notifies the platform users of congestion and traffic signal malfunctions via desktop user apps. Additionally, the city is deploying a network of smart controllers to automatically make second-by-second traffic lights adjustments, reacting to changing traffic conditions in real time.

**Smart parking**

With the help of GPS data from drivers' smartphones (or road-surface sensors embedded in the ground on parking spots), smart parking solutions determine whether the parking spots are occupied or available and create a real-time parking map. When the closest parking spot becomes free, drivers receive a notification and use the map on their phone to find a parking spot faster and easier instead of blindly driving around.

**Public transport**

The data from IoT sensors can help to reveal patterns of how citizens use transport. Public transportation operators can use this data to enhance traveling experience, achieve a higher level of safety and punctuality. To carry out a more sophisticated analysis, smart public transport solutions can combine multiple sources, such as ticket sales and traffic information.

In London, for instance, some train operators predict the loading of train passenger cars on their trips in and out of the city. They combine the data from ticket sales, movement sensors, and CCTV cameras installed along the platform. Analyzing this data, train operators can predict how each car will load up with passengers. When a train comes into a station, train operators encourage passengers to spread along the train to maximize the loading. By maximizing the capacity use, train operators avoid train delays.

**Utilities**

IoT-equipped smart cities allow citizens to save money by giving them more control over their home utilities. IoT enables different approaches to smart utilities:

- Smart meters & billing

With a network of smart meters, municipalities can provide citizens with cost-effective connectivity to utilities companies' IT systems. Now, smart connected meters can send data directly to a public utility over a telecom network, providing it with reliable meter readings. Smart metering allows utilities companies to bill accurately for the amount of water, energy and gas consumed by each household.

- Revealing consumption patterns

A network of smart meters enables utilities companies to gain greater visibility and see how their customers consume energy and water. With a network of smart meters, utilities companies can monitor demand in real time and redirect resources as necessary or encourage consumers to use less energy or water at times of shortage.

- Remote monitoring

IoT smart city solutions can also provide citizens with utility management services. These services allow citizens to use their smart meters to track and control their usage remotely. For instance, a householder can turn off their home central heating using a mobile phone. Additionally, if a problem (e.g., a water leakage) occurs, utilities companies can notify householders and send specialists to fix it.

**Street lighting**

IoT-based smart cities make maintenance and control of street lamps more straightforward and cost-effective. Equipping streetlights with sensors and connecting them to a cloud management solution helps to adapt lighting schedule to the lighting zone.

Smart lighting solutions gather data on illuminance, movement of people and vehicles, and combine it with historical and contextual data (e.g., special events, public transport schedule, time of day and year, etc.) and analyze it to improve the lighting schedule. As a result, a smart lighting solution "tells" a streetlight to dim, brighten, switch on or switch off the lights based on the outer conditions.

For instance, when pedestrians cross the road, the lights around the crossings can switch to a brighter setting; when a bus is expected to arrive at a bus stop, the streetlights around it can be automatically set brighter than those further away, etc.

**Waste management**

Most waste collection operators' empty containers according to predefined schedules. This is not a very efficient approach since it leads to the unproductive use of waste containers and unnecessary fuel consumption by waste collecting trucks.

IoT-enabled smart city solutions help to optimize waste collecting schedules by tracking waste

levels, as well as providing route optimization and operational analytics.

Each waste container gets a sensor that gathers the data about the level of the waste in a container. Once it is close to a certain threshold, the waste management solution receives a sensor record, processes it, and sends a notification to a truck driver's mobile app. Thus, the truck driver empties a full container, avoiding emptying half-full ones.

**Environment**

IoT-driven smart city solutions allow tracking parameters critical for a healthy environment in order to maintain them at an optimal level. For example, to monitor water quality, a city can deploy a network of sensors across the water grid and connect them to a cloud management platform. Sensors measure pH level, the amount of dissolved oxygen and dissolved ions. If leakage occurs and the chemical composition of water changes, the cloud platform triggers an output defined by the users. For example, if a Nitrate (NO3-) level exceeds 1 mg/L, a waterquality management solution alerts maintenance teams of contamination and automatically creates a case for field workers, who then start fixing the issue.

Another use case is monitoring air quality. For that, a network of sensors is deployed along busy roads and around plants. Sensors gather data on the amount of CO, nitrogen, and sulfur oxides, while the central cloud platform analyzes and visualizes sensor readings, so that platform users can view the map of air quality and use this data to point out areas where air pollution is critical and work out recommendations for citizens.

**Public safety**

For enhancing public safety, IoT-based smart city technologies offer real-time monitoring, analytics, and decision-making tools. Combining data from acoustic sensors and CCTV cameras deployed throughout the city with the data from social media feed and analyzing it, public safety solutions can predict potential crime scenes. This will allow the police to stop potential perpetrators or successfully track them.

For example, more than 90 cities across the United States use a gunshot detection solution. The solution uses connected microphones installed throughout a city. The data from microphones passes over to the cloud platform, which analyzes the sounds and detects a gunshot. The platform measures the time it took for the sound to reach the microphone and estimates the location of the gun. When the gunshot and its location are identified, cloud software alerts the police via a mobile app.

**Iterative approach to implementing smart city solutions**

The range of smart city applications is highly diverse. What they have in common is the approach to implementation. Whether municipalities plan to automate waste collection or improve street lighting, they should start with the foundation – a basic smart city platform. If a municipality prefers to expand the range of smart city services in future, it will be possible to

upgrade the existing architecture with new tools and technologies without having to rebuild it.

Here is a six-step implementation model to follow for creating an efficient and scalable IoT architecture for a smart city.

**Stage 1: basic IoT-based smart city platform**

To be able to scale, smart city implementation should start with designing a basic architecture – it will serve as a springboard for future enhancements and allow adding new services without losing functional performance. A basic IoT solution for smart cities includes four components:

- The network of smart things

A smart city – as any IoT system – uses smart things equipped with sensors and actuators. The immediate goal of sensors is to collect data and pass it to a central cloud management platform. Actuators allow devices to act - alter the lights, restrict the flow of water to the pipe with leakage, etc.

- Gateways

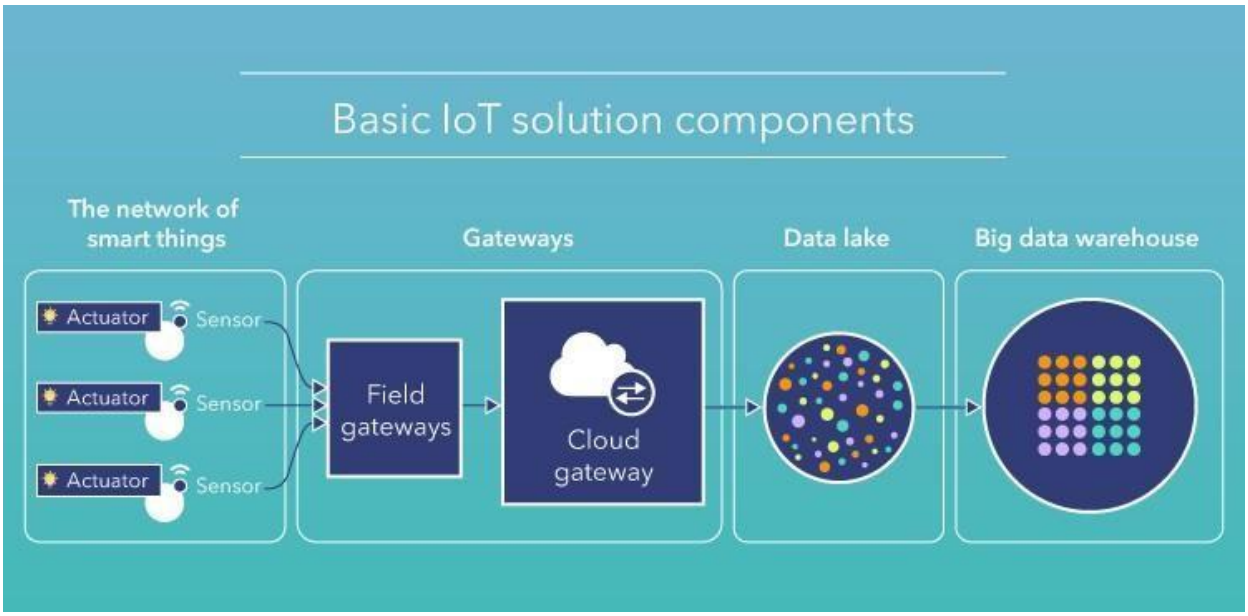
Any IoT system comprises two parts – a "tangible" part of IoT devices and network nodes and a cloud part. The data cannot simply pass from one part to the other. There must be doors – field gateways. Field gateways facilitate data gathering and compression by preprocessing and filtering data before moving it to the cloud. The cloud gateway ensures secure data transmission between field gateways and the cloud part of a smart city solution.

- Data lake

The main purpose of a data lake is to store data. Data lakes preserve data in its raw state. When the data is needed for meaningful insights, it's extracted and passed over to the big data warehouse.

- Big data warehouse

A big data warehouse is a single data repository. Unlike data lakes, it contains only structured data. Once the value of data has been defined, it's extracted, transformed and loaded into the big data warehouse. Moreover, it stores contextual information about connected things, e.g., when sensors were installed, as well as the commands sent to devices' actuators by control applications.




**Stage 2: Monitoring and basic analytics**

With data analytics, it is possible to monitor devices' environment and set rules for control applications (we cover them at stage 4) to carry out a particular task.

For example, analyzing the data from soil moisture sensors deployed across a smart park, cities can set rules for the electronic valves to close or open based on the identified moisture level. The data collected with sensors can be visualized on a single platform dashboard, allowing users to know the current state of each park zone.

**Stage 3: Deep analytics**

Processing IoT-generated data, city administrations can go beyond monitoring & basic analytics and identify patterns and hidden correlations in sensor data. Data analytics uses advanced techniques like machine learning (ML) and statistical analysis. ML algorithms analyze historical sensor data stored in the big data warehouse to identify trends and create predictive modelsbased on them. The models are used by control applications that send commands to IoT
devices' actuators. Here is how it applies in practice.

Unlike a traditional traffic light that is programmed to display a particular signal for a definite period, a smart traffic light can adapt signal timings to the traffic scenario. ML algorithms are applied to historical sensor data to reveal traffic patterns and adjust signal timings, helping to improve average vehicle speed and avoid congestions.

**Stage 4: Smart control**

Control applications ensure better automation of smart city objects by sending commands to

their actuators. Basically, they "tell" actuators what to do to solve a particular task. There are rule-based and ML-based control applications. Rules for rule-based control applications are defined manually, while ML-based control applications use models created by ML algorithms. These models are identified based on data analysis; they are tested, approved and regularly updated.

**Stage 5: Instant interacting with citizens via user applications**

Along with the possibility of automated control, there should always be an option for users to influence the behavior of smart city applications (for example, in case of emergency). This task is carried out by user applications.

User applications allow citizens to connect to the central smart city management platform to monitor and control IoT devices, as well as receive notifications and alerts. For example, using GPS data from drivers' smartphones, a smart traffic management solution identifies a traffic jam. To prevent even bigger congestion, the solution automatically sends a notification to the drivers in the area, encouraging them to take a different route.

At the same time, employees at a traffic control center who use a desktop user app receive a 'congestion alert.' To relieve the congestion and re-route part of the traffic, they send a command to the traffic lights' actuators to alter the signals.

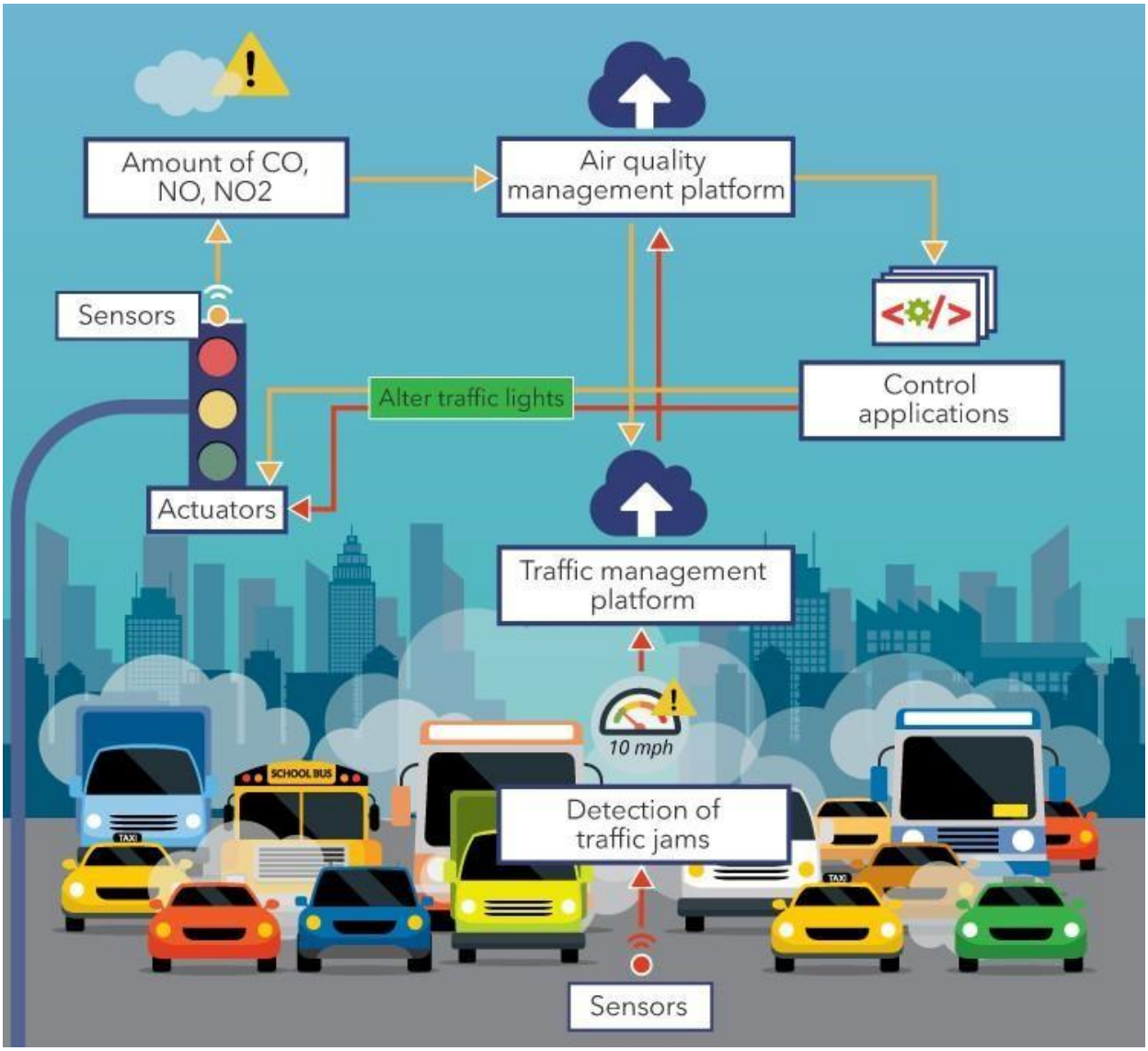
**Stage 6: Integrating several solutions**

Achieving "smartness" is not a one-time action – it is a continuous process. Implementing IoT-based smart city solutions today, municipalities should think of services they might like to implement tomorrow. It implies not only increasing the number of sensors but, more importantly, the number of functions. Let's illustrate this functional scalability with the example of a smart city solution for traffic monitoring.

A city deploys a traffic management solution to detect traffic jams in real time and manage traffic lights to reduce traffic in the areas with intensive traffic. After some time, the city decides to ensure city traffic doesn't harm the environment and integrates the traffic management solution with a smart air quality monitoring solution. Cross-solution integration allows controlling both traffic and air quality in the city dynamically.

For that, traffic lights or street lights along the roads can be equipped with sensors that monitor air quality. Sensors measure the amount of CO, NO, and $NO_2$ in the air and pass data records to a central air quality management platform for processing. If the amount of harmful gases in the air is critical, control applications apply rules or use models to take an output action, e.g., 'alter traffic lights.' Before that, there is a need to make sure that altering traffic lights won't cause accidents or blockages in other areas. It is possible due to the integration of the traffic management solution to the air quality management solution. The traffic management platform performs real-time analysis and identifies if it is possible to alter the traffic lights. If altering the lights is acceptable, control applications send a command to the traffic lights'

actuators, which execute the command.




Applying an iterative approach helps municipalities to reduce implementation costs, get a faster pay-off and make the benefits of smart solutions visible for citizens sooner.

Adapting IoT implementation strategy to the city size

Iterative approach can be leveraged in cities of different sizes. In larger ones, it helps to deal with the scale and complexity of implementation; in smaller ones, it helps to reduce investments in smart solutions and use constrained infrastructure resources more reasonably. However, starting a smart project in a smaller city, municipalities have some more points to consider.

On the way to smartness, midsized and small cities face many barriers, including budgetary and

procurement shortages, limited resources for public services, under-resourced IT infrastructure, etc. However, it doesn't mean a smaller city cannot be a smart city.

Starting a smart initiative in a city of medium or small size, it makes sense to begin with the projects that do not require huge investments and deliver tangible return on investments, such as smart parking or waste management, and use the established infrastructure to implement new services.

For example, the town of Vail, CO has less than 6,000 inhabitants but boasts an extensive smart infrastructure. The town started smart city development with connected streetlights. Later, they used the established infrastructure to broaden the range of services and topped it with smart parking and irrigation systems.

To determine which applications are a good fit for smaller cities, we've analyzed them by the volume of investments, required infrastructure, pay-off period, the visibility of benefits for citizens and came up with the following table:

THE RELEVANCE OF IOT APPLICATIONS FOR SMALLER SMART CITIES

| | Highly relevant | Can be implemented with certain restrictions | The value is questionable |
|---|---|---|---|
| Traffic management | | | ✓ |
| Parking | ✓ | | |
| Public transport | | ✓ | |
| Utilities | | | ✓ |
| Street lightning | ✓ | | |
| Waste management | ✓ | | |
| Environment | | ✓ | |
| Public safety | | ✓ | |

Another non-trivial way to enhance the affordability and accessibility of smart applications is sharing a common platform with a larger city. The cloud nature of IoT-enabled smart city solutions is suitable for that. This way, smart city solutions of both large and smaller smart cities are connected to and managed via a single cloud platform. By sharing the platform based on open data, several smart cities form a common urban ecosystem. One of the examples of such sharing is the Iberian Smart Cities Network, which currently includes 111 cities in Portugal and Spain. The network comprises cities of different sizes, which cooperate in multiple areas including smart energy, mobility, environment, and transport.

Let's sum it up, IoT helps cities connect and manage multiple infrastructure and public services. From smart lighting and road traffic to connected public transport and waste management – the range of use cases is highly diverse. What they have in common is the outcomes. Applying IoT solutions leads to reduced costs for energy, optimized use of natural resources, safer cities, and a healthier environment.

However, to enjoy these benefits, municipalities should take a consistent approach to design a functional and scalable smart city architecture. Well-designed, it will allow to reduce investments in IoT development and hasten the implementation of smart city solutions, still leaving space for expansion.

## IoT for smart home—a case study

IoT is the next step in the evolution of the internet and is being used in about everything you can think of. This project aimed to scope out use cases if LG were to start its own IoT wave for smart homes.

The Process


**Use Cases:**

• A new wave of connected appliances will enable better user experience, proactive alerts, and even safety notifications.

• Water and air treatment systems at home, their performance data, customized alerts on each device performance, automated consumable ordering and even automatically adjusts to your water usage patterns.

• Adding connectivity to fire safety devices can provide homeowners the ability to monitor appliances remotely and even send alerts to friends and neighbours in the case of an emergency.

**Research**:

For the research part, I googled a lot and tried to understand and grab as much I could about IoT's and its potential users.

"User-centered design means working with your users all throughout the project."— Don Norman

For personas, I interviewed various potential people at my workplace and at home.


Quick Summary :- Read the tutorial blog on how to implement Home Automation using IoT. It covers the software, hardware, sensors, protocols, architecture and platforms. Applications of IoT-enabled connectivity are home security, air quality monitoring, infotainment delivery, smart lock etc.

## Case Study 2: Home Automation Using the Internet of Things (IoT)

What really would compel someone to actually develop a product which is a complete IoT-based home automation system? Could it be the need to improve the safety of your home, could it be the desire to live a Jetson-like life that millennials always dreamt of.

It is difficult to say, often, it is even more difficult to visualize the technology that is required to build a home automation platform.

Due to the complexity introduced by software, hardware and networking ecosystems, it becomes extremely important to learn, understand and utilize the right home automation technology for your smart home product.

We hope to address some of the concerns with this article.

Home automation has three major parts:

- Hardware
- Software/Apps
- Communication protocols

Each of these parts is equally important in building a truly smart home experience for your customers. Having the right hardware enables the ability to develop your IoT prototype iteratively and respond to technology pivots with ease.

A protocol selected with the right testing and careful consideration helps your avoiding performance bottlenecks that otherwise would restrict the technology and device integration capabilities with sensors and IoT gateways.

Another important consideration is the firmware that resides in your hardware managing your data, managing data transfer, firmware OTA updates and performing other critical operations to make things talk.

**Applications of home automation**

Rebuilding consumer expectations, home automation has been projected to target wide array applications for the new digital consumer. Some of the areas where consumers can expect to see home automation led IoT-enabled connectivity are:

- Lighting control
- HVAC
- Lawn/Gardening management
- Smart Home Appliances
- Improved Home safety and security
- Home air quality and water quality monitoring
- Natural Language-based voice assistants
- Better Infotainment delivery
- AI-driven digital experiences
- Smart Switches
- Smart Locks
- Smart Energy Meters

The list is still not exhaustive and will evolve over the time to accommodate new IoT use cases.

Now that you are familiar with home automation applications, let's have a detailed look at what components are involved in building a typical home automation prototype.

**Home automation components**

We have talked about them before, but, let's clearly separate them into components that would finally help you build a realistic model of what major components are involved in building a smart home. The major components can be broken into:

- IoT Sensors
- IoT Gateways
- IoT Protocols
- IoT Firmware
- IoT Cloud and Databases
- IoT Middleware (if required)

IoT sensors involved in home automation are in thousands, and there are hundreds of home automation gateways as well. Most of the firmware is either written in C, Python, Node.Js, or any other programming language.

The biggest players in IoT cloud can be divided into a platform as a service (PaaS) and infrastructure as a service (LaaS).

**Major IoT platform as a service provider:**

- AWS IoT
- Azure IoT
- Thingworx
- Ubidots
- Thingspeak
- Carriots
- Konekt
- TempoIQ

- Xively
- IBM Bluemix

**Characteristics of IoT platforms**

Again, these platforms are extremely divided over the IoT application and security-related features that they provide. A few of these platforms are open source.

Let's have a look at what you should expect from a typical IoT platform:

- Device security and authentication
- Message brokers and message queuing
- Device administration
- Support towards protocols like CoAP, MQTT, HTTP
- Data collection, visualization, and simple analysis capabilities
- Integrability with other web services
- Horizontal and vertical scalability
- WebSocket APIs for real time for real-time information flow

Apart from what we mentioned above, more and more platform builders are open sourcing their libraries to developers. Take for example the Dallas temperature library for DS18B20 for Arduino was quickly ported because of open source development to a new version that helped developers to integrate DS18B20 with Linkit One. Understanding these things become crucial as IoT tends to evolve continuously and having an equally responsive platform makes it business safe to proceed.

Let's now deeply evaluate each of these components, starting with IoT sensors

**Home Automation Sensors**

There are probably thousands of such sensors out there that can be a part of this list. Since this is an introduction towards smart home technology, we will keep it brief. We will break down IoT sensors for home automation by their sensing capabilities:

- Temperature sensors
- Lux sensors
- Water level sensors

- Air composition sensors
- Video cameras for surveillance
- Voice/Sound sensors
- Pressure sensors
- Humidity sensors
- Accelerometers
- Infrared sensors
- Vibration's sensors
- Ultrasonic sensors

Depending upon what you need you may use one or many of these to build a truly smart home IoT product. Let's have a look at some of the most commonly used home automation sensors.

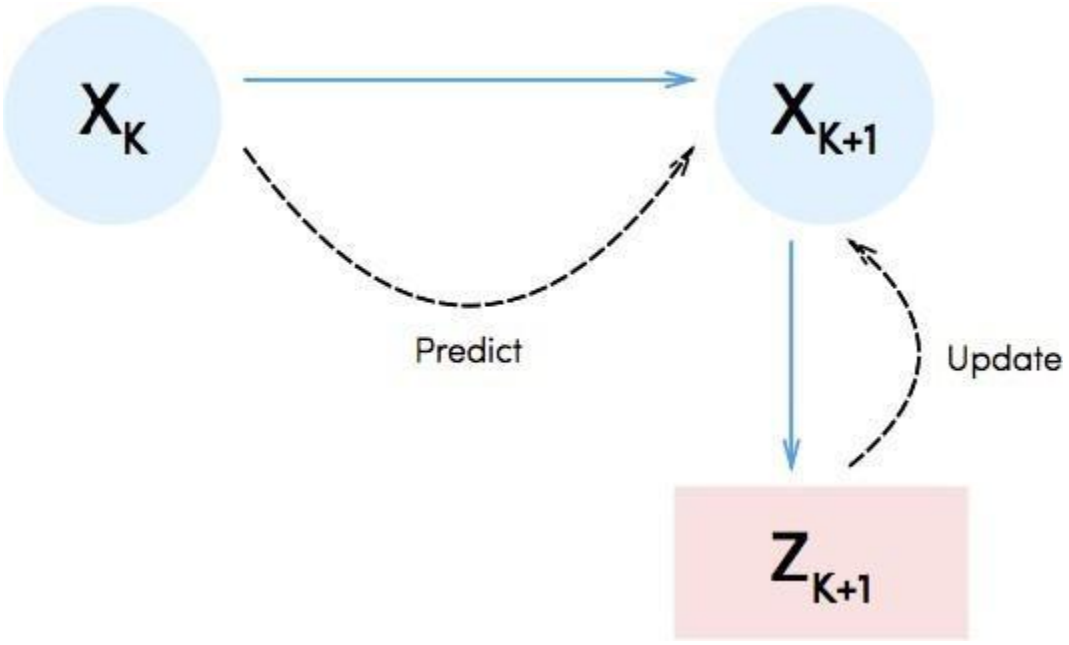
**Temperature sensors**

The market is full of them, but the famous temperature sensors are DHT11/22, DS18B20, LM35 and MSP430 series from TI. MSP430 series is more accurate than the rest but at the same time is one of the most expensive for prototyping or initial product testing purposes. MSP430 tops all temperature sensors as the precision and battery consumption is minimum with them.

MSP430 tops all temperature sensors as the precision and battery consumption is minimum with them.

DHT11 has a very restricted temperature range and suffers from accuracy issues. DHT22, on the other hand, is a little bit more accurate but still, doesn't make it as the preference.

DS18B20's, on the other hand, are more accurate, as opposed to digital temperature sensors like DHT22 and 11, Dallas temperature sensors are analog and can be extremely accurate down to 0.5 degrees.




Take note that often the temperatures that you directly sense from these sensors may not be very accurate and you would occasionally see 1000 F or greater values no matter what you are doing.

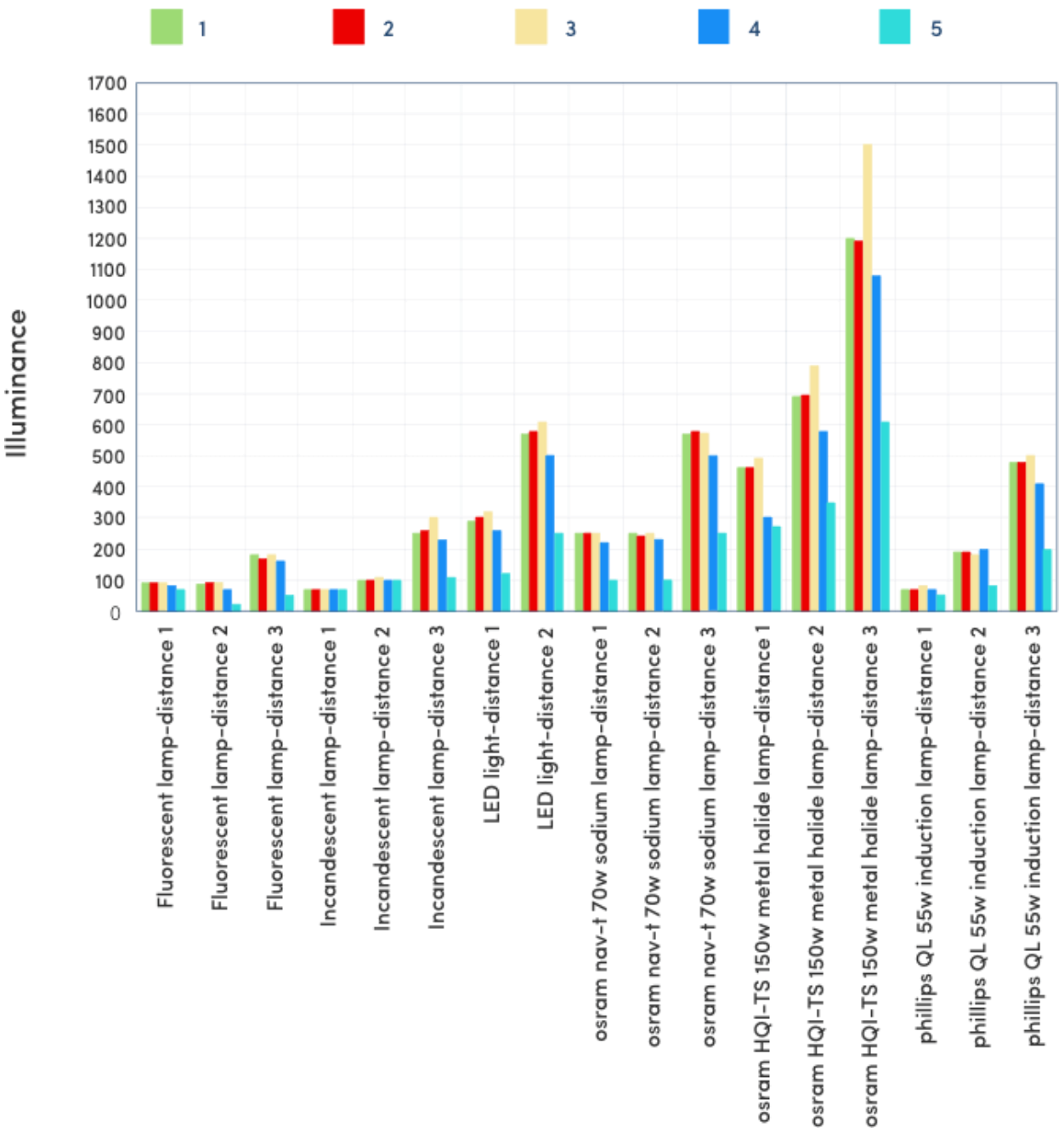
There's an entire logic that goes around building temperature sensors, that we will address in another blog post.

**Lux Sensors**

Lux sensors measure the luminosity and can be used to trigger various functions range from cross-validating movements to turn the lights on if it becomes too dark. Some of the most popular light sensors are TSL2591 and BH1750.

Recent tests to include TSL2591 and BH1750 into low-powered IoT devices have found them to be working fairly good for most of the use cases.

Here's a study was done by Robert and Tomas that shows how these two compare against a spectrometer and a photodiode.


1
2
3
4
5
Illuminance
1700
1600
1500
1400
1300
1200
1100
1000
900
800
700
600
500
400
300
200
100
0
Fluorescent lamp-distance 1
Fluorescent lamp-distance 2
Fluorescent lamp-distance 3
Incandescent lamp-distance 1
Incandescent lamp-distance 2
Incandescent lamp-distance 3
LED light-distance 1
LED light-distance 2
osram nav-t 70w sodium lamp-distance 1
osram nav-t 70w sodium lamp-distance 2
osram nav-t 70w sodium lamp-distance 3
osram HQI-TS 150w metal halide lamp-distance 1
osram HQI-TS 150w metal halide lamp-distance 2
osram HQI-TS 150w metal halide lamp-distance 3
phillips QL 55w induction lamp-distance 1
phillips QL 55w induction lamp-distance 2
phillips QL 55w induction lamp-distance 3

To get a good idea of whether these two sensors would suffice your needs we would suggest illuminance tests followed by normalization of the data to observe deviations under various situations.

**Water level sensors for Home Automation**

While building your prototype you may consider a solid state eTape liquid level sensor, or like others who just use an HC-SR04 ultrasonic sensor to measure the water level sensor.

On the other hand, in other cases where those two don't suffice, one has to utilize something that can deliver a much higher performance.

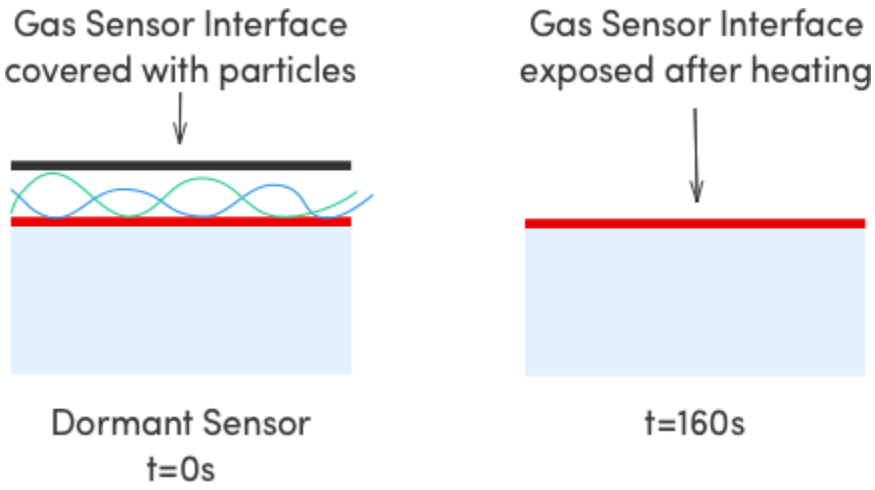
Float level sensors and other ICs like LM1830 offers a more precise measurement capability to IoT developers. Although, they are substantially much more expensive than others.

**Air composition sensors**

There are a couple of specific sensors that are used by developers to measure specific components in the air:

- CO monitoring by MiCS-5525
- MQ-8 to measure Hydrogen gas levels
- MiCS-2714 to measure nitrogen oxide
- MQ135 to sense hazardous gas levels (NH3, NOx, Alcohol, Benzene, smoke, CO2

Most of these are sensors have a heating time, which also means that they require a certain time before they actually start delivering accurate values.



These sensors mainly rely on their surface to detect gas components. When they initially start sensing, there's always something that's there on their surface, some sort of deposition that requires some heating to go away.
Hence, after the surface gets heated enough true values start to show up.

**Video cameras for surveillance and analytics**

A range of webcams and cameras specific to Hardware development kits are usually used in such scenarios. Hardware with USB ports offers to integrate and camera module to build functionalities.

But, utilizing USB ports in not very efficient, especially in the case of real-time video transfer or any kind of video processing.

Take RaspberryPi for example, it comes with a camera module (Pi cam) that connects using a flex connector directly to the board without using the USB port. This makes the Pi cam extremely efficient.

**Sound detection for Home Automation**

Sound detection plays a vital role from monitoring babies to turning on and off lights automatically to automatically detecting your dog's sound at the door and opening it up for them.

Some commonly used sensors for sound detection includes SEN-12462 and EasyVR Shield for rapid prototyping.

These sensors aren't as good as industrial grade sensors like those from 3DSignals which can detect even ultra-low levels of noise and fine tune between various noise levels to build even machine break up patterns.

**Humidity sensors for Home Automation**

These sensors bring the capability of sensing humidity/RH levels in air for smart homes. The accuracy and sensing precision depend a lot on multiple factors including the overall sensor design and placement.

But certain sensors like DHT22 and 11 built for rapid prototyping would always perform poorly when compared to high-quality sensors like HIH6100 and Dig RH.

While building a product to sense humidity levels, ensure that there's no localized layer of humidity that is obscuring the actual results. Also, keep into consideration that in certain small spaces, the humidity might be too high at one end as compared to the others.

When you look at free and open spaces where the air components can move much freely, the

distribution around the sensor can be expected to be uniform and subsequently would require very less number of corrective actions for the right calibration.

**Home Automation Protocols**

One of the most important parts of building a home automation product is to think about protocols, protocols that your device would use to communicate to gateways, servers, and sensors. A few years ago, the only way to do so was by either using Bluetooth, wifi or GSM. But due to added expenses on cellular sim cards, and low performance of Wifi, most such solutions didn't work.

A few years ago, the only way to do so was by either using Bluetooth, wifi or GSM. But due to added expenses on cellular sim cards, and low performance of Wifi, most such solutions didn't work.

Bluetooth survived and later evolved as Bluetooth Smart or Bluetooth low energy. This helped bring a lot of connectivity in the "mobile server powered economy", in this essentially your phone would act as a middleware to fetch data from BLE powered sensors and sent it over to the internet.

When looking at the major home automation protocols, the following tops the list:

- Bluetooth low energy or Bluetooth Smart: Wireless protocol with mesh capabilities, security, data encryption algorithms and much more. Ideal for IoT-based products for smart homes.
- Zigbee: Low cost, mesh networked and low power radio frequency-based protocol for IoT. Different Zigbee versions don't talk to each other.
- X10: A legacy protocol that utilizes powerline wiring for signaling and control
- Insteon: Communicates with devices both wirelessly and with wires
- Z-wave: Specializes in home automation with an emphasis on security
- Wifi: Needs no explanation
- UPB: Uses existing power lines installed in a home, reduces costs
- Thread: A royalty-free protocol for smart home automation, uses a 6lowpan
- ANT: An ultra-low power protocol helping developers build low-powered sensors with a mesh distribution capabilities.
- 6lowpan

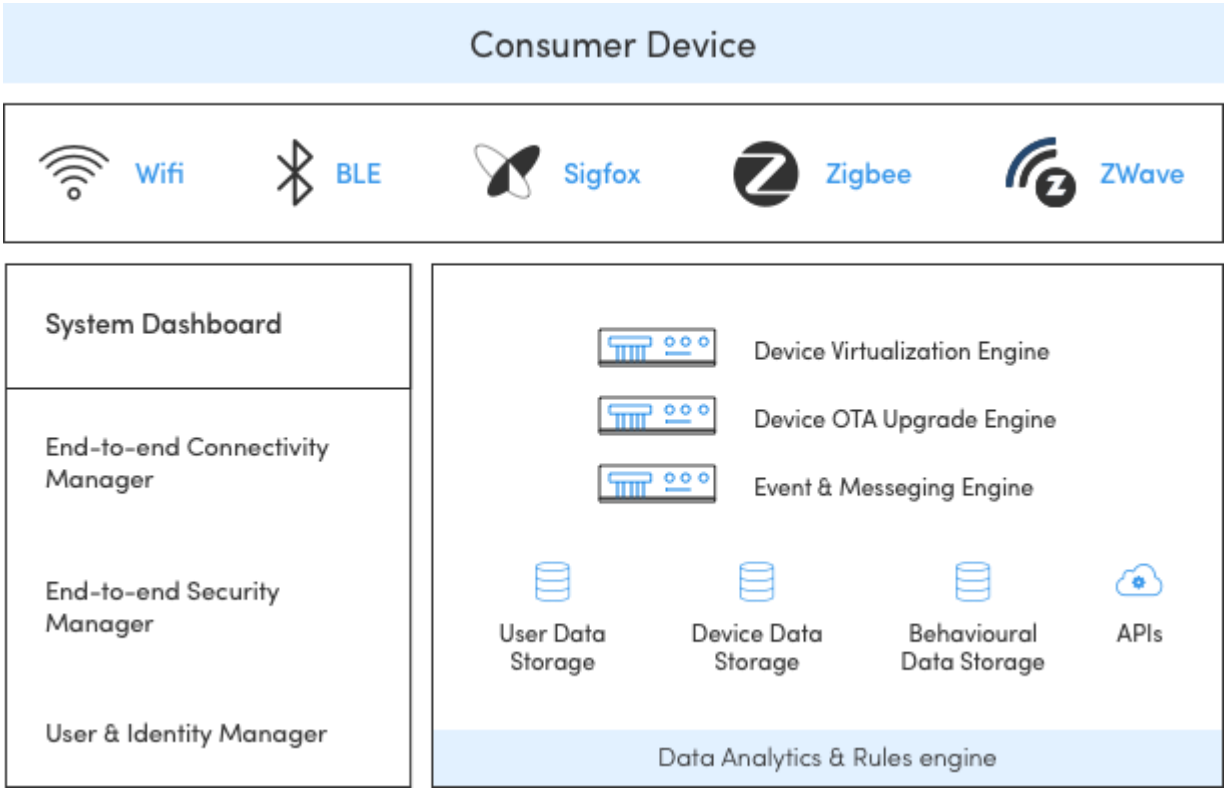
**Home Automation: Which protocol is the best?**

While there are some protocols that clearly offer much more than others, but it is always important to start from your smart home development needs and then move towards narrowing down the solutions.

The commonly preferred protocols are Bluetooth low energy, Z-wave, Zigbee, and Thread. The protocol selection can now be narrowed down by the following factors:

- Ability to perform identity verification
- Quality of sensor networks
- Data transfer rate
- Security level
- Network topology required
- Density of objects around
- Effective Distance to be covered

**Home Automation Architecture**



This architecture supports the following considerations for home automation solutions:

- End to end security mechanisms involving multilevel authentication
- End to end data encryption, including the link layer
- Flexible and configurable access and authorization control
- Powerful cloud infrastructure
- Network agnostic with built-in feedback loops
- Configurable cloud-based rules engine
- API endpoints
- Data scalability
- NoSQL databases

**Home Automation Gateways**

For developing a home automation product, often stand-alone product sending data to a server is not enough. Often due to battery and protocol limitations, the data from a sensor or sensors present in a home has been routed through an IoT gateway.

To select the perfect gateway for your IoT home automation, consider some of the factors including:

- Communication protocols supported
- Real-time capabilities
- MQTT, CoAP, HTTPS support
- Security and configuration
- Modularity

When it comes to building IoT gateways, modularity and hybrid IoT protocol support top that list when a product is in the early stages of market introduction.

To incorporate a gateway in your home automation stack you can consider the following options:

Either create a Gateway from the ground up using existing hardware stacks for prototyping (using Raspberry Pi, Intel Edison, etc). Then when a PoC is validated, you can create your own

custom hardware.

Or, you can use existing gateway modules like Ingincs BLE gateway. These gateways are extremely easy to customize and connect with your cloud services and devices. However, they may or may not offer the same level of support that you need to build certain features.

For example, a gateway with a bad networking queue may result in traffic congestion, or it may not support the required protocols that you wish to use.

Further, pivoting with these gateways to some other technology stack may become very difficult. It should have been emphasized that they are extremely good for robust prototyping needs.

**Home automation programming language for smart home developers**

The following programming languages dominated the home automation space: Python, Embedded C, C, Shell, Go, Javascript (node.js). This has mainly happened due to the sheer optimization of the languages for similar use cases.

**Home Automation frameworks**

If you think you can build everything from home automation (protocols, hardware, software, etc) on your own, it is a bit unrealistic. Everyone starting from high growth startups to billion-dollar consumer focused enterprises are now taking the help of home automation frameworks to build connected products to delight consumers.

Everyone starting from high growth startups to billion-dollar consumer focused enterprises are now taking the help of home automation frameworks to build connected products to delight consumers.

There are more than 15 different smart home frameworks available for IoT developers to use and build their next generation of connected home products. Some of these frameworks are open source and some are closed-source. Let’s have a look at some of them in the sections that follows.

Some of these frameworks are open source and some are closed-source. Let’s have a look at some of them in the sections that follows.

**Open source IoT platforms and frameworks for Home Automation**

Looking forward to doing a quick and dirty prototype? There’s no need to write down everything from scratch. Thanks to a bunch of awesome contributions by people like we have open-source platforms that can get your home automation products up and running in no time.

Our favorites are:

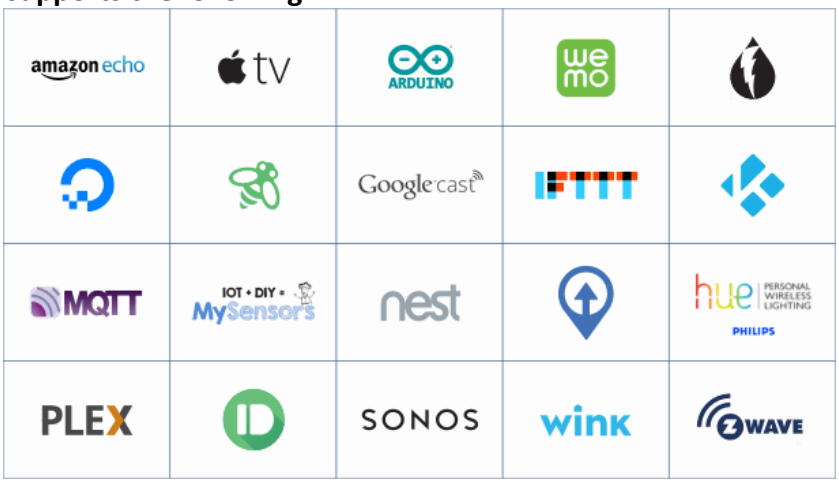
- Home Assistant

- Calaos
- Domoticz
- OpenHAB: Supports Raspberry Pi, written in Java and has design tools to build your own mobile apps by tweaking UI.
- OpenMotics[Asked their developer, waiting for them to respond(dev confirmed)]
- LinuxMCE
- PiDome
- MisterHouse
- Smarthomatic

**Home Assistant for smart home development:**

Supports RaspberryPi, uses Python with OS as Hassbian. It has simplified automation rules that developers can use to build their home automation product saving them thousands of lines of code.

**Home Assistant supports the following:**




How home assistant works involve the following:

- Home control responsible for collecting information and storing devices
- Home automation triggers commands based on user configurations
- Smart home triggers based on past user behavior

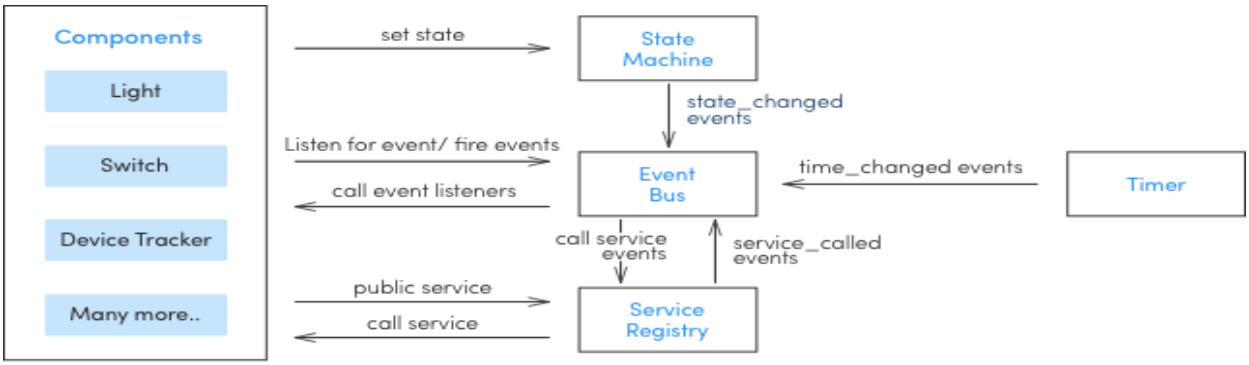
As developers, it is very important for us to understand the architecture of Home Assistant for

us to build high-performing products on top of it.

Let's have a look at Home control's architecture that makes control and information flow possible.

**Home control consists of five components:**

- Components
- State machine
- Event bus
- Service registry
- Timer




The core architecture of Home Assistant

All of these components working together create a seamless asynchronous system for smart home IoT. In the earlier version of Home Assistant core, the core often had to stop while looking for new device information.

But, with the new versions of home assistant, a backward compatible API, and ansyn core have been introduced making things a lot faster for IoT applications.

The best part about home assistant's core architecture is how carefully it has been designed and developed to support IoT at home.

## IoT in Healthcare: Applications and Use Cases

The growth of IoT into nearly every business arena from medical devices and healthcare applications to industrial IoT (IIoT) is amazing to behold. Our series highlighting the range of use cases for the Internet of Things illustrates how IoT products and services are being deployed around the globe, by industry. This article focuses on the range of IoT use cases in healthcare today, supporting patients, doctors, medical staff and first responders in achieving better

outcomes.

Why is IoT in healthcare a fast-growth industry? There are a number of reasons, including the capability of connected devices to monitor health vitals, route data, provide alerts, administer medications and automate critical processes. The medical industry is adopting Internet of Things technologies in everything from medical wearables to patient monitoring and pharmaceutical temperature monitoring in order to improve accuracy, promote efficiency, reduce costs, meet compliance requirements and enhance health and safety. In fact the term "healthcare IoT" or HIoT has been coined to describe this market niche. Digi solutions support development and deployment of a broad range of products and applications in this space.

Let's take a tour of some examples of IoT in medical and healthcare, including Digi customer case studies that help to demonstrate the breadth of IoT applications in healthcare patient support. You can find more examples of applications for a range of industries in the Customer Stories section of the Digi site.

**IoT in Healthcare - Promoting Hygienic Hospitals and Clinics**

There are many healthcare applications related to hygiene, and this became more imperative than ever as the COVID-19 pandemic took center stage around the world. As we shared in our article about how the pandemic accelerated the need for IoT solutions, the Internet of Things provides the right capabilities at the right time for no-contact applications and remote connectivity, all of which support more hygienic health management.

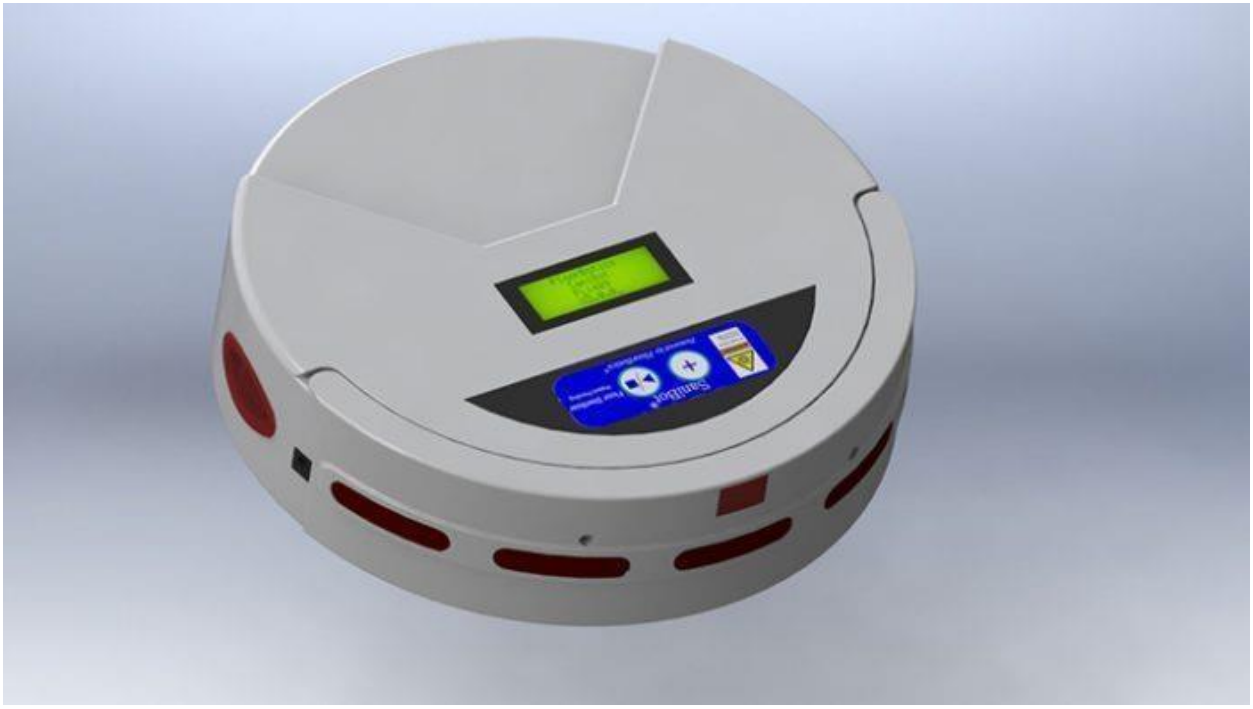
Examples of low-touch and no-touch health and medical applications include:

- Contact tracing
- Pathogen detection
- Thermal detection (elevated temperature)
- No-touch sanitation dispensers
- Automated hand hygiene
- Hygiene monitoring
- Workspace and floor sanitation
- Air quality sensors
- Biometrics scanners
- Vital signs monitoring
- Remote patient communications

- Instrument sterilization
- Medication dispensing

Here are a few examples of how Digi customers have built healthcare applications supporting sanitation and hygiene.

**Floorbotics**



FloorBotics, a robotic software and hardware development company, develops innovative solutions in response to the market demand for cost-effective alternatives to labor-intensive and non-sterile cleaning methodologies.

Their products include a line of set-and-forget robotic cleaners designed to combat hospital-acquired infections (HAIs).

**Clean Hands Safe Hands**

Clean Hands Safe Hands sought to help healthcare and medical institutions promote health and safety using wireless technology.

The Clean Hands Safe Hands product design uses wireless communication, via Bluetooth, in sanitizing stations. The stations are sensor-activated and provide staff with sanitation reminders. The sensors uniquely identify each employee and record hand hygiene events throughout the health system. As staff members enter or exit rooms, they have a specific amount of time to sanitize before the sensor records the event.

The sensors are connected using Digi XBee® Gateways which incorporate Zigbee mesh

technology. Because the sanitizing stations are connected by a mesh network, they maintain collective data. This helps to avoid unnecessary reminders for professionals who just used another hand sanitizer in a nearby location.

Get to Market Faster and Avoid Costly Mistakes: New FDA Guidance for RF Wireless Medical Devices

**BOS Technology**

BOS Technology found they were in the right place at the right time to provide monitoring solutions for critical environments like hospitals and clinics, but also in areas where individuals congregate, such as schools/colleges, tourism destinations and other sectors. Facilities managers can monitor data from BOS Technology sensors in critical areas, including temperature, humidity, tVOC (volatile organic compounds), ACPH (air changes per hour) and differential pressure.


For example, air supplies and air exchanges through HVAC systems can provide a range of data points that act as health metrics for a building's environment.

“In healthcare settings, construction teams must maintain negative pressure to trap unhealthy particles,” said John Lepak, chief executive officer of BOS Technology. “As the coronavirus pandemic started to spread rapidly, we realized that this technology would also work well in clinical settings. “Frankly, our device – the Differential Pressure Transmitter – is ideal for coronavirus patient rooms because it’s based on the same ideas and goals. In this case, you’re trying to trap pathogens using negative air pressure to prevent the spread of the virus into the hallway and beyond.”

**IoT Wearables: Health Monitoring, Injury Reduction and Contact Tracing**

With advances in Bluetooth technology, and the need for immediacy in feedback, wearables are an enormous growth area for IoT in healthcare. In this section, meet some of the Digi customers who are designing wearables for wellness, ergonomics, contact tracing and patient/doctor connectivity.

**Kinetic Wearables**


Healthcare today is not just about treatment. And its not just about care in hospitals and clinics. There is a growing belief in the adage that "prevention is the best medicine." Business leaders are increasingly promoting healthy habits to improve worker health and safety. These practices, in turn, can save companies enormous amounts annually in lost productivity and worker compensation due to injuries.

The Kinetic company developed a workplace wearable called REFLEX™ to improve worker safety through biofeedback. Designed for businesses whose employees perform workplace tasks that involve bending, twisting, and lifting, REFLEX is an ultra-compact device about the size of a small mobile phone. As the worker bends, the device provides feedback in the form of a light vibration if the employee performs any improper lifting or twisting that could lead to injury. At the end of the shift, the employee and supervisor can review the progress on a dashboard that collects data throughout the day.

**Kinetic Updates for Contact Tracing**

Additional worker safety concerns surfaced with the Covid-19 pandemic, especially in the close environments of industrial workplaces. Reducing worker virus transmission and quickly and accurately identifying potential risks have become key priorities in keeping industrial employees safe and operations open. “When Covid infections among industrial workers began forcing facility shutdowns, we saw a need to better leverage smart technology to connect essential employees and help protect them from illness,” said Bansal.

**LASARRUS Wearables for Physical Therapy**

Patient monitoring is one of the most rapidly growing IoT use cases in healthcare. For the founders of LASARRUS, a company that designed a patient monitoring device to support patients in physical therapy, better insights and improved outcomes for stroke victims were driving factors in development of their flagship WearME product.

LASARRUS (an acronym for Limb Activation Stimulation And Robotic Rehabilitation Unencumbered Services), uses battery-powered sensors configured to capture biometrics such as acoustic cardiography (through a built-in microphone), EKG, temperature, body position, and more. These sensors connect to Digi XBee Zigbee modules configured in a mesh network. "In the era of COVID-19, fewer patients or clinicians want to have in-person patient encounters," said co-founder Nelson Emokpae. "We’re recognizing that the LASARRUS WearME can play an important role in fighting the pandemic from a telehealth perspective. First, patients can wear our device from home and enable the clinician to quickly obtain a complete physiological assessment. That will improve patient outcomes without exposing them to unnecessary risk."

**IoT in Patient Care and Pain Medication Management**

IoT applications in healthcare today solve a range of critical needs. Monitoring and managing medications, ensuring that patient's dose correctly and on schedule are ongoing challenges in clinics, hospitals and care facilities.

An additional challenge is the ability of busy care staff to quickly respond to every patient need.

Avancen developed an IoT solution for healthcare that accurately and quickly dispenses pain medication in a PRN delivery method. PRN, which stands for the Latin "pro re nata," means "as needed," or as circumstances require. The product is called Medication On Demand (MOD®), and is the first patient-controlled analgesic (PCA) device that empowers patients to administer their own PRN oral pain medication.

While Avancen's MOD device puts the control in the hands of the patient, it comes with complete security and clinical control to prevent security breaches or overdoses. The patient waves an RFID wrist band in front of the locked, pre-programmed device to dispense the correct dose of pain medication as part of the prescribed treatment plan.

The technology behind the device is an embedded module in Digi's Embedded Systems line of products. The Digi ConnectCore® line of System-on-Modules is an ideal choice for new designs that require high performance processing, superior security, scalability and rapid time-to-market.

**Medical IoT: 3D Imaging Technology**


IoT technology in healthcare is taking a mjor leap forward faster application processors that can render and deliver medical imaging faster and at higher resolutions. One use case many of us may not be aware of is the need for accurate wound measurement. This is a concern with fresh wounds, in terms of assessing severity, as well as wounds that are progressing through the healing process.

Eykona, a UK-based medical imaging

company, developed a Wound Measurement System to meet this challenge. The system uses cameras and 3D imaging to photograph, measure and map wound progression over time. By observing changes in volume and tissue structure, clinicians can evaluate wounds and the effectiveness of treatment.

The product was built on a Digi ConnectCore® system-on-module (SOM) solution based on NXP application processors. By using a precise measurement system, healthcare providers can ensure they are not only identifying the severity of wounds, but that they administer the right care and treatment based on an accurate assessment.

**Pharmaceutical Temperature Monitoring and Compliance**

As the world learned during the COVID-19 pandemic, maintaining strict temperatures for pharmaceutical drugs and vaccines is imperative. SmartSense by Digi® temperature monitoring solutions for healthcare support this critical need. SmartSense is a division of Digi International that provides complete IoT monitoring solutions for supply chain and logistics, retail food service, grocery store and health and medical applications.

**Healthcare monitoring devices**

IoT devices offer a number of new opportunities for healthcare professionals to monitor patients, as well as for patients to monitor themselves. By extension, the variety of wearable IoT devices provide an array of benefits and challenges, for healthcare providers and their patients alike.

1. Remote patient monitoring

Remote patient monitoring is the most common application of IoT devices for healthcare. IoT devices can automatically collect health metrics like heart rate, blood pressure, temperature, and more from patients who are not physically present in a healthcare facility, eliminating the need for patients to travel to the providers, or for patients to collect it themselves.

When an IoT device collects patient data, it forwards the data to a software application where healthcare professionals and/or patients can view it. Algorithms may be used to analyze the data in order to recommend treatments or generate alerts. For example, an IoT sensor that detects a patient's unusually low heart rate may generate an alert so that healthcare professionals can intervene.

A major challenge with remote patient monitoring devices is ensuring that the highly personal data that these IoT devices collect is secure and private.

2. Glucose monitoring

For the more than 30 million Americans with diabetes, glucose monitoring has traditionally been difficult. Not only is it inconvenient to have to check glucose levels and manually record results, but doing so reports a patient's glucose levels only at the exact time the test is

provided. If levels fluctuate widely, periodic testing may not be sufficient to detect a problem.

IoT devices can help address these challenges by providing continuous, automatic monitoring of glucose levels in patients. Glucose monitoring devices eliminate the need to keep records manually, and they can alert patients when glucose levels are problematic.

Challenges include designing an IoT device for glucose monitoring that:
a. Is small enough to monitor continuously without causing a disruption to patients
b. Does not consume so much electricity that it needs to be recharged frequently.

These are not insurmountable challenges, however, and devices that address them promise to revolutionize the way patients handle glucose monitoring.

3. Heart-rate monitoring

Like glucose, monitoring heart rates can be challenging, even for patients who are present in healthcare facilities. Periodic heart rate checks don't guard against rapid fluctuations in heart rates, and conventional devices for continuous cardiac monitoring used in hospitals require patients to be attached to wired machines constantly, impairing their mobility.

Today, a variety of small IoT devices are available for heart rate monitoring, freeing patients to move around as they like while ensuring that their hearts are monitored continuously. Guaranteeing ultra-accurate results remains somewhat of a challenge, but most modern devices can deliver accuracy rates of about 90 percent or better.

4. Hand hygiene monitoring

Traditionally, there hasn't been a good way to ensure that providers and patients inside a healthcare facility washed their hands properly in order to minimize the risk of spreading contagion.

Today, many hospitals and other health care operations use IoT devices to remind people to sanitize their hands when they enter hospital rooms. The devices can even give instructions on how best to sanitize to mitigate a particular risk for a particular patient.

A major shortcoming is that these devices can only remind people to clean their hands; they can't do it for them. Still, research suggests that these devices can reduce infection rates by more than 60 percent in hospitals.

5. Depression and mood monitoring

Information about depression symptoms and patients' general mood is another type of data that has traditionally been difficult to collect continuously. Healthcare providers might periodically ask patients how they are feeling, but were unable to anticipate sudden mood swings. And, often, patients don't accurately report their feelings.

"Mood-aware" IoT devices can address these challenges. By collecting and analyzing data such

as heart rate and blood pressure, devices can infer information about a patient's mental state. Advanced IoT devices for mood monitoring can even track data such as the movement of a patient's eyes.

The key challenge here is that metrics like these can't predict depression symptoms or other causes for concern with complete accuracy. But neither can a traditional in-person mental assessment.

6. Parkinson's disease monitoring

In order to treat Parkinson's patients most effectively, healthcare providers must be able to assess how the severity of their symptoms fluctuate through the day.

IoT sensors promise to make this task much easier by continuously collecting data about Parkinson's symptoms. At the same time, the devices give patients the freedom to go about their lives in their own homes, instead of having to spend extended periods in a hospital for observation.

Other examples of IoT/IoMT

While wearable devices like those described above remain the most commonly used type of IoT device in healthcare, there are devices that go beyond monitoring to actually providing treatment, or even "living" in or on the patient. Examples include the following.

7. Connected inhalers

Conditions such as asthma or COPD often involve attacks that come on suddenly, with little warning. IoT-connected inhalers can help patients by monitoring the frequency of attacks, as well as collecting data from the environment to help healthcare providers understand what triggered an attack.

In addition, connected inhalers can alert patients when they leave inhalers at home, placing them at risk of suffering an attack without their inhaler present, or when they use the inhaler improperly.

8. Ingestible sensors

Collecting data from inside the human body is typically a messy and highly disruptive affair. No no enjoys having a camera or probe stuck into their digestive tract, for example.

With ingestible sensors, it's possible to collect information from digestive and other systems in a much less invasive way. They provide insights into stomach PH levels, for instance, or help pinpoint the source of internal bleeding.

These devices must be small enough to be swallowed easily. They must also be able to dissolve or pass through the human body cleanly on their own. Several companies are hard at work on ingestible sensors that meet these criteria.

9. Connected contact lenses

Smart contact lenses provide another opportunity for collecting healthcare data in a passive, non-intrusive way. They could also, incidentally, include micro cameras that allow wearers effectively to take pictures with their eyes, which is probably why companies like Google have patented connected contact lenses.

Whether they're used to improve health outcomes or for other purposes, smart lenses promise to turn human eyes into a powerful tool for digital interactions.

10. Robotic surgery

By deploying small Internet-connected robots inside the human body, surgeons can perform complex procedures that would be difficult to manage using human hands. At the same time, robotic surgeries performed by small IoT devices can reduce the size of incisions required to perform surgery, leading to a less invasive process, and faster healing for patients.

These devices must be small enough and reliable enough to perform surgeries with minimal disruption. They must also be able to interpret complex conditions inside bodies in order to make the right decisions about how to proceed during a surgery. But IoT robots are already being used for surgery, showing that these challenges can be adequately addressed.