Dayananda Sagar University

School of Engineering, Hosur Main Road, Kudlu Gate, Bengaluru-560 068

Introduction to IoT and Embedded Computing
20AM3610

20AM3610

SEMESTER – VI

Course Code: 20AM3610

Prof. Pradeep Kumar K

Dept. of CS&E (AIML)

DSU, Bangalore

20AM3610

Course Objectives:

- ● To understand the fundamental of IoT and appreciate the importance of communication between machines with reference to IoT.
- ● To understand the embedded systems including design techniques, controldriven architectures, and use of Internet for communication.
- ● See the mechanism of controls and sensing and use of Internet to take global decisions using IoT technology and see how it works.
- ● To understand, appreciate and develop ability to use various contemporary IOT communication protocols for transport, discovery and routing.
- ● To understand the methodologies to implement the software systems for embedded computing and methods of programming them.
- ● To appreciate the utilities of IoT through case studies

Course outcomes:

CO No.	Outcomes	Bloom's Taxonomy Level
CO1	Program an embedded computing device such as Arduino and Raspberry Pi.	L2
CO2	Build an IoT system for sensing and decision making.	L3
CO3	Implement standard communication protocols for IoT to build large systems	L4
CO4	Appreciate and understand the use of IoT in systems such as home automation, smart lighting, smart parking etc	L4

20AM3610

Text Documents Requirement

- **TEXT BOOKS:**

1. Jeeva Jose, “Internet of Things”, Khanna Book Publishing Company, 2021.

2. Samuel Greengard, “The Internet of Things”, 1st Edition, MIT Press, 2015.

3. Peter Waher, Pradeeka Seneviratne, Brian Russell, Drew Van Duren, “IoT: Building Arduino-Based Projects”, 1st Edition, Packt Publishing Ltd, 2016

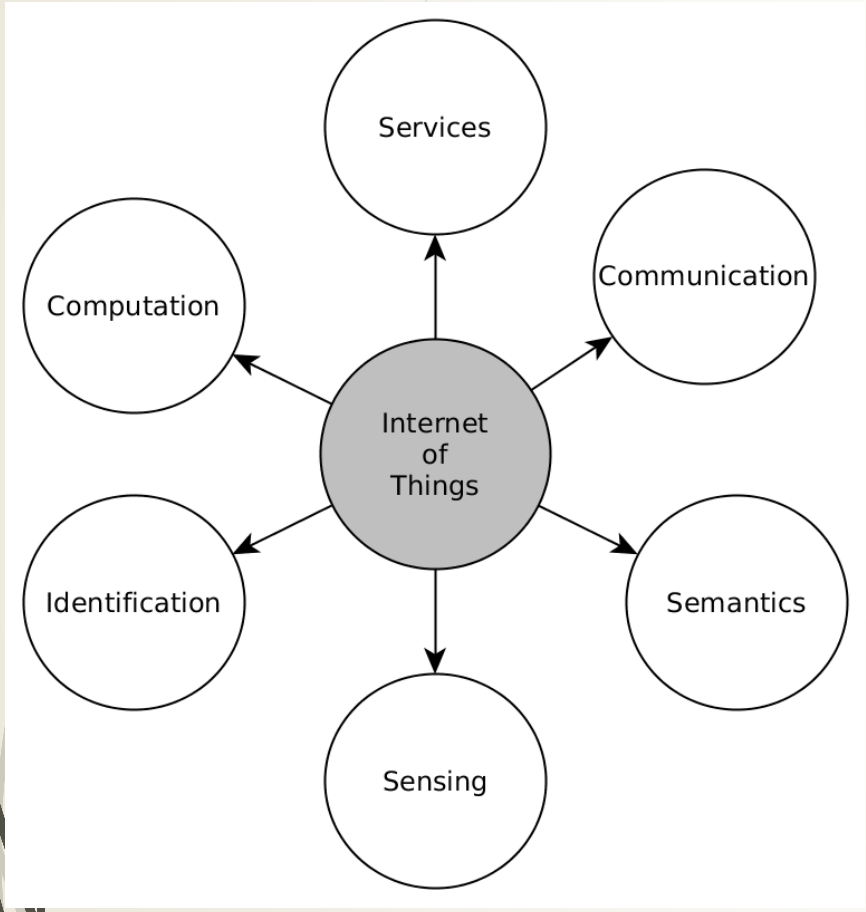
- **REFERENCES:**

Peter Waher, “Mastering Internet of Things: Design and create your own IoT applications using Raspberry Pi 3”, 1st Edition, Packt Publishing Ltd, 2018

David Hanes, Gonzalo Salgueiro, “IoT Fundamentals: Networking Technologies, protocols, and use cases for the Internet of Things”, Cisco Press, 8th impression 2021.

20AM3610

Module-2 : IoT Enabling Technologies



- Sensor Networks,
- Sensors and actuators,
- Analog/Digital Conversion,
- Communication Protocols,
- Embedded Computing Systems,
- Cloud Computing.

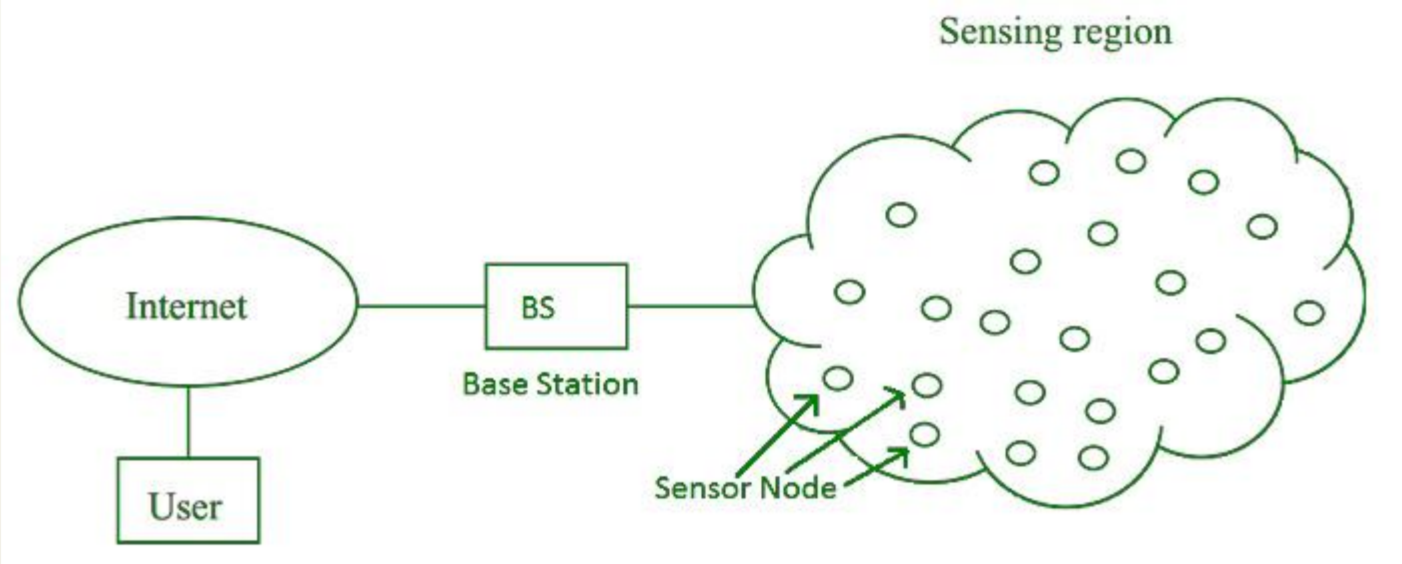
20AM3610

IoT(internet of things) enabling technologies are

- Wireless Sensor Network
- Cloud Computing
- Big Data Analytics
- Communications Protocols
- Embedded System

Wireless Sensor Network

- 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.

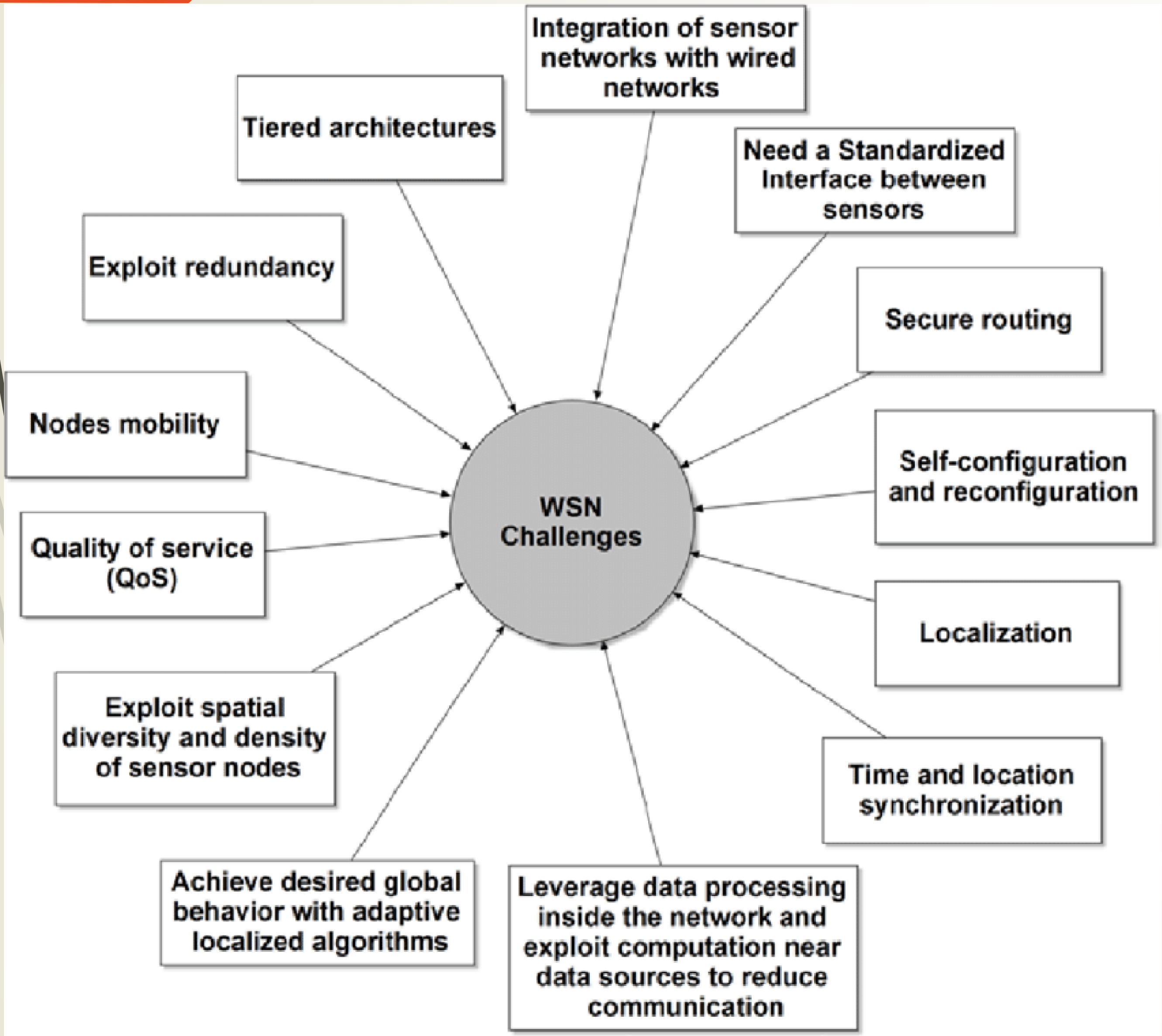


Applications of WSN:

- Internet of Things (IOT)
- Surveillance and Monitoring for security, threat detection
- Environmental temperature, humidity, and air pressure
- Noise Level of the surrounding
- Medical applications like patient monitoring
- Agriculture
- Landslide Detection

20AM3610

Challenges of WSN:



- Quality of Service
- Security Issue
- Energy Efficiency
- Network Throughput
- Performance
- Ability to cope with node failure
- Cross layer optimisation
- Scalability to large scale of deployment

20AM3610

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

- Limited power and energy:
- Limited processing and storage capabilities:
- 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

20AM3610

Components of WSN:

- **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.
- **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.
- **WLAN Access Point:** It receives the data which is sent by the Radio nodes wirelessly, generally through the internet.
- **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.

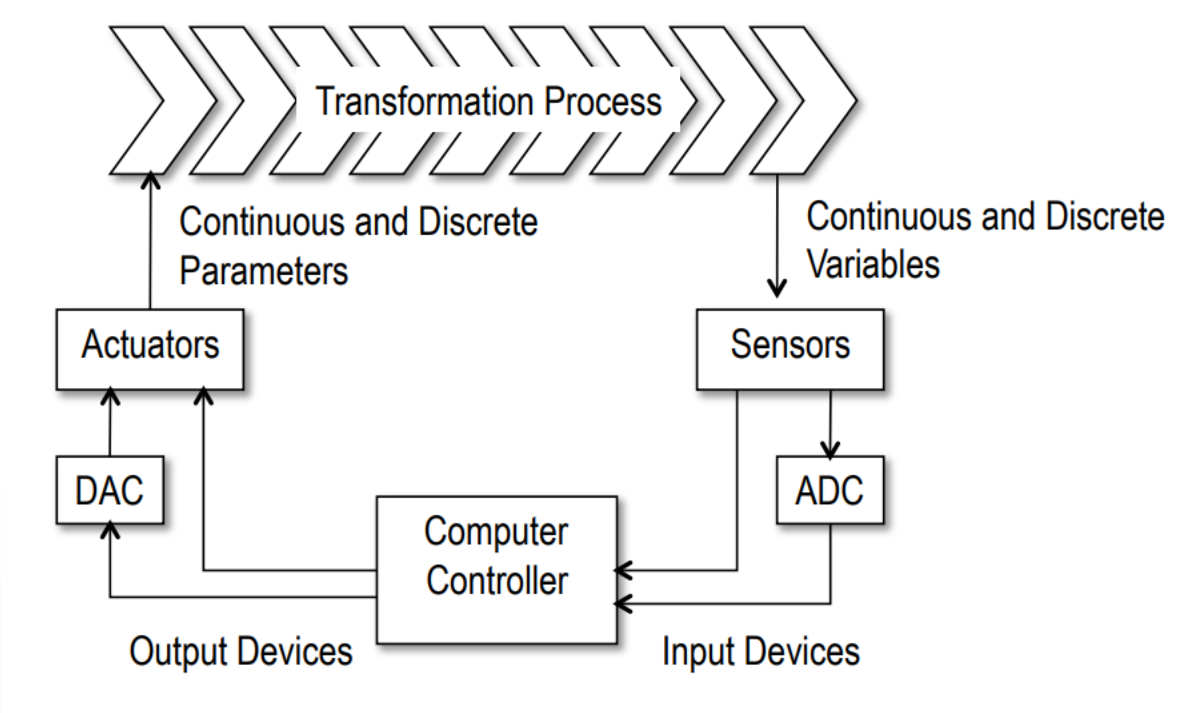
20AM3610

Sensors and actuators

- Computer-Process Interface

To implement process control, the computer must collect data from and transmit signals to the production process

1. Sensors to measure continuous and discrete process variables
2. Actuators to drive continuous and discrete process parameters
3. Devices for ADC and DAC
4. I/O devices for discrete data



- Sensors: A sensor is a transducer that converts a physical stimulus from one form into a more useful form to measure the stimulus

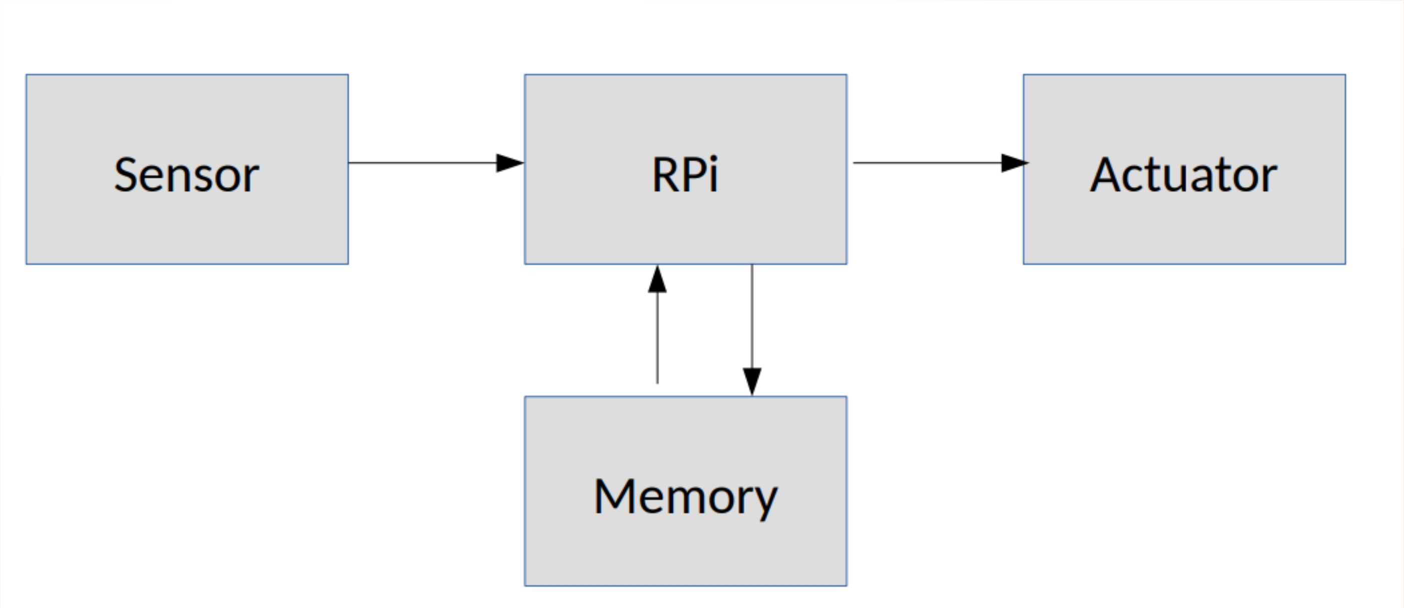
Sensor categories by stimulus

Stimulus	Example
Mechanical	Positional variables, velocity, acceleration, force, torque, pressure, stress, strain, mass, density
Electrical	Voltage, current, charge, resistance, conductivity, capacitance
Thermal	Temperature, heat, heat flow, thermal conductivity, specific heat
Radiation	Type of radiation (e.g. gamma rays, x-rays, visible light), intensity, wavelength
Magnetic	Magnetic field, flux, conductivity, permeability
Chemical	Component identities, concentration, pH levels, presence of toxic ingredients, pollutants

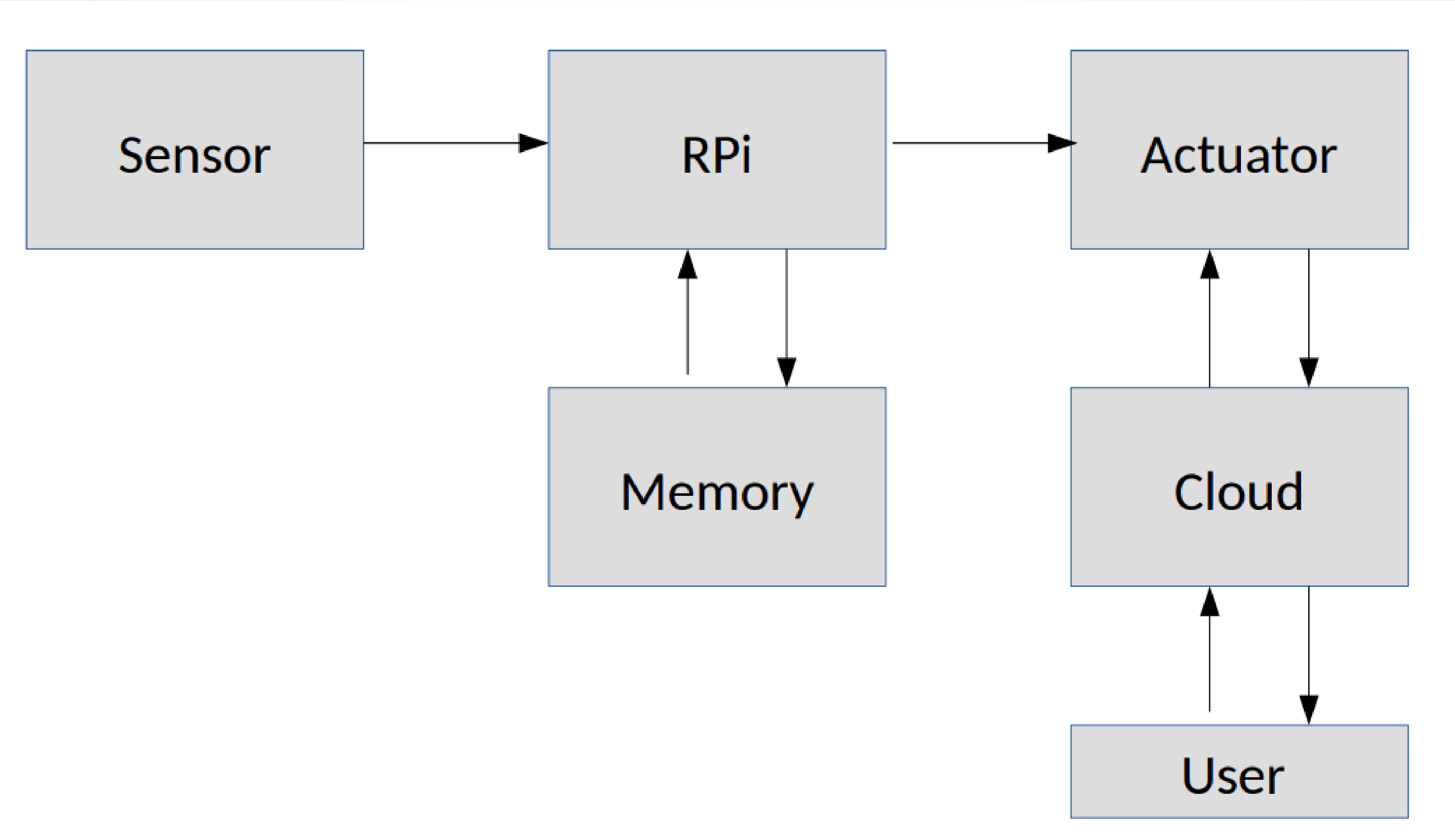
- Sensors can be classified into two basic categories:

1. Analog (continuous): thermocouple, strain gauges, potentiometers.
2. Discrete: Binary (on/off)- Limit switch, photoelectric switches.

   : Digital (pulse counter)- photoelectric array, optical encoder.



Simple IOT Architecture



20AM3610

Commonly used sensors

- Light Sensor
- Proximity Sensor
- GAS Sensor
- Bio-Medical Sensor
- Tilt Sensor
- Accelerometer Sensor
- Flow Sensor
- Level Sensor
- Speed Sensor
- Force Sensor
- Temperature Sensor
- Pressure Sensor
- Current Sensor
- Sound Sensor
- Hall effect Sensor
- Compass Sensor
- Humidity Sensor
- Motion Sensor
- RPM Sensor

Actuators

- An actuator is a component of a machine that is responsible for moving or controlling a mechanism or system.
- An actuator requires a control signal and a source of energy. The control signal is relatively low energy and may be electric voltage or current, pneumatic or hydraulic pressure, or even human power.
- When the control signal is received, the actuator responds by converting the energy into mechanical motion.
- Examples: LED, RGB LED, Buzzer, Servo Motor, DC Motor, Relay.

20AM3610

Analog/Digital Conversion,

- Assignment : Techniques for IOT ADC

20AM3610

Communication Protocols

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

20AM3610

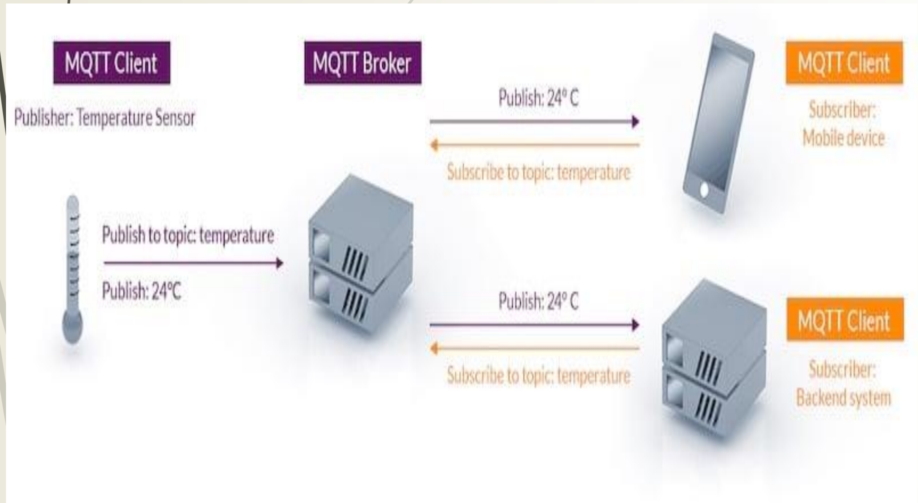
IoT Data Communication Protocols

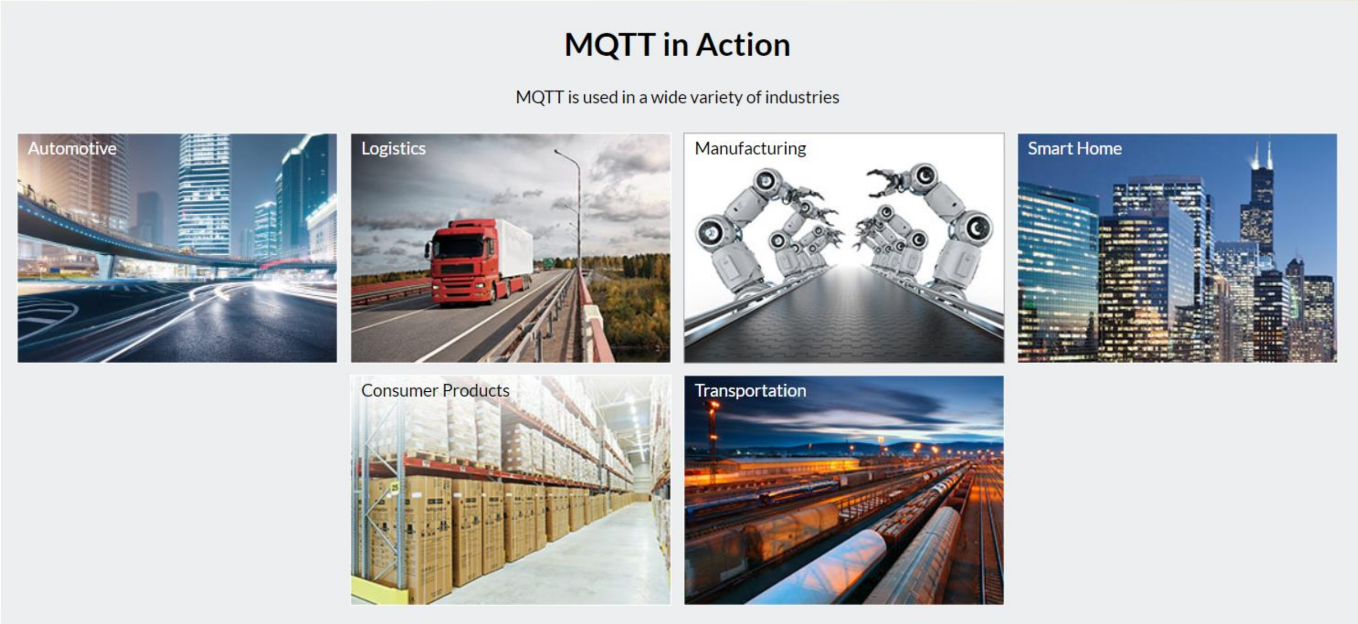
- Message Queue Telemetry Transport (MQTT)
- HyperText Transfer Protocol (HTTP)
- Constrained Application Protocol (CoAP)
- Data Distribution Service (DDS)
- WebSocket
- Advanced Message Queue Protocol (AMQP)
- Extensible Messaging and Presence Protocol (XMPP)
- OPC Unified Architecture (OPC UA)

Message Queue Telemetry Transport (MQTT)

- Designed to be lightweight, so it can work in very low bandwidth networks, MQTT allows communication between nodes in both reliable and unreliable networks.
- MQTT follows a publish/subscribe architecture, meaning that there are nodes (brokers) that make the information available, while others (clients) can read the available information after subscribing by accessing the corresponding URL.
- A use case of MQTT is in a smart factory where there are temperature sensors installed along with the production plant. The installed sensors will connect to the MQTT broker and will publish the data within sensor topics, as follows: sensors/temperature/assemblyLineInit
- the MQTT clients, which can be of several types and quantities, will subscribe to the same topic in order to read the temperature data. An example of an MQTT architecture can be seen in Figure.

20AM3610





- In addition, MQTT defines three levels of quality of service, depending upon the reliability, from lowest to highest:
- Level 0: there is no guarantee of the message delivery.
- Level 1: the delivery is guaranteed, but it is possible to receive duplicate messages.
- Level 2: the delivery is guaranteed and there will be no duplicates.

HyperText Transfer Protocol (HTTP)

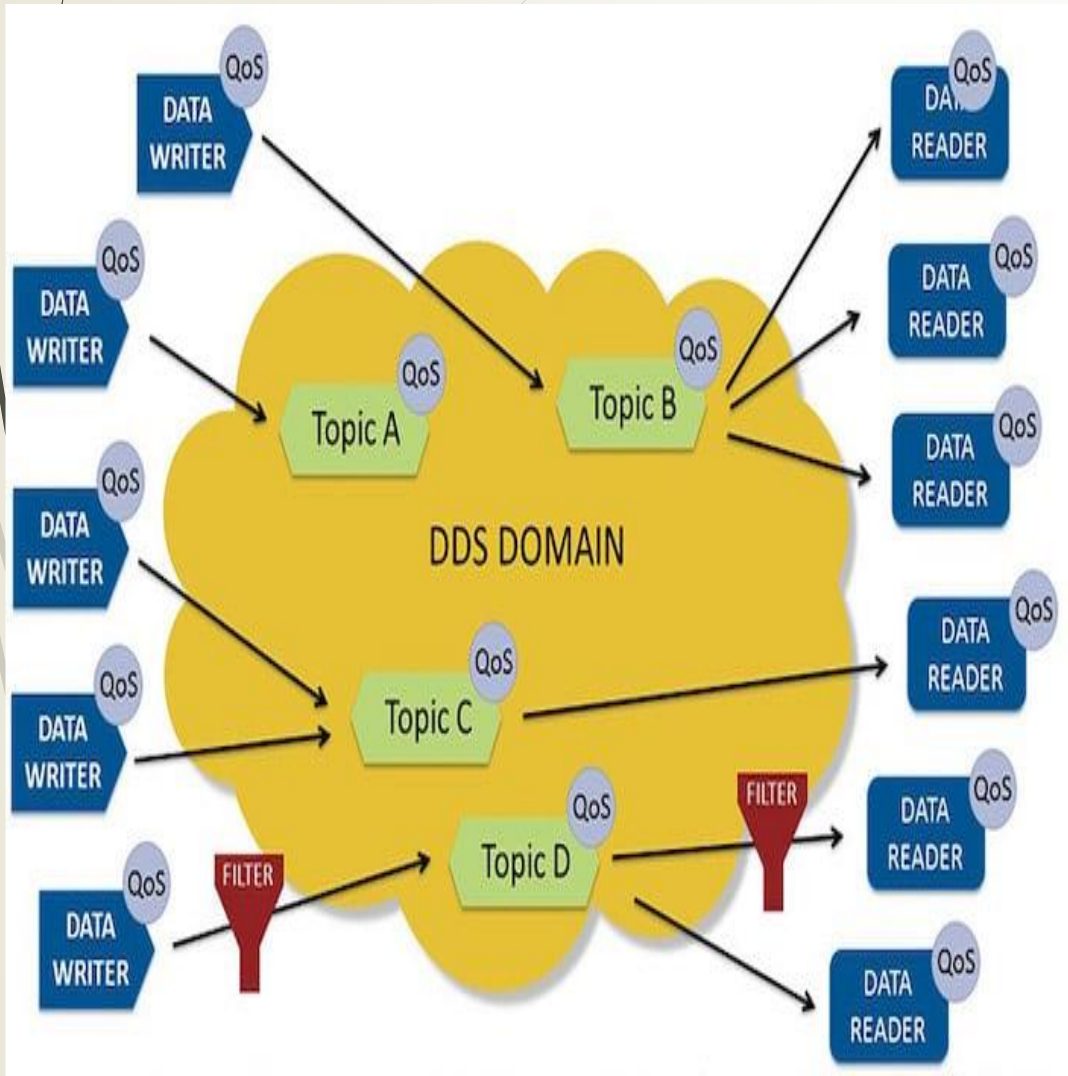
- This protocol has been the origin of data communication for the World Wide Web (WWW), so logically it is being used in the IoT world. However, it is not optimized for it because of the following:
- The HTTP is made for two systems communicating to each other at a time, not more, so it is time and energy-consuming to connect several sensors to get information.
- The HTTP is unidirectional, made for one system (client) to be sending one message to another one (server). This makes it quite hard to escalate an IoT solution.
- Power consumption: HTTP relies on Transmission Control Protocol (TCP), which requires a lot of computing resources, so it is not suitable for battery-powered applications.

Constrained Application Protocol (CoAP)

- CoAP is a web transfer protocol to be used with limited networks with low bandwidth and low availability. It follows a client/server architecture and is built similarly to HTTP, supporting the REST model: servers make resources available with an URL, and clients can make requests of types GET, POST, PUT and DELETE.
- The CoAP communication links are 1:1 and UDP-based, so the delivery is not guaranteed. CoAP is made to work in highly congested networks, where nodes do not have a lot of intelligence and are not always working.

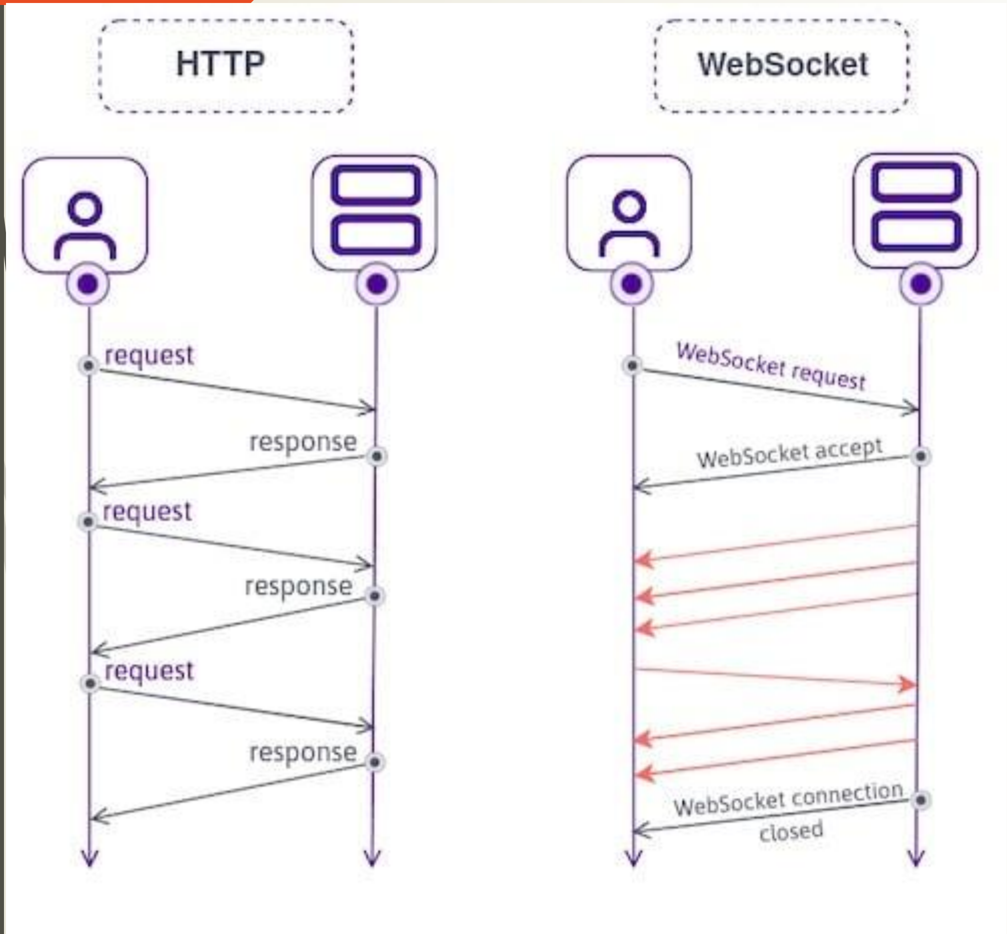
20AM3610

Data Distribution Service (DDS)



- Similar to MQTT, DDS follows a publish-subscribe methodology, with the main difference being that there are no brokers. It means that all publishers (i.e., temperature sensors) and subscribers (i.e., mobile phones) are all connected to the same network. This network is known as Global Data Space (GDS) and it interconnects each node with all the other ones to avoid bottlenecks. An example of the DDS GDS can be seen in Figure.

WebSocket



- Linked to the HTTP protocol, the WebSocket technology establishes a TCP connection between a browser and a server, and then both of them exchange information until the connection is closed. Figure 3 shows a high-level comparison between HTTP and WebSocket.
- Although this protocol can be seen as an improvement of the HTTP connection, the WebSocket is still very overloaded and heavy for IoT applications.

Advanced Message Queue Protocol (AMQP)

- In the beginning, AMQP was not initially created for IoT applications, but for banking environments. AMQP accepts publish/subscribe architectures, as well as request/response types. It is TCP-based, so delivery is guaranteed, as well as acknowledgment, which makes this protocol reliable, with the consequent overhead message reliability.
- Compared to MQTT, AMQP offers two Quality of Service levels:
- At most once: the sender does not wait until having an acknowledgment from the receiver to delete a message.
- At least once: for each message, the sender will receive an acknowledgment from the receiver before deleting the message. In a case where the acknowledgment is lost, the message is re-sent.
- Exactly once: the messages are sent only once. It requires special coordination between the sender and the receiver.

Extensible Messaging and Presence Protocol (XMPP)

- It is based on Extensible Markup Language (XML) and in the past, it was known as Jabber. It is an open-source, decentralized, secure protocol to exchange XML messages.
- A characteristic factor of XMPP is its addressing method and how nodes are identified. It uses a Jabber ID with the format jabberID@domain.com, which allows two nodes to interchange information regardless of the distance between them.

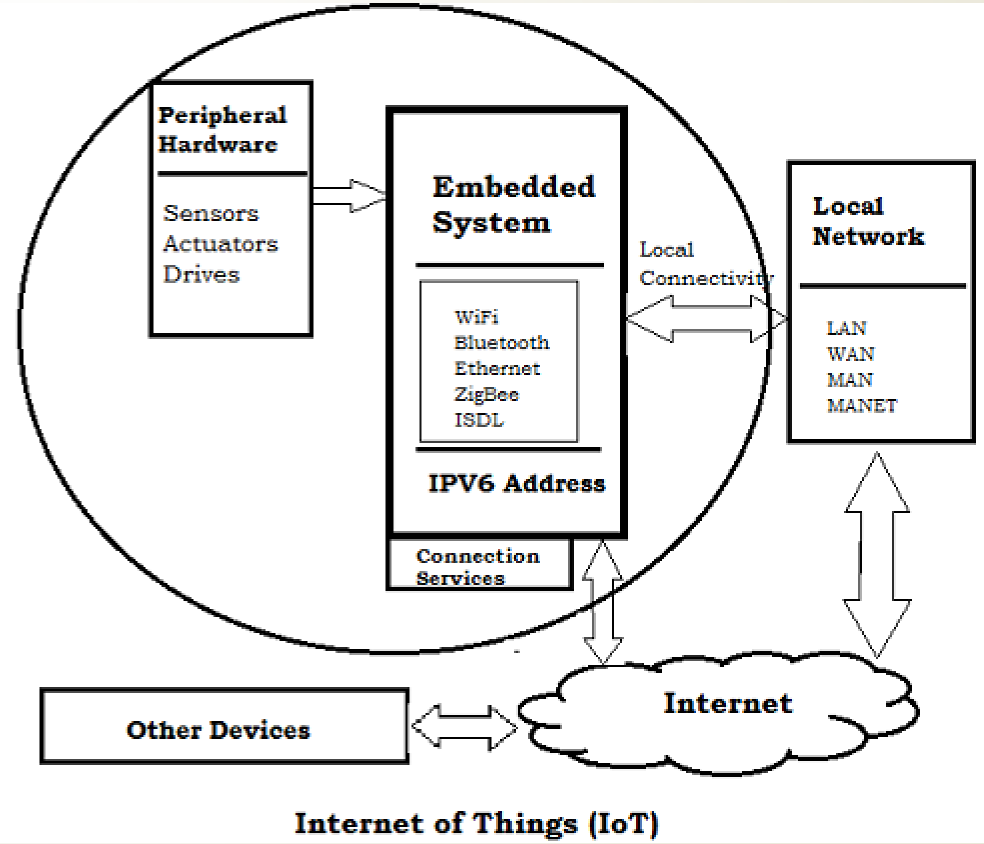
OPC Unified Architecture (OPC UA)

- It is a standard made for industrial communication, strongly oriented to guarantee interoperability between manufacturers, operating systems, and programming languages. The OPC foundation reported that many industrial vendors are currently (as of 2022) adopting the OPC UA as the open standard.
- All in all, the OPC UA is a transport-agnostic protocol, so it supports both of the previously used architectures: request/response (such as WebSocket or HTTP), as well as the publish/subscribe (such as the MQTT).

20AM3610

Embedded Systems :

- It is a combination of hardware and software used to perform special tasks.
- It includes microcontroller and microprocessor memory, networking units (Ethernet Wi-Fi adapters), input output units (display keyword etc.) and storage devices (flash memory).
- It collects the data and sends it to the internet.
- Embedded systems used in
- Digital camera
- DVD player, music player
- Industrial robots
- Wireless Routers etc.



Embedded Computing Systems

- Embedded computing systems are used every day, yet they are often overlooked and misunderstood.
- They are essential to the functioning of a huge array of products and applications, from consumer electronics to industrial devices.
- The stability of these embedded computing systems is essential to the appropriate functioning of significant aerospace, medical, and communication applications.

20AM3610

- How Do Embedded Systems Work?
- Embedded systems are task-specific, meaning they are built to perform one task rather than a variety of tasks. While the exact design of a particular embedded system depends on its intended task, it typically contains a processor, a power supply, and memory and communication ports. The communication ports transmit data between the processor and any peripheral devices, the processor interprets the data using the integrated software, and the power supply powers the assembly.
- The components within an embedded system can be broken down into hardware and software components. Hardware components include the processor (either a microprocessor or microcontroller that processes digital signals and stores them in memory), sensors (to convert physical sense data into electrical signals), analog-to-digital converters (to convert analog electrical signals to digital signals), digital-to-analog converters (to convert digital data to analog data), and actuators (to compare actual output to memory-stored data to choose the right one). Software components vary in complexity depending on the embedded system. However, industrial-grade embedded systems can use simple software with little memory capacity.

Types of Embedded Systems

Embedded systems can be categorized based on function. Some of the basic types available include:

- **Standalone embedded systems.** These systems are designed to perform specialized tasks without belonging to a host system. Examples include calculators and MP3 players.
- **Mobile embedded systems.** These systems are designed for portability. Examples include digital cameras.
- **Networked embedded systems.** These systems are designed to connect to a network and provide output data to other systems. Examples include point-of-sale (POS) systems and home security systems.
- **Real-time embedded systems.** These systems are designed to provide output within a defined time interval. Examples include braking systems and traffic control systems.

They can also be categorized by performance requirements:

- **Small-scale embedded systems.** These systems typically use up to an 8-bit microcontroller.
- **Medium-scale embedded systems.** These systems use a 16-bit or 32-bit microcontroller. They can use one or multiple units depending on the complexity of the system.
- **Sophisticated-scale embedded systems.** These systems may use complex software and hardware components.

Benefits of Embedded Systems

- They are highly customizable and low cost, which makes it easy and affordable to tailor them to exact functional and performance requirements.
- They also have low power consumption, which helps reduce energy costs for the overall system.

Applications of Embedded Systems

Embedded computing systems are found in the electronic devices and systems of a variety of industries. For example:

- In the healthcare industry, they are used for diagnostic and imaging devices, patient monitoring systems, and surgical and treatment tools.
- In the life sciences industry, they are used for bioinformatics, proteomics, and genome sequencing devices and systems.
- In the military and defense industry, they are used for autonomous and unmanned vehicles, C4ISR systems, and training and simulation systems.
- In the industrial automation industry, they are used for various manufacturing systems.

Embedded product solutions include:

- Control Processing. These platforms handle the machine or motion control programming for your application, whether it involves simple positioning or highly complex, fast, and precise movements.
- Image Generation. These platforms accommodate the rapidly evolving specifications and standards for advanced visual-based systems, such as AI, engineering simulation, immersive VR, and real-time ray tracing.
- Data and Image Processing. These platforms support applications that require real-time feedback, such as surgeries and chemical processing cycles.
- Planning and Viewing. The platforms have separate planning and viewing stations that allow workers to monitor embedded imaging output without being physically next to the device. They ensure workplace safety when imaging equipment would otherwise expose employees to radiation, X-rays, or chemicals.
- Display Systems. These platforms visually display output data. Examples include touch All-in Ones (AIO), embedded displays, monitors, and panel PCs.

What is IoT?

- IoT stands for Internet of Things and refers to the interconnected nature of devices that we use in our daily lives. These days "smart" devices can connect to the internet and to other devices to facilitate everyday actions. For example, a smart fridge can detect the items in it as well as expiry dates and provide information to the owner about it. This is a typical use-case for IoT in everyday use.

What is an embedded system?

- An embedded system is a tiny computer that was built for a custom purpose. Such a computer usually has a microprocessor or microcontroller, which is an integrated circuit that typically contains processor, memory, and I/O (input/output) peripherals on one chip.
- The defining factor of an embedded system is that it can carry out some form of digital processing. This is what distinguishes it from plain hardware devices that contain only some circuitry and possibly a battery.
- An embedded system often has sensors that let it monitor environmental factors.

What is the difference between the two?

- An embedded system does not necessarily have internet connectivity, but an IoT embedded system always does. The difference between an embedded system and IoT is that the embedded system can be a subset of IoT. Embedded system before IoT were not able to connect to the internet, therefore IoT embedded systems generally encompass a broader range of devices and use cases.

- What is embedded system in IoT?
- An IoT embedded system is an embedded system that has internet connectivity. Another word for what is IoT embedded system is a "smart" device. A touch screen and a keyboard are not necessary to define a device as an IoT embedded system, although these peripherals can also be attached.

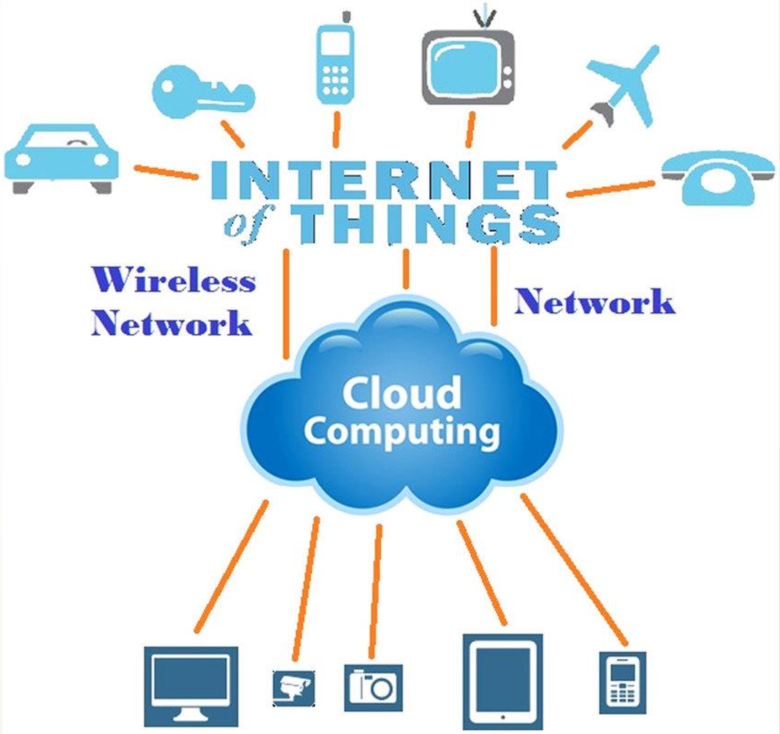
20AM3610

Cloud Computing

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

Characteristics –

- Broad network access
- On demand self-services
- Rapid scalability
- Measured service
- Pay-per-use



Different Services Provided by Cloud Computing

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

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

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

Cloud Computing and IoT: How Do They Work Together?

- Cloud computing provides hosted services through the Internet. On the other hand, the Internet of Things (IoT) connects adjacent smart devices to the network to share and evaluate data.
- Cloud computing and the internet of things are the perfect examples of innovative technologies. Cloud computing provides the tools and services needed to create IoT applications.
- It also supports the creation of dependable and accurate IoT-based applications.

Cloud Computing and Internet of Things: Head-to-head

20AM3610

Basis	Internet of Things	Cloud Computing
Definition	The Internet of Things (IoT) is a network of networked devices, machines, and other 'things' that may be equipped with sensors, circuitry, and software to gather and exchange data.	Individuals and companies may use cloud computing to access computing resources and apps on demand.
Objective	The basic goal of IoT is to build an ecosystem of networked devices that can detect, touch, interact, and communicate with one another.	Cloud computing, on the other hand, enables us to make use of all of the data created by IoT by allowing us to interact with our organization from anywhere, at any time.
Application s	Smartwatches, fitness bands, smartphones, smart home devices, smart cities, automated transit, smart monitoring, conversational interfaces, self-driving cars, thermostats, implants, lighting, and other devices are among the most essential and widely used IoT applications.	Antivirus apps, online data keeping, data processing, email applications, streaming video software, web conferencing applications, and so on are concrete examples of cloud computing.
Purpose	The basic goal is to establish an ecosystem of networked items that can detect, touch, regulate, and communicate with one another.	The goal is to provide virtual access to enormous quantities of computer power while providing a unified system perspective.

20AM3610

Role	IoT's function is to create large volumes of data.	Role of cloud computing in IoT is to give a way to store IoT data and provide the needed tools to create IoT applications.
Salary	IoT Engineer salary ranges from $120,460 to $151,687 with the average base salary of $136,021.	Cloud Developer pay levels vary from in the Range: $80k - $179k, with an average of $141,798
Skills	Node.js development. Mobile app development.	Cloud service platform expertise. Programming languages.
Education	After qualifying, a BE/B. Tech/MCA/M. Sc/MS with at least three years of particular IT experience is required.	After qualifying, a BE/B. Tech/MCA/M. Sc/MS with at least three years of particular IT experience is required.
Integration	The IoT Cloud is a centralized server that contains computer resources that may be accessed at any time.	Cloud integration, in its simplest form basic, entails bringing together several cloud environments — either as a hybrid deployment or as numerous public clouds — so to allow them to operate as a single, unified IT infrastructure for a company.

20AM3610

Role of Cloud computing in IoT

1. Facilitates remote computing

- The Internet of Things eliminates the dependency on on-site infrastructure with its large storage capacity. It has become mainstream due to the continued development of internet-based technology such as the internet and devices that support advanced cloud solutions.

2. A secure and private environment

- By automating tasks using cloud technology and IoT, organizations are able to reduce security threats considerably. Furthermore, it provides strong security measures for users with effective authentication and encryption protocols.

3. Integrated data management

- Modern technology not only integrates IoT and cloud seamlessly but also allows real-time communication and connectivity. This enables real-time information collection and real-time data integration with 24/7 connectivity, as well as on-spot analysis of key business processes.

4. Continuity of operations

- A cloud service uses a network of data servers located across multiple geographical locations that store multiple copies of backed-up data. With cloud computing, it is easy to recover data from IoT-based operations during any emergency.
- Cloud computing aids an enhanced IoT, but cloud service deployment might not always be easy and should be considered during planning.

20AM3610

How do I connect IoT to cloud?

- Create an AWS project.
  - Make a "thing"
  - Link the "object" to a gadget.
  - Choose a programming language.
  - Create a certificate as well as policy, then include them in the project.
- Install the AWS IoT SDKs on the client/device.
- Develop the device-side application using the following code:
  - Include the AWS module in the client code.
  - In code, create a client and provide it with the private key but also a certificate.
  - Begin a connection
  - To receive information, subscribe to a discussion and define a callback.
  - Publish messages as your app requires.

20AM3610

20AM3610

20AM3610

20AM3610

ME-1

Thank You