UNIT-I

Introduction to Internet of Things (IoT) |

- **What Is IoT:**

IoT stands for Internet of Things. It refers to the interconnectedness of physical devices, such as appliances and vehicles, that are embedded with software, sensors, and connectivity which enables these objects to connect and exchange data. This technology allows for the collection and sharing of data from a vast network of devices, creating opportunities for more efficient and automated systems.

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

IoT is network of interconnected computing devices which are embedded in everyday objects, enabling them to send and receive data.

Over 9 billion 'Things' (physical objects) are currently connected to the Internet, as of now. In the near future, this number is expected to rise to a whopping 20 billion.

Main components used in IoT:

- **Low-power embedded systems:** Less battery consumption, high performance are the inverse factors that play a significant role during the design of electronic systems.
- **Sensors:** Sensors are the major part of any IoT application. It is a physical device that measures and detects certain physical quantities and converts it into signal which can be provided as an input to processing or control unit for analysis purpose.

Different types of Sensors:

1. Temperature Sensors
2. Image Sensors
3. Gyro Sensors
4. Obstacle Sensors
5. RF Sensor
6. IR Sensor
7. MQ-02/05 Gas Sensor
8. LDR Sensor
9. Ultrasonic Distance Sensor

- **Control Units:** It is a unit of small computer on a single integrated circuit containing microprocessor or processing core, memory and programmable input/output devices/peripherals. It is responsible for major processing work of IoT devices and all logical operations are carried out here.
- **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.

- **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.
- **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.

There are two ways of building IoT:

1. Form a separate internet work including only physical objects.

2. Make the Internet ever more expansive, but this requires hard-core technologies such as rigorous cloud computing and rapid big data storage (expensive).

In the near future, IoT will become broader and more complex in terms of scope. It will change the world in terms of

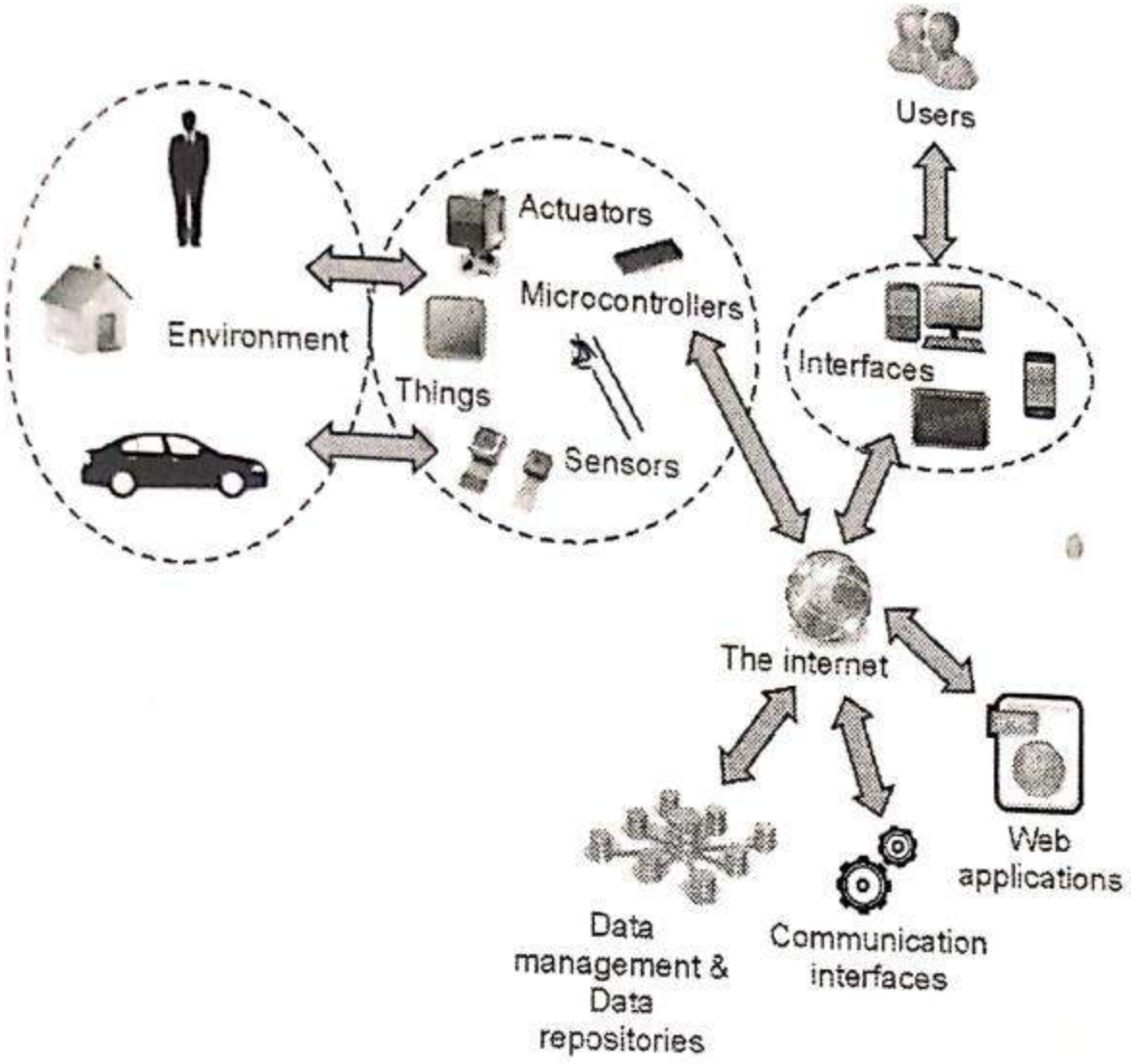
"anytime, anyplace, anything in connectivity."

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 tiny devices with dimensions usually less than a hundred nanometers.
- **Smart networks:** (ex: mesh topology).

Working with IoT Devices:

- *Collect and Transmit Data :* For this purpose sensors are widely used they are used as per requirements in different application areas.
- *Actuate device based on triggers produced by sensors or processing devices:* If certain conditions are satisfied or according to user's requirements if certain trigger is activated then which action to perform that is shown by Actuator devices.
- *Receive Information:* From network devices, users or devices can take certain information also for their analysis and processing purposes.
- *Communication Assistance:* Communication assistance is the phenomenon of communication between 2 networks or communication between 2 or more IoT devices of same or different networks. This can be achieved by different communication protocols like: MQTT, Constrained Application Protocol, ZigBee, FTP, HTTP etc.



Working of IoT

Characteristics of IoT:

- Massively scalable and efficient
- IP-based addressing will no longer be suitable in the upcoming future.
- An abundance of physical objects is present that do not use IP, so IoT is made possible.
- Devices typically consume less power. When not in use, they should be automatically programmed to sleep.

- A device that is connected to another device right now may not be connected in another instant of time.
- Intermittent connectivity – IoT devices aren't always connected. In order to save bandwidth and battery consumption, devices will be powered off periodically when not in use. Otherwise, connections might turn unreliable and thus prove to be inefficient.
- **Desired Quality of any IoT Application:**
- **Interconnectivity**

It is the basic first requirement in any IoT infrastructure. Connectivity should be guaranteed from any devices on any network then only devices in a network can communicate with each other.

- **Heterogeneity**

There can be diversity in IoT enabled devices like different hardware and software configuration or different network topologies or connections, but they should connect and interact with each other despite so much heterogeneity.

- **Dynamic in nature**

IoT devices should dynamically adapt themselves to the changing surroundings like different situations and different prefaces.

- **Self-adapting and self configuring technology**

For example, surveillance camera. It should be flexible to work in different weather conditions and different light situations (morning, afternoon, or night).

- **Intelligence**

Just data collection is not enough in IoT, extraction of knowledge from the generated data is very important. For example, sensors generate data, but that data will only be useful if it is interpreted properly. So intelligence is one of the key characteristics in IoT. Because data interpretation is the major part in any IoT application because without data processing we can't make any insights from data. Hence, big data is also one of the most enabling technologies in IoT field.

- Scalability

The number of elements (devices) connected to IoT zones is increasing day by day. Therefore, an IoT setup should be capable of handling the expansion. It can be either expand capability in terms of processing power, storage, etc. as vertical scaling or horizontal scaling by multiplying with easy cloning.

- Identity

Each IoT device has a unique identity (e.g., an IP address). This identity is helpful in communication, tracking and to know status of the things. If there is no identification then it will directly affect security and safety of any system because without discrimination we can't identify with whom one network is connected or with whom we have to communicate. So there should be clear and appropriate discrimination technology available between IoT networks and devices.

- Safety

Sensitive personal details of a user might be compromised when the devices are connected to the Internet. So data security is a major challenge. This could cause a loss to the user. Equipment in the huge IoT network may also be at risk. Therefore, equipment safety is also critical.

- **Architecture**

It should be hybrid, supporting different manufacturer's products to function in the IoT network.

As a quick note, IoT incorporates trillions of sensors, billions of smart systems, and millions of applications.

Application Domains: IoT is currently found in four different popular domains:

1) Manufacturing/Industrial business - 40.2%
2) Healthcare - 30.3%
3) Security - 7.7%
4) Retail - 8.3%

Modern Applications:

1. Smart Grids and energy saving
2. Smart cities
3. Smart homes/Home automation
4. Healthcare
5. Earthquake detection
6. Radiation detection/hazardous gas detection
7. Smartphone detection
8. Water flow monitoring
9. Traffic monitoring
10. Wearables
11. Smart door lock protection system
12. Robots and Drones
13. Healthcare and Hospitals, Telemedicine applications
14. Security
15. Biochip Transponders (For animals in farms)
16. Heart monitoring implants (Example Pacemaker, ECG real time tracking)

Advantages of IoT:

1. Improved efficiency and automation of tasks.
2. Increased convenience and accessibility of information.
3. Better monitoring and control of devices and systems.
4. Greater ability to gather and analyze data.
5. Improved decision-making.
6. Cost savings.

Disadvantages of IoT:

1. Security concerns and potential for hacking or data breaches.
2. Privacy issues related to the collection and use of personal data.
3. Dependence on technology and potential for system failures.
4. Limited standardization and interoperability among devices.
5. Complexity and increased maintenance requirements.
6. High initial investment costs.
7. Limited battery life on some devices.
8. Concerns about job displacement due to automation.
9. Limited regulation and legal framework for IoT, which can lead to confusion and uncertainty.

physical design knowledge is crucial for selecting suitable devices and sensors, ensuring seamless integration, and optimizing connectivity options in IoT systems. It enables power efficient strategies, facilitates edge computing, and enhances reliability and resilience through

redundancy and failover mechanisms. This knowledge ensures robust, efficient, and reliable IoT ecosystems.

In this article, we will discuss more about physical design of IoT. Let's start.

Physical Design of Internet of Things (IOT)

Posted on9 April 2021 ByYASH PAL 1 Commenton Physical Design of Internet of Things (IOT)

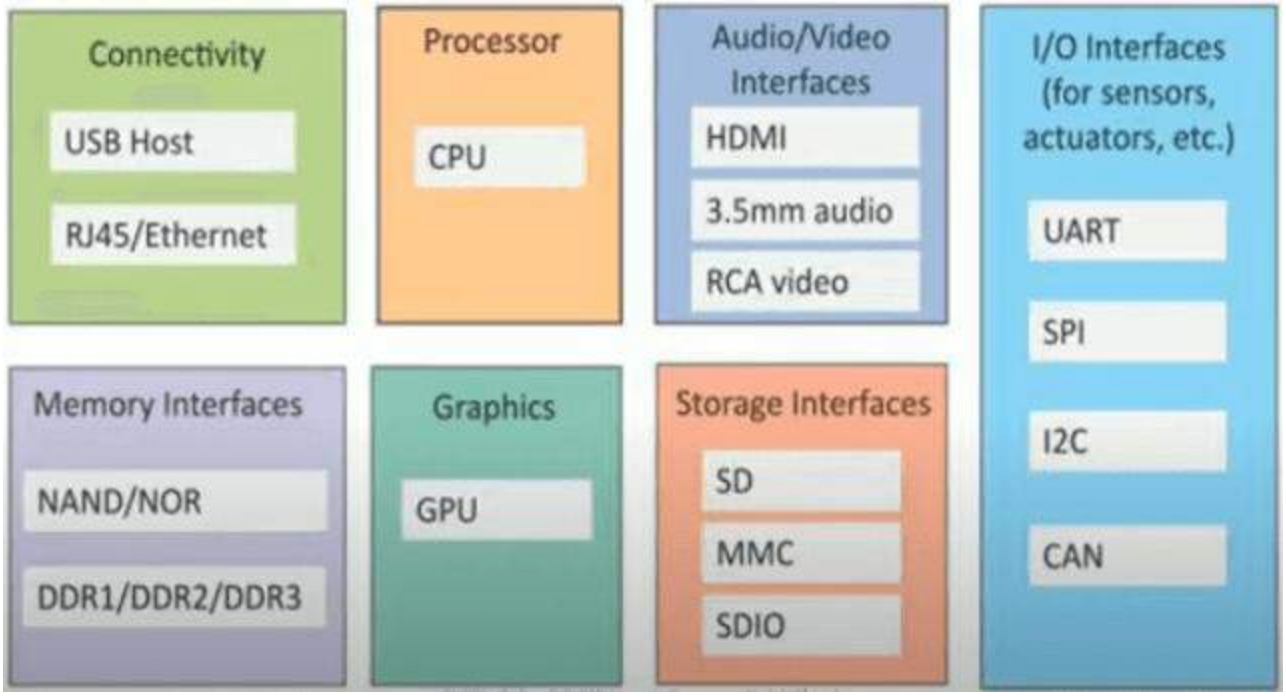
The **physical design** of an **IoT system is referred to as the Things/Devices and protocols that are used to build an IoT system. All these things/Devices are called Node Devices and every device has a unique identity that performs remote sensing, actuating,** and monitoring work. and the protocols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

Physical Design of IoT

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 is then determined by algorithms.



devices in IoT(Internet of things)

Connectivity

Devices like USB hosts and ETHERNET are used for connectivity between the devices and the 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 give input and output signals to sensors, and actuators we use things like UART, SPI, CAN, etc.

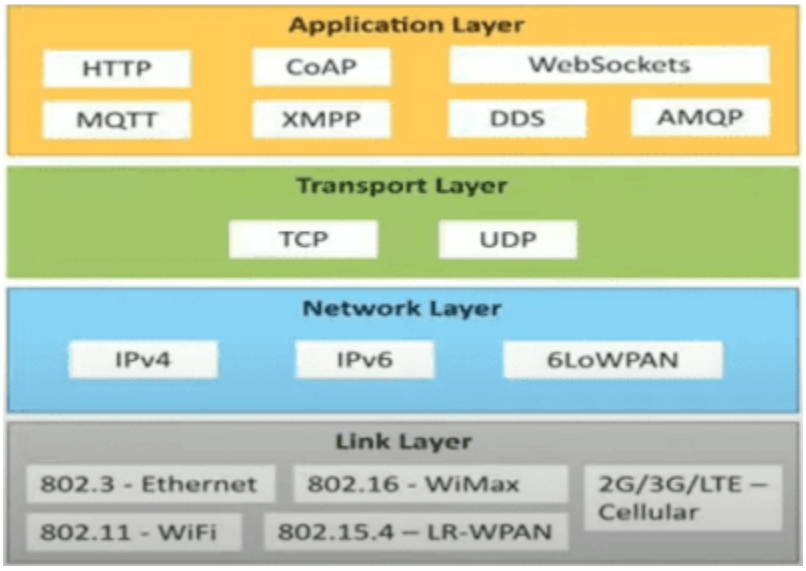
Storage Interfaces

Things like SD, MMC, and SDIO are used to store the data generated from an IoT device.

Other things like DDR and 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 a 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 are present on both the server and client side and these protocols are managed by network layers like application, transport, network, and link layer.



IoT(Internet of Things) protocols

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 include HTTP, WebSocket, XMPP, MQTT, DDS, and AMQP protocols.

HTTP

Hypertext transfer protocol is a protocol that presents 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 the 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

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.

Transport Layer

This layer is used to control the flow of data segments and handle 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 part of an internet protocol called the connectionless protocol. this protocol is 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 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 to 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 long-anticipated problems.

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.

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.

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.

Overview of Physical Design

Devices and Sensors

Types of IoT devices and their functionalities:

IoT devices encompass a wide range of physical objects that are connected to the internet and communicate with each other. Some common types of IoT devices include:

- **Wearable Devices**: These include smartwatches, fitness trackers, and health monitoring devices. They collect data about an individual's activity, health, and location.
- **Smart Home Devices:** These devices automate and control various aspects of a home, such as lighting, security systems, thermostats, and appliances. They enable remote monitoring and control through internet connectivity.
- **Industrial IoT Devices**: These devices are used in industrial settings to monitor and control machinery, optimize processes, and improve operational efficiency. Examples include sensors in factories, logistics tracking systems, and remote monitoring equipment.
- **Smart Appliances**: These are traditional home appliances, such as refrigerators, washing machines, and ovens, enhanced with IoT capabilities. They can be controlled remotely, gather usage data, and offer features like predictive maintenance.
- **Connected Vehicles:** IoT devices in vehicles enable features like real−time GPS tracking, remote diagnostics, and vehicle−to−vehicle communication for enhanced safety and efficiency.

The functionalities of IoT devices vary depending on their intended use, but common features include data collection, remote control and monitoring, automation, and connectivity to other devices or cloud platforms.

Sensor technologies used in IoT

Sensors play a crucial role in IoT by collecting data from the physical environment and converting it into digital information. Different sensor technologies are employed in IoT devices, including:

- Temperature Sensors: Measure and monitor temperature variations.
- Humidity Sensors: Detect and measure humidity levels in the environment.
- Proximity Sensors: Detect the presence or absence of objects within a certain range.
- Motion Sensors: Detect motion or movement in their surroundings.
- Light Sensors: Measure light intensity or detect changes in light levels.
- Pressure Sensors: Measure pressure variations in gases or liquids.
- Accelerometers: Detect and measure acceleration, tilt, and vibration.
- GPS (Global Positioning System) Sensors: Provide location information using satellite signals.

These sensors enable IoT devices to collect real−time data, monitor the environment, and respond to specific conditions or triggers.

Examples of IoT devices in different industries

IoT devices have applications in various industries, offering numerous benefits. Here are a few examples:

- Healthcare: Wearable devices and medical sensors enable remote patient monitoring, continuous health tracking, and early detection of health issues. They can help improve patient outcomes and reduce healthcare costs.
- Agriculture: IoT devices are used for precision farming, monitoring soil moisture levels, temperature, and weather conditions. They optimize irrigation, automate pest control, and enhance crop yield.
- Manufacturing: IoT−enabled sensors and devices are used for predictive maintenance, real−time monitoring of equipment, inventory management, and supply chain optimization. They improve operational efficiency and reduce downtime.
- Transportation and Logistics: Connected vehicles, tracking devices, and smart logistics solutions enable real−time tracking of shipments, route optimization, and efficient fleet management. They enhance supply chain visibility and reduce transportation costs.
- Energy Management: IoT devices monitor energy consumption, optimize energy usage, and enable remote control of devices to save energy and reduce costs in buildings and homes.

Connectivity

Communication protocols for IoT

Communication protocols are essential for IoT devices to exchange data and information. Some commonly used protocols in IoT include:

- MQTT (Message Queuing Telemetry Transport): A lightweight protocol designed for efficient communication in constrained networks, suitable for low−power devices and unreliable connections.
- HTTP (Hypertext Transfer Protocol): A standard protocol used for communication between web browsers and servers, also employed in IoT for web−based interactions and data transfer.
- CoAP (Constrained Application Protocol): Designed for resource−constrained devices, CoAP enables efficient communication and is often used in IoT applications that require low power and low bandwidth.
- AMQP (Advanced Message Queuing Protocol): A protocol for reliable messaging between devices and applications, capable of supporting complex messaging scenarios.
- WebSocket: A communication protocol that enables full−duplex communication over a single, long−lived connection, facilitating real−time data transfer between IoT devices and servers.

Wireless connectivity options

Wireless connectivity is a key aspect of IoT, providing flexibility and mobility. Some common wireless connectivity options used in IoT devices include:

- Wi−Fi: A widely used wireless networking technology that enables high−speed data transfer over short to medium distances. It is suitable for applications with power availability and the need for high bandwidth.
- Bluetooth: A short−range wireless technology used for connecting devices in proximity. It is commonly used for IoT devices that require low power consumption and intermittent data transfer, such as wearable devices and home automation systems.
- Zigbee: A low−power, low−data−rate wireless communication protocol designed for applications with low power consumption requirements and a large number of devices. It is commonly used in home automation, smart lighting, and industrial applications.
- LPWAN (Low−Power Wide Area Network): LPWAN technologies, such as LoRaWAN and NB−IoT, offer long−range connectivity with low power consumption, making them suitable for IoT applications that require wide area coverage, such as smart city deployments and agricultural monitoring.

Wired connectivity options

While wireless connectivity is prevalent in IoT, there are also cases where wired connectivity is preferred for its reliability and stability. Some common wired connectivity options include:

- Ethernet: A standard wired networking technology that provides reliable and highspeed data transfer over local area networks (LANs). Ethernet is commonly used in industrial settings and for devices requiring high bandwidth and low latency.
- Powerline Communication: This technology allows data transmission over existing power lines, eliminating the need for additional wiring. Powerline communication is often used in home automation systems and smart meters.

Power and Energy Management

Power requirements of IoT devices can vary based on factors such as their functionality, processing capabilities, and communication needs. IoT devices typically fall into two categories:

- Battery−powered devices: These devices operate on limited battery power and must be designed to optimize energy consumption to extend battery life. They often employ low−power components, sleep modes, and efficient power management techniques.
- Line−powered devices: Devices that are connected to a power source have more flexibility in terms of power requirements. However, energy efficiency is still a consideration to minimize operating costs and environmental impact.

Battery life and energy−efficient designs

Extending battery life is crucial for many IoT devices to ensure uninterrupted operation and minimize maintenance. Energy−efficient designs for IoT devices may include:

- Low−power components: Using low−power microcontrollers, sensors, and wireless modules helps reduce power consumption.
- Sleep modes: Devices can be programmed to enter sleep or idle modes when not actively performing tasks, conserving energy.
- Optimized data transmission: Transmitting data in a compressed or aggregated format reduces the amount of data transferred, saving power.

Conclusion

In conclusion, understanding the physical design of IoT is essential for successful implementation and utilization of IoT systems. This understanding encompasses various aspects, including devices and sensors, connectivity options, power and energy management, and edge computing. By comprehending these elements, organizations and individuals can make informed decisions and optimize their IoT solutions for different industries and use cases.

The physical design of IoT involves a diverse range of devices and sensors, each serving specific functionalities and enabling data collection, remote control, automation, and connectivity. Sensor technologies play a crucial role in collecting real−time data from the environment, while IoT devices facilitate data exchange and communication.

IOT Functional Blocks

The Internet of Things Functional Blocks is the foundation of the IoT ecosystem. Companies are fast discovering ways to leverage the Internet of Things (IoT) to improve their efficiency as it grows in popularity. While the Internet of Things has numerous advantages, businesses are trying to comprehend how to incorporate technology into their work and daily lives. This article will look at that the IoT functional blocks and how they interact to produce a functional IoT system**.**

The Internet of Things (IoT) is a rapidly growing industry, with more and more devices becoming connected on a regular basis. The Internet of Things devices range from small sensors to huge machines and systems, but they all have one thing in common: they generate data. IoT systems use functional blocks to make sense of this data and extract value from it. These blocks are distinct components of the IoT system that carry out specialized functions.

IoT systems are composed of a number of building blocks, including sensors/actuators, connectivity, security, services, etc. The functional blocks are responsible for sensing, verification, actuation, management, and communication.

These functional blocks are made up of devices that handle interactions between a web server and the client, enable controls and monitoring functions, manage data transfer, secure the IoT system through authentication and various purposes, and offer an interface for monitoring and managing various concepts. Let’s gather more information about the IoT Functional Blocks.

Sensor/Actuator block

The sensor/actuator block serves as the data entry point in an IoT system. Sensors collect data from their surroundings, whereas actuators drive physical processes. Sensors gather data on temperature, humidity, light, motion, and other variables, whereas actuators turn on lights, open doors, and control machines. These gadgets work together to collect data and operate in the physical world.

Connectivity Block

Once the sensor/actuator block has collected data, it must be sent to the remainder of the system. This is when the connection block enters the picture. The connectivity block is in charge of creating and managing communication channels amongst IoT system devices. This can be accomplished with the use of several technologies such as Wi-Fi, Bluetooth, ZigBee, and cellular networks.

Data Processing Block

The obtained data is examined and processed in the data processing block. This block is in charge of filtering out noise and irrelevant data, converting the data into an easily studied format, and recognizing patterns and anomalies in the data. This block can also execute real-time analysis, enabling speedy data-driven decisions.

Application Block

The application block is the component of the IoT system that gives value to the end user. This block is in charge of utilizing the processed data to provide a specified function or service. An application block, for example, could be used to provide insights into energy usage in a building or to adjust the temperature in a greenhouse.

Security Block

The security block is in charge of assuring the IoT system's security and protection against illegal access. This block is in charge of authentication and authorization, as well as data encryption during transmission and storage. It also handles intrusion detection and response, assisting in the prevention and mitigation of threats.

Management Block

The management block is in charge of overseeing the overall operation of the IoT system. This block is capable of handling device configuration, firmware updates, and system monitoring. It

can also give analytics and reporting, allowing system administrators to understand how the system is operating and find areas for improvement.

Advantages of IoT Functional Blocks

IoT Functional Blocks provide various advantages to enterprises and people interested in implementing IoT solutions.

These advantages include −

Scalability

IoT Scalability is built into Functional Blocks, allowing enterprises to add new devices and services to their IoT system as needed. The capacity to scale assures that an IoT system can grow and react to changing business needs and future technology.

Interoperability

IoT Functional Blocks is a standardized architecture for developing IoT solutions. This standardization means that devices and services from various suppliers may function seamlessly together, enhancing interoperability and lowering integration costs.

Modularity

Because IoT Functional Blocks are modular, they may be swapped, modified, or added as needed. Because of this versatility, enterprises can select the optimal components for their IoT system and easily swap out components as needed.

Flexibility

IoT Functional Blocks offer a variety of deployment choices. Depending on their needs, businesses can implement an IoT system on-premises, in the cloud, or in a hybrid approach.

Conclusion

The IoT functional blocks collaborate to build a strong and functional IoT system. It is easier to comprehend how the system works and how different components interact with one another when it is broken down into discrete components. This might be useful for troubleshooting and diagnosing system difficulties. Furthermore, the usage of functional blocks in IoT systems

provides for increased modularity and flexibility, making it easier to add new components or upgrade current ones.

As the Internet of Things expands, it is critical to comprehend the role that each functional block plays in the larger system. Understanding the functioning and purpose of each piece makes designing, building, and maintaining IoT systems that fulfill the demands of businesses and consumers alike easier.

Types of IoT Networks

An Internet of Things (IoT) network is a group of hardware, including sensors, gadgets, appliances, and software, that interact with one another and share data and information without the need for human interaction. Businesses may now gather new insights from devices through IoT networks thanks to cloud and edge computing capabilities. Organizations may now monitor environmental, geospatial, and atmospheric variables in real time because of this bridging of the digital and physical worlds. Businesses can quickly respond to environmental changes when combined with automation, resulting in less downtime, more significant insights, and increased productivity.

How does an IoT Network work?

IoT Sensors

Small, low-cost sensors are used by IoT networks to gather data about the surroundings. For instance, farmers employ IoT sensors to track moisture levels, while industrial facilities utilize the same sensors to track pipe pressure. IoT sensors provide a wide range of configuration options and can track hundreds of distinct changes.

IoT Connectivity

IoT sensors feed information back to the cloud or an edge computing device constantly for processing. Instead of sending vast data streams, IoT devices usually utilize less power and provide smaller quantities of data. Since edge computing reduces the distance between the sensor and the server, it is frequently chosen by businesses that need the lowest latency and quickest reaction time. Businesses may select from a variety of IoT networks depending on the technology and use case to achieve their objectives. WiFi or cellular connections are the two methods through which sensors often convey their data. IoT Processing

The software then analyses and stores the data in the cloud or on an edge server after it has been collected. Several systems employ artificial intelligence and machine learning to take action when particular data is transmitted from a sensor. Businesses combine automation and IoT networks to coordinate device management in a low-cost, predictable, and scalable way. Enterprises are able to monitor anything from machine maintenance to the weather outdoors thanks to IoT management solutions' ability to handle data from diverse platforms.

Types of IoT Networks

Cellular

IoT devices may interact using cellular networks, the same mobile networks used by smartphones. These networks weren't always thought to be the most excellent option for IoT devices because they were first created for power-hungry gadgets like smartphones. Later, the cellular sector created new technologies that were more suited for IoT use cases. In the modern day, this kind of wireless network is widely used and regarded as a dependable and secure form of IoT communication. The majority of the U.S. has access to cell service, and this kind of network has an extensive coverage area. The locations where monitoring sensors are most necessary, such as within utility closets, elevator shafts, basements, etc., frequently lack cell coverage.

WiFi

WiFi is a standard option for IoT networks since many companies already have WiFi coverage across their infrastructure. For stationary IoT sensors that must communicate data over a medium distance, WiFi is a reliable solution. To assist and improve the dependability of their sensors, WiFi administrators could segment IoT sensors on a distinct subnet and apply quality of service. WiFi IoT networks do have certain disadvantages, though. WiFi networks don't have as much coverage as done by cellular networks because of their power restrictions. Mobile IoT sensors may have connection problems on WiFi networks since WiFi networks don't handle device handover as efficiently as cellular networks do.

Local and Personal Area Networks (LAN/PAN)

Personal area networks (PAN) and local area networks are networks that only span relatively limited distances (LAN). Although data transport via PAN and LAN networks is often thought to be cost-effective, it is not always dependable. WiFi and Bluetooth are two wireless personal and local area network technologies that are often used in IoT connectivity solutions. When

numerous access points are included in a more extensive network, WiFi may be utilized for dispersed applications in addition to local ones. A single battery powered by Bluetooth Low Energy (BLE) might last up to five years if the device is not continually receiving data. BLE is a more energy-efficient wireless network protocol.

Low Power Wide Area Networks (LPWAN)

IoT devices that use LPWANs transmit little data packets, rarely over great distances. This kind of wireless network was created in response to the early difficulties with cellular communication. LPWAN is marketed as having a more excellent range than WiFi and Bluetooth while consuming less power than cellular. LoRaWAN, which operates on the LoRa (long-range) communication network, is a well-known and widely used IoT network protocol in this category. LoRaWAN has benefits for IoT devices, including reduced power consumption (for longer battery life) and relatively affordable chipsets. A single base station or gateway operating on a long-range network is capable of delivering service to a very vast area-a few kilometers in congested metropolitan areas-under the right circumstances.

Mesh Networks

The connection configuration of mesh networks-how the parts communicate with one another—is the most effective way to characterize them. In mesh networks, all sensor nodes work together to share data among themselves so that it may reach the gateway. One illustration of an IoT wireless network technology is Zigbee. Mesh networks have a relatively limited range, so you might need to add more sensors throughout a building or utilize repeaters to achieve the coverage you need for your application. Additionally, the nature of how these networks interact can lead to excessive power consumption, particularly if you want fast communications, as in the case of an application for intelligent lighting. Mesh networks are a standard option since they are also very resilient, adept at locating the data transmission pathways that are both quick and reliable, and simple to set up.

Conclusion

The five types of IoT networks in this article are an absolute fit for most businesses looking for solutions to their problems in the Internet of Things. These networks are a combination of wired and wireless networks for IoT-connected devices.

IoT Communication Protocols

The prime focus of the internet of things is to offer communication between various objects that are not traditional computers. IoT allows these objects to send and receive data over a network. There are various protocols in IOT established to offer this communication. The most popular communication protocols offered by IOT are described in detail below. Let's start!!

IoT Communication Protocols

1. Bluetooth

Standard communication protocols help in unleashing the full potential of the internet of things. More than one third of Iot devices contain bluetooth connectivity. Bluetooth is a form of wireless technology used for device communication and to make personal area networks(PANs). The latest version is the bluetooth5 and it has features such as high range, speed and data broadcasting.

Bluetooth connects to devices with smaller distances and changes the device interaction. It has some empowering features that support IoT devices. Bluetooth low energy(BLE) supports devices that require less power and ideal for IoT enabled projects.

2. Zigbee

ZIgbee is a WLAN and a wireless technology that aims to support extremely low power devices. It supports these kinds of devices and makes it possible to connect them to the internet. It is an open global standard and works on IEEE 802.15.4 physical radio standards.

IoT devices do not require extra functionality and Zigbee is an ideal protocol for transferring data from one communication point to another. Zigbee makes data flow easy. It is used to send small amounts of data using very less power which is why it is used in machine to machine communication(M2M) and IoT.

3. Bluetooth Low Energy

BLE is a bluetooth that uses less power. It is designed to support the internet of things. BLE is energy efficient and offers better connectivity compared to other forms of technology such as Zigbee or LoRa. BLU fits the need of data transfers as they are the only function that takes place

in Iot sensors. BLE is used in the making of smartwatches, medical devices, fitness trackers, beacons and home automation devices. BLE consumes less power and has less bandwidth.

4. Wifi

Wifi is a form of local area network for wireless communication. It is a better option for data transfers as it easily fits into a variety of standards. It plays an important role in IoT communication and intercommunication with other cellular networks such as bluetooth. Wifi supports high bandwidth and low latency.

5. Z-Wave

Z-wave is a wireless messaging protocol to communicate between various IoT devices. It is useful especially in home automation to connect appliances in smart homes. Z-wave offers a two-way mode of communication empowered with mesh networking and message received acknowledgment.

Z-wave is a low-cost technology that eliminates issues caused by Wifi and Bluetooth. The network of Z-wave includes the internet of things devices and control called the primary hub. When z-hub receives a message via a smartphone or tablet it sends this message to the relevant smart home appliance.

RFID(radio frequency identification)

Radio frequency identification system is a technology that supports the identification of objects via radio waves. By connecting the RFID reader to the terminal of the internet, users can identify, monitor and track the object with tags. It was first used in World War 2.

RFID is fast, dependent and does not require physical interaction between the user and the tagged item. It identifies IoT objects by tagging and labelling them. The tag or label replaces the object. RFID is useful to identify, track and monitor remote IoT objects with time and location.

6. Cellular

Cellular connects the devices to anything and everything without the need for smartphones or gateway. This means it connects devices directly to the base station without any intermediaries. These connections are always present even in remote areas. Cellular IoT makes it possible to construct less power-consuming devices that connect to the internet which was not possible

before. You can easily send small packs of data through a cellular network. 5G is the latest cellular technology that is taking over Iot devices with rapid speed.

7. Sigfox

Sigfox was the first to introduce LPWAN technologies in the development of IoT projects. It uses a low-power wide area network to intercommunicate between IoT devices via the internet. It supports long-distance communication for sending and receiving small messages.

8. Ethernet

Ethernet is a communication standard that was developed in the early 80s to network between local devices and computers. The local environment is labeled as a local area network(LAN). LAN creates a common environment for devices to receive and share information among one another. Ethernet however offers a wired form of communication. Since it does not offer wireless communication the set becomes a bit costly and is not the ideal option for IoT communication.

9. NFC

NFC stands for near-field communication. It is a wireless technology for short-distance communication. However, the NFC-enabled devices must be in close proximity to each other so that they can communicate via radio waves. One of the devices should be an active device such as a smartphone or tablet and the other device can be passive such as an NFC tag. The active devices require an external power supply while the passive devices do not.

10. LPWAN

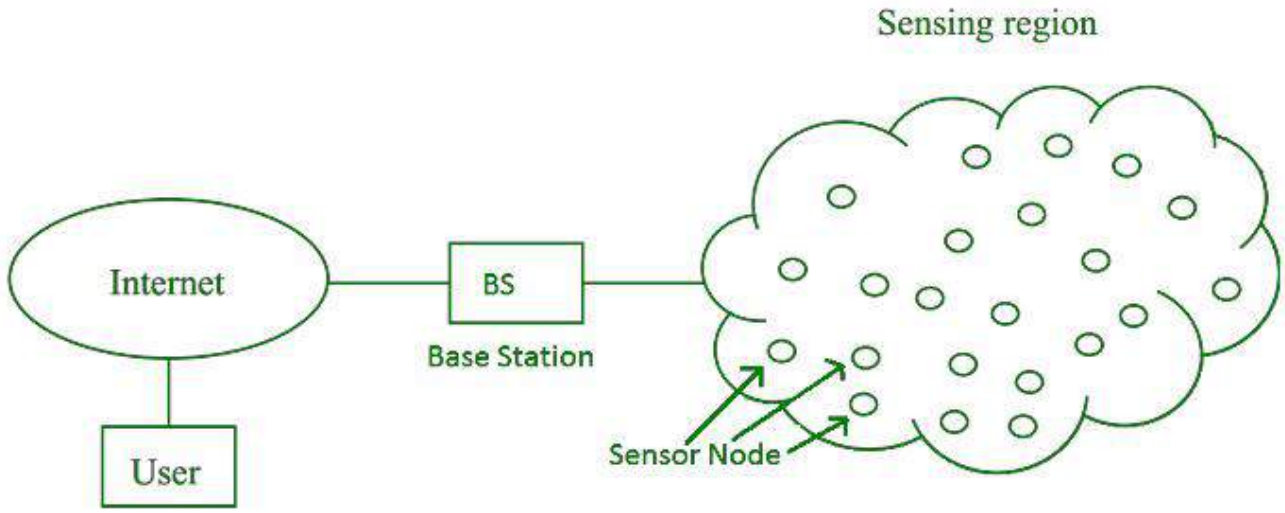
LPWAN stands for low power wide area network. It offers wireless communication between devices that consume less amounts of power. It connects these devices to the internet to send and receive messages from devices within the same network. Some examples of LPWAN are sigfox and LoRa.

11. LoRaWAN

Low range wide area network or LoRaWAN is a wide area network protocol. It was constructed to connect objects to the internet and to act as a mode of communication between these objects. These objects could be home automation devices, smart cars, thermostats and so on.

Wireless Sensor Network (WSN)

- **Wireless Sensor Network (WSN)** is an infrastructure-less wireless network that is deployed in a large number of wireless sensors in an ad-hoc manner that is used to monitor the system, physical or environmental conditions.
  Sensor nodes are used in WSN with the onboard processor that manages and monitors the environment in a particular area. They are connected to the Base Station which acts as a processing unit in the WSN System.
  Base Station in a WSN System is connected through the Internet to share data.



WSN can be used for processing, analysis, storage, and mining of the data.

Applications of WSN:

1. Internet of Things (IoT)
2. Surveillance and Monitoring for security, threat detection
3. Environmental temperature, humidity, and air pressure
4. Noise Level of the surrounding
5. Medical applications like patient monitoring
6. Agriculture
7. Landslide Detection

Challenges of WSN:

1. Quality of Service
2. Security Issue
3. Energy Efficiency
4. Network Throughput
5. Performance
6. Ability to cope with node failure
7. Cross layer optimization
8. Scalability to large scale of deployment

A modern Wireless Sensor Network (WSN) faces several challenges, including:

- **Limited power and energy:** WSNs are typically composed of battery-powered sensors that have limited energy resources. This makes it challenging to ensure that the network can function for
  long periods of time without the need for frequent battery replacements.
- **Limited processing and storage capabilities:** Sensor nodes in a WSN are typically small and have limited processing and storage capabilities. This makes it difficult to perform complex tasks or store large amounts of data.
- **Heterogeneity:** WSNs often consist of a variety of different sensor types and nodes with different capabilities. This makes it challenging to ensure that the network can function effectively and
  efficiently.
- **Security:** WSNs are vulnerable to various types of attacks, such as eavesdropping, jamming, and spoofing. Ensuring the security of the network and the data it collects is a major challenge.
- **Scalability:** WSNs often need to be able to support a large number of sensor nodes and handle large amounts of data. Ensuring that the network can scale to meet these demands is a significant
  challenge.
- **Interference:** WSNs are often deployed in environments where there is a lot of interference from other wireless devices. This can make it difficult to ensure reliable communication between sensor nodes.
- **Reliability:** WSNs are often used in critical applications, such as monitoring the environment or controlling industrial processes. Ensuring that the network is reliable and able to function correctly
  in all conditions is a major challenge.

Components of WSN:

1. **Sensors:**
   Sensors in WSN are used to capture the environmental variables and which is used for data acquisition. Sensor signals are converted into electrical signals.
2. **Radio Nodes:**
   It is used to receive the data produced by the Sensors and sends it to the WLAN access point. It consists of a microcontroller, transceiver, external memory, and power source.
3. **WLAN Access Point:**
   It receives the data which is sent by the Radio nodes wirelessly, generally through the internet.
4. **Evaluation Software:**
   The data received by the WLAN Access Point is processed by a software called as Evaluation Software for presenting the report to the users for further processing of the data which can be used for processing, analysis, storage, and mining of the data.

Advantages of Wireless Sensor Networks (WSN):

Low cost: WSNs consist of small, low-cost sensors that are easy to deploy, making them a cost-effective solution for many applications.

Wireless communication: WSNs eliminate the need for wired connections, which can be costly and difficult to install. Wireless communication also enables flexible deployment and reconfiguration of the network.

Energy efficiency: WSNs use low-power devices and protocols to conserve energy, enabling long-term operation without the need for frequent battery replacements.
Scalability: WSNs can be scaled up or down easily by adding or removing sensors, making them suitable for a range of applications and **environments.**
Real-time monitoring: WSNs enable real-time monitoring of physical phenomena in the environment, providing timely information for decision making and control.
Disadvantages of Wireless Sensor Networks (WSN):
Limited range: The range of wireless communication in WSNs is limited, which can be a challenge for large-scale deployments or in environments with obstacles that obstruct radio signals.
Limited processing power: WSNs use low-power devices, which may have limited processing power and memory, making it difficult to perform complex computations or support advanced applications.
Data security: WSNs are vulnerable to security threats, such as eavesdropping, tampering, and denial of service attacks, which can compromise the confidentiality, integrity, and availability of data.
Interference: Wireless communication in WSNs can be susceptible to interference from other wireless devices or radio signals, which can degrade the quality of data transmission.
Deployment challenges: Deploying WSNs can be challenging due to the need for proper sensor placement, power management, and network configuration, which can require significant time and resources.
while WSNs offer many benefits, they also have limitations and challenges that must be considered when deploying and using them in real-world applications.

UNIT-II

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 telemetrics 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:

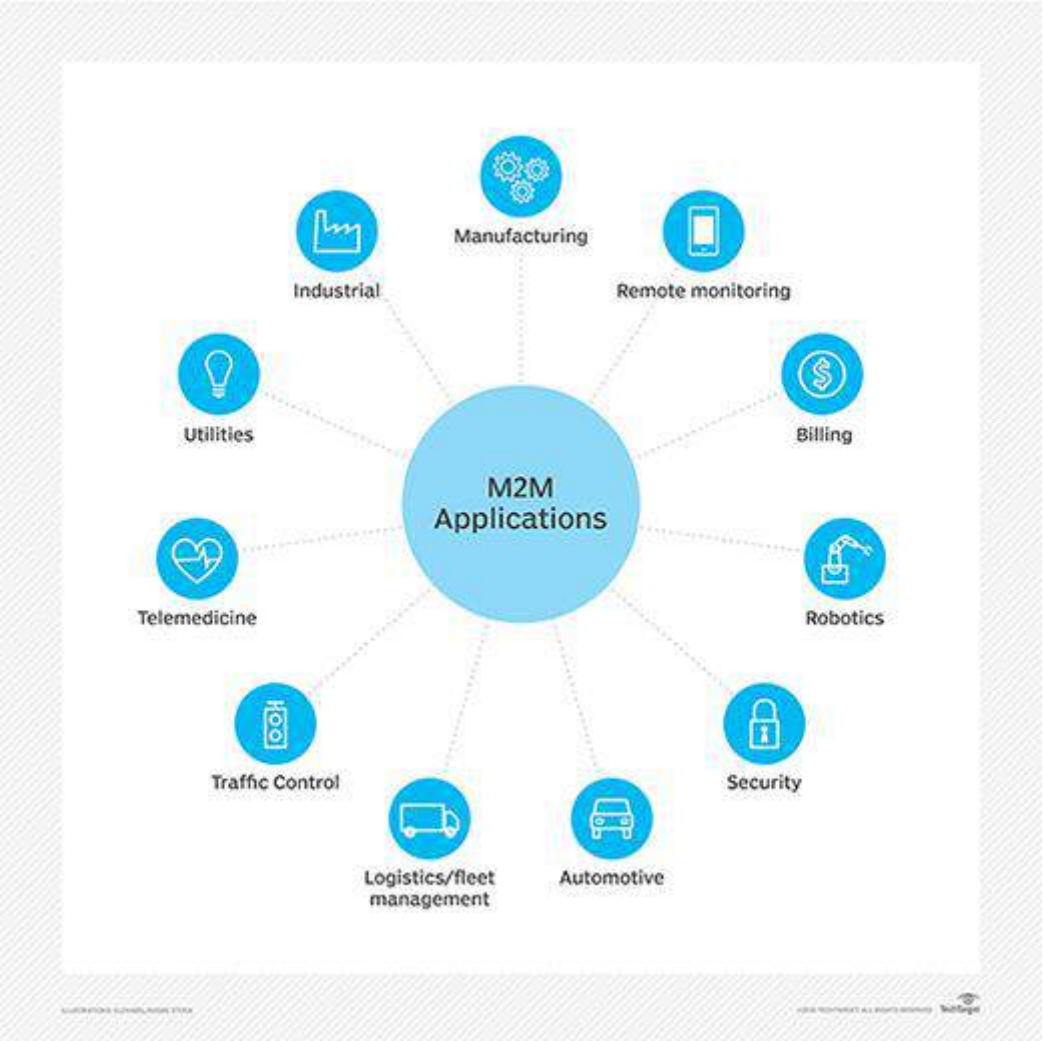
- reduced costs by minimizing equipment maintenance and downtime;
- boosted revenue by revealing new business opportunities for servicing products in the field; and

- 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 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.

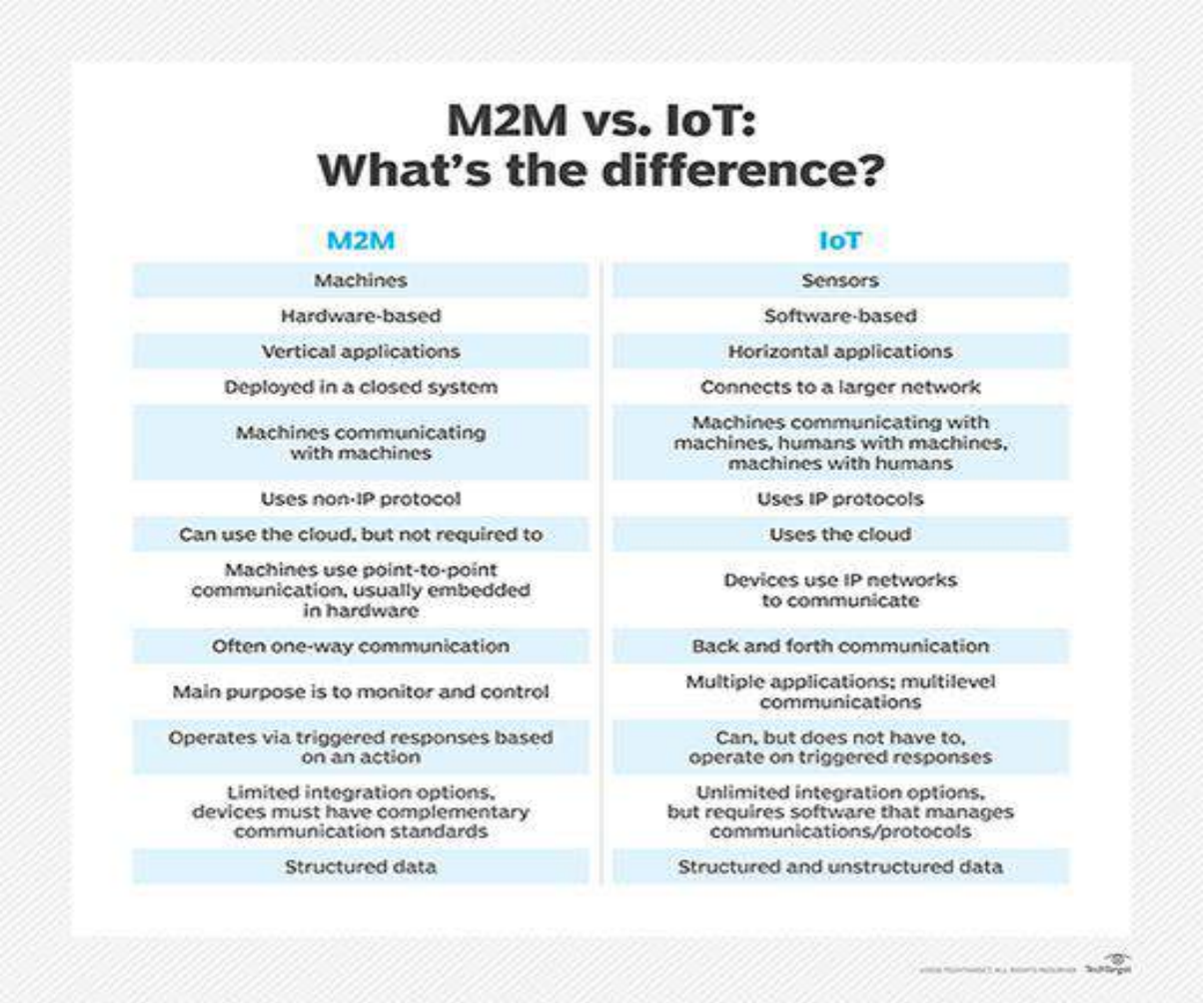
- M2M application communication principles - M2M systems should enable communication between M2M applications in the network and the M2M device or gateway using communication techniques, such as short message service (SMS) and IP Connected devices should also be able to communicate with each other in 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 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.

M2M standards: Machine-to-machine technology does not have a standardized device platform, and many M2M systems are built to be task- or device-specific. Several key M2M standards, many of which are also used in IoT settings, have emerged over the years, including:

- OMA DM (Open Mobile Alliance Device Management), a device management protocol
- OMA LightweightM2M, a device management protocol
- MQTT, a messaging protocol
- TR-069 (Technical Report 069), an application layer protocol
- HyperCat, a data discovery protocol
- OneM2M, a communications protocol
- Google Thread, a wireless mesh protocol
- AllJoyn, an open source software framework

Concerns about M2M

The major concerns surrounding M2M are all related to security. M2M devices are expected to operate without human direction. This increases the potential of security threats, such as hacking, data breaches and unauthorized monitoring. In order to repair itself after malicious attacks or faults, an M2M system must allow remote management, like firmware updates.

The necessity of remote management also becomes a concern when considering the length of time M2M technology spends deployed. The ability to service mobile M2M equipment becomes unrealistic since it is impossible to send personnel to work on them.

The inability to properly service the M2M equipment creates various unique security vulnerabilities for the M2M systems and the wireless networks they use to communicate.

History of M2M

While the origins of the acronym are unverified, the first use of machine-to-machine communication is often credited to Theodore Paraskevakos, who invented and patented technology related to the transmission of data over telephone lines, the basis for modern-day caller ID.

Nokia was one of the first companies to use the acronym in the late 1990s. In 2002, it partnered with Opto 22 to offer M2M wireless communication services to its customers.

In 2003, *M2M Magazine* launched. The publication has since defined the six pillars of M2M as remote monitoring, RFID, sensor networking, smart services, telematics and telemetry.

Interoperability for IoT devices

WIRELESS SOLUTIONS FOR INTEROPERABLE IOT DEVICES

Internet of Things (IoT) is an ever-growing network of physical devices embedded with sensors, actuators,and wire-less connectivity to communicate and share their informationamong themselves.
The application of IoT is in diverse areas such as agriculture, poultry and farming, smart city, and health care, where a sensor node must support heterogeneous sensors/actuators, and varying types of wireless connectivity.
Interoperability is the ability of two or more devices, systems, platforms or networks to work in conjunction.
Interoperability enables communication between heterogeneous devices or system in order to achieve a common goal. However, the current devices and systems are fragmented with respect to the communication technologies, protocols, and data formats.
This diversity makes it difficult for devices and systems in the IoT network to communicate and share their data with one another.
The utility of IoT network is limited by the lack of interoperability.
We work towards achieving and implementing interoperability in IoT-based systems and environment. We propose solutions to enable seamless integration of peripheral with IoT device towards building a global IoT network of heterogeneous sensors and actuators. We study and analyze dynamic integration of heterogeneous devices in IoT environment.

HardwareComponents-Computing(Arduino,RaspberryPi)

IoT (Internet of Things) is no longer a buzzword. With several inspiring use cases, emanating daily, multiple firms are now discovering how they could leverage the technology for business growth. It is fast becoming an important feature for new devices to be IoT based, irrespective of the other technologies implemented, and according to Gartner, by 2020, 95% of new devices and systems will use the IoT.

Each is a part of an IoT hardware platform — a combination of hardware, connectivity tools, and software development environment for IoT projects.

Arduino

- ✓ Audrino is an open-source electronic device that can read inputs (such as light on asensor,fingeronabutton,oraTwittermessage),andbasedontheseinputsproduces output(suchas turningonanLEDoractivatingamotor).
- ✓ Arduino boards are microcontrollers, not full computers with their own operatingsystem like Raspberry Pi. They simply execute code written in C/C++, stored in theirfirmware.
- ✓ Arduino Integrated Development Environment (IDE) is an open-source software

usedtowritecodesand uploadthemtoanAudrinoboard.

Features

- MostoftheArduinoboardscomewithan8-bitMicrocontroller.
- 32kbytesofflashmemoryand2kbytesofSRAM(StaticRandomAccessMemory).
- Inputvoltagerequired- 7V– 12V
- ArduinousesC/C++asaprogramminglanguage.
- DigitalI/Opins-14
- AnalogInputpins-6
- Clockfrequency-
- Processorspeedrangesfrom8MHzto400MHz.TheaveragespeedofmostoftheArduinosis 16MHz.
- Itislimited toIDE(IntegratedDevelopmentEnvironment)

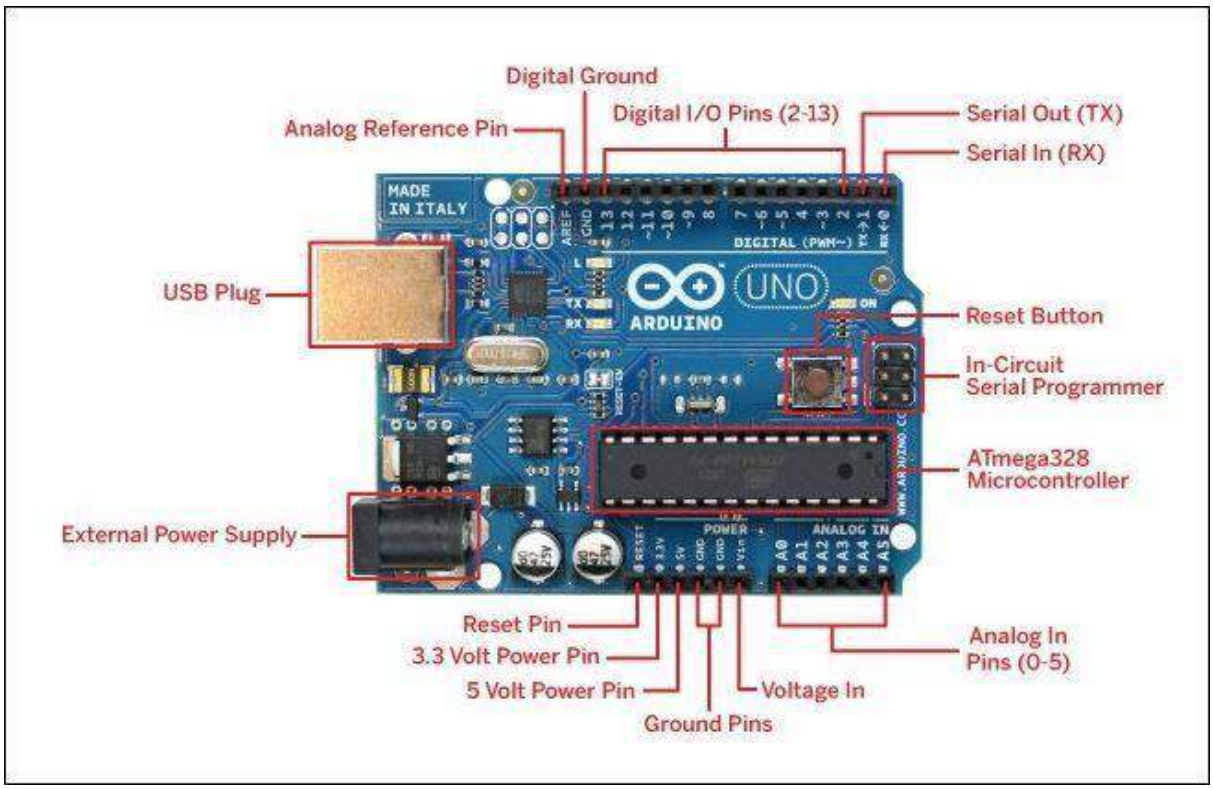
ArduinoUnoR3Specifications

The**ArduinoUnoR3board**includesthe followingspecifications.

- ItisanATmega328PbasedMicrocontroller
- TheOperatingVoltageoftheArduinois5V
- Therecommended inputvoltagerangesfrom7Vto12V

- Thei/pvoltage(limit) is6Vto20V
- Digitalinputandoutputpins-14
- Digitalinput&output pins(PWM)-6
- Analogi/ppinsare6
- DCCurrentforeach I/OPinis20mA
- DCCurrentusedfor 3.3VPin is50mA
- FlashMemory -32KB,and 0.5KBmemoryisusedbythebootloader
- SRAMis2KB
- EEPROMis1KB
- ThespeedoftheCLKis16MHz
- InBuiltLED
- Lengthandwidth oftheArduinoare68.6mmX53.4mm
- TheweightoftheArduinoboard

is25gArduinoUnoR3PinDiagram



PowerSupply

The power supply of the Arduino can be done with the help of an exterior power supplyotherwise USB connection. The exterior power supply (6 to20 volts) mainly includes abattery or an AC to DC adapter. The connection of an adapter can be done by plugging acenter-positive plug (2.1mm) into the power jack on the board. The battery terminals can beplaced in the pins of Vin as well as GND. The power pins of an **Arduino board** include thefollowing.

Vin: The input voltage or Vin to the Arduino while it is using an exterior power supplyopposite to volts from the connection of USB or else **RPS (regulated power supply)**. By usingthispin,onecansupply thevoltage.

5Volts: The RPS can be used to give the power supply to the microcontroller as well ascomponentswhichareusedontheArduinoboard.Thiscanapproachfromtheinputvoltagethrough aregulator.

3V3: A 3.3 supply voltage can be generated with the onboard regulator, and the highestdrawcurrentwill be50mA.

GND:GND(ground)pins

Memory

The memory of an ATmega328 microcontroller includes 32 KB and 0.5 KB memory is utilizedfortheBootloader),and also itincludes SRAM-2 KBaswellas EEPROM-1KB.

InputandOutput

We know that an arguing Uno R3 includes 14-digital pins which can be used as an inputotherwise output by using the functions like pin Mode (), digital Read(), and digital Write().These pins can operate with 5V, and every digital pin can give or receive 20mA, & includes a20k to 50k ohm pull up resistor. The maximum current on any pin is 40mA which cannotsurpass for avoiding the microcontroller from the damage. Additionally, some of the pins ofanArduinoincludespecificfunctions.

SerialPins

The serial pins of an Arduino board are TX (1) and RX (0) pins and these pins can be used totransfer the TTL serial data. The connection of these pins can be done with the equivalentpins oftheATmega8U2USBtoTTLchip.

ExternalInterruptPins

The external interrupt pins of the board are 2& 3, and these pins can be arranged toactivate an interrupt on a rising otherwise falling edge, a low-value otherwise a modify invalue

PWMPins

The PWM pins of an Arduino are 3, 5, 6, 9, 10, & 11, and gives an output of an 8-bit PWMwiththefunctionanalogWrite().

SPI(SerialPeripheralInterface)Pins

The SPI pins are 10, 11, 12, 13 namely SS, MOSI, MISO, SCK, and these will maintain the **SPIcommunication**withthehelpoftheSPI library.

LEDPin

An arguing board is inbuilt with a LED using digital pin-13. Whenever the digital pin is high,theLEDwillglowotherwiseitwillnotglow.

TWI(2-WireInterface)Pins

The TWI pins are SDA or A4, & SCL or A5, which can support the communication of TWI withthehelpofWirelibrary.

AREF(AnalogReference)Pin

An analog reference pin is the reference voltage to the inputs of an analog i/ps using thefunction likeanalogReference().

Reset(RST) Pin

This pin brings a low line for resetting the microcontroller, and it is very useful for using anRSTbutton toward shields which canblocktheoneovertheArduinoR3board.

Communication

ThecommunicationprotocolsofanArduinoUnoincludeSPI,I2C,and **UARTserialcommunication**.

UART

An Arduino Uno uses the two functions like the transmitter digital pin1 and the receiverdigitalpin0.Thesepinsaremainlyusedin UARTTTLserialcommunication.

I2C

AnArduinoUNOboardemploysSDApinotherwiseA4pin&A5pinotherwiseSCLpinisused for I2C communication with wire library. In this, both the SCL and SDA are CLK signalanddatasignal.

SPIPins

TheSPIcommunicationincludesMOSI,MISO,andSCK.

MOSI(Pin11)

Thisisthemasteroutslaveinthepin,usedtotransmitthedatatothedevices

MISO(Pin12)

This pin is a serial CLK, and the CLK pulse will synchronize the transmission of which isproducedby themaster.

SCK(Pin13)

The CLK pulse synchronizes data transmission that is generated by the master. Equivalentpins with the SPI library is employed for the communication of SPI. ICSP (in-circuit serialprogramming) headers can be utilized for programming **ATmega microcontroller** directlywiththebootloader.

```
}
```

INTEGRATION OF SENSOR AND ACTUATORS WITH ARDUINO in IoT

Sensors

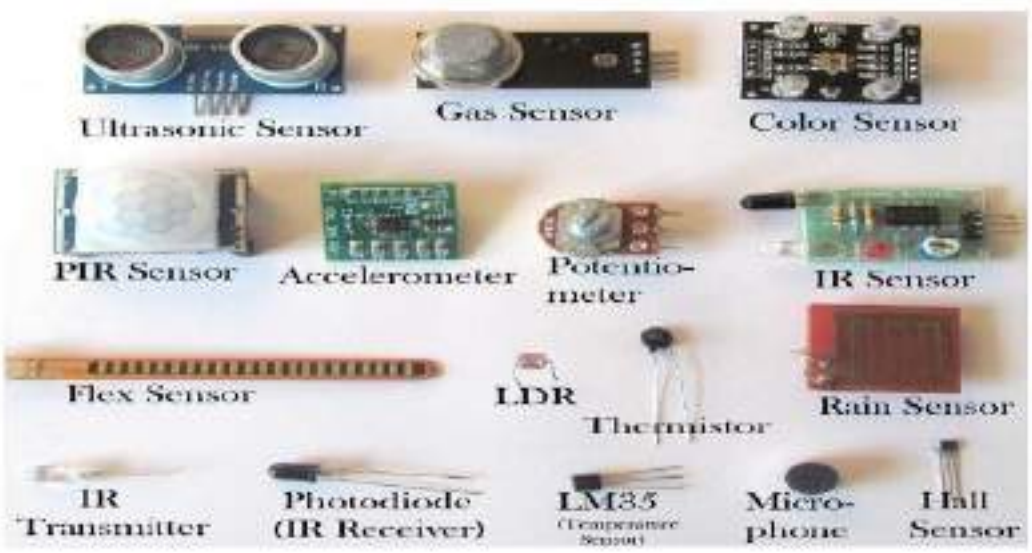
A better term for a sensor is a transducer. A transducer is any physical device that convertsone form of energy into another. So, in the case of a sensor, the transducer converts somephysical phenomenon into an electrical impulse that determines the reading. A microphoneisasensorthattakesvibrationalenergy(soundwaves),andconvertsittoelectricalener

S.No	Sensor	Applications	Technology
1.	Inertial sensors	Industrial machinery, automotive, human activity	MEMS and Gyroscope
2.	Speed Measuring Sensor	Industrial machinery, automotive, human activity	Magnetic, light
3.	Proximity sensor	Industrial machinery, automotive, human activity	Capacitive, Inductive, Magnetic, Light, Ultrasound
4.	Occupancy sensor	Home/office monitoring	PassiveIR, Ultrasound most common
5.	Temperature/humidity sensor	Home/office HVAC control, automotive, industrial	Solid state, thermocouple
6.	Light sensor	Home/office/industrial lighting control	Solid state, photocell, Photo resistor, photodiode

gyinausefulwayfor othercomponentsinthesystemtocorrelatebacktotheoriginalsound.

7.	Power (current) sensor	Home/office/industrial powermonitoring/control Technology	Coil (Faraday's law), Hall effect
8.	Air/fluid pressure sensor	Industrial monitoring/control, automotive, agriculture	Capacitive, Resistive
9.	Acoustic sensor	Industrial monitoring/control, human interface	Diaphragm condenser
10.	Strain sensor	Industrial monitoring/control, civil infrastructure	Resistive thin films

- ✓ Inthefirstclassification ofthesensors,theyaredividedin to**ActiveandPassive**.
- ✓ **Active Sensors** are those which require an external excitation signal or a power signal.**PassiveSensors,**ontheotherhand,donotrequireanyexternalpowersignalanddirectly generatesoutputresponse.
- ✓ The other type of classification is based on the means of detection used in the sensor.SomeofthemeansofdetectionareElectric,Biological,Chemical,Radioactiveetc.
- ✓ The next classification is based on conversion phenomenon i.e. the input and the output.SomeofthecommonconversionphenomenaarePhotoelectric,Thermoelectric,Electrochemical,Electromagnetic, Thermo-optic, etc.
- ✓ The final classification of the sensors are Analog and Digital Sensors. Analog Sensorsproduce an analog output i.e. a continuous output signal with respect to the quantitybeing measured. Digital Sensors, in contrast to Analog Sensors, work with discrete ordigital data. The data in digital sensors, which is used for conversion and transmission, isdigitalinnature.



1. IRLED:

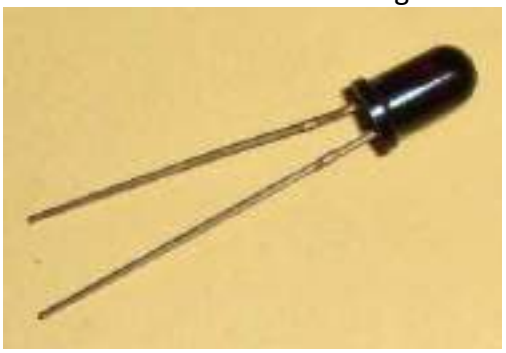
- ✓ ItisalsocalledasIRTransmitter.
- ✓ Itis usedtoemitInfraredrays.
- ✓ Therangeofthesefrequenciesaregreaterthanthemicrowavefrequencies(i.e.>300GHztofewhundreds of THz).

- ✓ TheraysgeneratedbyaninfraredLEDcanbesensedbyPhotodiodeexplainedbelow.
- ✓ **Thepairof**IRLEDandphotodiodeiscalled<u>IRSensor</u>**.**

Fig:LEDsensor

2. PhotoDiode(Light Sensor):

- ✓ It is a semiconductor device which*is* used to detect the light rays and mostlyusedas IRReceiver.
- ✓ ItsconstructionissimilartothenormalPNjunctiondiodebuttheworkingprinciplediffers fromit.
- ✓ As we know a PN junction allows small leakage currents when it is reverse biasedso,this property isusedtodetectthelightrays.
- ✓ A photodiode is constructed such that light rays should fall on the PN junctionwhich makes the leakage current increase based on the intensity of the light thatwehaveapplied.
- ✓ So, in this way, a photodiode can be used to sense the light rays and maintain thecurrentthroughthecircuit.CheckheretheworkingofPhotodiodewithIRsensor.



3. **Proximity Sensor:**A Proximity Sensor is a non-contact type sensor that detects thepresenceofanobject.ProximitySensorscanbeimplementedusingdifferenttechniqueslikeOptical(likeInfraredorLaser),Ultrasonic,HallEffect,Capacitive,etc.

SomeoftheapplicationsofProximitySensors areMobilePhones,Cars(ParkingSensors),industries (objectalignment),**GroundProximity inAircrafts,etc.Proximity**

Sensor in Reverse Parking is implemented in this Project: Reverse Parking SensorCircuit**.**

4. # LDR(LightDependentResistor):

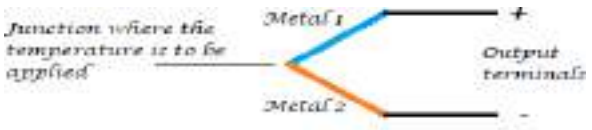
As the name itself specifies that the resistor that depends upon the light intensity. Itworks on the principle of photoconductivity which means the conduction due to thelight. It is generally made up of Cadmium sulfide. When light falls on the LDR**,** itsresistance decreases and acts similar to a conductor and when no light falls on it, itsresistance is almost in the range of MΩ or ideally it acts as an open circuit**.** One noteshould be considered with LDR is that it won‘t respond if the light is not exactlyfocusedonits surface.

With a proper circuitry using a transistor it can be used to detect the availability oflight. A voltage divider biased transistor with R2 (resistor between base and emitter)replaced withanLDRcanworkasalightdetector.

5. **Thermistor (Temperature Sensor):** A thermistor can be used to detect the variationin temperature**.**It hasanegative temperaturecoefficient that meanswhen thetemperature increases the resistance decreases. So, the thermistor‘s resistance canbe varied with the rise in temperature which causes more current flow through it.This change in current flow can be used to determine the amount of change intemperature.Anapplicationforthermistoris,itisusedtodetecttheriseintemperature and control the leakage current in a transistor circuit which helps inmaintaining its stability. Here is one simple application for Thermistor to control theDCfanautomatically.

6. # Thermocouple(TemperatureSensor):

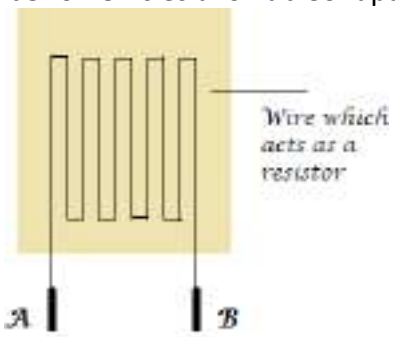
Another component that can detect the variation in temperature is a thermocouple**.**In its construction, two different metals are joined together to form a junction. Itsmain principle is when the junction of two different metals are heated or exposed tohigh temperatures a potential across their terminals varies. So, the varying potentialcanbefurther usedtomeasuretheamountofchangeintemperature.



Thermocouple

7. StrainGauge(Pressure/ForceSensor) :

A strain gauge is used to detect pressure when a load is applied**.** It works on theprinciple of resistance, we know that the resistance is directly proportional to thelength of the wire and is inversely proportional to its cross-sectional area (R=ρl/a).The same principle can be used here to measure the load. On a flexible board, a wireisarrangedinazig-zagmannerasshowninthefigurebelow.So,whenthepressureis applied to that particular board, it bends in a direction causing the change inoverall length and cross- sectional area of the wire. This leads to change in resistanceof the wire. The resistance thus obtained is very minute (few ohms) which can bedetermined with the help of the Wheatstone bridge. The strain gauge isplaced inone of the four arms in a bridge with the remaining values unchanged. Therefore,when the pressure is applied to it as the resistance changes the current passingthrough the bridge varies and pressure can be calculated. Strain gauges are majorlyusedtocalculatetheamountofpressurethatanairplanewingcanwithstandanditis alsousedto measurethenumber ofvehicles allowableonaparticular roadetc.

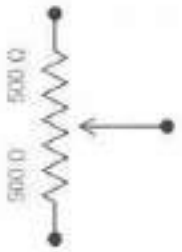


Strain Gauge

8. LoadCell(WeightSensor):

Loadcellsaresimilartostraingaugeswhichmeasurethephysicalquantity like force and give the output in form of electrical signals. Whensome tension is applied on the load cell it structures varies causing thechangeinresistanceandfinally,itsvaluecanbecalibratedusingaWheatsto ne bridge. Here is the project on how to measure weight usingLoadcell.

9. **Potentiometer:**Apotentiometerisusedtodetectthepositon.Itgenerallyhasvariousrang esofresistorsconnectedtodifferentpolesoftheswitch.Apotentiometercanbeeitherrotar yorlineartype.Inrotarytype,awiperisconnected to a long shaft which can be rotated. When the shaft has rotated theposition of the wiper alters such that the resultant resistance varies causing thechange in the output voltage. Thus the output can be calibrated to detect the changeitsposition.



Potentiometer

10. **Encoder :** To detect the change in the position an encoder can also be used. It has acircular rotatable disk-like structure with specific openings in between such thatwhen the IR rays or light rays pass through it only a few light rays get detected.Further,theseraysareencodedintoadigitaldata(intermsofbinary)whichrepresentsthespecificposition.

11. **HallSensor:**

The name itself states that it is the sensor which works on the Hall Effect. It can bedefined as when a magnetic field is brought close to the current carrying conductor(perpendicular to the direction of the electric field) then a potential difference isdeveloped across the given conductor. Using this property, a Hall sensor is used todetect the magnetic field and gives output in terms of voltage. Care should be takenthattheHall sensorcandetectonlyonepoleof themagnet.

The hall sensor is usedinfew smartphones whichare helpful in turning off thescreen when the flap cover (which has a magnet in it) is closed onto the screen. Hereisonepracticalapplicationof HallEffectsensorinDoorAlarm.

12. **Flex Sensor:** A FLEX sensor is a transducer which changes its resistance when itsshape is changed or when it is bent. A FLEX sensor is 2.2 inches long or of fingerlength. Simply speaking the sensor terminal resistance increases when it‘s bent. Thischange in resistance can do no good unless we can read them. The controller at handcan only read the changes in voltage and nothing less, for this, we are going to usevoltagedividercircuit,withthatwecanderivetheresistancechangeasavoltage

change.

Fig:Flexsensor

13. **Microphone (Sound Sensor):** Microphone can be seen on all the smartphones ormobiles. It can detect the audio signal and convert them into small voltage (mV)electrical signals. A microphone can be of many types like condenser microphone,crystal microphone, carbon microphone etc. each type of microphone work on theproperties like capacitance, piezoelectric effect, resistance respectively. Let us seethe operation of a crystal microphone which works on the piezoelectric effect. Abimorph crystal is used which under pressure or vibrations produces proportionalalternating voltage. A diaphragm is connected to the crystal through a drive pin suchthat when the sound signal hits the diaphragm it moves to and fro, this movementchanges the position of the drive pin which causes vibrations in the crystal thus analternatingvoltageisgeneratedwithrespecttotheappliedsoundsignal.Theobtained voltage is fed to an amplifier in order to increase the overall strength of thesignal.

14. **Ultrasonicsensor:**

Ultrasonic means nothing but the range of the frequencies. Its range is greater thanaudible range (>20 kHz) so even it is switched on we can‘t sense these sound signals.Onlyspecificspeakersandreceiverscansensethoseultrasonicwaves.Thisultrason ic sensor is *used to calculate the distance between the ultrasonic transmitterandthetarget andalsoused tomeasurethevelocityofthetarget***.**

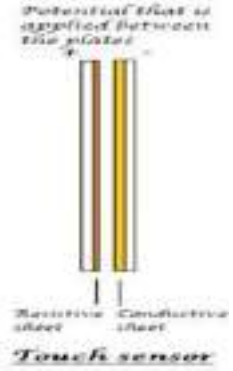
Ultrasonic sensor HC-SR04 can be used to measure distance in the range of 2cm-400cm with an accuracy of 3mm. Let‘s see how this module works. The HCSR04module generates a sound vibration in ultrasonic range when we make the ‗Trigger‘pin high for about 10us which will send an 8 cycle sonic burst at the speed of soundand after striking the object, it will be received by the Echo pin. Depending on thetime taken by sound vibration to get back, it provides the appropriate pulse output.Wecancalculatethedistanceoftheobjectbasedonthetimetakenbytheultrasonic wavetoreturnbacktothesensor.

There are many applications with the ultrasonic sensor. We can make use of it avoidobstacles for the automated cars, moving robots etc. The same principle will be usedin the RADAR for detecting the intruder missiles and airplanes. A mosquito can sensetheultrasonicsounds.So,ultrasonicwavescanbeusedasmosquitorepellent.

15. TouchSensor

In this generation, we can say that almost all are using smartphones which havewidescreen that too a screen which can sense our touch. So, let‘s see how thistouchscreen works. Basically, there are two types of touch sensors *resistive basedand a capacitive based touch screens.* Let‘s know about working of these sensorsbriefly.

The resistive touch screen has a resistive sheet at the base and a conductive sheetunder the screen both of these are separated by an air gap with a small voltageapplied to the sheets. When we press or touch the screen the conductive sheettouches the resistive sheet at that point causing current flow at that particular point,thesoftwaresenses thelocation and relevantactionisperformed.



16. PIRsensor

PIRsensorstandsfor**PassiveInfraredsensor**.Theseareusedtodetectthemotionofhumans,animalsorthings.Weknowthatinfraredrayshaveapropertyofreflection. When an infrared ray hits an object, depending upon the temperature ofthe target the infrared ray properties changes, this received signal determines themotion of the objects or the living beings. Even if the shape of the object alters, theproperties of the reflected infrared rays can differentiate the objects precisely. HereisthecompleteworkingorPIRsensor.

17. Accelerometer(TiltSensor):

Anaccelerometersensor*can*sensethetiltormovementofitinaparticulardirection**.** It works based on the acceleration force caused due to the earth‘s gravity.The tiny internal parts of it are such sensitive that those will react to a small externalchange in position. It has a piezoelectric crystal when tilted causes disturbance in thecrystal and generates potential which determines the exact position with respect toX, Y and Z axis. These are commonly seen in mobiles and laptops in order to avoidbreakage of processors leads. When the device falls the accelerometer detects thefallingconditionanddoes respectiveactionbasedonthesoftware.

18. Gassensor:

In industrial applications gas sensors plays a major role in **detecting the gas leakage**.If no such device is installed in such areas it ultimately leads to an unbelievabledisaster. These gas sensors are classified into various types based on the type of gasthat to be detected. Let‘s see how this sensor works.Underneath a metal sheetthere exists a sensing element which is connected to the terminals where a current isapplied to it. When the gas particles hit the sensing element, it leads to a chemicalreaction such that the resistance of the elements varies and current through it alsoalters which finally can detect the gas. So finally, we can conclude that sensors arenot only used to make our work simple to measure the physical quantities, makingthedevices automatedbutalsousedto help living beingswithdisasters.

19. ResistiveSensors:

Resistive sensors, such as the potentiometer, have three terminals: power input,grounding terminal, and variable voltage output. These mechanical devices havevariedresistancethatcanbechangedthroughmovablecontactwithitsfixedresistor. Outputfromthesensorvariesdependingonwhetherthemovablecontactisneartheresistor'ssuppleendorgroundend.Thermistorsarealsovariableresistors,althoughtheresistanceofthesensor varies withtemperature

20. Voltagegeneratingsensors:

Voltage-generating sensors, such as piezo electrics, generate electricity by pressurewith types of crystals like quartz. As the crystal flexes or vibrates, AC voltage isproduced.Knocksensorsutilizethistechnologybysendingasignaltoanautomobile'son-boardcomputerthatengineknockishappening.Thesignalisgenerated through crystal vibration within the sensor, which is caused by cylinderblock vibration. The computer, in turn, reduces the ignition timing to stop the engineknock.

21. SwitchSensors

Switch sensors are composed of a set of contacts that open when close to a magnet.Areedswitchisacommonexampleofaswitchsensorandismostcommonlyusedas aspeedorpositionsensor.Asaspeedsensor,amagnetisattachedtothespeedometer cable and spins along with it. Each time one of the magnet's polespasses the reed switch, it opens and then closes. How fast the magnet passes allowsthesensortoreadthevehicle's speed.

Actuator

Another type of transducer that you will encounter in many IoT systemsis an actuator. Insimple terms, an actuator operates in the reverse direction of a sensor. It takes an electricalinput and turns it into physical action. For instance, an electric motor, a hydraulic system,andapneumaticsystemareall differenttypes ofactuators.

Examplesofactuators

- Digitalmicro mirrordevice
- Electricmotor
- Electroactivepolymer
- Hydrauliccylinder
- Piezoelectricactuator
- Pneumaticactuator
- Screwjack

- Servomechanism
- Solenoid
- Steppermotor

Types of Actuators

1. **Hydraulic actuator**

A hydraulic actuator is a mechanical device that employs hydraulic power to complete a task. A cylinder or a fluid motor drives them. According to the needs of the IoT device, mechanical motion is translated to rotary, linear, or oscillatory motion. Hydraulic actuators are used in construction equipment because they can create a considerable amount of force.

Advantages

- Hydraulic actuators have the ability to generate significant amounts of force at a high rate.
- Used in welding, clamping, and other applications.
- In car transport carriers, it's used to lower or raise the vehicles.

Disadvantages

- Leaks in hydraulic fluid can reduce performance and complicate cleanup.
- Noise reduction equipment, heat exchangers, and high-maintenance systems are all required.
- It is expensive.

2. Pneumatic Actuators

A pneumatic actuator converts energy created by vacuum or high-pressure compressed air into linear or rotary motion. For example, sensors that act like human fingers and are powered by compressed air are used in robotics.

Advantages

- They are a low-cost solution that is employed in extreme temperatures where employing air rather than chemicals is a safer option.
- They require little maintenance, are long-lasting, and have a lengthy service life.
- It is quite quick to initiate and stop the action.

Disadvantages

- It can become less efficient if there is a loss of pressure.
- The air compressor should be turned on all the time.
- It is possible for air to be polluted, and it must be maintained.

3. Electrical actuators

An electric actuator works by converting electrical energy into mechanical torque and is usually powered by a motor. A solenoid-based electric bell is an example of an electric actuator.

Advantages

- It can automate industrial valves, which makes it useful in a variety of sectors.
- It makes less noise and is completely safe to use because there are no fluid leaks.
- It has the ability to be reprogrammed and delivers the highest level of control and precision positioning.

Disadvantages

- It's not cheap.
- It is highly dependent on the surrounding environment.

UNIT-III

PYTHON IN IOT(INTERNET OF THINGS)

Python programming powers intuitive interfaces of intelligent and effective Internet of Things (IoT) systems that are paramount in remote sensor networks, big data and data analysis, automation, and machine learning. IoT applications function efficiently with the help of Python **libraries/packages** which include:

NUMPY

Numpy is a scientific computing package that helps to create datasets to test with the time series data in IoT. Numpy features are used in IoT to read sensor bulk data from the database inbuilt functions in the system

SOCKETS AND MYSQLDB

Sockets that facilitate networking in IoT devices include TCP/IP and UDP, which are compatible to work with Python packages. TCP/IP and UDP act as transport layer protocols for communication. The MySQLdb is a go-to relational format database that helps in the development of remote stores for the IoT system.

MATPLOTLIB

To get data insights, matplotlib visualizes the most paramount operations by giving a variety of graphs to represent the data.

REQUESTS, TKINTER AND TENSORFLOW

To make HTTP calls and parse responses in Python, the **request package** acts as a major protocol for data exchanges. **Tkinter GUI** puts the aspects of Python script in a controlled distribution, which enables functional testing and repeated executions in IoT Python devices.

Therefore, the numerical computations of machine learning initiated into the IoT systems utilize the representation in data flow graphs dealing with huge non-linear datasets and deep learning aspects.

AZURE IOT SDK IN PYTHON

Azure IoT hub offers a variety of features for IoT SDK usage which provides the ability to connect devices and services. The IoT SDK is supported by the MQTT protocol which facilitates the data exchange processes. The device requirements to be used along with Python include:

- Python version 3.7+: helps in both asynchronous and synchronous API
- Azure-iot-device library

The IoT hub SDK helps with the following aspects: access, processing, and analysis of data for machine learning applications.

The Azure IoT hub helps collect messages and feedback data collected by IoT devices and is displayed in the code below:

COUNTLY IOT RASPBERRY PI SDK

Sending data and visualizing data on a dashboard is simplified by involving the Countly IoT Pi SDK, which relies on internet connectivity for efficient and effective data insights from the device.

The code below is used to start the process of collecting data using Countly IoT Pi SDK in Python. Install by running: **pip install Raspberry_SDK**:

```
from Raspberry_SDK.Countly
import Countly
#intiate the SDK
Countly = Countly(“SERVER_URL”, “APP_KEY”, 0)
#Send an event
countly.event(“NAME”, VALUE)
```

Countly SDK also helps to retrieve data events for both analog and digital circuits. Use case of Countly IoT Raspberry Pi SDK is applicable in **temperature room measuring** and **Bulb light**. For instance, the server gets to pass the application key to collect the data and data is being manipulated by **GroveAPI** for raspberry IoT as displayed below:

What is Raspberry Pi | IoT Raspberry Pi Tutorial for Beginners

In **IoT Tutorials**, we saw different types of applications
like **Health**, **Education**, **Government** etc. But today, we will talk about a new device called Raspberry Pi that can be incorporated into IoT systems to make work easy.
So, in this Raspberry Pi tutorial, we are going to learn about IoT Raspberry Pi introduction with its innovation. Moreover, we will discuss the difference between Raspberry Pi models in IoT. At last, we will see how to buy IoT Raspberry Pi.

So, let's start with Introduction to IoT Raspberry Pi.

What is a Raspberry Pi?

The Raspberry Pi is a very small computer that is almost the size of your credit card. It costs between Rs 750 and Rs 4000. It can function as a proper desktop computer or use to build smart devices and is available anywhere in the world.

The Pi changed into what initially was meant to be a microcomputer to teach kids coding. Its scope can expand after hobbyists and engineers noticed its capacity, and it's far now one of the most famous objects inside the international era.

IoT Raspberry Pi Tutorial – What is Raspberry Pi

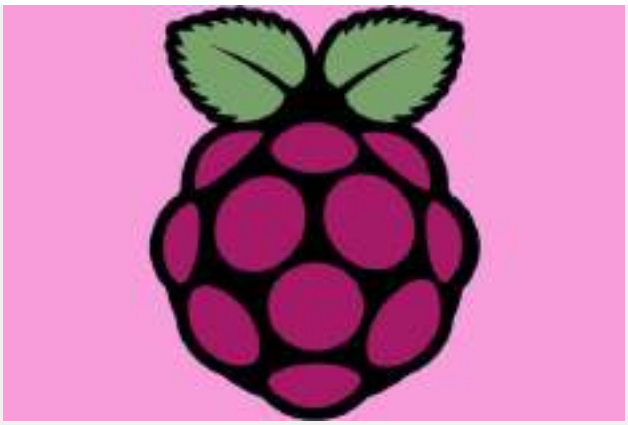
Who Invented IoT Raspberry Pi?

The Raspberry Pi Foundation was formed in the year 2008 after a group of technicians and academics who were concerned about students' interest gradually drifting and declining in computer sciences. So they came up with a low-cost computer as a solution to inspire children and make it more accessible.

The basic motive was that these tiny computer systems might allow for very basic and simple programming. Its low electricity utilization and value expect to make Pis more easily to be used in school rooms.

Why is it called Raspberry Pi?

The "Raspberry" name is a homage to computer companies in early times that were being named after a fruit, like Apple, Apricot Computers, Tangerine Computer Systems. The idea to make a small computer to run only the **Python programming language** is where the "Pi" derives from.



What is Raspberry Pi in IoT

When was IoT Raspberry Pi launched?

The first Raspberry Pi unit which was available commercially was launched on February 19, 2012. This version featured 256MB of RAM, could run on **Linux**-based desktop operating systems, had one USB port, and no Ethernet port. This was named the Model A.

Follow DataFlair on WhatsApp & Stay updated with latest technology trends.

What's the Difference Between Raspberry Pi Models?

IoT Raspberry Pi models can be confusing because there are so many of them and there are two levels to the naming system.

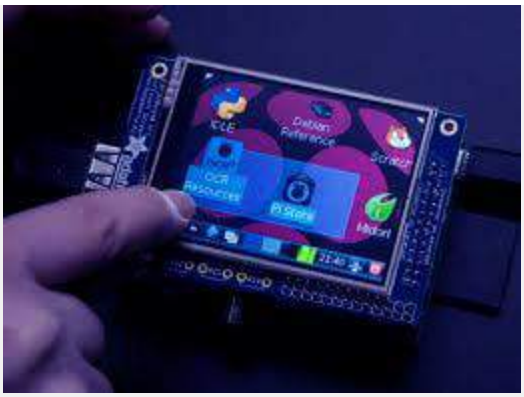
The "generation" of the model, represent by Pi 1, Pi 2, and Pi 3 where Pi 1 is for models between 2012-14, Pi 2 is 2015 models, and Pi 3 is 2016 models. So 3 is the most recent which is better than 2, which is better than 1.

The power and features indicate by model A, A+, B, and B+. It's not like grades though, A is lower than B.

Where is IoT Raspberry Pi's used?

IoT Raspberry Pi can be used in a wide variety of tasks. It's ideal and best suitable for projects where there is a computer requirement but you don't require much processing power, you want to keep the costs low and want to save on space. Here's a brief list of some ideal uses of the Pi.

- Teach kids (or yourself) the way to code.



How IoT Raspberry Pi Used

- Use it as a desktop pc.
- Construct a movement seize safety digital camera or a DIY pan and tilt digital camera with Raspberry Pi.

How IoT Raspberry Pi Uses

- Make your very own retro gaming console.

How IoT Raspberry Pi Uses

- You can make an FM radio or a global clock.
- Prepare time-lapse pictures digital camera with the digital camera module.

How IoT Raspberry Pi Used

IoT and the Maker Movement

Another factor driving the momentous adoption of the IoT system is the rise of the maker culture. The maker culture encourages hobbyists (and professionals alike) to create their own devices as well as tinker with existing ones to find solutions to solve their specific problems.

With the maker movement comes a host of DIY electronic platforms, such as Arduino and Raspberry Pi. Arduino (see **Figure 2**) is a small and inexpensive electronic board that allows you to connect to various external accessories (such as sensors) and create applications to use the data collected.

Figure 2: The Arduino UNO board

Another open-source hardware platform that has gotten very popular with hobbyists these days is Raspberry Pi. It's really a computer, by all definitions. Raspberry Pi is a low-cost, credit card-sized computer that connects to a computer monitor or TV using HDMI, and uses a standard keyboard and mouse. It can run a host of operating systems, such as Raspbian (Debian Linux), Android, Windows 10, IoT Core, etc.

Raspberry Pi has gone through a few iterations and **Table 1** shows the list of Raspberry models released over the years and their prices.
Of the various models, Raspberry Pi 3 (see **Figure 3**) and Raspberry Pi Zero (see **Figure 4**) stand out.

Figure 3: The Raspberry Pi 3

Figure 4: The Raspberry Pi Zero

Raspberry Pi 3 is the third generation of Raspberry Pi and it packs quite a formidable punch in its credit card-sized package. Most notably, in addition to the standard features of the Raspberry Pi (such as four USB 2.0 ports and built-in Ethernet), it has:

- A 1.2GHz 64-bit quad-core ARMv8 CPU
- 802.11n Wireless LAN
- Bluetooth 4.1 Low Energy (BLE)

The powerful CPU coupled with Wireless LAN and Bluetooth 4.1 radio makes it an ideal candidate for IoT projects, because multiple sensors can be connected to it simultaneously. In addition, the Raspberry Pi has a 40-pin GPIO (General Purpose I/O) connector for interfacing with external sensors.

The Raspberry Pi Zero is the smallest Raspberry Pi ever made, and although it doesn't have a processor that's as powerful as the Pi 3, its small size is especially suited for embedded projects (such as wearables, etc.), where space is a premium.

Powering the Raspberry Pi

One of the most popular OSs used for the Raspberry Pi is the Raspbian Operating system. The Raspbian OS is based on the Debian OS, optimized for the Raspberry Pi hardware. The easiest way to install the Raspbian OS for the Raspberry Pi is to download NOOBS from https://www.raspberrypi.org/help/noobs-setup/. NOOBS stands for New Out Of Box Software.

The easiest way to install the Raspbian OS for the Raspberry Pi is to download NOOBS from: https://www.raspberrypi.org/help/noobs-setup/.
The Raspbian OS boots off a micro-SD card and the entire operating system runs off the card. A typical Class 4 8GB micro-SD card is sufficient for most purposes, but you have the option to connect it to an external hard disk or flash drive for more storage.

Once the Raspbian OS is installed, you can proceed to log into it and see a full windowed system (see **Figure 5**). The default username is `pi` and the password is `raspberry`.

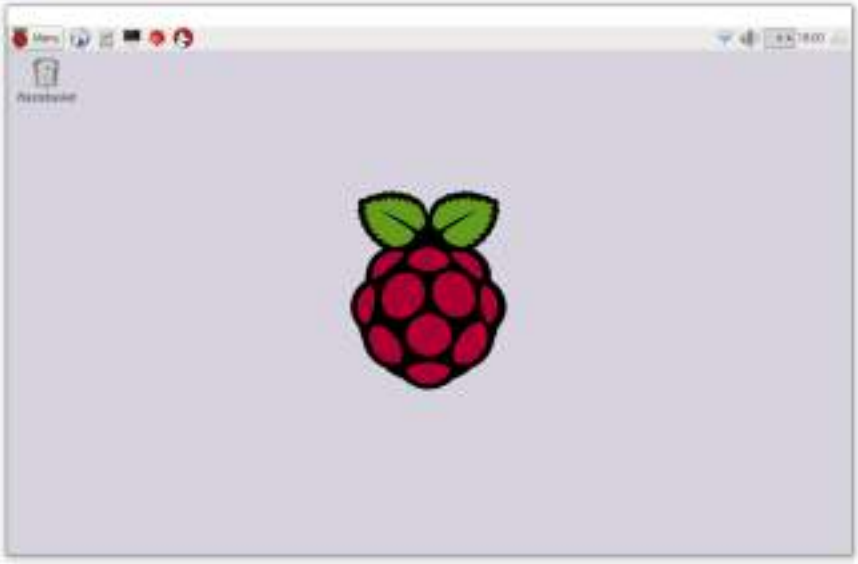


Figure 5: The Raspbian OS uses the LXDE (Lightweight X11 Desktop Environment) for managing user interactions

Connecting the Raspberry Pi to the Outside World - GPIO Pins

The Raspberry Pi has a 40-pin GPIO (General Purpose Input/Output) connection, which makes it very easy to connect to the outside world. To connect the GPIO to external sensors, you can:

- Connect the sensors directly to the GPIO pins using jumper wires
- Connect the GPIO pins to a ribbon cable, which in turn connects it to a breadboard. The Adafruit Pi T-Cobbler Plus - Breakout + Cable for Raspberry Pi A+/B+/Pi 2/Pi 3 (see **Figure 6**) is one such product. This option is ideal during the prototyping phase.

Figure 6: The Adafruit Pi T-Cobbler Plus Breakout + Cable for Raspberry Pi

For project prototyping, my favorite is using the second option: the Adafruit Pi T-Cobbler Plus. The Adafruit Pi T-Cobbler Plus connects to the Raspberry Pi via a ribbon cable (see **Figure 7**).

Figure 7: Connecting the Adafruit Pi T-Cobbler Plus to the Raspberry Pi

Examining the GPIO Pins

One of the advantages of using the Adafruit Pi T-Cobbler Plus is that you have a clear labeling of the various GPIO pins (see **Figure 8**).

Figure 8: The labels on the various pins on the Adafruit Pi T-Cobbler Plus

The GPIO pins on the Raspberry Pi are divided into the following groups:

- **Power:** Pins that are labeled 5.0v supply 5 volts of power and those labeled 3V3 supply 3.3 volts of power. There are two 5V pins and two 3V3 pins.
- **GND:** These are the ground pins. There are eight ground pins.
- **Input/Output pins:** These are the pins labeled with the # sign, for example, #17, #27, #22, etc. These pins can be used for input or output.
- **I2C:** I2C is a serial protocol for a two-wire interface to connect low-speed devices like microcontrollers, EEPROMs, A/D and D/A converters, I/O interfaces, and other similar peripherals in embedded systems. These pins are labeled **SDA** and **SCL**.
- **UART:** The *Universal Asynchronous Receiver/Transmitter* allows your Raspberry Pi to be connected to serial peripherals. The UART pins are labeled **TXD** and **RXD**.
- **SPI:** The *Serial Peripheral Interface* is a synchronous serial communication interface specification used for short distance communication, primarily in embedded systems. The SPI pins are labeled **MOSI**, **MISO**, **SCLK**, **CE0**, and **CE1**.
- **ID EEPROM:** *Electrically Erasable Programmable Read-Only Memory* is a user-modifiable read-only memory that can be erased and written to repeatedly through the application of higher than normal electrical voltage. The two **EEPROM** pins on the Raspberry Pi (**EED** and **EEC**) are also secondary I2C ports that primarily facilitate the identification of Pi Plates (e.g., Raspberry Pi Shields/Add-On Boards) that are directly attached to the Raspberry Pi.

Connecting to a Sensor to Detect Motion

To demonstrate how to use the **GPIO** to connect to an external sensor, we'll now use a **PIR** motion sensor to detect motion. For this, I used the **Parallax PIR Motion Sensor** .The PIR Sensor detects motion by measuring changes in the infrared (heat) levels emitted by surrounding objects of up to three meters.

Figure 9: The Parallax PIR Motion Sensor

The Parallax Motion sensor has three pins (see **Figure 10**):

- **GND:** The ground pin. Connect this pin to the GND on the GPIO.
- **VCC:** The voltage pin. Connect this pin to one of the 5V pins on the GPIO.
- **OUT:** The output pin. Connect this to one of the Input/Output pins on the GPIO.

Figure 10: The layout of the various pins on the PIR Motion Sensor

When the PIR Motion sensor detects motion, it outputs a high signal on its output pin. You need to write an application to read the value of this output pin. **Figure 11** shows a PIR Motion sensor connected to the T-Cobbler Plus.
Depending on the PIR Motion Sensor that you're using, the arrangement of the various pins isn't always in the same order as described. It's important to verify and connect the correct pins to the correct GPIO pins. Connecting the wrong pins to the Raspberry Pi can permanently damage the PIR Motion Sensor.

In the figure, the red line is the VCC and should be connected to the 5V pin on the GPIO.

The yellow line is the OUTPUT and is connected to pin #4 on the GPIO. The black line is

the GND and should be connected to GND on the GPIO. Implementation of IoT with

Raspberry Pi

Internet of Things

- Creating an interactive environment
- Network of devices connected together

Sensor

- Electronic element
- Converts physical quantity into electrical signals
- Can be analog or digital

Actuator

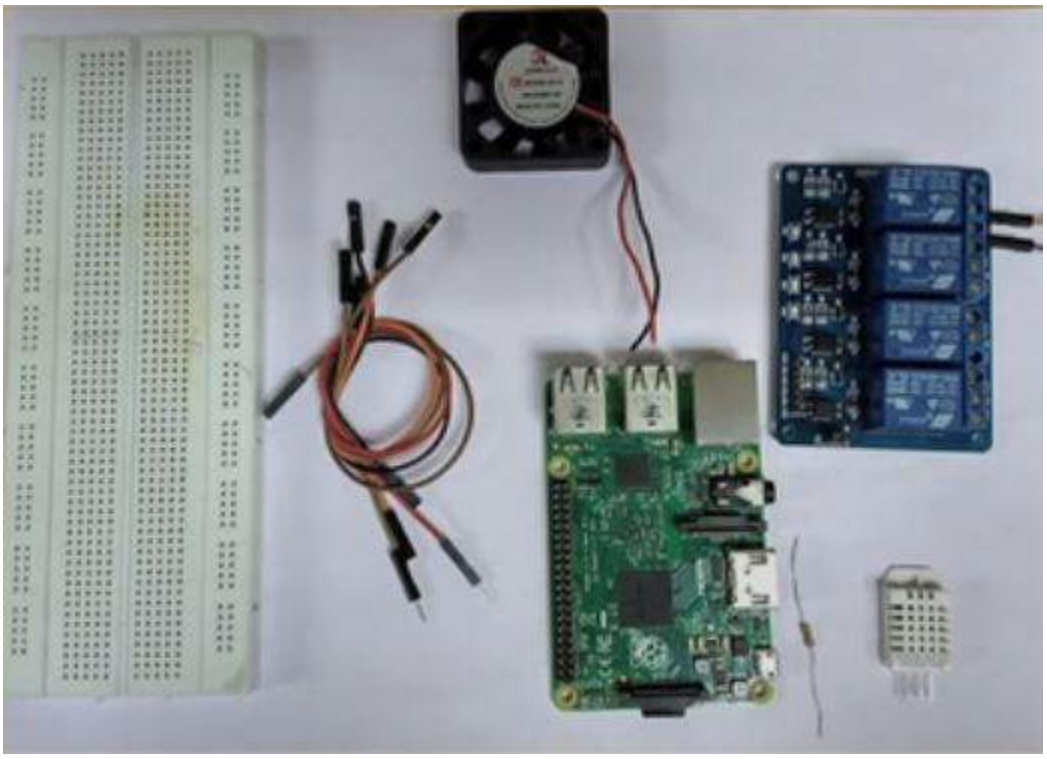
- Mechanical/Electro-mechanical device
- Converts energy into motion
- Mainly used to provide controlled motion to other components

System Overview

- Sensor and actuator interfaced with Raspberry Pi
- Read data from the sensor
- Control the actuator according to the reading from the sensor
- Connect the actuator to a device

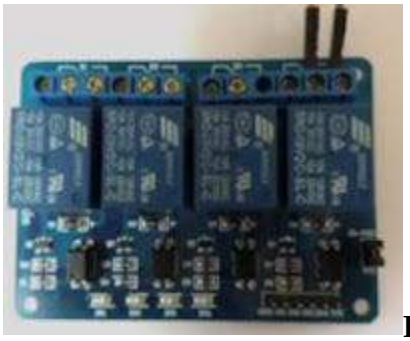
Requirements

- DHT Sensor
- 4.7K ohm resistor
- Relay
- Jumper wires
- Raspberry Pi
- Mini fan



DHT Sensor

- Digital Humidity and Temperature Sensor (DHT)
- PIN 1,2,3,4 (from left to right)
- PIN 1-3.3V - 5V Power supply
- PIN 2- Data
- PIN 3- Null
- PIN 4 - Ground

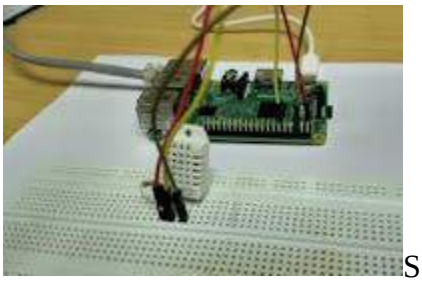
 **Relay**

Mechanical/electromechanical switch

3 output terminals (left to right)

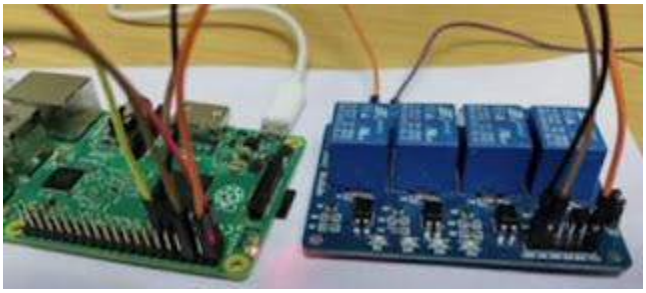
- NO (normal open) :
- Common
- NC (normal close)

Temperature Dependent Auto Cooling System

 Sensor interface with Raspberry Pi

- Connct pin 1 of DHT sensor to the 3.3V pin of Raspberry Pi
- Connect pin 2 of DHT sensor to any input pins of Raspberry Pi, here we have used pin 7
- Connect pin 4 of DHT sensor to the ground pin of the Raspberry Pi

Relay interface with Raspberry Pi



- Connect the VCC pin of relay to the 5V supply pin of Raspberry Pi
- Connect the GND (ground) pin of relay to the ground pin of Raspberry Pi
- Connect the input/signal pin of Relay to the assigned output pin of Raspberry Pi (Here we have used pin 11)

Adafruit provides a library to work with the DHT22 sensor

Install the library in your Pi -

- Get the clone from GIT

  → **git clone https://github.com/adafruit/Adafruit_Pythone_DHT.git**

- Go to folder Adafruit_Python_DHT

  → **cd Adafruit_Python_DHT**

- Install the library

  → **sudo python setup.py install**

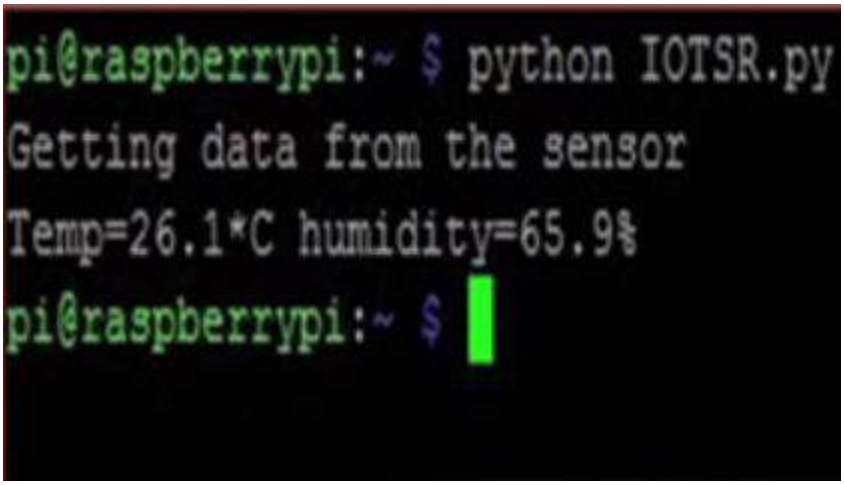
Program : DHT22 with Pi

```
import RPi.GPIO as GPIO
from time import sleep
import Adafruit_DHT                                  #importing the Adafruit library

GPIO.setmode(GPIO.BOARD)
GPIO.setwarnings(False)
sensor = Adafruit_DHT.AM2302                         # create an instance of the sensor type
print ('Getting data from the sensor')
#humidity and temperature are 2 variables that store the values received from the sensor

humidity, temperature = Adafruit_DHT.read_retry(sensor,4)
print ('Temp={0:0.1f}*C humidity={1:0.1f}%'.format(temperature, humidity))
```

Output :



Connection : Relay

- Connect the relay pins with the Raspberry Pi as mentioned in previous slides
- Set the GPIO pin connected with the relay's input pin as output in the sketch

  GPIO.setup (11,GPIO.OUT)

- Set the relay pin high when the temperature is greater than 30

```
if temperature > 20:

  print ('Temp > 20')

 GPIO.output (FAN,0)

  print('Fan on')

  sleep(5)

  print('Fan off ')

 GPIO.output(FAN,1)

else:

 GPIO.output(FAN,1)

 print ('Temp below max value.FAN OFF')
```

Connection : Fan

Connect the Li-po battery in series with the fan

- No terminal of the relay -> positive terminal of the Fan.
- Common terminal of the relay -> Positive terminal of the battery
- Negative terminal of the battery -> Negative terminal of the fan.

Run the existing code. The fan should operate when the surrounding temperature is greater than the threshold value in the sketch

What is Raspberry Pi?

Raspberry Pi is a microcomputer that we can connect with any keyboard, mouse or a monitor. Raspberry Pi is able to perform all tasks like creating word documents, spreadsheet, browsing, coding, games and much more. It is a tiny Linux computer that is used by many companies for performing all computer-based activities, learning and electronic projects. The Raspberry Pi uses a 32 bit ARM processor and was first developed by Raspberry Pi Foundation. The low cost makes it more popular among young people.

How does the Raspberry Pi works?

The memory card slot is used for inserting an SD card that acts as the hardware of Raspberry Pi. The USB port, HDMI port, and the audio/video port help us connect with a monitor, TV or any other device. This is how Raspberry Pi is capable of working anytime

and anywhere. The processor gives the correct speed for running our computer programs all the time

How is Raspberry Pi used IoT?

Raspberry Pi can be used as a platform to develop many Internet of Things project. It is simple to use Raspberry Pi because it uses Linux OS in a small card like a computer. It is also available in low cost. IoT, on the other hand, refers to a network of devices and software that help us to connect and exchange data. Raspberry Pi is easy to set up, so it is recommended for IoT.

What are the different components of a Raspberry pi board?

The Raspberry Pi board contains a 700 or 900 MHz processor with a minimum memory provision of 128 MB. It has an additional slot for memory card too. There is a graphics set up, USB port for connecting keyboard or mouse. Raspberry Pi comes along with an audio or video output option to connect your monitor. There is an HDMI port too for connecting with TV.

What is the NOOBS Software all about?

NOOBS or New Out Of The Box Software is used for initializing Raspberry Pi for the first time. This is the easiest way possible to install Raspberry Pi anywhere. It requires no additional access for network or any imaging software. To set up Raspberry Pi, we need the NOOBS zip file which is to be downloaded. After that we can insert an SD card, connect Monitor, keyboard or mouse and start using Raspberry Pi.

What are GPIO pins used in Raspberry pi boards?

GPIO is the acronym for General Purpose Input Output Pin. They are more like switches on the board that gives an output voltage when high and give no voltage when turned to low. This is used in the Raspberry Pi boards to make an interface between the Raspberry Pi and all the components of the board. This enables multiple interactions and makes internal properties of devices easily available on board.

Can Raspberry Pi be used as a server?

Yes, like we use Raspberry Pi for the desktop we can use it for the server as well. Raspberry Pi allows us to utilise it like a Web server that can help us create a simple site or even store data in the cloud. This can be accessed at any time. We can develop new templates without paying any monthly hosting fee. Raspberry Pi is thus an affordable web server too. This is very suitable for small business and for people who want to learn easy web languages.

What is the language used by Raspberry Pi?

Raspberry Pi uses Python as its official programming language. Python is one of the most user-friendly programming language used. It is also preferred by many companies for system development. Raspberry Pi helps us to quickly release our projects with Python. Raspberry Pi is preloaded with Python which has comprehensive syntax.

How is Raspberry Pi different from Arduino?

Raspberry Pi uses Linux OS and is a general purpose microcomputer. It is capable of running multiple programs at a time, while the Arduino is a simple microcomputer that is capable of running one program only.

Why do you think we must use the Raspberry Pi?

Raspberry Pi is ideal for creating projects. We can use it when we need computers but not advanced processing and hence save space. Raspberry Pi can be used as an alternative to a desktop or even use it as a photography tool at a low cost. There are options to create Google Home, light sensing switch and a retro gaming console. The Raspberry Pi has a wide range of uses that are essential in companies that idealize innovation

Are we likely to confront any overheating problems with Raspberry Pi?

No, Raspberry Pi is free of overheating issues and does not require a fan for cooling too. It is designed in such a way that it can withstand about 100 degree Celsius. This will help us finish projects in the safest way possible.

How can you measure power consumption used by Raspberry Pi?

The power consumption for Raspberry Pi varies from model to model. However, it is relatively low than a complete computer system. It can be measured using a multi-meter.

What are the generations of Raspberry Pi available?

Starting from Raspberry Pi Zero, there have been many improvements in the versions. Raspberry Pi 1, Pi 2, Pi 3 are available and A, A+, B indicate the power consumption.

Explain the architecture of Raspberry Pi and how it differs from a traditional computer system.

Raspberry Pi utilizes ARM-based architecture, differing from traditional x86 or x64 architectures found in most computers. It's a single-board computer (SBC) with all components – CPU, GPU, RAM, and I/O ports – on one circuit board, unlike traditional systems where these are separate entities connected via buses. Raspberry Pi uses System-on-Chip (SoC) design, integrating the CPU, GPU, and memory onto a single chip, Broadcom BCM2837 SoC, for compactness and efficiency. Traditional PCs use discrete components which allow for upgrades but increase size and power consumption. The Raspberry Pi also lacks certain features standard in traditional computers like storage drives, relying instead on microSD cards for booting and storage.

Difference between Arduino and Raspberry Pi

S No.	Arduino	Raspberry Pi
1.	In the year 2005, the classrooms of the Interactive Design Institute in Ivrea, Italy, first introduced the Arduino board.	In the year 2012, Eben Upton first introduced the Raspberry Pi device in February.
2.	Control unit of the Arduino is from the Atmega family.	The control unit of Raspberry Pi is from the ARM family.
3.	Arduino is based on a microcontroller.	While Raspberry Pi is based on a microprocessor.
4.	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.
5.	Arduino boards have a simple hardware and software structure.	While Raspberry Pi boards have a complex architecture of hardware and

S No.	Arduino	Raspberry Pi
		software.
6.	CPU architecture: 8 bit.	CPU architecture: 64 bit.
7.	It uses very little RAM, 2 kB.	While Raspberry Pi requires more RAM, 1 GB.
8.	It clocks a processing speed of 16 MHz.	While Raspberry Pi clocks a processing speed of 1.4 GHz.
9.	It is cheaper in cost.	While Raspberry Pi is expensive.
10.	It has a higher I/O current drive strength.	While Raspberry Pi has a lower I/O current drive strength.
11.	It consumes about 200 MW of power.	While it consumes about 700 MW of power.
12.	Its logic level is 5V.	Its logic level is 3V.
13.	It does not have internet support.	It has inbuilt Ethernet port and WiFi support.
14.	It has higher current drive strength.	It has lower current drive strength.
15.	Some of the applications of Arduino are traffic light countdown timer , Weighing machines , etc.	Some of the applications of Raspberry Pi are Stop motion cameras , Robot Controllers , Game Servers.
16.	Operating systems are required in Arduino.	Operating System is required in Raspberry Pi.

S No.	Arduino	Raspberry Pi
17.	Two tiny cores Arduino with 32 Mhz	Single core and 700 MHz

What are advantages and disadvantages of Raspberry Pi?

Advantages of Raspberry Pi:

- This microcomputer is useful for small business that run on a lower budget to use their product or to invent new technology that embeds the product. Small business owners can use it to automate any small task, i.e.; such as using the Pi to run a website or use it as a small database and media server.
- The product does not require user to have extensive programming experience since it is aimed for the younger generation to learn about programming. Python, the programming language i.e.; Pi uses, is a smaller amount complex than other languages available. It has better code readability and allows the user to type concepts using fewer number of lines. Python also has a automatic memory management function.
- The product gives a lot of room to experiment and turn it into something else i.e.; entirely different. The SD cards on the board can be easily switched, i.e.; which allows to change the functions of the device without spending a lot of time re-installing the software.
- The Raspberry Pi is perfect for adaptive technology and it is able to display images or play videos i.e.; at high-definition resolution to building systems such as prototyping embedded systems. This product makes it possible to build complex and effective at a cheaper price.
- The product is efficient and i.e.; provides an ethical alternative to small businesses. This small card sized product i.e.; makes it easy to recycle and does not release as much carbon dioxide emissions into the environment, i.e.; unlike big servers that need lots of energy and extensive cooling systems.

Disadvantages of Raspberry Pi:

- It does not replace the computer, and the processor is not as fast. It is a time consuming to download and install software i.e.; unable to do any complex multitasking.
- Not compatible with the other operating systems such as Windows.
- This is fit for those who want a gadget that they can tailor to their own needs and tastes, i.e.; not for those that just wants to urge a job done fast. Business owners need to consider the extra hassle if is worth it.

- This product not be useful for bigger business that already have big servers, i.e.; which would already do everything that the Raspberry Pi does. So, it would not be worth and it take time to get to put it together.

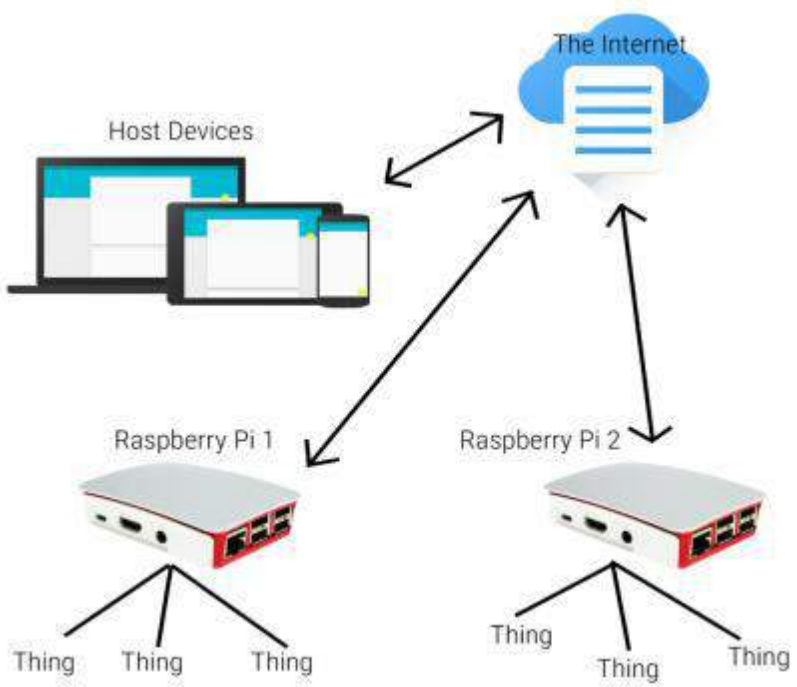
UNIT-IV

Implementation of IoT with Raspberry Pi

BASIC IMPLEMENTATION OF IOT USING RASPBERRY PI

The basic implementation of IOT includes usage of a host device, a Remote Controllable Device and connectivity between them. In this paper, the host device can be a computer or a mobile phone and the remote controllable device is a Raspberry Pi, which executes the commands given by the master host. The implementation mechanism can be understood by the following figure

Fig 1: Block diagram of implementing the Internet of Things



The implementation requires a close association with both hardware and software

2.1 HARDWARE IMPLEMENTATION: The system that implements the Internet of Things includes clusters of hardware components that we are familiar with. Firstly, we need a host like a Personal Computer or a mobile phone that can be used to pass commands to a remotely operable device. As the brain of the system we are using a Raspberry Pi that can be used to control and obtain a desired result from a device. The "things" that we use here are basically day-to-day objects like a bulb, a fan, a washing machine etc., Our intention is to show the operation of the Internet of Things in a concise way. As the Raspberry Pi is more like a compact computer itself, it cannot control "things" directly. It needs an interface to communicate the with them. Fortunately, Raspberry Pi comes with a 40-pin GPIO set that could efficiently be utilized to communicate with the "things". As we need an interface between them, a "Daughter Board" is to be designed. This Daughter Board will enable us to dim and glow a light source. Switch ON/OFF electrical devices and receive feedback from sensors

. **2.2 SOFTWARE IMPLEMENTATION:** Hardware without proper software is nothing but a piece a brick. When it comes to Raspberry Pi, an OS must be installed to control and configure it. And in the case of the Daughter Board, python scripts are to be coded to work with the "things". We have, a communications platform for IOT devices that

enables device setup and user interaction from mobile devices and the web, can be used to accomplish communication between Host device and the Raspberry Pi.

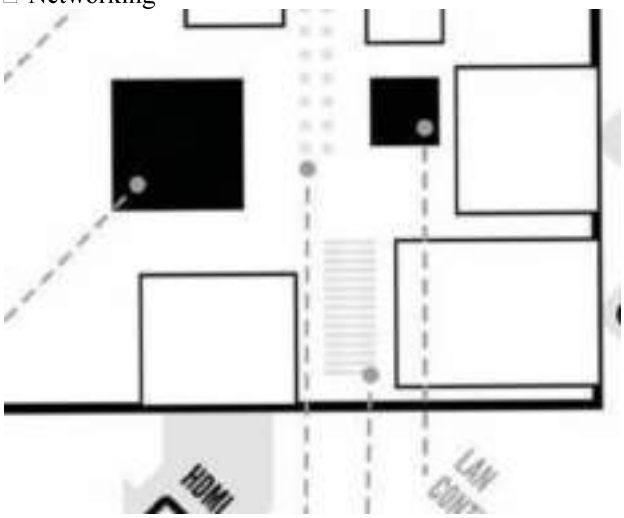
2.3 OPERATIONS TO DEMOSTRATE The operations to demonstrate include:

- Remotely dim and glow a light source (an LED).
- Switch electrical devices ON/OFF based upon their state remotely (Staircase Control).
- Receive feedback from sensing elements connected to the daughter board.

III. PURPOSE OF SELECTING A RASPBERRY PI

Since a Raspberry Pi is basically a mini, credit card-sized computer, These are the few positive aspects that we got to learn while working with the Raspberry Pi, they are:

- low power consumption
- No moving parts
- Compactness
- Cost effective
- No noise
- Status lights
- Built-in HDMI
- The GPIO ports
- Remote control
- Over-clocking capability
- Multiple uses
- Networking



IV. INTERFACING MODULES TO RASPBERRY PI

In this paper we have presented the

- Remote controlling the dimmer LED.
- Remote controlling the stair-case switched devices connected to the daughter board, along with their feedback.
- Interfacing the sensors and Sensing the state of the several sensors as feedback.
- Interfacing the camera module.

And power saving by solar energy

Solar powered Raspberry Pi – Instead of using a power supply adaptor we can charge the Raspberry Pi from the stored solar energy.

Software defined Networking(SDN)

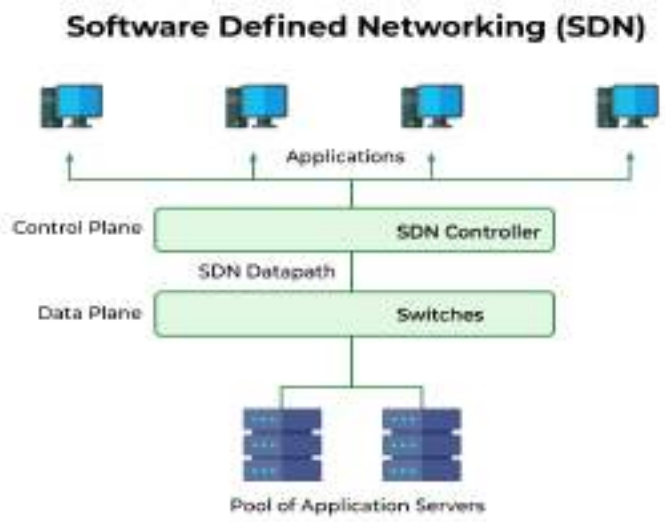
- SDN stands for Software Defined Network which is a networking architecture approach. It enables the control and management of the network using software applications. Through Software Defined Network (SDN) networking behavior of the entire network and its devices are programmed in a centrally controlled manner through software applications using open APIs.

  To understand software-defined networks, we need to understand the various planes involved in networking.

  1. Data Plane
  2. Control Plane

  Data plane: All the activities involving as well as resulting from data packets sent by the end-user belong to this plane. This includes:
  - Forwarding of packets.
  - Segmentation and reassembly of data.
  - Replication of packets for multicasting.

  Control plane: All activities necessary to perform data plane activities but do not involve end-user data packets belong to this plane. In other words, this is the brain of the network. The activities of the control plane include:
  - Making routing tables.
  - Setting packet handling policies.



Software Defined Networking

How does Software-Defined Networking work?

Software-Defined Networking (SDN) is a network architecture that allows for the centralized control of network traffic flow. SDN controllers use protocols like OpenFlow to communicate with network devices and determine the best way to route traffic. This allows for more flexibility and easier management of complex network environments.

. Why should we use Software-Defined Networking?

There are many reasons to use Software-Defined Networking (SDN). SDN can provide greater flexibility and agility in your network, since it allows you to programmatically control the flow of traffic. This can be especially useful in dynamic or cloud environments where network conditions can change rapidly. SDN can also help to improve network security, since it allows you to centrally manage and monitor network traffic. Finally, SDN can help to reduce network costs by simplifying the network infrastructure and making it easier to manage.

Why SDN is Important?

- **Better Network Connectivity:** SDN provides very better network connectivity for sales, services, and internal communications. SDN also helps in faster data sharing.
- **Better Deployment of Applications:** Deployment of new applications, services, and many business models can be speed up using Software Defined Networking.

- **Better Security:** Software-defined network provides better visibility throughout the network. Operators can create separate zones for devices that require different levels of security. SDN networks give more freedom to operators.
- **Better Control with High Speed:** Software-defined networking provides better speed than other networking types by applying an open standard software-based controller.

In short, it can be said that- SDN acts as a "Bigger Umbrella or a HUB" where the rest of other networking technologies come and sit under that umbrella and get merged with another platform to bring out the best of the best outcome by decreasing the traffic rate and by increasing the efficiency of data flow.

Where is SDN Used?

- Enterprises use SDN, the most widely used method for application deployment, to deploy applications faster while lowering overall deployment and operating costs. SDN allows IT administrators to manage and provision network services from a single location.
- Cloud networking software-defined uses white-box systems. Cloud providers often use generic hardware so that the Cloud data center can be changed and the cost of CAPEX and OPEX saved.

Components of Software Defining Networking (SDN)

The three main components that make the SDN are:

1. **SDN Applications:** SDN Applications relay requests or networks through SDN Controller using API.
2. **SDN controller:** SDN Controller collects network information from hardware and sends this information to applications.
3. **SDN networking devices:** SDN Network devices help in forwarding and data processing tasks.

SDN Architecture

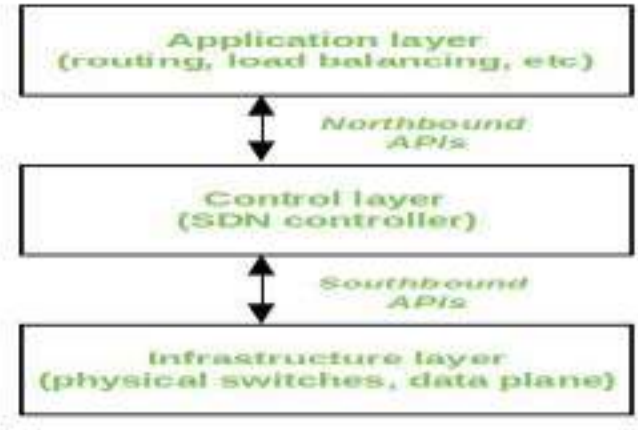
In a traditional network, each switch has its own data plane as well as the control plane. The control plane of various switches exchange topology information and hence construct a forwarding table that decides where an incoming data packet has to be forwarded via the data plane. Software-defined networking (SDN) is an approach via which we take the control plane away from the switch and assign it to a centralized unit called the SDN controller. Hence, a network administrator can shape traffic via a centralized console without having to touch the individual switches. The data plane still resides in the switch and when a packet enters a switch, its forwarding activity is decided based on the entries of flow tables, which are pre-assigned by the controller. A flow table consists of match fields (like input port number and packet header) and instructions. The packet is first matched against the match fields of the flow table entries. Then the instructions of the corresponding flow entry are executed. The instructions can be forwarding the packet via one or multiple ports, dropping the packet, or adding headers to the packet. If a packet doesn't find a corresponding match in the flow table, the switch queries the controller which sends a new flow entry to the switch. The switch forwards or drops the packet based on this flow entry.

A typical SDN architecture consists of three layers.

- **Application layer:** It contains the typical network applications like intrusion detection, firewall, and load balancing
- **Control layer:** It consists of the SDN controller which acts as the brain of the network. It also allows hardware abstraction to the applications written on top of it.

- **Infrastructure layer:** This consists of physical switches which form the data plane and carries out the actual movement of data packets.

The layers communicate via a set of interfaces called the north-bound APIs(between the application and control layer) and southbound APIs(between the control and infrastructure layer).



SDN Architecture

security issues in software-defined networks?

security issues are a major concern in software-defined networks. Because these networks are so new, there are still many unknowns when it comes to potential vulnerabilities. Additionally, because software-defined networks rely heavily on software, there is a greater potential for malicious actors to exploit vulnerabilities in the code. As these networks become more widespread, I believe that it will be increasingly important to address these security concerns.

How can you improve latency in software-defined networks?

There are a few ways to improve latency in software-defined networks. One way is to use a lower-latency network fabric, such as Ethernet. Another way is to use Quality of Service (QoS) to prioritize latency-sensitive traffic. Finally, you can use traffic engineering to route traffic in a way that minimizes latency.

What are the central tasks of the Control Plane with its Network Controller?

The central tasks of the Control Plane with its Network Controller are the following:

- The management of the different network components
- The configuration of the hardware
- The configuration of network security relevant security specifications
- The management of access to the network components
- The control of data forwarding by the hardware

Creating the routing specifications for forwarding the data packets to the desired destination

What is the advantage of central intelligence over distributed intelligence in SDN?

Software defined networking moves away from the concept of distributed intelligence and the use of different operating systems. In the SDN, the intelligence of the network is moved to a central instance and the configuration of individual devices or operating systems is superfluous.

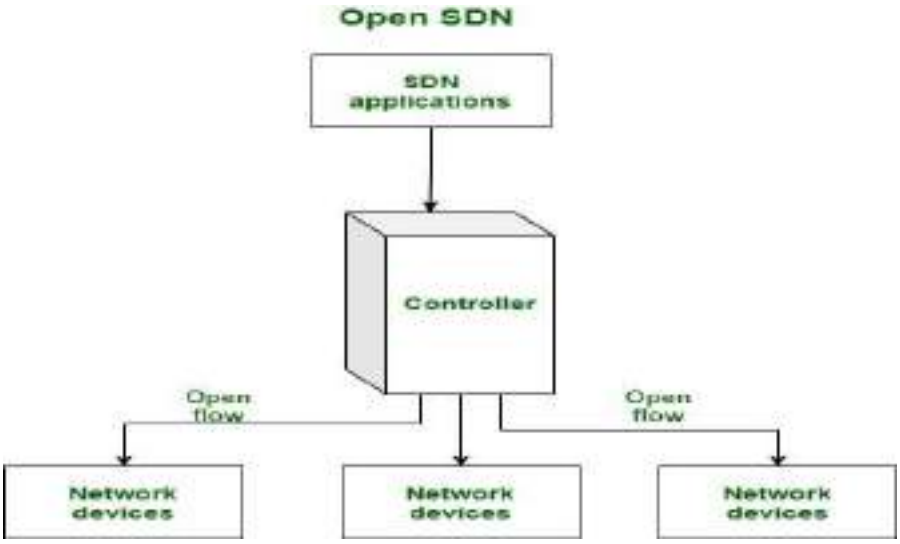
The goal of the concept is to reduce network maintenance and administration while increasing intelligence and flexibility. In addition, the hardware can concentrate on its actual task, the data forwarding, and is relieved of control and management functions.

Different Models of SDN

There are several models, which are used in SDN:

1. Open SDN
2. SDN via APIs
3. SDN via Hypervisor-based Overlay Network
4. Hybrid SDN

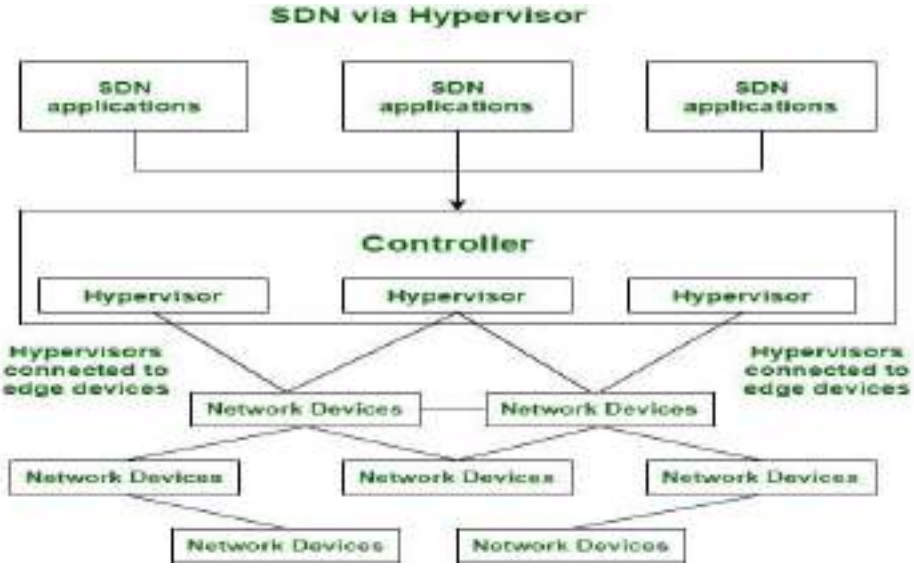
1. Open SDN: Open SDN is implemented using the OpenFlow switch. It is a straightforward implementation of SDN. In Open SDN, the controller communicates with the switches using south-bound API with the help of OpenFlow protocol.



Open SDN

2. SDN via APIs: In SDN via API, the functions in remote devices like switches are invoked using conventional methods like SNMP or CLI or through newer methods like Rest API. Here, the devices are provided with control points enabling the controller to manipulate the remote devices using APIs.

3. SDN via Hypervisor-based Overlay Network: In SDN via the hypervisor, the configuration of physical devices is unchanged. Instead, Hypervisor based overlay networks are created over the physical network. Only the devices at the edge of the physical network are connected to the virtualized networks, thereby concealing the information of other devices in the physical network.



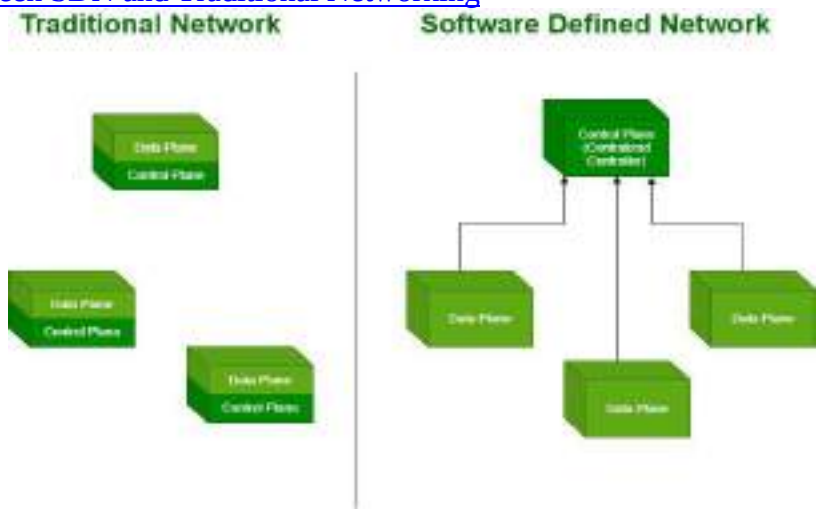
SDN via Hypervisor-based Overlay Network

4. Hybrid SDN: Hybrid Networking is a combination of Traditional Networking with software-defined networking in one network to support different types of functions on a network.

Difference between SDN and Traditional Networking

Software Defined Networking	Traditional Networking
Software Defined Network is a virtual networking approach.	A traditional network is the old conventional networking approach.
Software Defined Network is centralized control.	Traditional Network is distributed control.
This network is programmable.	This network is nonprogrammable.
Software Defined Network is the open interface.	A traditional network is a closed interface.
In Software Defined Network data plane and control, the plane is decoupled by software.	In a traditional network data plane and control plane are mounted on the same plane.

differences between SDN and Traditional Networking



Difference between SDN and Traditional Networking

Advantages of SDN

- The network is programmable and hence can easily be modified via the controller rather than individual switches.

- Switch hardware becomes cheaper since each switch only needs a data plane.
- Hardware is abstracted, hence applications can be written on top of the controller independent of the switch vendor.
- Provides better security since the controller can monitor traffic and deploy security policies. For example, if the controller detects suspicious activity in network traffic, it can reroute or drop the packets.

Disadvantages of SDN

- The central dependency of the network means a single point of failure, i.e. if the controller gets corrupted, the entire network will be affected.
- The use of SDN on large scale is not properly defined and explored.
    - **Software-defined Networking in IoT is an** architecture that easily abstracts many different layers of a network[1]**. It offers** software-based controllers to manage hardware infrastructure and traffic flow on a network effectively[2]**. SDN aims to** improve network control by enabling enterprises and service providers to respond quickly to changing business requirements[1]**. SDN is useful to** manage and control IoT network, wireless sensor nodes, and network performance[3]**.**

requires change in entire network infrastructure to implement SDN protocol and SDN controller. Hence it requires complete reconfiguration of the network. This increases cost due to reconfiguration.

➨Staff need to be trained.

➨New management tools need to be procured and everyone should be trained to use it.

➨Security is a big challenge in SDN.

➨Single point of failure.

- 

Data handling and analytics in the Internet of Things (IoT) refers to the process of collecting, storing, and analyzing data generated by IoT devices and systems. IoT devices are connected

devices that generate large amounts of data, including sensor data, location data, and device usage data. This data is used to monitor and control the devices, as well as to extract insights and make informed decisions.

To handle and analyze IoT data effectively, it is important to have robust data management systems and tools in place. These may include:

Data storage systems: IoT data needs to be stored in a way that allows it to be accessed and analyzed efficiently. This may involve using data lakes, data warehouses, or other types of data storage systems.

Data processing and analytics tools: IoT data is often generated at a high volume and velocity, and may be structured or unstructured. To analyze this data effectively, it is important to have tools and platforms that can process and analyze the data in real-time or near-real-time. This may involve using stream processing systems, data visualization tools, or machine learning algorithms.

Data security and privacy: IoT data often contains sensitive or personal information, and it is important to ensure that this data is protected from unauthorized access and misuse. This may involve using secure data transmission protocols, data encryption, and other security measures.

By handling and analyzing IoT data effectively, organizations can gain insights, optimize operations, and improve decision-making.

UNIT-V

Role of Cloud Computing in IoT

Innovations lead to a continuing expansion of technologies. IoT and cloud computing are now two upcoming internet technologies that are closely linked, with one providing the groundwork for the success of the other.

Cloud computing is helping the IoT in getting success. Cloud is a big factor in the success of IoT. As cloud enable user to carry and access all thing over internet without any storage, IoT is related with cloud computing. Future users of these technologies will gain a number of benefits. As was already mentioned, cloud computing allows for scalability in the delivery of applications and software as a service by enabling businesses to manage and store data across cloud platforms.

What is IoT?

In, IoT we do not need interaction between human or between human and computer. We can communicate data over a network of linked devices like objects, computers, or digital devices.

A heart monitor implant is an example of IoT usage. An in-built sensor that gives alert to driver on path danger in car is also example of IoT. IoT device is an object which can transfer or receive the data across network and can have IP address. IoT object should be capable of having IP address.

Increased data output has led to the growth of IoT. Due to the Internet of Things Cloud Service's excessive communication between cheap sensors in the IoT, there will soon be billions of connected machines and devices joining human users.

Why Cloud Computing Is Essential For IoT?

As a result of cloud computing, storage options for personal and professional use have undergone tremendous change. Data is also available from a distance, thanks to cloud solutions' scalability and data dynamics. It has thus shown to be a successful solution for data transfer across internet channels and via specialized direct links, depending on the organization's needs.

The cloud is an excellent IoT enabler that satisfies the data-driven requirements of the company. Cloud also offers technology framework. Using that framework, we can develop better IoT devices.

Speed and scale are two essential aspects of cloud computing, and they work in unmatched harmony with IoT networking and mobility. So, user can benefit more by combine use of cloud computing and IoT. Unquestionably, some factors show that the cloud is necessary for the success of IoT, and here are some of them.

Cloud Functions as a Distant Computing Power

On-premises infrastructure reliance is no longer a viable option. As usage of cloud and IoT devices are increasing day by day, we generated large amount of data. We need to process that data quickly using big data. The advantage of having a tonne of storage capacity in this situation comes from the cloud. As we are heading towards 5G from 4G, cloud computing also enables developer more speed in getting access to data.

IoT Data is More Secure And Private as a Result of Cloud Computing.

IoT involves significant data generation. And when you work with data, the data security and data privacy become issue. IoT also makes use of mobility. Cloud uses advance encryption algorithms and authentication. Which enable cloud to provide its user high security.

No Requirement For Hosting on-premises

For IoT devices, plug-and-play hosting services are necessary. This will become quite expensive due to plug-and-play hosting services. This will cost more to organizations. This type of hosting services needs hardware system. Due to the combined power of cloud computing and IoT, you do not need to depend on substantial machinery. As cloud computing infrastructure make it ready to use without having hardware storage device set-up offline. This makes it easy for IoT hosting organizations.

Improved Device-To-Device Communication

We can use cloud technology for the communication using the IoT. Smart device can easily connect with each other using IoT APIs. It also makes internal communication between devices fast and easy.

Less Cost of Ownership

While preventing enterprises from putting up the infrastructure, cloud technology also provides many resources. As a result, it saves lot of money on infrastructure construction. Additionally, because there is no idea of local systems, hardware, and software in the cloud, the IT teams are abler to concentrate on their regular tasks.

Program For Business Continuity

Business continuity is guaranteed by cloud computing, even if unexpected disasters occur while it is being used. There is no danger of data loss because data is maintained on additional distinct servers, which is considerably more crucial in the case of IoT-based architecture.

IoT innovations with low entry barriers require hassle-free hosting options. As a result, cloud computing in IoT is a suitable solution. IoT players can use the power of distant data centers due to cloud computing without requiring on-premises gear and software. IoT cloud computing is the best option financially because users need to adhere to the pay-as-you-go concept. It also saves a tonne of money upfront.

This helps businesses can launch massive IoT projects with ease. This removes many obstacles to entry for the majority of IoT-based organizations.

Communication Between Devices

By using cloud computing in proper way, IoT devices can communicate with each other seamlessly. As a result, connected devices and smart devices can communicate with various reliable APIs. In this way, networked technologies are made possible by cloud computing.

Conclusion

Every firm works individually to keep up with this evolving technology's pace. IoT technology is predicted to connect billions of devices, and the information that these devices produce will be challenging to handle and process using the current methods.

Sensor-Cloud for IoT

The technological advancement of wireless sensor networks (WSNs) empowers numerous real-life applications such as target tracking, battlefield monitoring, tele monitoring, ubiquitous monitoring, and several other applications. However, these WSNs are single-user centric. On the

other hand, for certain applications such as environment monitoring and tele monitoring, the data from the single sensor networks can be shared among multiple applications. In these scenarios, **Sensor-Cloud** can play a huge role to provision the **Sensors-as-a-Service** (**Se-aaS**) platform, while satisfying the requirements of multiple applications by forming virtual sensors in a cloud platform. The sensor-cloud architecture has been conceived as a potential solution for multiorganization WSN deployment and data access.
We can study the various aspects of the sensor-cloud architecture such as **Virtualization, Caching, QoS, and Pricing**. We can also develop and analyze various models to characterize the performance of sensor-cloud.

IOT Application in Smart Homes and Smart Cities

The two most prospering domains with the help of the internet of things are smart homes and smart cities. Iot in smart homes and smart homes is increasing an urbanised world and improving energy efficiency.

Smart homes

Smart homes connect the devices and home appliances together in order to improve efficiency. These interconnect devices under one roof such as geysers, ovens, smart TVs, thermostats and allow communication between the devices.

IoT connects these devices to the internet and these devices constantly send and receive information about the surroundings. The devices send the data to giant cloud servers mostly via IoT gateways. Smart homes allow users to remotely monitor their devices via mobile applications. Applications of smart homes also include home security systems, smart thermostats and smart refrigerators.

Smart homes cities integrate with the entire cities by creating and controlling a network.

Smart thermostats

The Smart homes include thermostats that are capable of sensing and controlling the temperature. This controls the flow.

Location-based smart devices

Smart devices can track your location and instantly send messages to other devices to operate. For example, your smart thermostats can track your location from a smart car and switch on the ACs before you arrive

Voice-enabled devices

These devices can interpret human voices and convert these into text that is interpreted by the machines. Machines then perform the necessary task. Examples include alexa and siri.

Smart security systems

Security systems that are made using IOT use features such as facial recognition, iris scanners and other security modes.

Facial recognition

This is one of the most propelling uses of the IoT. facial recognition models track the features of the face such as the yes, noses, chin and lips to predict the output of the person. Based on the accuracy the machine is further trained or sent for development.

Motion detection

Similar to facial detention but here the sensors record the movements or changes on the surroundings. These behavioural changes are then sent to the model for further analysis.

Biometric access control

Biometrics have become the backbone of any organisation's security systems. They are easy to manage and hassle-free. The employees and the staff members simply record their thumb or iris impressions on arrival and the machine marks their attendances. This model saves time and cost.

Benefits for smart homes

Smart homes allow you to add improvised functionality into regular homes. These make lives easier. For instance, smart vacuums clean up your entire home when you are away and smart refrigerators stock up your fridge and manage your die charts. Smart homes offer security, stability, flexibility and peace of mind.

Challenges in smart home systems

- Although there is no doubt that smart homes immensely make the lives of humans easier. It comes with the following challenges
- It becomes difficult to integrate the system when each one of the IoT devices is sold by different vendors

- Many times the IOt devices are not compatible with each other. Each of the devices must be connected to the WiFi in addition to being connected to devices such as Google Assistant.
- The prices of setting up smart homes are highly expensive and so the customers may not prefer to make such a hefty investment when regular homes just work fine without IoT.
- Security is still a big threat in smart homes. Smart homes must have a powerful farewell in order to avoid system hacking

Smart city

The internet of things technology allows smart cities to stay connected worldwide. Smart cities are driven by technology to make smart cities more safer, modern and reliable. It leads to a boost in the economy.

Smart cities include services, devices and technology that work with IoT. These offer services to improve water, electricity, roads, transportation, public areas, buildings and digital services such as broadbands. These replace regular machines with smart machines that contain sensors to sense and collect data and actuators generate efficient responses on the basis of the data incoming from sensors.

Smart cities make the lives of the citizens more comfortable and easy. Smart cities with traffic sensors have decreased road accidents and deaths to a large extent.

Features of smart cities

1. Smart water management

Smart eater management uses the internet of things to gain access to real-time information about the water systems and the water bodies. This allows humans to better manage their water resources and their requirements.

2. Smart lighting

Smart lighting contains sensors that can sense when the night falls and automatically switch on the street lighting. These sensors can also sense when the sun comes and they automatically switch off. Smart lighting also is capable of reaching with the help of solar energy.

3. Smart traffic management

Smart traffic management integrates with smart cities in order to better control and manage traffic in the cities. It offers different routes to regulate the traffic and it sends help quickly in case of a road incident. These systems are in constant contact with the traffic police and deliver the officers with real-time updates.

4. Smart parking

Smart parking is a technology that makes use of a combination of efforts by iOt devices and humans. This allows drivers to locate empty spots in parking areas, detect the cars around them and alert the drivers in case they are blocking someone elses driveways.

5. Smart waste management

Smart waste management involves the use of sensors that detect full garbage bins to notify the city authorities. These management systems collect and store data over the course of years to create better drinking routes and reduce the cost of operation.

6. Smart policing

Smart policing is one of the important features of smart cities. It involves the use of IOT based technologies to manage the citizens of the city. These technologies collect data about the cities in order to better manage the city. For example, cameras in local areas can provide data about the areas with more burglary cases in order for the police to take the next step of actions.

Benefits of Smart Cities

Smart cities have revolutionized the lives of communities. Sensors can now sense the percentage of pollutants in the air. Sensors send data about the traffic in a locality to manage the city.

Smart homes in smart cities

Smart homes in smart cities are going to become a much more common occurrence as housing is one of the most important features of any city.

These are capable of reducing the cost of living and the cost of construction. As we move into the near future we are likely to notice that smart homes connect to smart cities to provide further benefits to the entire city. These may sound like an alien topic to discuss as of now. But as the

year moves by and the technology develops into a higher level we are likely to notice inter-connected smart cities and smart homes on a wider scale and network.

IoT Smart homes for smart citizens

A city with smart homes does not necessarily mean expensive houses with tight security. However, it is necessary to have some security on personal and public networks so the city is safe from crimes such as theft.

By ensuring devices are protected we ensure the fundamental growth of the city. We can monitor our personal assets remotely and this is all possible due to the internet of things. Smart homes help in saving time and costs for example thermostats manage the temperature of your home and control your expenses. Similarly, if we apply this concept on a city level we can reuse and save energy to a much larger extent.

Smart cities are much safer. These include smart traffic control, pollution control, smoke detectors, security cameras near every public area, and so on. Smart security systems reduce crime rates in any city by a large percentage.

What is a Smart Grid?

Electric grids are the complex system of networks that deliver energy from its production origin, like power plants, to users such as residential consumers and businesses. In the United States, the traditional electric grid was built over a century ago and relies on a one-way flow of electricity from source to destination.

However, technology is now changing the way energy is produced, stored, and saved on the grid—and opening the door to a burgeoning smart energy infrastructure. The addition of intelligent Internet of Things sensors, microgrids, digitization, distributed renewable energy sources, and automation is forming the new smart energy ecosystem, of which the smart grid is part.

In the same way the Internet facilitates a flow of information and data between computers tapped into a single network, the smart grid system is powered by a web of interconnected devices.

But digitizing and automating energy communication signals isn't the only thing the smart grid does. The new smart energy ecosystem also brings about a fundamental shift in the transmission of energy. Consumers, who were previously limited as recipients of energy, are now able to locally generate and store energy themselves at the edge—via commercial wind turbines and solar farms, but also through consumer solar storage systems in residential settings.
As Forbes put it: "Now, as consumers become producers of energy due to maturing solar panels, wind turbines, and other sources of energy, the power flow is 2-way."

As we'll see, IoT applications through the smart grid and overarching smart energy infrastructure are poised to change the way energy solutions are conceived—both now and in the future.

How IoT Makes the Smart Grid 'Smart' - From Open to Closed Loops

Within the confines of the traditional grid, electric utility providers have little insight into how consumers actually use electricity. The one-way flow of a traditional grid is purely demand-based—when there's an uptick in demand for electricity, operators send more to the grid.

Smart grids offer something radically different: **a bi-directional flow of information between consumers and utility companies**.

At Smart Energy Summit 2022, Dr. Kenneth Wacks shared that the promise of the smart grid lies in helping infrastructure **evolve from an open loop to a closed loop**. By adding data collection capabilities at the edge—which includes **smart meters, smart home technologies, EV chargers, solar/wind farms, and more**—usage and condition data can be shared across the value chain. This makes the grid more resilient because grid operators can detect outages and issues without relying on customer complaint volume to tell them something is wrong.

Similar to traditional meters, smart meters record and store information related to electricity usage. However, smart energy solutions take this to the next level by using wireless networks to send this information directly back to the energy supplier. This approach provides a much more nuanced view of the way energy is used. With smart solutions deployed from the edge all the way back to the utility, the **entire value chain** will gain insight into usage patterns that can change based on time-of-day, seasonality, and other factors.

In this way, smart energy solutions help both the utility provider and consumer to make more informed choices. Utilities can better anticipate energy needs, and even provide consumers with incentives that save both of them money.

How IoT Can Enable Smart Energy Solutions That Strengthen Smart Grids — Use Cases

In general, the IoT is a collection of internet-enabled devices enabling information collection, data pipelines, and the real-time transmission of that information between those devices and other people.

Smart grids represent the application of IoT technology in the energy sector. When done well, smart grids resolve several problems associated with traditional grids: outages, security concerns, high carbon emissions, and other factors.

The following list includes references to specific solutions incorporating IoT and smart grid applications.

Smart Meters

Advanced metering infrastructure is one of the key components of smart grid technology, and smart meters are the devices that bring the solution to life.

Smart metering works by providing a line of bi-directional communication between the devices themselves and the utility with the purpose of gathering, disseminating and analyzing user energy consumption data.

The sort of advanced metering solutions provided by smart meters are a vast improvement over automatic meter reading, which involve a one-way communication that limits the potential for improvements to future analysis. With automatic meter reading, data is collected, sent, and analyzed by the utility after the energy usage event.

With advanced metering, however, the insights recorded and communicated by the smart meter can be implemented by automated processes in real-time.

In that sense, smart meters can help utilities:

- Provide real-time alerts around meter or grid damage or outages
- Modify pricing and supply on the fly based on data insights
- Monitor and control power quality
- Increase energy saving
- Install software updates

Solar Farm Monitoring

IoT technologies like smart meters can also help individuals and companies alike better the efficiency of their solar farms.

Solar farms are not only great for ROI—they also make a big difference toward the reduction of Co2 emissions. Iot-based technologies like smart grids take this a step further by helping solar farms improve operations.

Examples include:

- Improving predictive analysis by collecting and analyzing yield data while adjusting for variables including time of year, weather, and individual panel performance
- Streamlining maintenance by attaching monitors to individual panels providing feedback on performance and structural deficiencies in real-time
- Getting more out of each panel by optimizing for factors like tilt angle and direction

IoT-Based Electric Vehicle Charging

Replacing vehicles that run on fossil fuels with electric vehicles (EVs) is one of the key indicators for reducing carbon emissions in years to come. But the explosive growth in the EV market presents its own set of unique challenges, including but not limited to the charging infrastructure needed to support millions of new EV drivers and their vehicles.

An IoT smart grid–based approach to EV charging can alleviate the pressure from one of its biggest challenges: identifying and coordinating optimal charging strategies for drivers.

In one use case, smart grids deployed to individual EVs can continuously monitor charge levels over the course of a journey. Simultaneously, these monitors connect to a GPS network of other charging stations. The goal? An EV assistant that can recommend the optimal time and place for refueling based on a variety of competing factors, including:

- The EV's charge level
- The EV's location and destination
- Location of available charging stations
- Availability and business of nearby charging stations

IoT-based assistive technology for EV charging could accelerate the adoption of EVs for both consumer and commercial uses—contributing to wider goals related to emissions reduction.

Battery Monitoring Systems

The switch to renewables based on sources like wind and solar energy can leave businesses and consumers alike grappling with the inherent variability these sources introduce.

Batteries are increasingly used for excess energy storage. The excess energy can be redistributed to others on a grid. But when batteries are under or overcharged with energy, it not only decreases performance—it can lead to a diminished battery life cycle for the business or consumer relying on it to effectively store energy.

Smart systems that monitor a battery's state-of-charge (SOC) can help prevent premature failure due to under or overcharging.

How Smart Grids Benefit from IoT-Enabled Capabilities

The IoT supports the technology and communication required to make "smart grids" smart.

In the context of the smart grid, IoT has concrete applications for monitoring electricity generation, gauging intelligent power consumption, managing energy efficiency, and much more.

Below, we break down some of the key benefits and use cases for IoT in the smart grid.

Prevention of Energy Theft

The energy sector loses billions of dollars in value due to fraud each year, resulting in higher prices for consumers and increased taxes for taxpayers supporting government energy subsidies.

It's been estimated that as much as $100 billion is lost due to energy theft and other non-technical losses every year.

Energy theft can be the result of direct theft—consumers connecting directly to the main supply and bypassing metering efforts—or by tampering with meters. Before the introduction of advanced metering infrastructure, it was more difficult to detect fraud without making physical inspections of units or auditing records.

Now, IoT solutions exist that bring theft detection and prevention into the 21st century. By monitoring key indicators, such as energy availability and consumption, down to the meter in real time, utilities can help their consumers save money by correcting non-technical losses in metering and billing.

Remote Control

Remote shut-off features find a natural application in combating energy theft, as they allow utilities to automatically restrict access to energy and even cut off services in the event of a delinquent account.

But the practical uses for remote control IoT functions apply to much more than energy utilities. Companies and consumers alike can use remote control functionality to control remote devices, and even entire systems, such as industrial air quality monitors, smart home products, and other devices with smart capabilities.

For users reliant on far-flung grids, the ability to toggle remote assets on and off or otherwise change their states can be a huge time and cost-saving measure—especially if the alternative is sending out a technician.

Another key feature in an IoT remote control function is the ability to remotely download and install core software updates via the cloud, as well as view and manage vital asset data from anywhere.

Preventative Maintenance

As the name suggests, preventative maintenance is about addressing issues before they happen with proactive monitoring and fixes. Every time you take your car in for an oil change, for example, the mechanic adheres a sticker to the inside part of your windshield reminding you to return either before you drive a certain number of miles or an amount of time passes.

But for the same reasons a skeptical consumer might feel the recommended amount of time between visits mainly advantages the mechanic they're paying, it's important to remember that scheduled maintenance isn't without perceived costs. Too much maintenance means you might be dealing with unwieldy and frequent checkups. Too little maintenance, of course, might also mean paying for a costly replacement part—or worse, a reduction in equipment performance or security over time.

The IoT response to preventative maintenance? Real-time asset monitoring through remote, interconnected devices.

With IoT, monitoring becomes a responsive process. And for some applications, they represent an enormous improvement over traditional solutions.

One example is in IoT-connected HVAC systems, whose traditional monitoring systems often represent a prohibitively expensive barrier to entry. With the introduction of internet-connected microcontrollers, however, key data points, such as voltage, current, tilt, power, irradiance, and others are used to gauge when components read a breaking point and send out an alert.

IoT allows businesses to get real-time alerts for system deterioration and other features at a fraction of the cost, sparing them time waiting for repairs to crucial energy infrastructure by notifying suppliers faster about the need for a fix—improving the overall consumer experience.

Performance Optimization

Power generation is the second-leading cause of greenhouse gas emissions, behind only the transportation sector.

In a sector aching for innovation, smart grid technology powered by the IoT is leading the digital transformation for utilities and consumers.

Some of the ways that smart grids help with performance optimization include:

Demand Response and Demand-Side Management

What if consumers could save money by relegating usage to non-peak hours? Companies using smart grids to optimize their demand response can create incentives for consumers to run their dishwashers or do a load of laundry during times when there is low energy demand, which would save them money and decrease unnecessary emissions.

Environmentally and/or budget–conscious consumers, including businesses, can use the data applications from smart grids to be better informed of their own energy consumption levels. At the same time, suppliers can better tailor their power to service actual needs of consumers—instead of relying on estimates.

Optimizing the Grid

Utilities and suppliers can use smart grids to analyze the complex relationship between pricing, availability, efficiency and supply to better optimize all four —faster.

Monitoring

IoT tools can help smart grids monitor key components, alert stakeholders, and identify solutions to problems.

Storing Energy

Utilities and even consumers can store electrical energy through smart-enabled batteries that promote healthy life cycles and distribute energy evenly to others on the grid.

More Accurate Billing

Traditionally, billing has often been the most odious element of running utilities. Fortunately, with IoT-powered smart grid technology, utilities can bring their billing into the 21st century.

Getting Maximum Value from AMIs

Investing in a smart grid involves updating and transforming old infrastructure. But it also represents a chance to maximize ROI. Smart grids analyze key data and automate finding to ensure that you're generating the most possible revenue out of your smart meter implementation.

Get Creative with Rates

One of the main advantages of the smart grid for utilities is that it allows them to provide incentives for consumers to monitor their consumption. With smart billing, you can easily come up with more creative offers that will make consumers want to reduce their consumption during peak hours—a win-win for everyone.

Suppliers can create pricing strategies based on daily electricity demand that incentivize users to shift their consumption to outside of peak hours. When suppliers are able to manage peak times without generating excess energy, it not only results in savings for the supplier, but decreased Co2 emissions.

Billing at Scale

Usage-based billing in the era of IoT can get complicated. With the help of smart grids technology, you can exploit innovative billing solutions at scale without missing a beat.

Improve Billing Transparency

When they can't get to the traditional meter to take a reading, energy companies have been known to bill based on estimated usage. The problem is both suppliers and consumers lose out when actual usage deviates from the expected—as this results in inaccurate billing for consumers and usage bottlenecks for suppliers.

Smart grids, with their improved data capture and communication features, fix this problem.

Building IoT-Enabled Smart Energy Solutions with Particle

Learn more about how Particle gives you the foundation to build robust IoT smart energy solutions.

As a flexible integrated IoT Platform-as-a-Service, Particle is ideal for building IoT-enabled smart energy solutions. Here's why:

Connectivity That Just Works

Particle offers both cellular and WiFi connectivity options so you can choose the one that makes sense for your use cases. For remote or moving energy assets, cellular connectivity is a must-have.

Cellular IoT is enormously difficult to manage in-house. Particle is one of the few IoT platforms that offers global cellular coverage and will pick the best carrier based on your geographic location to ensure you get the best reception and secure the best roaming agreements. Our cellular devices make connectivity management easy because they're built for simplicity and fast deployment. Just turn them on and they'll instantly connect to the best network available.

If you're not sure what connectivity option is best for your project, check out our guide to cellular vs. WiFi for IoT.

An Integrated Platform-as-a-Service

Most IoT solutions fall into one of two categories:

- They provide hardware and SIM cards, forcing you to figure out how to develop on the hardware, certify your IoT deployment, and make everything "talk to each other."
- They provide an off-the-shelf solution that is quick to deploy but offers little configurability or customization options for you to build your solution.

Particle's IoT Platform-as-a-Service hits the sweet spot of offering a fully integrated solution that handles the hard parts of making hardware, connectivity, and software work together while being flexible enough for you to build your product your way.

Reliable Over-the-Air Updates

IoT platforms that only offer hardware or fleet management make device management and over-the-air updates difficult. They don't give you insight into if the device is being used, and they can fail if the connection is lost. Worse, they require complex integrations between the device, operating system, and cloud solutions.

Particle provides an industry-leading OTA solution, which includes:

- Compatibility verification that automatically verifies firmware-hardware compatibility to prevent devices from bricking.
- Intelligent updates that automatically deliver updates when devices indicate they are ready to receive them.
- Dynamic delivery that automatically adjusts the speed of data transfer to the capabilities of the network.

- Update confirmation that will automatically fall back to the prior firmware if the update is unsuccessful.

Security

Central to the promise of the smart grid is the idea of a more secure electrical grid. In times of geopolitical uncertainty, nations across the world are investing in cybersecurity to protect critical infrastructure.

IoT-enabled smart energy solutions, as part of smart grids, should be secure-by-default. Particle allows you to build secure solutions confidently with key IoT security certifications like SOC II, GDPR, Privacy Shield, and CCPA compliance.

All Particle devices use end-to-end encryption between the Particle Cloud and our customers. When customers pull data from our cloud service into their own clouds, those links also encrypted. In other words, we securely apply standard network encryption to protect all end-to-end communications with encryption keys. Learn more about how Particle makes securing IoT devices easy.

Connected vehicles are a key component of the Internet of Things (IoT) ecosystem. They are equipped with sensors and communication technologies that enable them to share data with other vehicles, infrastructure, and the cloud. Here are some examples of connected vehicles at work in the IoT:

1. **Adaptive cruise control**: This technology uses sensors to detect the distance between vehicles and automatically adjusts the speed of the vehicle to maintain a safe following distance.
2. **Automatic route planning based on real-time conditions**: Connected vehicles can use real-time data about traffic, weather, and road conditions to optimize their routes and avoid congestion.
3. **Traffic redirected away from congestion**: Connected vehicles can communicate with each other and with infrastructure to redirect traffic away from congested areas.
4. **Automatically updated road signage, to report traffic or conditions**: Connected vehicles can provide real-time data about traffic and road conditions, which can be used to update road signage and provide drivers with up-to-date information.
5. **Communications to drivers, notifying them of crash sites or wrong-way drivers ahead**: Connected vehicles can communicate with each other and with infrastructure to provide drivers with real-time information about accidents, construction, and other hazards.
6. **Automatic vehicle braking to prevent collisions (in trials)**: Connected vehicles can use sensors and communication technologies to detect potential collisions and automatically apply the brakes to prevent accidents.
7. **Autonomous and semi-autonomous vehicle control (in trials)**: Connected vehicles can use sensors, communication technologies, and artificial intelligence to enable autonomous or semi-autonomous driving.

Connected vehicles have the potential to improve safety, reduce congestion, and enhance the overall driving experience.

Industrial IoT (IIoT) refers to the use of Internet of Things (IoT) technologies in manufacturing and other industrial sectors to improve the efficiency, safety, and productivity of industrial processes. IIoT involves the use of sensors, actuators, and other devices to collect data from machines and equipment, which is then analyzed to optimize performance and reduce downtime. The data collected can also be used to identify potential problems before they occur, enabling predictive maintenance and reducing the risk of equipment failure. IIoT is expected to have a significant impact on the manufacturing industry, with estimates suggesting that it could generate up to $15 trillion in global GDP by 2030