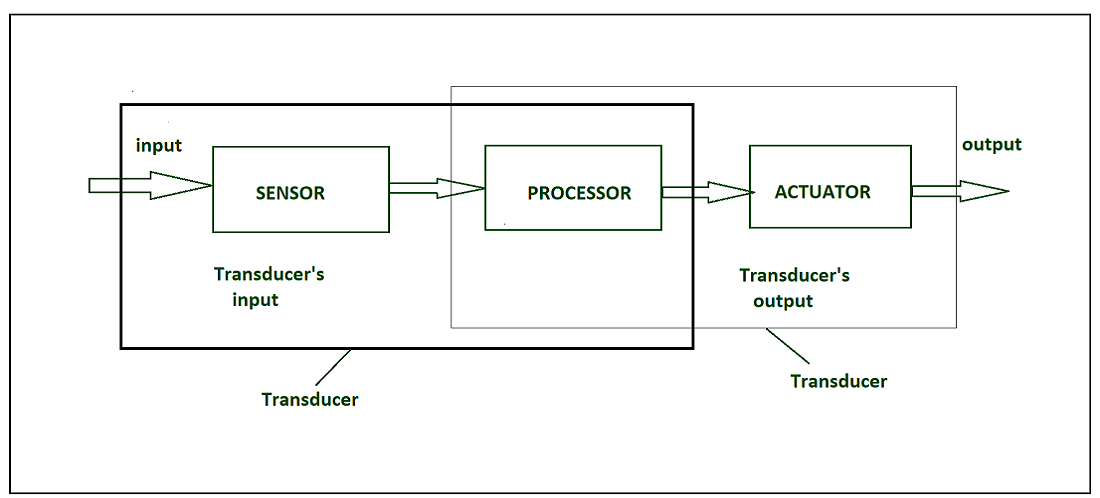
**Hardware for IoT**

**Sensors in Internet of Things(IoT)**

**Sensors** are used for sensing things and devices etc.

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

****

**Transducer :**

* A transducer converts a signal from one physical structure to another.
* It converts one type of energy into another type.
* It might be used as actuator in various systems.

**Sensors characteristics :**

1. Static
2. Dynamic

**1. Static characteristics :**  
It is about how the output of a sensor changes in response to an input change after steady state condition.

* **Accuracy:**Accuracy is the capability of measuring instruments to give a result close to the true value of the measured quantity. It measures errors. It is measured by absolute and relative errors. Express the correctness of the output compared to a higher prior system. Absolute error = Measured value – True value  
  Relative error = Measured value/True value
* **Range:**Gives the highest and the lowest value of the physical quantity within which the sensor can actually sense. Beyond these values, there is no sense or no kind of response.  
  e.g. RTD for measurement of temperature has a range of -200`c to 800`c.
* **Resolution:**Resolution is an important specification for selection of sensors. The higher the resolution, better the precision. When the accretion is zero to, it is called the threshold.  
  Provide the smallest changes in the input that a sensor is able to sense.
* **Precision:**It is the capacity of a measuring instrument to give the same reading when repetitively measuring the same quantity under the same prescribed conditions.  
  It implies agreement between successive readings, NOT closeness to the true value.  
  It is related to the variance of a set of measurements.  
  It is a necessary but not sufficient condition for accuracy.
* **Sensitivity:**Sensitivity indicates the ratio of incremental change in the response of the system with respect to incremental change in input parameters. It can be found from the slope of the output characteristics curve of a sensor. It is the smallest amount of difference in quantity that will change the instrument’s reading.
* **Linearity:**The deviation of the sensor value curve from a particularly straight line. Linearity is determined by the calibration curve. The static calibration curve plots the output amplitude versus the input amplitude under static conditions.   
  A curve’s slope resemblance to a straight line describes linearity.
* **Drift:**The difference in the measurement of the sensor from a specific reading when kept at that value for a long period of time.
* **Repeatability:**The deviation between measurements in a sequence under the same conditions. The measurements have to be made under a short enough time duration so as not to allow significant long-term drift.

**Dynamic Characteristics :**  
Properties of the systems

* **Zero-order system:**The output shows a response to the input signal with no delay. It does not include energy-storing elements.  
  Ex. potentiometer measure, linear and rotary displacements.
* **First-order system:**When the output approaches its final value gradually.  
  Consists of an energy storage and dissipation element.
* **Second-order system:**Complex output response. The output response of the sensor oscillates before steady state.

**Sensor Classification :**

* Passive& Active
* Analog & digital
* Scalar & vector

1. **Passive Sensor –**  
   Can not independently sense the input. Ex- Accelerometer, soil moisture, water level and temperature sensors.
2. **Active Sensor –**  
   Independently sense the input. Example- Radar, sounder and laser altimeter sensors.
3. **Analog Sensor** **–**  
    The response or output of the sensor is some continuous function of its input parameter. Ex- Temperature sensor, LDR, analog pressure sensor and analog hall effect.
4. **Digital sensor –**  
   Response in binary nature. Design to overcome the disadvantages of analog sensors. Along with the analog sensor, it also comprises extra electronics for bit conversion. Example – Passive infrared (PIR) sensor and digital temperature sensor(DS1620).
5. **Scalar sensor –**   
   Detects the input parameter only based on its magnitude. The answer for the sensor is a function of magnitude of  some input parameter. Not affected by the direction of input parameters.  
   Example – temperature, gas, strain, color and smoke sensor.
6. **Vector sensor –**  
   The response of the sensor depends on the magnitude of the direction and orientation of input parameter. Example – Accelerometer, gyroscope, magnetic field and motion detector sensors.

### Types of sensors –

* **Electrical sensor :**

Electrical proximity sensors may be contact or non contact.

Simple contact sensors operate by making the sensor and the component complete an electrical circuit.

Non- contact electrical proximity sensors rely on the electrical principles of either induction for detecting metals or capacitance for detecting non metals as well.

* **Light sensor:**

Light sensor is also known as photo sensors and one of the important sensor.

Light dependent resistor or LDR is a simple light sensor available today.

The property of LDR is that its resistance is inversely proportional to the intensity of the ambient light i.e when the intensity of light increases, it’s resistance decreases and vise versa.

* **Touch sensor:**

Detection of something like a touch of finger or a stylus is known as touch sensor.

It’s name suggests that detection of something.

They are classified into two types:

1. Resistive type
2. Capacitive type

Today almost all modern touch sensors are of capacitive types.

Because they are more accurate and have better signal to noise ratio.

* **Range sensing:**

Range sensing concerns detecting how near or far a component is from the sensing position, although they can also be used as proximity sensors.

 Distance or range sensors use non-contact analog techniques. Short range sensing, between a few millimetres and a few hundred millimetres is carried out using electrical capacitance, inductance and magnetic technique.

 Longer range sensing is carried out using transmitted energy waves of various types eg radio waves, sound waves and lasers.

* **Mechanical sensor:**

Any suitable mechanical / electrical switch may be adopted but because a certain amount of force is required to operate a mechanical switch it is common to use micro-switches.

* **Pneumatic sensor:**

These proximity sensors operate by breaking or disturbing an air flow.

 The pneumatic proximity sensor is an example of a contact type sensor. These cannot be used where light components may be blown away.

* **Optical sensor:**

In there simplest form, optical proximity sensors operate by breaking a light beam which falls onto a light sensitive device such as a photocell. These are examples of non contact sensors. Care must be exercised with the lighting environment of these sensors for example optical sensors can be blinded by flashes from arc welding processes, airborne dust and smoke clouds may impede light transmission etc.

* **Speed Sensor:**

Sensor used for detecting the speed of any object or vehicle which is in motion is known as speed sensor .For example – Wind Speed Sensors, Speedometer ,UDAR ,Ground Speed Radar .

* **Temperature Sensor:**

Devices which monitors and tracks the temperature and gives temperature’s measurement as an electrical signal are termed as temperature sensors .These electrical signals will be in the form of voltage and is directly proportional to the temperature measurement .

* **PIR Sensor:**

PIR stands for passive infrared sensor and it is an electronic sensor that is used for the tracking and measurement of infrared (IR) light radiating from objects in its field of view and is also known as Pyroelectric sensor .It is mainly used for detecting human motion and movement detection .

* **Ultrasonic Sensor:**

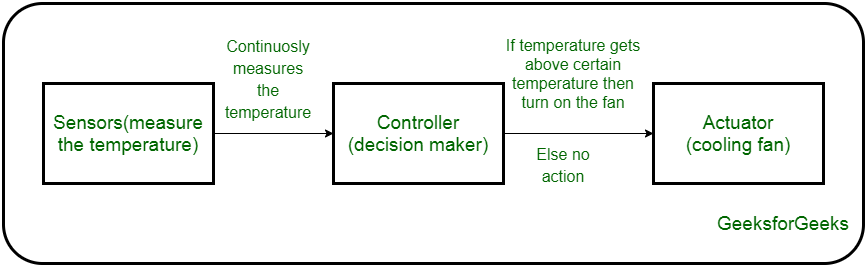
The principle of ultrasonic sensor is similar to the working principle of SONAR or RADAR in which the interpretation of echoes from radio or sound waves to evaluate the attributes of a target by generating the high frequency sound waves .

# Actuators in IoT

An [IoT](https://www.geeksforgeeks.org/introduction-to-internet-of-things-iot-set-1/) device is made up of a Physical object (“thing”) + Controller (“brain”) + [Sensors + Actuators](https://www.geeksforgeeks.org/difference-between-sensor-and-actuator/) + Networks (Internet). An actuator is a machine component or system that moves or controls the mechanism of the system. Sensors in the device sense the environment, then control signals are generated for the actuators according to the actions needed to perform.

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

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

****

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

**Types of Actuators :**

**1. Hydraulic Actuators –**

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

**Advantages :**

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

**Disadvantages :**

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

**2. Pneumatic Actuators –**

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

**Advantages :**

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

**Disadvantages :**

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

**3. Electrical Actuators –**

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

**Advantages :**

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

**Disadvantages :**

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

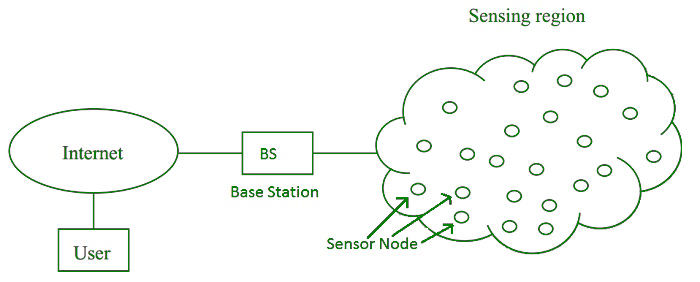
Other actuators are –

* **Thermal/Magnetic Actuators –**  
  These are actuated by thermal or mechanical energy. Shape Memory Alloys (SMAs) or Magnetic Shape‐Memory Alloys (MSMAs) are used by these actuators. An example of a thermal/magnetic actuator can be a piezo motor using SMA.
* **Mechanical Actuators –**   
  A mechanical actuator executes movement by converting rotary motion into linear motion. It involves pulleys, chains, gears, rails, and other devices to operate. Example – A crankshaft.
* Soft Actuators
* Shape Memory Polymers
* Light Activated Polymers
* With the expanding world of IoT, sensors and actuators will find more usage in commercial and domestic applications along with the pre-existing use in industry.

# Wireless Sensor Network (WSN)

**Wireless Sensor Network (WSN)**, is an infrastructure-less wireless network that is deployed in a large number of wireless sensors in an ad-hoc manner that is used to monitor the system, physical, or environmental conditions.

Sensor nodes are used in WSN with the onboard processor that manages and monitors the environment in a particular area. They are connected to the Base Station which acts as a processing unit in the WSN System. The base Station in a WSN System is connected through the Internet to share data. WSN can be used for processing, analysis, storage, and mining of the data.

****

## **Wireless Sensor Network Architecture**

A Wireless Sensor Network (WSN) architecture is structured into three main layers:

* **Physical Layer**: This layer connects sensor nodes to the base station using technologies like radio waves, [infrared](https://www.geeksforgeeks.org/difference-between-radio-wave-microwave-and-infrared-waves/), or [Bluetooth](https://www.geeksforgeeks.org/bluetooth/). It ensures the physical communication between nodes and the base station.
* **Data Link Layer**: Responsible for establishing a reliable connection between sensor nodes and the base station. It uses protocols such as IEEE 802.15.4 to manage data transmission and ensure efficient communication within the network.
* **Application Layer**: Enables sensor nodes to communicate specific data to the base station. It uses protocols like[ZigBee](https://www.geeksforgeeks.org/introduction-of-zigbee/) to define how data is formatted, transmitted, and received, supporting various applications such as environmental monitoring or industrial control.

These layers work together to facilitate the seamless operation and data flow within a Wireless Sensor Network, enabling efficient monitoring and data collection across diverse applications.

## **WSN Network Topologies**

Wireless Sensor Networks (WSNs) can be organized into different network topologies based on their application and network type. Here are the most common types:

* **Bus Topology**: In a [Bus Topology,](https://www.geeksforgeeks.org/advantages-and-disadvantages-of-bus-topology/)multiple nodes are connected to a single line or bus. Data travels along this bus from one node to the next. It’s a simple layout often used in smaller networks.
* **StarTopology**:[Star Topology](https://www.geeksforgeeks.org/advantages-and-disadvantages-of-star-topology/) have a central node, called the master node, which connects directly to multiple other nodes. Data flows from the master node to the connected nodes. This topology is efficient for centralized control.
* **Tree Topology**: [Tree Topology](https://www.geeksforgeeks.org/advantages-and-disadvantages-of-tree-topology/)arrange nodes in a hierarchical structure resembling a tree. Data is transmitted from one node to another along the branches of the tree structure. It’s useful for expanding coverage in hierarchical deployments.
* **Mesh Topology**:[Mesh Topology](https://www.geeksforgeeks.org/advantage-and-disadvantage-of-mesh-topology/)feature nodes interconnected with one another, forming a mesh-like structure. Data can travel through multiple paths from one node to another until it reaches its destination. This topology offers robust coverage and redundancy.

Each topology has its advantages and is chosen based on factors such as coverage area, scalability, and reliability requirements for the specific WSN application.

## Types of Wireless Sensor Networks (WSN)

### Terrestrial Wireless Sensor Networks

* Used for efficient communication between base stations.
* Consist of thousands of nodes placed in an ad hoc (random) or structured (planned) manner.
* Nodes may use solar cells for energy efficiency.
* Focus on low energy use and optimal [routing](https://www.geeksforgeeks.org/types-of-routing/)for efficiency.

### Underground Wireless Sensor Networks

* Nodes are buried underground to monitor underground conditions.
* Require additional sink nodes above ground for data transmission.
* Face challenges like high installation and maintenance costs.
* Limited battery life and difficulty in recharging due to underground setup.

### Underwater Wireless Sensor Networks

* Deployed in water environments using sensor nodes and autonomous underwater vehicles.
* Face challenges like slow data transmission, bandwidth limitations, and[signal attenuation.](https://www.geeksforgeeks.org/attenuation/)
* Nodes have restricted and non-rechargeable power sources.

### Multimedia Wireless Sensor Networks

* Used to monitor multimedia events such as video, audio, and images.
* Nodes equipped with [microphones](https://www.geeksforgeeks.org/what-is-a-microphone/) and cameras for data capture.
* Challenges include high power consumption, large bandwidth requirements, and complex data processing.
* Designed for efficient wireless data compression and transmission.

### Mobile Wireless Sensor Networks (MWSNs)

* Composed of mobile sensor nodes capable of independent movement.
* Offer advantages like increased coverage area, energy efficiency, and channel capacity compared to static networks.
* Nodes can sense, compute, and communicate while moving in the environment.

# IoT Protocols

## IoT Protocols Introduction

The IoT system can only perform and transfer information when it’s in online mode, which means the devices in IoT must be safely connected to communication networks. The question that arises is how these devices can connect and communicate with each other. The answer lies in PROTOCOLS. Protocols enable these devices to communicate effectively, and new protocols continue to be introduced regularly. In this article, we will discuss the IoT network requirements, and the different types of protocols used in IoT, and provide a brief description of commonly used protocols in the Internet of Things.

## Why do we need IoT Protocols?

IoT protocols are essential for enabling seamless and efficient communication between Internet of Things (IoT) devices, applications, and services. They serve as the rules and standards that govern how devices in an IoT ecosystem exchange data and interact with each other. The need for IoT protocols arises due to several reasons:

Interoperability: In IoT, devices come from various manufacturers, and they may use different communication technologies and data formats. IoT protocols ensure that devices from different vendors can communicate and work together seamlessly, fostering interoperability and creating a unified IoT ecosystem.

Efficient Communication: IoT protocols are designed to optimize data transmission and reduce unnecessary overhead, leading to efficient use of resources like power and bandwidth. This efficiency is crucial, especially for IoT devices with limited processing capabilities and battery life.

Security: IoT devices often handle sensitive data and are vulnerable to cybersecurity threats. IoT protocols incorporate security mechanisms like encryption, authentication, and data integrity checks to ensure secure communication and protect against unauthorized access.

Scalability: IoT deployments can involve a large number of devices spread across wide areas. IoT protocols are designed to handle scalability, allowing the network to accommodate a growing number of devices without compromising performance.

Low Power Consumption: Many IoT devices are battery-powered or operate in energy-constrained environments. IoT protocols, especially those designed for low-power communication, help extend the battery life of devices and minimize energy consumption.

Real-time Communication: Some IoT applications, such as industrial automation and healthcare monitoring, require real-time data exchange. IoT protocols cater to such scenarios and ensure timely delivery of critical information.

Standardization: IoT protocols are standardized to ensure consistency and compatibility across different implementations. This standardization facilitates the development of a diverse range of IoT devices and applications while ensuring they can work together smoothly.

Data Management: IoT protocols define how data is structured, transmitted, and processed, facilitating efficient data management and analysis. Proper data handling is essential for making informed decisions and deriving meaningful insights from IoT-generated data.

## IoT Protocol Types

### MQTT Protocol

[MQTT](https://microcontrollerslab.com/what-is-mqtt-and-how-it-works/)is a lightweight protocol specifically designed for sending data from sensors to middleware and applications. It operates on top of TCP/IP, ensuring reliable data delivery. MQTT consists of three main components: Subscriber, Publisher, and Broker. Let’s explore how these components work together in the basic workflow of this protocol, where the publisher generates and transmits information to subscribers through a broker.

#### Understanding MQTT

MQTT, short for Message Queue Telemetry Transport, is an IoT protocol that excels in lightweight data transfer. Its primary purpose is to enable efficient communication by transmitting data from sensors to middleware and applications.

The Three Main Components

MQTT comprises three key components: Subscriber, Publisher, and Broker, each with specific roles.  
Subscribers receive data from the publisher through the broker, facilitating seamless data flow.  
The Basic Workflow

In MQTT’s basic workflow, the publisher is responsible for generating and transmitting information.  
The broker acts as an intermediary, managing the data and delivering it to the relevant subscribers.

## What Is The CoAP Protocol?

he CoAP Protocol, short for Constrained Application Protocol, is a specialized internet application protocol for constrained devices. It was designed to allow small, low-power devices to join the Internet of Things (IoT). The protocol allows these devices to communicate with the wider Internet using minimal resources.

The CoAP protocol has a small base specification that can be extended with additional functionality when needed. It operates over UDP and provides a request/response interaction model between application endpoints, enabling interoperability among different types of devices.

CoAP is also highly reliable, with mechanisms in place to ensure message delivery, even in cases of limited network connectivity or device power. This makes it suitable for IoT devices, which often operate in challenging network environments.

## Key Features of CoAP

### RESTful Architecture

CoAP uses a RESTful (Representational State Transfer) architecture. It follows a set of constraints that allow it to operate efficiently over a large, distributed network. In a RESTful system, data and functionality are considered resources, and these resources are accessed using a standard, uniform interface.

In the case of CoAP, this RESTful architecture allows it to provide a high level of interoperability among different types of devices. It also makes it easy for developers to build applications that use the protocol, as they can use standard HTTP methods (such as GET, POST, PUT, and DELETE) to interact with resources.

### Built-In Discovery

The CoAP protocol’s built-in discovery mechanism allows devices to discover resources on other devices without requiring any prior knowledge of their existence. This is especially useful in IoT networks, where devices may be constantly joining and leaving the network.

The built-in discovery feature in CoAP is achieved through the use of a well-known 'core' resource that provides a list of available resources on a device. This resource can be queried by other devices on the network, allowing them to discover what resources are available and how to interact with them.

### Asynchronous Message Exchanges

CoAP supports asynchronous message exchanges, which is crucial for IoT networks where devices may not always be connected or available. With asynchronous message exchanges, a device can send a request to another device and then continue with other tasks without waiting for a response. The response can be processed once it arrives, even if delayed.

This feature uses a message identifier in each CoAP message, which allows a device to match responses with requests. This, in conjunction with the ability to retransmit lost messages, ensures a high level of reliability in message exchanges.

### Optional Reliability with Confirmable Messages

CoAP offers optional reliability through the use of confirmable messages. When a device sends a confirmable message, it expects an acknowledgement from the recipient. If no acknowledgement is received within a certain time, the message is retransmitted.

This feature allows CoAP to provide reliable communication in environments where network connectivity is unreliable. Devices can ensure that critical messages are received and processed.

## Use Cases of CoAP

### Smart Home Automation

CoAP is increasingly being used in smart home automation systems due to its low overhead and high reliability. In these systems, various devices such as lights, thermostats, and security cameras can all communicate using the CoAP protocol. This allows for a high level of interoperability and makes it easy to add new devices to the network.

### Industrial IoT

In [**industrial IoT applications**](https://www.emqx.com/en/blog/industrial-iot-applications), reliability and efficiency are crucial. Devices such as sensors and actuators can communicate using CoAP, allowing for real-time monitoring and control of industrial processes. The protocol's support for multicast communication is particularly useful in these scenarios, as it allows a single device to communicate with multiple others simultaneously.

### Wearables and Healthcare

CoAP is becoming increasingly popular in wearable devices and healthcare applications. These applications often involve small, battery-powered devices that need to communicate with each other or with a central server. CoAP's low overhead and power requirements make it useful for these types of applications.

### Energy Management

CoAP is used in energy management systems, where it allows for real-time monitoring and control of energy usage. Devices such as smart meters and energy management controllers can use the protocol to communicate with each other and with a central server, providing a high level of control over energy usage.

# XMPP Protocol

XMPP is a short form for Extensible Messaging Presence Protocol. It’s protocol for streaming [XML elements](https://www.geeksforgeeks.org/xml-basics/) over a network in order to exchange messages and present information in close to real-time.

Let’s dive into each character of word **XMPP**:

* **X :** It means eXtensible. XMPP is an open-source project which can be changed or extended according to the need.
* **M :** XMPP is designed for sending messages in real time. It has very efficient push mechanism compared to other protocols.
* **P :** It determines whether you are online/offline/busy. It indicates the state.
* **P :** XMPP is a protocol, that is, a set of standards that allow systems to communicate with each other.

These are the basic requirements of any Instant Messenger which are fulfilled by XMPP:

1. Send and receive messages with other users.
2. Check and share presence status
3. Manage subscriptions to and from other users.
4. Manage contact list
5. Block communications(receive message, sharing presence status, etc) to specific users.

# Block Diagram of Embedded System

An embedded system is a specialized computing system that is designed to perform dedicated functions or tasks, often in real-time and in a resource-constrained environment.

[An Embedded system](https://www.analog.com/en/design-center/glossary/embedded-system.html#:~:text=An%20embedded%20system%20is%20a,%2C%20files%2C%20or%20operating%20systems.) is defined as a way of working, organizing, performing single or multiple tasks according to a set of rules. In an embedded system, all the units assemble and work together according to the program.

Unlike general-purpose computers, which are designed to run a variety of applications, embedded systems are integrated into specific devices or products to control particular functions.

## Key Characteristics of Embedded Systems

Key characteristics of embedded systems include:

1. **Dedicated Functionality:** Embedded systems are built to perform specific tasks or functions within a larger system. Examples include the control systems in appliances, automotive control systems, industrial automation, medical devices, and more.
2. **Real-Time Operation:** Many embedded systems are required to respond to external events or inputs in real-time. This means they must meet specific timing constraints to ensure accurate and timely operation.
3. **Resource Constraints:** Embedded systems often have limited resources, including processing power, memory, and storage. These constraints necessitate efficient design and optimization of software and hardware components.
4. **Integration:** Embedded systems are integrated into larger systems or products. They may consist of a combination of hardware and software components, and they often work alongside other systems within the same device.
5. **Reliability:** Many embedded systems are designed for critical applications where reliability is essential. For example, embedded systems in automotive applications must operate reliably to ensure the safety and proper functioning of the vehicle.
6. **Low Power Consumption:** Embedded systems are commonly used in battery-powered devices or applications where power consumption is a critical consideration. Optimizing power usage is often a key design goal.

## Block Diagram of Embedded System

### **Processor**

* At the heart of the embedded system is the central processing unit or processor. It is the hardware that executes the software and brings life to the embedded system. It also controls the activities of all the other circuits.
* There are varieties of processors available for embedded systems, and the main criteria for selection is to provide the processing power needed to perform the tasks within the system.
* Besides, the system cost, power consumption, software development tools, and component availability are also important factors to be considered while selecting a processor for embedded system design.

### **Memory**

* The embedded system also has memory, often several different types in one system. The memory is an important part of any embedded system because how the software is designed, written, and developed is decided by memory.
* The memory is used to store the software that the processor will run. It also provides storage for data such as program variables, intermediate results, status information and any other data generated throughout the operation.

### **Peripherals**

An embedded system communicates with the outside world through peripherals. The main types of peripherals that are used include:

* **Digital inputs and outputs:** These are simple external pins whose logic state can be controlled by the processor to either be a logic 1 or a logic 0. They can also be used as input pins so that the processor can receive binary inputs from the external world. They can be used individually or grouped together to create parallel ports.
* **Serial interfaces:** These are interfaces that send or receive data using one or two pins in a serial mode. They are less complex to connect but are more complicated to program because the data may also be augmented with additional information as required by the data transfer protocol.
* **Analog to digital converters:**While processors operate on the digital data, the surrounding world is analog in nature. Therefore, interfaces between the system and the external world requires analog to digital conversions and vice versa.
* **Displays:** Displays are used by the processor to display the status information, error messages, and output results. They could be simple LEDs, seven segment displays, or character LCD panels.
* **Keypads:** Keypads are used by the end user to provide inputs to the embedded system. The inputs could be anything such as entering the password, changing functional settings, switching between menu items, etc.

# IoT Platforms Overview: Arduino, Raspberry Pi

### **Arduino**

The Arduino platform was created back in 2005 by the Arduino company and allows for open source prototyping and flexible software development and back-end deployment while providing significant ease of use to developers, even those with very little experience building IoT solutions.

Arduino is sensible to literally every environment by receiving source data from different external sensors and is capable to interact with other control elements over various devices, engines and drives. Arduino has a built-in micro controller that operates on the Arduino software.

Projects based on this platform can be both standalone and collaborative, i.e. realized with use of external tools and plugins.  The integrated development environment (IDE) is composed of the open source code and works equally good with Мac, Linux and Windows OS. Based on a processing programming language, the Arduino platform seems to be created for new users and for experiments. The processing language is dedicated to visualizing and building interactive apps using animation and Java Virtual Machine (JVM) platform.

Let's note that this programming language was developed for the purpose of learning basic computer programming in a visual context. It is an absolutely free project available to every interested person. Normally, all the apps are programmed in C/C++, and are wrapped with avr-gcc (WinAVR in OS Windows).

Arduino offers analogue-to-digital input with a possibility of connecting light, temperature or sound sensor modules. Such sensors as SPI or I2C may also be used to cover up to 99% of these apps’ market.

Arduino is a microcontroller (generally it is the 8-bit ATmega microcontroller), but not a mini-computer, which makes Arduino somehow limited in its features for advanced users. Arduino provides an excellent interactivity with external devices and offers a wide range of user manuals, project samples as well as a large community of users to learn from / share knowledge with.

### **Raspberry Pi**

Raspberry Pi is a mono-board computing platform that's as tiny as a credit card. Initially it was developed for computer science education with later on progress to wider functions

Since the inception of Raspberry, the company sold out more than 8 million items. Raspberry Pi 3 is the latest version and it is the first 64-bit computing board that also comes with built-in Wi-Fi and Bluetooth functions. According to Raspberry Pi Foundation CEO Eben Upton, "it's been a year in the making". The Pi3 version is replaced with a quad-core 64-bit 1.2 GHz ARM Cortex A53 chip, 1GB of RAM, VideoCore IV graphics, Bluetooth 4.1 and 802.11n Wi-Fi. The developers claim the new architecture delivers an average 50% performance improvement over the Pi 2.

Another peculiarity of Raspberry Pi is the GPIO (General Purpose Input-Output), which is a low-level interface of self-operated control by input-output ports. Raspberry has it as a 40-pin connector.

Raspberry Pi uses Linux as its default operating system (OS). It’s also fully Android compatible. Using the system on Windows OS is enabled through any virtualization system like XenDesktop. If you want to develop an application for Raspberry Pi on your computer, it is necessary to download a specific toolset comprised of ARM-compiler and some libraries complied down to ARM-target platform like glibc.