Chapter: 1

(1) What is IOT and CPS? Explain its Characteristics and its advantage and disadvantages.

- IOT stands internet of Things. It is a network of physical devices, vehicle, home appliances and other item embedded with electronic software, sensor, and connectivity which enables these objects to connect and exchange data.
- IOT architecture is the foundation of connecting physical devices to the internet to enable communication and data exchange. It consists of multiple component that work together to facilitate this communication.
- IOT, also called the internet of objects, refers to a wireless network between objects, usually the network will be wireless and self-configuring, such as house hold appliances.
- Device connect and communicate in many ways. Example of this is smart interact with other smart phones, Vehicle to Vehicle communication, connected video cameras, and connected medical devices. They are able to communicate with consumers, collect and transmit data to companies, and compile large amount of data for third parties.
- IOT will help a business to gain efficiency, harness intelligence from a wide range of equipment, improve operations and increase customer satisfaction.

❖ **Characteristics:**

- **Connectivity:** IoT and devices are connected to each other and the internet allowing data to be transmitted and received.
- **Sensors:** IoT devices are equipped with sensor that gather data about the environment or usage.
- **Processing Power:** IoT devices have the ability to process data to make decision based on the data.
- **Autonomous:** IoT devices can perform tasks on their own, without direct human intervention.
- **Interoperability:** IoT devices should be able to communicate with each other and other devices for seamless communication for their OS.

- **Scalability:** IoT has the potential to scale up to billions of devices and to support a huge number of users.
- **Real-time:** IoT devices can transmit data in real-time, allowing for real-time monitoring and control.

Advantages:

- **Improve efficiency:** Automated task and real time data analysis increased efficiency and productivity.
- **Enhances User experienced:** Device can control remotely, making life easier and more convenient.
- **Increase Connectivity:** Ability to connect devices, people and machines creates new opportunity for communication.
- **Real-time Monitoring:** Real-time data analyze enables organization to quickly identify and address problems.

Disadvantages:

- **Security & Privacy:** IoT devices are vulnerable to hacking and data breaches, and privacy concerns are major issue.
- **Complexity:** The integration of multiple components and technologies can be complex and challenging.
- **Interoperability:** Different devices and system sometime may not work together because of compatibility issues.
- **Cost:** Implementing and IoT solution can be expensive due to the need for hardware, software and network infrastructure.

CPS (Cyber Physical Systems):

- CPS are a subset of IoT that refers to physical systems that are integrated with computer-based algorithms, networks, and communication protocols to achieve a desired outcome.
- CPSs involve the integration of physical systems with the cyber world, creating a tight feedback loop between the two.
- The goal of CPS is to create a seamless and efficient interaction between the physical and virtual worlds.
- CPSs are used in a wide range of applications, from industrial control systems to healthcare and transportation.

(2) Explain Component and Application of IoT.

❖ **Components:**

- **Devices:** Physical devices such as sensor, actuators, and embedded systems that collect and transmit data.
- **Connectivity:** Network infrastructure such as Wi-Fi, Bluetooth and cellular networks to connect devices to the internet.
- **Gateway:** Acts as a bridge between the devices and the cloud, responsible for collecting, filtering, and forwarding data.
- **Cloud Platform:** Provides storage, computing, and analytics capabilities for the collected data.
- **Application:** Software that runs on top of the cloud platform to perform operation such as data analysis, visualization and decision making.

❖ **Applications:**

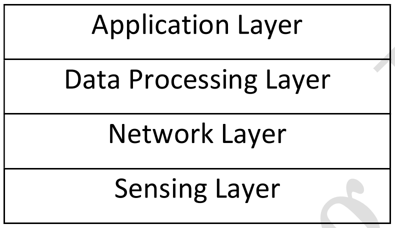
- **Smart Home:** IoT devices such as smart locks, smart thermostats, and smart lighting can be used to control and monitor home environment, providing increased security and convenience.
- **HealthCare:** IoT devices such as wearable health monitors and remote patient monitoring systems can be used to collect and transmit patient data to healthcare providers, improving quality of care.
- **Transportation:** IoT devices such as connected cars, smart traffic systems, and HPS tracking can be used to improve traffic flow, reduce fuel consumptions and enhance safety on the roads.
- **Agriculture:** IoT devices such as smart sensor, drones, and weather monitoring system can be used to optimize crop production, reduce waste and improve the efficiency of farming operation.
- **Worksites:** It is custom production environments like mining, oil and gas construction: operating efficiencies, predictive maintenance, health and safety.

(3) Explain architecture of IoT.

❖ **Architecture:**

➢ IoT technology has a wide variety of applications and use of internet of things is growing so faster. Depending upon different application areas of internet of things, it works accordingly as per it has been developed.

➢ There are different phases in the architecture of IoT but vary according to the situations but generally, there are these four phases in the architecture of IoT:



- **Sensing Layer:** The first stage of IoT includes sensors, devices, actuators etc. Which collect data from the physical environment, processes it and then sends it over the network.
- **Network Layer:** The second stage of the IoT consists of network gateways and data acquisition systems. DAS converts the data collected from sensors into digital data. It also performs malware detection and data management.
- **Data Processing Layer:** The third stage of IoT is the most important stage. Here, data is pre-processed on its variety and separated accordingly. After this, it is sent to Data Centers. Here edge IT comes into use.
- **Application Layer:** The fourth stage of IoT consists of Cloud/Data centers where data is managed and used by applications like agriculture, defense, health care, etc.

(4) Explain IoT Stack.

Number	Name	Function & Applications
Layer: 1	Physical or Sensor	• Sensor is main component of this layer. • Various sensor like temp, heat, humidity etc.
Layer: 2	Processing and Control action	• Micro controller or processor are used in this layer. • Microcontroller receive data from sensor. • OS play important role in this layer
Layer: 3	Hardware Interface	• Component for hardware and serial communication. • RS232, CAN, SPI, etc are used.
Layer: 4	RF	• Protocol used for communication and transport data using short and long range.
Layer: 5	Session or message	• It uses messaging protocol for broadcasting messages. • Protocol: MQTT, FTP, HTTP, CoAP and SSH
Layer: 6	User Experience	• It is related to user interface after product is designed. • It uses object and procedure-oriented technologies. • It uses various database and SQL.
Layer: 7	Application	• Simple application • Smart city application • Smart Home • Smart Agriculture

(5) Explain IoT Levels.

- Levels of IoT systems are IoT Level 1 to 6.
- **Level - 1:**
  - Physical devices and controllers that might control multiple devices. These are the “things” in the IoT, and they include a wide range of endpoint devices that send and receive information.
  - These are suitable for modeling low-cost and low complexity solution where the data involved is not big and the analysis requirements are not computationally intensive.
  - The system consists of a single node that allows controlling the lights and appliances in a home remotely. The device can be used in the system interface with the light and appliances using electronic relay switches. The status information of each appliance is maintained in local database. The controller service continuously monitors the state of each appliance and trigger the relay switch accordingly.

- **Level – 2:**
  - In this level single node performs sensing and local analysis.
  - Both data and application are stored on the cloud. This type of system is suitable for big data used for analysis and that to performed on local side.
  - Smart integration is an example of this level. System uses single sensor for monitoring the soil moisture level and control irrigation system.
  - Controller service monitors continuously the moisture level. If the moisture level drops below set level, the system is turned ON.

- **Level – 3:**
  - Single node is used in this level. Collected data is stored on the cloud and processed on the cloud. Application is also stored on data cloud.
  - This is suitable for huge data analysis requirement are computationally intensive.

- Example of tracking package handling: Accelerometer and gyroscope sensor is used. System allows shippers tracking cargo to communicate with the devices, changing reporting frequencies and investigating alerts generated by attached sensor.
- Controller device sends the sensor data to the cloud in real-time using web socket service. Data is stored on the cloud and analysis is also performed on the cloud.
- Cloud sends alerts if the vibration levels is greater than threshold levels.

Level – 4:

- This system uses multiple sensor and processing is performed on local node. Data is stored on the cloud and application is stored on the cloud storage.
- It uses two observer sensors for local and cloud. These sensors subscribe and receive information from IoT device to cloud.
- The observer sensor can process information and used for various purposes. Control function is not performed by this observer sensor.
- Noise monitoring system is Used in this. Sensors are not depending upon each other. Each sensor runs its own controller service that sends the data to cloud.

Level – 5:

- It contains multiple end sensor and one coordinator sensor.
- The end sensor performs sensing. Coordinator sensor collect data from the end sensor and sends to the cloud.
- Data is stored and analysis in the cloud and application is also cloud based. Forest fire detection system uses this Level system.
- Multiple sensors are kept at different location to monitor temperature, humidity, carbon dioxide level. Coordinator sensor collect data from end node and these node act as gateway and provides internet services to the IoT system.

- **Level – 6:**
  - It contains multiple independent end nodes and it performs sensing function. It sends data to the cloud.
  - Data is stored on the cloud and application is also cloud based. Result is displayed on the cloud. Centralized controller knows the status of monitoring node and sends control commands to nodes.
  - Weather monitoring system uses level 2 IoT system. Multiple nodes are kept at different location to monitor temperature and humidity level.
  - Each node sends real time data to cloud using web socket service. Clous database is used for storing data. Data analysis is performed on cloud side. Cloud based application is used for display data.

(6) Explain Challenges in IoT.

- **Security:** Protecting the vast amount of data generated by IoT devices, and the device themselves, from cyber-attacks and unauthorized access.
- **Interoperability:** Ensuring the different IoT devices can communicate with each other and with other systems, for their manufacturer or OS.
- **Scalability:** Managing the exponential growth of connected devices and data, ensuring the IoT infrastructure can accommodate this growth without becoming overwhelmed.
- **Energy Efficiency:** Minimizing the power consumption of IoT devices to extend their battery life and reduce their impact on the environment.
- **Data Management:** Sorting, Processing and analyzing the vast amount of data generated by IoT devices in real-time, without overwhelming existing systems.
- **Privacy & Data Protection:** Ensuring that personal data collected by IoT devices is protected from unauthorized access and misuse.
- **Regulation:** Ensuring that IoT devices and system comply with existing laws and regulation regarding data protection, privacy and security.
- **Design Based Challenge:** With the development in technology design challenges are increasing at faster rate. There have been issues regarding design like limited computation power, limited energy and limited memory which need to be sorted out.

(7) What is WNS in IoT?

- WNS stands for Wireless Network Services.
- It provides communication between different IoT devices and networks. It enables the seamless and secure transfer of data and information between connected devices.
- WNS enables IoT devices to communicate with each other and with other devices, regardless of their manufacturer or operating system.
- WNS includes different communication technologies such as Wi-Fi, Zigbee, Bluetooth, NFC, and cellular networks, among others.
- The combination of these technologies provides a versatile and flexible communication platform for the IoT.
- WNS also provides the means for remote management and control of IoT devices, making it possible to monitor and manage the devices from a central location.
- WNS is an essential component of the IoT architecture, providing the means for communication and connectivity between IoT devices.

- **Component of WNS:**

  - **Network management:** This component deals with the provisioning, configuration and monitoring of IoT devices, networks and applications.
  - **Connectivity management:** This component provides reliable and secure data transmission between IoT devices, gateways and back-end systems.
  - **Security management:** This component deals with the protection of IoT devices and networks against security threats such as hacking, data theft and unauthorized access.
  - **Device management:** This component provides a centralized platform for managing and monitoring the health, status and behaviour of IoT devices.

- **Application:** For Application of WNS refers to (Q.2) => Application of IOT.

Chapter: 2

(1) What is Sensor? Explain its specifications and its types.

- A sensor in IoT is a device that detects and measures physical properties such as temperature, light, humidity, pressure, etc. and converts them into electrical signal for processing and analysis.
- Sensor can be classified as active and passive sensor:
- Active sensor has its own energy source, such as a battery or an external power source. It generates an output signal actively and sends data to the receiving device. It is more expensive and high-power consuming.
- A passive sensor does not have its own energy sources and relies on the energy from physical phenomenon being measured. It is less expensive and less power consuming.
- In embedded system, sensor and actuator are used for controlling the system. Sensor is connected to input port. Actuators are connected to the output port.
- Sensor captures the changes in the environment variable. Middle system processes the information. Actuators are changed according to input variable. It displays the output.

❖ Specifications:

- **Sensitivity:** The minimum change in the physical property that the sensor can detect.
- **Accuracy:** The degree to which the sensor's measurement agrees with the actual value of the physical property being measured.
- **Range:** The minimum and maximum values of the physical property that the sensor can measure.
- **Response time:** The time takes by the sensor to respond to a change in the physical property is being measured.
- **Sampling Rate:** The number of measurements that a sensor can take per unit time.

- **Types of Sensors:**

  - **Temperature Sensor:** Measure temperature and convert it into electrical signal.
  - **Light Sensor:** Detect and measure light intensity and convert it into electrical signal.
  - **Humidity Sensor:** Measure the amount of moisture in the air and convert it into electrical signals.
  - **Pressure Sensor:** Measure the force applied per unit area and convert it into electrical signal.
  - **Motion Sensor:** Detect movement and convert it into electrical signal.
  - **Sound Sensor:** Detect and measure sound waves and convert them into electrical signal.
  - **Gas Sensor:** Detect and measure various gases and convert it into electrical signal.

(2) What is ARM? Explain special features of ARM processor.

- ARM stands for Advanced RISC Machine.
- ARM processor is a type of microprocessor that is commonly used in consumer electronic devices such as smartphones, wearable, and tablets.
- It is known for its low power consumption, compact size, efficient use of resources.

- The architecture of ARM processor is based on the reduced instruction set computing (RISC) principle, which simplifies the instruction set and reduce power consumption.
- The processor is customizable and can be used in various designs such as 32-bit devices and embedded systems.
- The main feature of ARM processor include support for multiprocessing, tightly coupled memory, efficient memory management, Thumb 2 Technology, one cycle executing time, pipelining and a large number of registers.
- **Multiprocessing Systems:** ARM processor are designed so that they can be used in case of multiprocessing systems where more than one processor are used to process information.
- **Tightly Coupled Memory:** Memory of ARM processor is tightly coupled. This has very fast response time. It has low latency and quick response time so that can be used in cases of cache memory.
- **Memory Management:** ARM processor has management section. This includes memory management unit and memory protection unit. This management system become very important in managing memory efficiently.
- **Thumb-2 Technology:** It was introduced in 2003 and was used to create variable length instruction set. It extends 16-buts instruction of initial thumb technology to 32-bit instructions. It has better performance than previously used in thumb technology.
- **One cycle execution time:** ARM processor is optimized for each instruction on CPU. Each instruction is of fixed length that allow time for fetching future instruction before executing present instruction.
- **Pipelining:** Processing of instruction is done in parallel using pipelines. Instructions are broken down and decoded in one pipeline stage. The pipeline advances one step at a time to increase throughput (rate of processing).
- **Large Number of Registers:** Large number of registers are used in ARM processor to prevent large amount of memory interaction. Register contains data and addresses. These act as local memory store for all operations.

(3) Explain Heart sensor, Heat sensor, Gyro Sensor, GPS Sensor and LDR Sensor.

➢ **Heartbeat Sensor:**

- A heartbeat sensor in the context of IoT refers to a device that measures the heart rate of a person.
- It typically uses photoplethysmography (PPG) or electrocardiography (ECG) to detect the electrical signals produced by the heart.
- The sensor is usually connected to a wearable device or smartphone and can be used to track heart health over time, monitor physical activity levels, and alert the user if there are any abnormal readings.
- The data collected by the heartbeat sensor is typically sent to a cloud-based server for analysis and storage, allowing users to track their heart health and access the data from multiple devices.
- In the context of IoT, heartbeat sensors are considered a type of wearable health monitoring device that can provide valuable insights into a person's health and wellness.

➢ **Heat Sensor:**

- A heat sensor is a type of temperature sensor that measures the temperature of a physical object or environment.
- They are commonly used in IoT applications to monitor temperature changes and detect anomalies.
- Heat sensors can be based on various technologies, including thermistors, thermocouples, and resistance temperature detectors (RTDs), and can have different form factors and measurement ranges.
- They can be integrated into a wide range of IoT devices, including home automation systems, environmental monitoring systems, and industrial control systems.

- **Gyro Sensor:**
  - A gyro sensor, also known as a gyroscope, is a device that measures the angular velocity of an object.
  - It typically uses a spinning wheel or a vibrating structure to measure changes in orientation, and is commonly used in navigation and control systems to provide stability and orientation information.
  - In the context of IoT, gyro sensors are often integrated into wearable devices, drones, and other mobile devices to provide information about their orientation and movement.
  - They can be used to track physical activity levels, detect changes in posture, and provide navigation information.
- **GPS Sensor:**
  - A GPS (Global Positioning System) sensor is a device that determines the location and time of a specific location on the Earth's surface.
  - It works by receiving signals from a network of satellites orbiting the Earth and then uses the triangulation method to determine the exact location.
  - GPS sensors are commonly used in navigation systems in vehicles, smartphones, and other consumer electronics devices. It provides information about the location, speed, and direction of travel.
- **Light Dependent Sensor:**
  - A Light Dependent Resistor (LDR) is a type of photoelectric sensor that measures light intensity and converts it into resistance. The resistance of an LDR decreases as the intensity of light increases and vice versa.
  - The LDR is connected in a voltage divider circuit, and the resistance changes will result in a change in the voltage across the LDR, which can be measured and used to determine the light level.
  - LDRs are widely used in various applications such as light control, automatic street lighting, and security systems.
  - In the Internet of Things (IoT), LDRs can be used in smart home applications for controlling lighting and measuring light levels for environmental monitoring.

(4) Difference Microprocessor vs Microcontroller.

Microprocessor	Microcontroller
Microprocessors Are Mainly Used in Computers. It Is the CPU Of the Computer. E.G 8085,8086 Etc.	Microcontrollers Are Used in Embedded Systems. Thus, It Is Like a Mini-Computer That Performs Its Own Tasks. E.G. 8051,8951 Etc.
Since It Is Only a Processor Hence Memory And Other Peripherals Are Connected Externally.	Peripherals Such As RAM, ROM, I/O Ports and Timers, Are In-Built in A Microcontroller.
However, The Overall Cost Of System Is High.	Thus, The Overall Cost of System Is Less.
Since Microprocessors Have External Components, Total Power Consumption Is High.	Since Microcontrollers Do Not Have Many External Components, Total Power Consumption Is Low.
Microprocessors Use External Busses to Access RAM, ROM, And Other Peripherals.	Microcontroller Uses an Internal Controlling Bus.
Microprocessors Have a Small Number of Registers Due To Which Operations Are Memory-Based.	Microcontroller Has More Registers Due to Which Programs Are Easier To Write In Them.
Microprocessors Do Not Have Power-Saving Features.	Microcontrollers Have Power-Saving Features.
Microprocessor Requires an External Memory for Program and Data Storage.	Microcontrollers Have an On-Chip Memory Embedded. Hence, It Does Not Require Any External Memory for Program and Data Storage.
Microprocessor Is Complex and Expensive.	Microcontroller Is Simple and Inexpensive.
Microprocessors Are Also Used for General Purpose Applications That Allow Us to Store Large Amounts Of Data.	Microcontrollers Are Thus Used for Application-Specific Systems.

Chapter: 3

(1) What is COAP? Explain its features with advantages and disadvantages.

- COAP stands for **Constrained Application Protocol** is specialized web transfer protocol use with constrained nodes and constrained networks.
- CoAP is designed for simplicity, low overhead and multicast support in resource constrained environment.
- CoAP is a web protocol that runs over UDP in Iot. Datagram Transport Layer Security (DTLS) is used to protect CoAP transmission.
- CoAP is simplified version of HTTP, designed for low-power devices such as sensor, microcontrollers and other constrained hardware.
- It is used for communication between devices on the same network, between devices and nodes on the internet, and between different networks.
- CoAP is not yet standard protocol, but the major standardization work is being done by the IETF CoRE Working Group.
- It is used in low power, lossy network and can also be used via other mechanism like SMS on mobile networks.
- **Features:**
  - It is very efficient RESTful protocol.
  - Easy to proxy to and from HTTP.
  - It is embedded web transfer protocol.
  - It uses asynchronous transaction model.
  - It uses small and simple 4-byte header.
  - Uses built in discovery mechanism.
  - Use four Method similar to HTTP: Get, Put, Post and Delete.
  - Four Message types: Confirmable (CON), Non-Confirmable (NON), Acknowledgement (ACK) and Reset (RST).
  - Use simple proxy and caching capabilities.

Advantage	Disadvantage
Runs over UDP avoid overhead of TCP.	Constrained associate with DTLS.
Easy to do HTTP – CoAP translation.	No standard framework for authorization.
Light weight application layer protocol.	No explicit support for ream time IoT application at present.

(2) Explain MQTT and quality of service with its characteristics.

- MQTT stands for Message Queuing Telemetry Transport.
- It is a publish-subscribe based messaging protocol for IoT communication. It provides way for IoT devices to efficiently send and receive data in real-time between each other and from/to central server.
- It is designed to be lightweight, low overhead and able to support limited-resources devices such as microcontrollers, sensors and mobile devices.
- In MQTT there are two main components: **Broker and Client.**
  - A broker, which acts as a central server. Broker ensure data is delivered to all interested parties in a secure and reliable manner.
  - A client which are IoT devices that send and receive data through broker. Client subscribes topic od interest and receive any data published to those topics.
- MQTT is widely adopted in IoT because of its low bandwidth usage, low latency, and efficient publish-subscribe mechanism.

- **<u>Characteristics:</u>**
  - MQTT uses a publish-subscribe model where devices can publish message to a server, and other devices can subscribe to receive those messages.
  - MQTT is a light weight protocol, making it well suited for devices with limited processing power and memory.
  - MQTT uses a compact binary format for message, which helps reduce network overhead and improve efficiency.
  - MQTT includes features like QoS levels and message acknowledgements, which help ensure reliable delivery of messages even in unreliable network condition.
  - MQTT is designed to be scalable, allowing large number of devices communicate with a single broker or multiple brokers to work together.
  - MQTT includes security like SSL/TSL encryption, username/password authentication, and access control.
  - MQTT is support variety of platforms, including Linux, Windows, IOS, Android.

- **Quality of Service (QoS):**

  - **QoS 0 (At Most Once):** The message is delivered at most once, and may be lost if the network is congested. It is the quickest and most efficient QoS level, but has the lowest level of reliability.

  - **QoS 1 (At Least Once):** The message is guaranteed to be delivered at least once, but may be duplicated in the event of a failure. This level of QoS is slower, but provides higher level of reliability than QoS 0.

  - **QoS 2 (Exactly Once):** The message is guaranteed to be delivered exactly once, but it is the slowest and most resource-intensive of the three QoS levels. It provides the highest level of reliability and is typically used for critical messages.

(3) Difference between CoAP and MQTT.

CoAP	MQTT
Stands for Constrained Application Protocol	Stands for Message Queuing Telemetry Transport.
It uses Request-Response model.	It uses Publish-Subscribe model.
This uses both asynchronous and synchronous.	This uses only Asynchronous.
Mains uses UDP.	Mainly uses TCP.
Header size is 4 bytes.	Header size is 2 bytes.
Uses REST principles.	Does not use REST principle.
It is used in utility area network and has secured mechanism.	It used in IoT applications and is secure.
Effectiveness in LNN is excellent.	Effectiveness in LNN is low.
Communication model is one-one.	Communication model is many-one.
Security type is DTLS.	Security type is SSL/TSL.

(4) Explain BLE and its component.

- BLE stands for Bluetooth Low Energy.
- It is a wireless communication technology designed for low-power, short-range communication.
- It is a variation of Bluetooth classic and uses less power, making it ideal for IoT device and wearable technology.
- BLE is used for variety of application including smart home devices, wearable technology, fitness tracker, healthcare devices and many more.
- Its low power consumption and short-range capabilities make it a popular choice for IoT application.
- BLW is a subset of classic Bluetooth technology, designed specifically for use in small. Low-power devices, making it ideal for IoT application.
- BLW operates in 2.4 GHz ISM band and has a range of approximately 5 meters in ideal connection.
- BLE offers three different communication models: Broadcast, point-to-point, and mesh networking, enabling it to be used for a wide range of IoT applications.
- **Client:** A device that initiates commands and requests, and accepts responses. Ex. Smartphone, Laptop.
- **Server:** A device that receives commands and requests, and returns responses. Ex. Heart rate monitor, Heat Sensor.
- **<u>Features:</u>**
    - Lowest power consumption.
    - Cost efficient and compatible.
    - Ease of use and integration.
    - Robustness, security and reliability.
    - License free
    - Support multi-brand mobile.

❖ **<u>Component:</u>**

<table>
<tr><td colspan="2">Applications</td></tr>
<tr><td colspan="2">Generic access profile</td></tr>
<tr><td colspan="2">Generic attribute profile</td></tr>
<tr><td>Attribute Protocol</td><td>Security Manager</td></tr>
<tr><td colspan="2">Logical link control and adaptation protocol</td></tr>
<tr><td colspan="2">Host Controller Interface</td></tr>
<tr><td colspan="2">Link Layer</td></tr>
<tr><td colspan="2">Physical Layer</td></tr>
</table>

- **Physical Layer:** The physical layer (PHY) in communication systems deals with the analog side of communication. It involves converting analog signals into digital form through modulation and demodulation, and using source coding technique.
- **Link Layer:** The Link layer in system combines hardware and software to handle the advertising, scanning, and connection management. It is the intermediary between the Physical layer and the rest of the communication stack.
- **Host Controller Interface:** It provides communication between controller and host through standard interface types. This HCI layer can be implemented either using API or by interfaces such as SPI/USB.
- **Logical Link Control and Adaptation Protocol:** This layer offer data encapsulation services to upper layers. This allows logical end to end data communication.
- **Attribute Control:** This layer allows BLE device to expose certain piece of data or attributes.
- **Security Manager:** This provides methods for device pairing and key distributions. It offers service to other protocol stack layers in order to securely connect and exchange data between BLE devices.
- **ATT:** Defines format of the data exposes its data to client and how this data is structured.
- **GATT:** Defines format of the data exposed by BLE devices. It also defines the procedures needed to access the data exposed by device.
- **Application Layer:** BLE protocol stack layers interact with application and profiles as desired. Application interoperability in the Bluetooth system is accomplished by Bluetooth profiles.

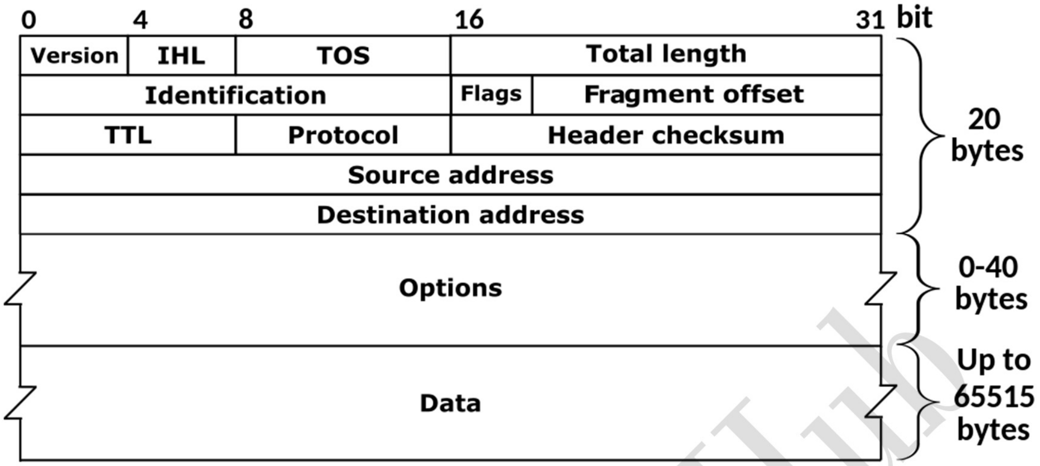
(5) Difference between Li-Fi and Wi-Fi.

Li-Fi	Wi-Fi
Transmit data using light with the help of LED bulbs.	Transmit data using radio waves with the help of Wi-Fi router.
Cannot penetrate through wall.	Can penetrate through wall.
Cover distance about 10 meters.	Covers range of 30 meters.
Power consumption is less.	Power consumption is more.
Speed of 1 Gbps in use commercially.	Speed up to 2 Gbps can be achieved commercially.
Uses Standard IEE 802.15.7	Uses standard UEEE 802.11
Works with high dense environment.	Works with low dense environment..

(6) Difference between IPV4 and IPV6.

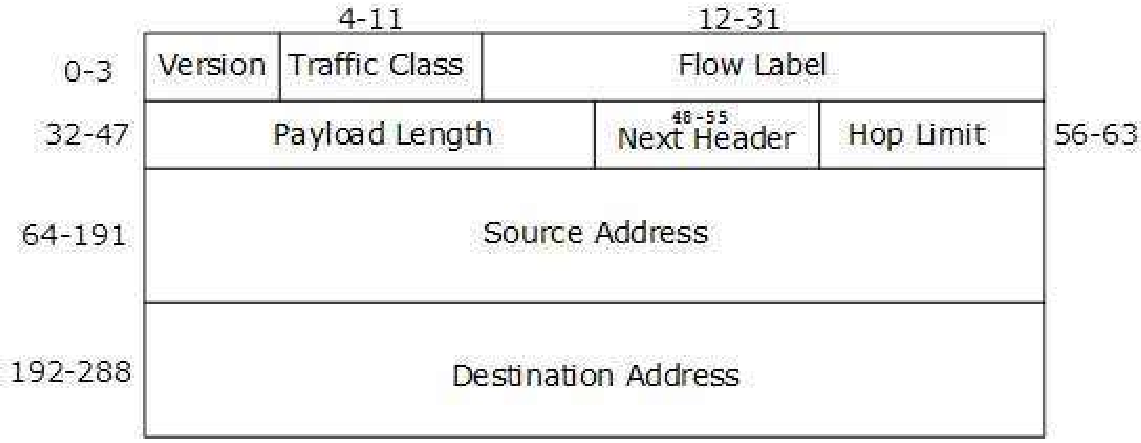
IPv4	IPv6
32-bit length address.	128-bit length address.
Supports manual and DHCP configuration.	Support auto and renumbering address configuration.
Address representation in decimal.	Address representation in hexadecimal.
Checksum field is available.	Checksum field is not available.
Has broadcast message transmission scheme.	Has multicast and anycast message transmission scheme.
Can not support real-time application.	Can support real-time application.
No security at network layer.	Provide security at network layer.
It has Limited number of IP Address.	It has large number of IP Address.
Does not provide encryption and authentication	It provides encryption and authentication.
IPv4 is broadcasting.	IPv6 is multicasting.
IPv4 can be converted to IPv6.	Not all IPv6 can be converted to IPv4.
IPv4 consist 4 field which is separated by (.).	IPv6 consists of 8 fields, which are separated by colon (:)
IPv4 divided into 5 classes: Class A to Class E.	IPv6 does not have any classes of IP Address.

(7) Explain IPv4 Header.



-
- **Version:** Version of IP Protocol which is 4 bits.
- **IHL:** Ip Header Length. Which is the number of 32-bit word in header. Minimum value of this field is 5 and maximum is 15.
- **Total Length:** Length of header + data, which has minimum value of 20 bytes and maximum is 65,535 bytes.
- **Identification:** Unique packet for identifying the group of fragments of single IP Datagram.
- **Flags:** 3 flags of 1 bit each: reserved bit, do not fragment flag, more fragment flag.
- **Fragment Offset:** Represent number of data bytes ahead of the particular fragment in the particular diagram. Specified in terms of number of 8 bytes.
- **Time to live:** Datagram lifetime is 8-bits, it prevents the datagram to loop through the network by restricting number of hops taken by a packet before delivering to destination.
- **Protocol:** Name of the protocol to which the data is to be passed.
- **Header Checksum:** 16 bits header checksum for checking errors in data gram header.
- **Source IP address:** 32 bits IP address of the sender.
- **Destination IP address:** 32 bits IP address of the receiver.
- **Option:** Optional information such as source route, record route. Used by the network administrator to check weather a path is working or not.

(8) Explain IPv6 Header.



- **Version (4-bits):** It represents the version of IP.
- **Traffic Class (8-bits):** These bits are divided into two parts. The most significant 6 bits are used for types of service to let router known what services should be provided to this packet. The least significant two bits are used for Explicit Congestion Notification (ECN).
- **Flow Label (20-bits):** This label is used to maintain the sequential flow of the packets belonging to a communication, The source labels the sequence to help the router identify that a particular packet belongs to a specific flow of information.
- **Payload Length (16-bits):** This field is used to tell the routers how much information a particular packet contains in its payload. Payload is composed of extension header and upper layer data.
- **Next Header (8-bits):** This is field is used to indicate either the type of extension header, or if the extension header is not present then it indicates the upper layer PDU. The values for upper layer PDU are same as IPv4.
- **Hop Limit (8-bits):** This field is used to stop packet to loop in network infinitely. This is same as TTL in IPv4. The value of HOP limit field is decremented by 1 as it passes a link. When the field reaches 0 the packet is discarded.
- **Source address (128-bits):** Indicates address of the sender.
- **Destination address (128-bits):** Indicates address of the receiver.

(9) Explain URI (Uniform Resources Identifier).

- **Definition:** A URI (Uniform Resource Identifier) is a string of characters that defines a standardized way of identifying resources on the internet. It provides a mechanism to locate, identify, and access these resources over the internet.
- **Purpose:** The purpose of a URI is to provide a unique identifier for resources on the internet, such as web pages, images, videos, and other digital assets. This allows the resources to be easily located, accessed, and manipulated by applications and systems that need to use them.
- **Types:** There are two main types of URIs - URLs (Uniform Resource Locators) and URNs (Uniform Resource Names). URLs are used to specify the location of a resource on the internet, while URNs provide a unique identifier for a resource without specifying its location.
- **Structure:** URIs consist of a scheme and a path. The scheme specifies the type of resource being identified (e.g., "http" for a web page, "ftp" for a file transfer), while the path provides the specific identifier for the resource (e.g., the URL "http://www.example.com" specifies the location of the web page "www.example.com" on the internet).
- **Relative vs. Absolute URIs:** URIs can be either relative or absolute. A relative URI refers to a resource in relation to the current location of the resource being accessed, while an absolute URI specifies the complete location of the resource.
- **Usage in Web Browsers:** URIs are widely used in web browsers to access web pages and other resources on the internet. When a user types a URI into a web browser, the browser sends a request to the server hosting the resource, which returns the requested information to the browser.
- **Usage in APIs:** URIs are also commonly used in APIs (Application Programming Interfaces) to provide access to resources over the internet. APIs allow applications and systems to interact with resources on the internet by sending and receiving information using URIs.
- **Encoding:** URIs can contain special characters that are not suitable for use in URLs. To handle this, special encoding rules are used to convert these characters into a form that can be safely included in a URL.

Chapter:4

(1) What is Cloud computing? Explain its component.

- Cloud computing involves storing and accessing data, software and processing resources through the internet instead of local computer.
- It provides access to data and applications from anywhere, while also offering enhanced innovation, scalability, and adaptive resources.
- The data and software is hosted by external parties on secure network location, making it easier for user to access and share information from multiple devices.
- This result in more efficient use of computing resources and simplified data management.
- Cloud computing refers to applications and services that run on a distributed network using virtualized resources and accessed by common internet protocols and networking standard.
- **Cloud Provider:** A cloud provider is the company that offers cloud computing service such as Amazon Web Services (AWS), Microsoft Azure, Google cloud platform.
- **Cloud Consumer:** It is the end-user who uses the cloud service provided by the cloud provider, either as an individual or and organization.
- **Service Owner:** It is responsible for managing the cloud service and ensuring it meets the need of consumer. They are responsible for providing technical support, ensuring security and privacy and ensuring service is reliable and available.
- **Resource administrator:** It is responsible for managing the resources used by the cloud service, including hardware, software and data. They also handle tasks such as capacity planning resource allocation and monitoring resources usage to ensure that the cloud service meets the needs of the consumer.

Architecture of Cloud Computing:

- Architecture is divided into two parts: Front-End & Back-end.
- **Front-End:** It is used by the client. It contains client-side interfaces and applications that are required to access the cloud computing platforms. The front end includes web server, thin & fat clients, tablet and mobiles.

- **Back-End:** It is used by the service provider. It manages all the resources that are required to provide cloud computing services. It includes a huge amount of data storage, security mechanism, virtual machines, deploying models, servers, traffic control mechanism, etc.

❖ **Components:**

- **Infrastructure as a Service (IaaS):** provides virtualized computing resources such as storage, servers, and networking.
- **Platform as a Service (PaaS):** provides a platform for the development, testing, and deployment of applications and services.
- **Software as a Service (SaaS):** provides software applications over the Internet, often on a subscription basis.
- **Data as a Service (DaaS):** provides access to data through the cloud, often through APIs and data analytics services.
- **Function as a Service (FaaS):** provides a way to execute functions in the cloud without managing infrastructure.
- **Client Infrastructure:** It is a front-end component, It provides GUI (Graphical User Interface) to interact with the cloud.
- **Application:** It may be any software or platform that clients want to access.
- **Storage:** It is one of the most important components of cloud computing. It provides a huge amount of storage capacity in the cloud to store and manage data.
- **Management:** It is used to manage components such as application, service, run-time cloud, storage and other security issues in the backend and establish coordination between them.
- **Security:** It is an in-built backend component of cloud computing. It implements a security mechanism in the backend.
- **Internet:** It is medium through which front end and backend can interact and communicate with each other.

(2) Explain Cloud deployment model.

- Cloud deployment models are refers to the location and management of the cloud's infrastructure.
- Deployment models are defined by the ownership and control architectural design and the degree of available customization. Cloud deployment models are private, public, hybrid and community clouds.
- **Public Cloud:** The cloud infrastructure is made available to the general public or large industry group and is owned by an organization selling cloud services.
  - Public cloud is a huge data center that offers the same service to all its users. The services are accessible for everyone and much used for consumer segment.
  - Advantage: Minimal investment, No setup cost, No maintenance
  - Disadvantage: Less secure, Low customization
- **Private Cloud:** It is a cloud computing service that is dedicated to a single organization. This type of deployment is usually owned and managed by the organization itself, and is designed to meet the specific needs of the organization.
  - Advantage: Better control, Data security and privacy, Customization
  - Disadvantage: Less scalable, Costly
- **Hybrid Cloud:** It is a combination of public and private clouds, and designed to provide the benefits of both types of deployments. This type of deployment allow organization to use public cloud resources for non-sensitive applications and data, while still retaining control over critical and sensitive data and applications by using a private cloud.
  - Advantage: Flexible and control, High secure, only pay when used
  - Disadvantage: Difficult to manage, slow data transmission
- **Community Cloud:** It is shared by serval organizations and supports a specific community that has shared concerns and tasks. It may be managed by the organizations or a third party and may exist on-premises or off-premises.
  - Advantage: Cost effective, Security, Shared resources
  - Disadvantage: Limited scalability, Rigid in customization

PUBLIC	PRIVATE	HYBRID
It is cost-effective.	It is expensive.	It is expensive.
Security depends on the cloud provider.	It is most secure.	It is secure.
It is highly scalable.	Scalability is limited.	Highly scalable with right architecture.
Accessibility is to everyone.	Accessibility to limited.	Accessibility to medium.
Low maintenance.	Highest maintenance.	Medium maintenance.
Owned by cloud provider.	Owned by organization.	Owned by organization.
Ex. AWS, Azure, IBM	Ex. HPE, Dell, Open stack	Ex. Rackspace, VMware

(3) Explain Challenges and application of cloud computing.

- **Challenges:**
  - **Security:** Protecting sensitive data and ensuring privacy is a major challenge for cloud computing, as cloud-based systems store large amounts of sensitive data.
  - **Interoperability:** Interoperability issues can arise when different cloud services are used and need to communicate with each other.
  - **Data Loss:** Data loss is a significant concern, as the data is stored in remote servers and can be lost due to system failures or data breaches.
  - **Performance:** Performance can be an issue in cloud computing as the cloud infrastructure may be located far away from the end-user.
  - **Reliability:** Reliability is another challenge in cloud computing, as the cloud-based systems can suffer from downtime, network issues, or hardware failures.
  - **Compliance:** Compliance with regulations and standards, such as data protection regulations, is a challenge in cloud computing.
  - **Cost:** Cost is a challenge in cloud computing, as it can be difficult to predict and manage costs, especially in multi-tenant and pay-per-use cloud models.

Applications:

- **Business Processes:** Cloud computing enables businesses to automate and streamline their operations, from customer relationship management to supply chain management.
- **Data Analytics:** Cloud computing provides scalable, cost-effective and flexible data storage and processing capabilities, allowing businesses to store and analyse large amounts of data to make informed decisions.
- **Web-based Applications:** Cloud computing provides a platform for developing and deploying web-based applications, which can be accessed from anywhere with an internet connection.
- **Disaster Recovery and Business Continuity:** Cloud computing offers a secure and reliable platform for backing up and recovering critical data and applications in the event of a disaster or system failure.
- **Collaboration and Productivity:** Cloud computing enables employees to collaborate on projects and share files in real-time, leading to increased productivity and teamwork.
- **Education and E-Learning:** Cloud computing provides a platform for delivering online courses, enabling students to access educational content
- **Gaming:** Cloud gaming services enable users to play video games on their devices without the need for high-performance hardware, as the games are run on remote servers.
- **Government and Public Sector:** Cloud computing provides government agencies with a platform for storing, managing, and sharing sensitive data and information, improving service delivery and reducing costs.

(4) What is Fog Computing? Explain its characteristics and its advantages.

- Fog computing is a decentralized way of processing data and providing services closer to the data source, between the source and the cloud. It helps connect IoT devices to cloud computing smoothly.
- It is used when only selected data is required to send to the cloud. This selected data us chosen for long-term storage and is less frequently accessed by the host.
- It is used when the data should be analyzed within a fraction of seconds.
- It is used whenever large number of services need to be provided over a large area at different geographical locations.
- **Characteristics:**
  - **Distributed Nature:** Fog computing operates at the edge of the network, close to IoT devices and sensors.
  - **Latency-Sensitive:** It aims to reduce latency by processing data closer to the edge, reducing the need to transmit large amounts of data to cloud.
  - **Efficient use of Resources:** Fog computing nodes are designed to be lightweight and resource-efficient, allowing for efficient deployment scale.
  - **Flexibility:** Fog computing can be deployed in a variety of network environments, making it adaptable to different use case and requirements.
- **Advantages:**
  - **Improved performance:** Fog computing can reduce the latency associated with cloud computing by processing data closer to the edge.
  - **Increased security:** Sensitive data can be processed locally, reducing the risk of data breaches and unauthorized access.
  - **Enhanced scalability:** Fog computing enables efficient scaling of compute and storage resources, making it suitable for use in large-scale IoT deployments.
  - **Cost-effectiveness:** Fog computing reduces the cost associated with transmitting large amounts of data over long distances, making it a cost-effective solution for IoT deployments.

Chapter: 5

(1) Explain IoT application in food.

➢ IoT applications in food industry involve the use of IoT devices and technologies to improve various processes in food production, distribution, and consumption.

- **Food Traceability**: IoT sensors and devices can be used to track food products from farm to table, providing transparency and accountability in the food supply chain.
- **Inventory Management:** IoT devices can monitor food inventory levels and alert suppliers and distributors when it's time to restock, reducing waste and maximizing efficiency.
- **Agricultural Monitoring:** IoT sensors and devices can be used to monitor soil moisture, temperature, and other environmental conditions in agriculture, helping farmers make informed decisions about when to plant, irrigate, and harvest crops.
- **Quality Control:** IoT sensors and devices can be used to monitor the temperature and humidity of food products during storage and transportation, ensuring food safety and quality.
- **Smart Kitchens:** IoT devices can be used to monitor the cooking process, reducing energy consumption and enabling the creation of customized, healthy and balanced meals.
- **Smart packaging:** IoT sensors can be integrated into food packaging, providing information on product freshness and expiration dates, as well as helping to prevent counterfeiting and fraud.
- **Food safety monitoring:** IoT devices can be used to monitor temperature and other conditions of food products to ensure that they are stored, transported and cooked at safe levels, reducing the risk of food-borne illnesses.

(2) Explain IoT application in HealthCare.

- IoT technology brings numerous applications in healthcare, from remote monitoring to smart sensors to medical device integration. It keeps the patients safe and healthy as well as improves the physician delivers care towards patients.
    - **Wearable devices:** Wearable IoT devices such as fitness trackers, smartwatches, and heart rate monitors help people monitor their health and fitness levels. They can also track vital signs such as heart rate, sleep patterns, and steps taken.
    - **Remote patient monitoring:** IoT-powered devices can be used to monitor patients remotely, which can help reduce hospital stays and improve care for patients with chronic conditions.
    - **Electronic health records:** IoT can be used to securely store and access electronic health records, which can improve the accuracy of patient data and reduce the risk of medical errors.
    - **Telemedicine:** IoT can be used to connect patients with healthcare providers remotely, which can help improve access to care and reduce costs.
    - **Medical equipment management:** IoT can be used to monitor and manage medical equipment such as ventilators, infusion pumps, and imaging machines, which can improve the efficiency of care delivery and reduce equipment downtime.
    - **Clinical decision support:** IoT can be used to provide real-time data to healthcare providers, which can help inform clinical decision-making and improve patient outcomes.
    - **Clinical decision support systems:** IoT sensors and devices can gather data on patients, including vital signs and medical histories, and use that data to provide real-time recommendations to healthcare providers.
    - **Asset tracking:** IoT devices can be used to track and manage medical equipment, supplies, and drugs, helping to improve efficiency and reduce waste.

(3) Explain IoT application in warehouse.

- Many organizations are investing in IoT-enable warehouse for better automated control system (ACS) and warehouse management system (WMS) to improve their operational efficiency by reducing cost.
- IoT enabled warehouse gives businesses real-time data on product locations, transportation details, packaging, and routing. Due to this instant update, store manager ensures no inventory is lost during transportation. Also, they ensure supply chain vendors manage deliveries responsibly.
    - **Inventory Management:** IoT sensors and devices can be used to track the real-time location and movement of products, thereby improving inventory accuracy and reducing the risk of stock-outs.
    - **Asset Tracking:** IoT devices can be used to track the movement of assets like pallets, containers, and trucks within the warehouse, providing real-time visibility into their location and status.
    - **Predictive Maintenance:** IoT sensors can be used to monitor the health of warehouse equipment, such as conveyor systems, forklifts, and cranes, and predict when maintenance is required.
    - **Environmental Monitoring:** IoT sensors can be used to monitor and control temperature, humidity, and other environmental factors that affect product quality and safety.
    - **Labour Management:** IoT devices can be used to monitor the performance and productivity of workers, enabling real-time decision making and improving overall warehouse efficiency.

(4) Explain IoT application in retail with Inventory and Smart Payment.

- The Internet of Things (IoT) has revolutionized the retail industry by enabling retailers to collect and analyse massive amounts of data from various sources, including customers, devices, and sensors.
- IoT has brought about new opportunities for retailers to improve inventory management, enhance the shopping experience, and boost revenue.
- **Inventory Management:** IoT devices like RFID (Radio-Frequency Identification) sensors, barcode scanners, and beacons can be used to track the movement of products in real-time and provide accurate inventory information to retailers. This helps retailers to avoid overstocking and stockouts, leading to better inventory management and increased profits.
- **Smart Payment:** IoT enabled smart payment systems like contactless payments, mobile payments, and wearable payments allow customers to make purchases faster and more securely. This improves the shopping experience and increases the speed of checkout, leading to higher customer satisfaction and reduced checkout lines.
- **Customer Engagement:** IoT devices like beacons, smartphones, and in-store sensors can be used to engage with customers and provide them with relevant information and offers in real-time. This helps to increase customer engagement and build brand loyalty.
- **Smart Shelves:** IoT-enabled smart shelves can be used to monitor the levels of stock and automatically reorder items when they are running low. This ensures that retailers never run out of stock and can provide a seamless shopping experience to customers.
- **Smart Carts:** IoT-enabled smart carts help shoppers to keep track of their purchases as they move around the store. The carts use RFID technology to automatically add items to the shopping list and track the total cost.

(5) Explain IoT application in Driver Assistance.

- The IoT technology has been widely adopted in the automotive industry, leading to the development of smart driver assistance systems.
- These systems use sensors ana other connected device to provide real-time information to drivers and improve their overall driving experience.

  - **Navigation and mapping:** IoT-powered navigation systems use real-time traffic data to provide drivers with the most efficient route to their destination. Additionally, these systems can provide alerts for road conditions, traffic congestion, and construction, helping drivers to avoid delays.
  - **Parking assistance:** IoT sensors and cameras can help drivers find available parking spaces, avoid congestion and reduce the amount of time spent searching for a spot.
  - **Blind spot monitoring:** IoT sensors can detect vehicles in a driver's blind spot, alerting them to potential dangers and helping to prevent accidents.
  - **Driver fatigue monitoring:** IoT devices can monitor a driver's eye movements and drowsiness levels, providing them with reminders to rest when necessary.
  - **Vehicle diagnostics:** IoT technology can also be used to monitor the performance of a vehicle, alerting drivers to potential problems and helping to keep their vehicles in good working order.

Chapter: 6

(1) What is Arduino? explain its features.

- Arduino is a project, open-source hardware, and software platform used to design and build electronic devices. It designs and manufactures microcontroller kits and single-board interface for building electronic project.
- Arduino boards were initially created to help the students with the non-technical background.
- The design of Arduino boards uses a variety of controller and microprocessor.
- It is an open-source electronics platform that is designed for creating interactive electronic devices, such as robots, home automation systems, and wearable technologies.
- **Features:**
  - **Open-source platform:** Arduino is an open-source platform that encourages users to contribute to the development of its hardware and software components. This has created a large community of users who share their projects, tutorials, and code examples.
  - **Easy-to-use programming language:** Arduino uses a simplified version of C++, making it easy to learn and use for those who have limited programming experience.
  - **Wide variety of hardware components:** There are many different hardware components available for the Arduino platform, including sensors, actuators, motors, and displays.
  - **Cross-platform compatibility:** Arduino can be used with a wide range of operating systems, including Windows, MacOS, and Linux, making it a versatile platform for users.
  - **Strong community support:** There is a large and active community of users who are always willing to help with questions, provide tutorials and support, and share their projects and code.
  - **Cost-effective:** Arduino hardware components are affordable and easily accessible, making it a cost-effective solution for individuals and small businesses who want to build their own digital devices.

(2) Explain Raspberry Pi and its Interface.

- Raspberry Pi is a small, low-cost, single-board computer that was first introduced in 2012.
- It was created with the goal of promoting the teaching of computer science and programming in schools, but it has since become popular among makers and developer for variety of projects.
- The raspberry Pi is slower than modern laptop and computer but is still a complete Linux computer and can provide all the expected abilities to implies, at a low-power consumption level.

- **INTERFACES:**

- **GPIO (General Purpose Input/Output) -** The Raspberry Pi has 40 GPIO pins that can be used to control and interface with external devices, such as LEDs, sensors, and motors.
- **Display -** The Raspberry Pi has a dedicated display interface for connecting an HDMI monitor, as well as a Composite Video output for older CRT televisions.
- **Audio -** The Raspberry Pi has a 3.5mm audio jack that can be used to output audio to a speaker or connect headphones.
- **USB -** The Raspberry Pi has four USB 2.0 ports that can be used to connect a variety of devices, such as keyboards, mice, and Wi-Fi adapters.
- **Ethernet -** The Raspberry Pi has a 10/100 Ethernet port that can be used to connect to the internet or other networks.
- **Camera -** The Raspberry Pi has a dedicated camera interface that allows the connection of a Raspberry Pi camera module for use in projects such as time-lapse photography or security cameras.
- **SD Card -** The Raspberry Pi uses a micro-SD card for its operating system and data storage.

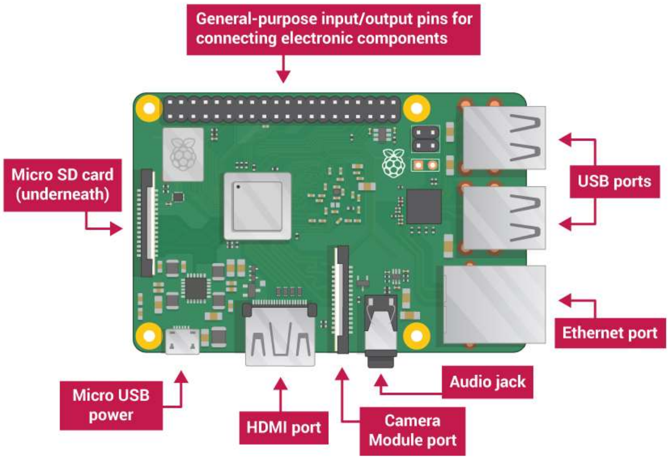
(3) Explain Interface of LED with raspberry Pi.

- Interfacing an LED with Raspberry Pi involves connecting the LED to the GPIO pins of the Raspberry Pi and then controlling the LED using a programming language such as Python. The steps for interfacing an LED with Raspberry Pi are:

  - Connect the positive leg of the LED to a GPIO pin on the Raspberry Pi.
  - Connect the negative leg of the LED to a resistor, which is then connected to the ground pin of the Raspberry Pi.
  - Install the required software and libraries, such as the Raspberry Pi GPIO library, which will be used to control the GPIO pins.
  - Write a Python program that sets the GPIO pin connected to the LED to output mode and then turns the LED on and off by setting the pin to high or low respectively.
  - Run the program and observe the LED turning on and off according to the program.

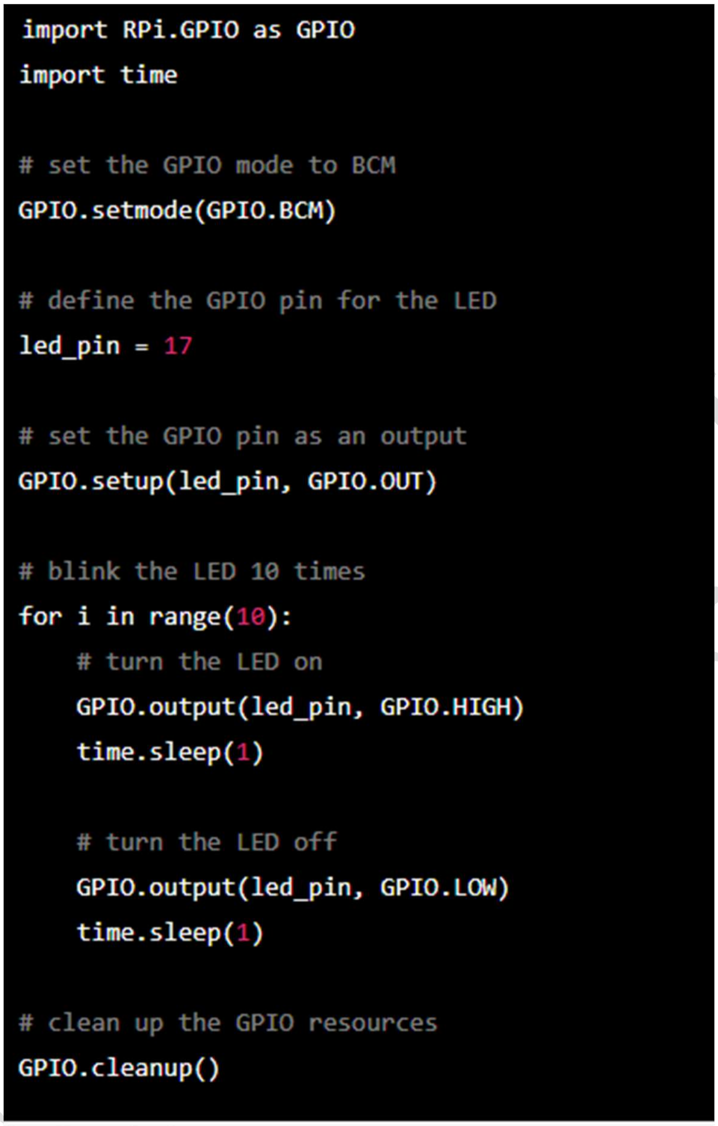
- By interfacing an LED with Raspberry Pi, you can create simple electronic projects, such as a blinking LED, that can be used as a building block for more complex projects. Additionally, by controlling the GPIO pins, you can interface with a wide variety of sensors and other electronic components.

(4) Explain Raspberry Pi diagram and explain its component.


General-purpose input/output pins for connecting electronic components
Micro SD card (underneath)
USB ports
Ethernet port
Audio jack
Micro USB power
HDMI port
Camera Module port

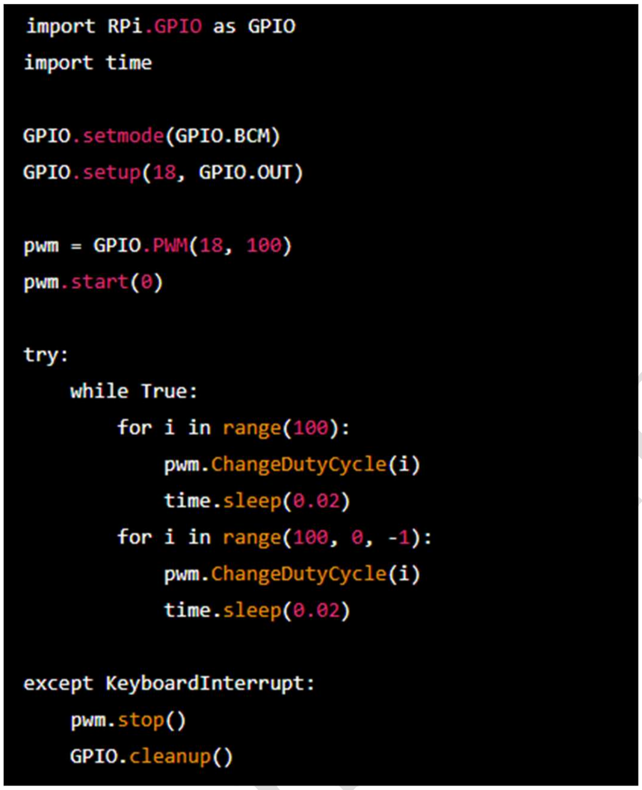
- **Central Processing Unit (CPU):** The CPU is the main processor that runs the operating system and other software applications on the Raspberry Pi.
- **Graphics Processing Unit (GPU):** The GPU is responsible for rendering graphics and video on the Raspberry Pi.
- **Memory (RAM):** The Raspberry Pi has a fixed amount of memory (RAM) which is used for temporary storage of data.
- **USB Ports:** The Raspberry Pi has multiple USB ports for connecting devices such as keyboards, mice, and storage devices.
- **Ethernet Port:** The Ethernet port provides connectivity to a local network or the internet.
- **HDMI Port:** The HDMI port allows the Raspberry Pi to be connected to a display device.
- **Audio Jack:** The audio jack provides audio output.
- **GPIO Pins:** The GPIO (General Purpose Input/Output) pins are used for connecting and controlling peripheral devices, such as sensors and actuators.
- **Micro-SD Card Slot:** The Micro-SD card slot is used for loading the operating system and storing data.
- **Power Input:** The Raspberry Pi is powered by a 5V DC power supply, which is connected to the board through the power input.

(5) Write Raspberry Pi code to blink LED ON and OFF.



```
import RPi.GPIO as GPIO
import time

# set the GPIO mode to BCM
GPIO.setmode(GPIO.BCM)

# define the GPIO pin for the LED
led_pin = 17

# set the GPIO pin as an output
GPIO.setup(led_pin, GPIO.OUT)

# blink the LED 10 times
for i in range(10):
    # turn the LED on
    GPIO.output(led_pin, GPIO.HIGH)
    time.sleep(1)

    # turn the LED off
    GPIO.output(led_pin, GPIO.LOW)
    time.sleep(1)

# clean up the GPIO resources
GPIO.cleanup()
```

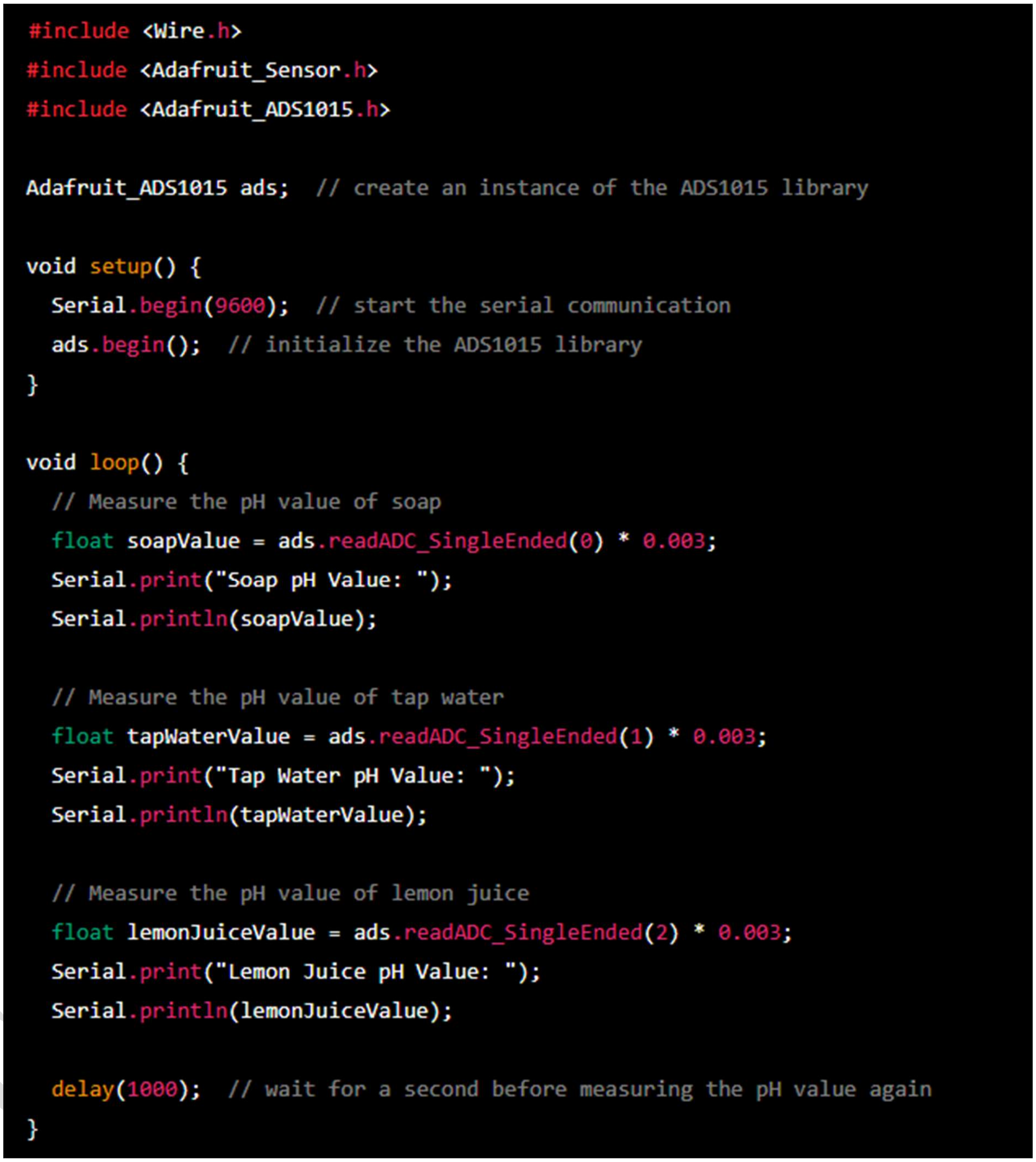
(6) Write Raspberry Pi code to change the brightness of LED.



```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)
GPIO.setup(18, GPIO.OUT)

pwm = GPIO.PWM(18, 100)
pwm.start(0)

try:
    while True:
        for i in range(100):
            pwm.ChangeDutyCycle(i)
            time.sleep(0.02)
        for i in range(100, 0, -1):
            pwm.ChangeDutyCycle(i)
            time.sleep(0.02)

except KeyboardInterrupt:
    pwm.stop()
    GPIO.cleanup()
```

- This code uses the ‘**RPi.GPIO’** library to control the GPIO pins of the Raspberry Pi. The ‘**GPIO.setup’** function sets the GPIO pin 18 as an output.
- The **‘GPIO.PWM’** function creates a PWM (pulse width modulation) object that can be used to control the brightness of the LED.
- The **‘pwm.start’** function starts the PWM with a duty cycle of 0, meaning the LED is off.
- The for loops in the code increase and decrease the duty cycle, causing the LED to change its brightness.
- The **‘time.sleep’** function provides a delay between the changes in brightness.
- The try statement and except clause are used to stop the PWM and clean up the GPIO pins when the user presses **CTRL + C.**

(7) Write Arduino code to identify PH value of Soap , Tap water and Lemon juice using PH sensor.



```
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_ADS1015.h>

Adafruit_ADS1015 ads;  // create an instance of the ADS1015 library

void setup() {
  Serial.begin(9600);  // start the serial communication
  ads.begin();  // initialize the ADS1015 library
}

void loop() {
  // Measure the pH value of soap
  float soapValue = ads.readADC_SingleEnded(0) * 0.003;
  Serial.print("Soap pH Value: ");
  Serial.println(soapValue);

  // Measure the pH value of tap water
  float tapWaterValue = ads.readADC_SingleEnded(1) * 0.003;
  Serial.print("Tap Water pH Value: ");
  Serial.println(tapWaterValue);

  // Measure the pH value of lemon juice
  float lemonJuiceValue = ads.readADC_SingleEnded(2) * 0.003;
  Serial.print("Lemon Juice pH Value: ");
  Serial.println(lemonJuiceValue);

  delay(1000);  // wait for a second before measuring the pH value again
}
```

Chapter:7

(1) Explain architecture of IOT security.

- The architecture of IoT security can be divided into several layers, including device security, network security, cloud security, and application security.
    - **Device security:** This layer focuses on securing individual IoT devices, including sensors and actuators, from unauthorized access, tampering, and malware. It includes hardware-based security measures such as secure boot, trusted execution environments, and hardware-based encryption.
    - **Network security:** This layer focuses on securing the communication between IoT devices and the cloud, as well as protecting the data that is transmitted between these devices. It includes measures such as encryption, authentication, and access control.
    - **Cloud security:** This layer focuses on securing the cloud infrastructure that stores and processes IoT data. It includes measures such as data protection, access control, and threat detection.
    - **Application security:** This layer focuses on securing the applications that run on the cloud and process IoT data. It includes measures such as secure coding practices, secure data storage, and access control.
- Each layer of the IoT security architecture is critical to protecting IoT devices, networks, and data from malicious attacks and ensuring the confidentiality, integrity, and availability of data. The implementation of these security measures should be done in a systematic and comprehensive manner to ensure the security of the entire IoT system.

(2) Explain IOT security Challenges.

- **Data privacy and protection:** As IoT devices collect and transmit sensitive data, it is important to protect this data from unauthorized access or theft. This includes ensuring secure communication between devices, using encryption, and implementing proper authentication methods.
- **Device security:** IoT devices often have limited processing power and memory, making them vulnerable to hacking and malware attacks. Ensuring that devices are secure and free from vulnerabilities is critical to protecting the overall network.
- **Network security:** IoT networks are often decentralized and spread across multiple locations, making them difficult to secure. This requires implementing robust security measures at each layer of the network, including firewalls, intrusion detection and prevention systems, and secure protocols.
- **Cloud security:** IoT data is often stored and processed in the cloud, which requires additional security measures to protect against data breaches and unauthorized access.
- **Interoperability:** Ensuring that different IoT devices can communicate and work together is a critical component of IoT security. This requires standardizing protocols and ensuring that all devices are compatible with each other.
- **Human factor:** People play an important role in IoT security, and it is important to educate users on how to securely use IoT devices and follow best practices to minimize security risks.
- **Software and firmware updates:** As IoT devices and software evolve, it is important to ensure that devices receive timely and secure updates to address security vulnerabilities and protect against attacks.

(3) Explain IOT Security algorithm.

- **AES (Advanced Encryption Standard)**: It is a symmetric encryption algorithm that is used to encrypt and decrypt data in IoT systems. It uses a shared secret key that must be known by both the sender and the receiver.
- **RSA (Rivest–Shamir–Adleman):** It is an asymmetric encryption algorithm that uses a pair of public and private keys for encryption and decryption. The public key is used to encrypt the data, while the private key is used to decrypt it.
- **SHA (Secure Hash Algorithm):** It is a cryptographic hash function that is used to generate a fixed-length digest of a message. The digest is used to verify the integrity of the message and to detect any changes made to the original message.
- **Elliptic Curve Cryptography (ECC):** It is a public-key cryptography algorithm that is used to provide secure communication over the internet. It provides high security with low computational overhead and is particularly useful for IoT devices with limited processing power.
- **HMAC (Hashed Message Authentication Code):** It is a keyed-hash message authentication code that is used to ensure the authenticity and integrity of data in IoT systems. The HMAC is generated using a secret key and a hash function, and it is included in the message along with the data.