INTERNET OF THINGS FUNDAMENTALS: CONCEPTS, DEVICES, AND APPLICATIONS

Dr. Nin Hayati Mohd Yusoff

INTERNET OF THINGS FUNDAMENTALS: CONCEPTS, DEVICES, AND APPLICATIONS

Nin Hayati Mohd Yusoff

INTERNET OF THINGS FUNDAMENTALS: CONCEPTS, DEVICES, AND APPLICATIONS

Nin Hayati Mohd Yusoff

Hak Cipta setiap bahagian penerbitan ini tidak boleh diterbitkan semula atau diedarkan dalam sebarang bentuk dengan sebarang cara atau sistem perolehan semula tanpa kebenaran bertulis terlebih dahulu.

PUBLISHED BY:
Politeknik Merlimau Melaka
KM2.5, Jalan Merlimau-Jasin
77300 Merlimau Melaka

PREFACE

The Internet of Things (IoT) has become one of the most transformative technologies of the Fourth Industrial Revolution (IR 4.0). It connects billions of devices worldwide—ranging from simple sensors to complex smart systems—creating an ecosystem that collects, exchanges, and analyzes data to enhance human life, industrial efficiency, and global sustainability.

This book, Internet of Things Fundamentals: Concepts, Devices, and Applications, is designed to introduce students and educators to the essential concepts and practical aspects of IoT. It covers the fundamental topics outlined in the course, including the evolution and framework of IoT, the use of microcontrollers and sensors, connectivity standards, data and network protocols, and real-world applications in areas such as smart homes, agriculture, and healthcare.

Each chapter blends theory with applied learning, providing readers with the knowledge to understand IoT architectures and the skills to build simple connected systems. This book aims to support TVET and higher education learners in developing the technical foundation needed to participate in the growing IoT industry, while inspiring innovation toward smart, sustainable, and connected communities.

Nin Hayati Mohd Yusoff

"The future belongs to those who innovate with purpose and connect with intelligence."

Connect with the Author

About Author

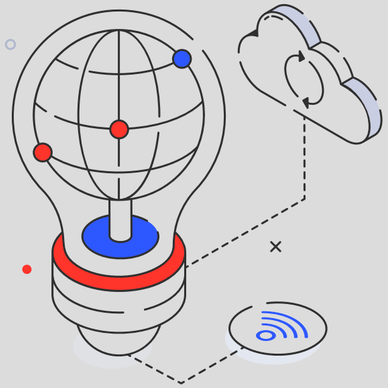
Website

Table of Contents

01

Chapter 1

INTERNET OF THINGS CONCEPTS



02

Introduction to Industrial Revolution

The Industrial Revolution (IR) refers to a series of technological and industrial transformations that changed how people work, produce goods, and live. Each revolution introduced new technologies that significantly increased efficiency and productivity and leading us to today's Industry 4.0, where the Internet of Things (IoT) plays a major role.

Industry 1.0

Mechanization and the introduction of stream and water power

Industry 2.0

Electric Powered Assembly Line, Mass Production

Industry 3.0

Automation, Computers and Electronics

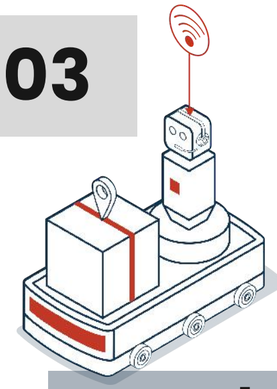
Industry 4.0

Automation, Computers and Electronics

Industry 5.0

Human-robot collaboration, cognitive systems, customization

03



Introduction to

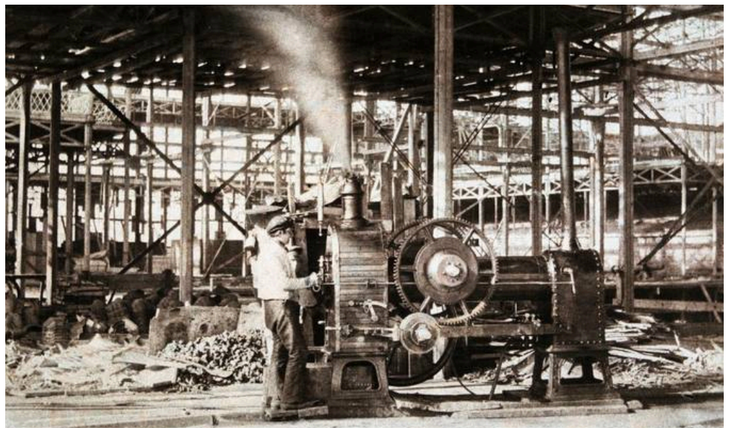
Industrial Revolution

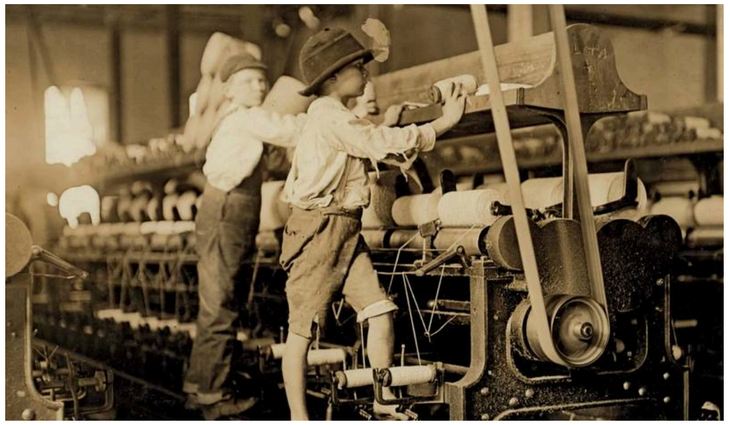
Industry 1.0

MECHANIZATION AND THE INTRODUCTION OF STREAM AND WATER POWER

The Industrial Revolution in Britain marked a major transformation in production methods between 1760 and 1840. During this period, industries began shifting from manual labor to machine-based manufacturing powered by steam engines and water.

This transition significantly boosted agricultural productivity and introduced the concept of factories. Among the various sectors, the textile industry was the first to embrace mechanized production, playing a crucial role in driving Britain's economy at the time.





Main Feature: Introduction of mechanical production powered by steam and water.

Key Technologies:

- Steam engines
- Textile machines (spinning jenny, power loom)
- Water-powered factories

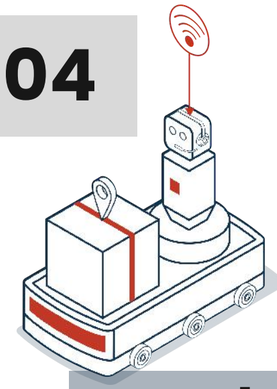
Impact:

- Manual labor was replaced by machines.
- Factories began to form, starting the industrial era.
- Urbanization increased as people moved to cities for work.

Example:

Steam engines powered trains and ships, revolutionizing transportation and trade.

04



Introduction to

Industrial Revolution

Industry 2.0

ELECTRIC POWERED ASSEMBLY LINE, MASS PRODUCTION

The Second Industrial Revolution took place between 1870 and 1914 and brought major changes to industries. During this time, inventions such as the telegraph, railways, and electricity were used to improve factory operations and communication.

Factories became more efficient with the use of electrical power, and mass production started to replace manual work. The steel industry grew quickly and helped expand railway systems. There were also new discoveries in chemistry, such as synthetic dyes. Although World War I slowed progress, the ideas and technologies from this period continued to influence modern industry and manufacturing.

Main Feature: Introduction of electricity, assembly lines, and mass production.

Key Technologies:

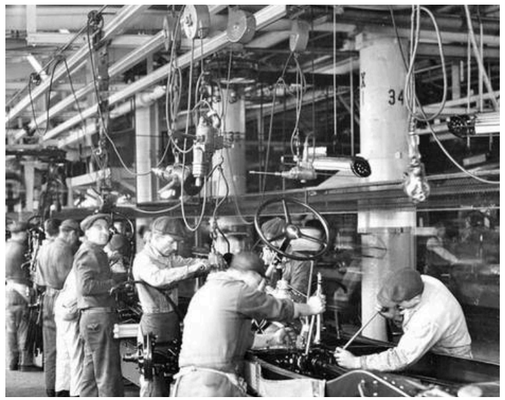
- Electrical power systems
- Conveyor belts
- Internal combustion engines
- Telephone and telegraph

Impact:

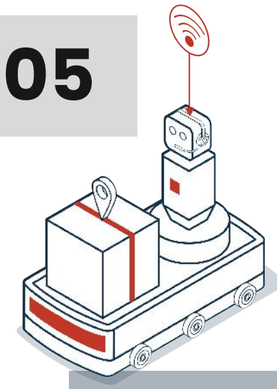
- Factories could run longer hours with electric lighting.
- Products became cheaper due to large-scale manufacturing.
- The concept of “division of labor” was introduced (each worker specialized in one task).

Example:

Henry Ford’s automobile production line (1913) is a famous symbol of IR 2.0 efficiency.

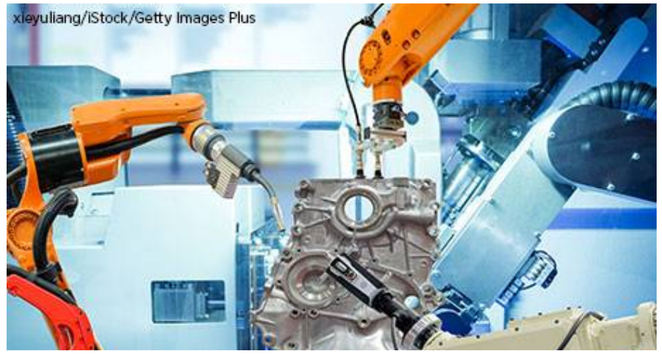


05



Introduction to

Industrial Revolution



Industry 3.0

The Third Industrial Revolution, which occurred between 1950 and 1970, is also known as the **Digital Revolution**. This period marked the shift from mechanical and analog systems to digital technology.

It introduced computers, automation, and information and communication technology (ICT) into industries, transforming how products were designed and manufactured. Often called the Information Age, this revolution laid the foundation for modern digital systems and continues to influence today's industrial and technological development.

AUTOMATION, COMPUTERS AND ELECTRONICS

Main Feature: The rise of computers, electronics, and automation systems.

Key Technologies:

- Microprocessors and programmable logic controllers (PLCs)
- Industrial robots
- Internet and information technology (IT)

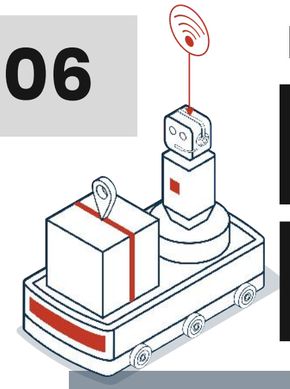
Impact:

- Machines could perform repetitive tasks automatically.
- Data processing became faster, improving decision-making.
- The foundation of the digital economy was established.

Example:

Factories started using robot arms for welding and assembly, improving accuracy and speed.

06



Introduction to

Industrial Revolution

Industry 4.0

The Fourth Industrial Revolution, or Industry 4.0, builds on digital technologies to create smart, connected systems. It uses the Internet of Things (IoT), real-time data, and cyber-physical systems to link machines, people, and processes. This integration connects the physical and digital worlds, allowing better communication and automation across industries.

Industry 4.0 helps businesses monitor and control operations more efficiently, make data-driven decisions, and improve productivity. It represents a new era of intelligent and interconnected manufacturing, where technology drives innovation and growth.

AUTOMATION, COMPUTERS AND ELECTRONICS

Main Feature: Integration of cyber-physical systems, IoT, AI, and Cloud Computing to create smart, autonomous, and data-driven industries.

Key Technologies:

- Internet of Things (IoT)
- Artificial Intelligence (AI) and Machine Learning (ML)
- Cloud and Edge Computing
- Big Data Analytics
- Cyber-Physical Systems (CPS)
- Robotics and 5G connectivity

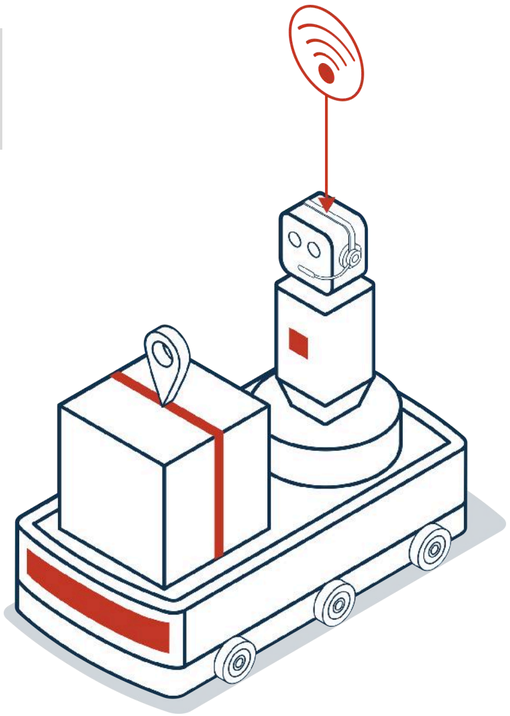
Impact:

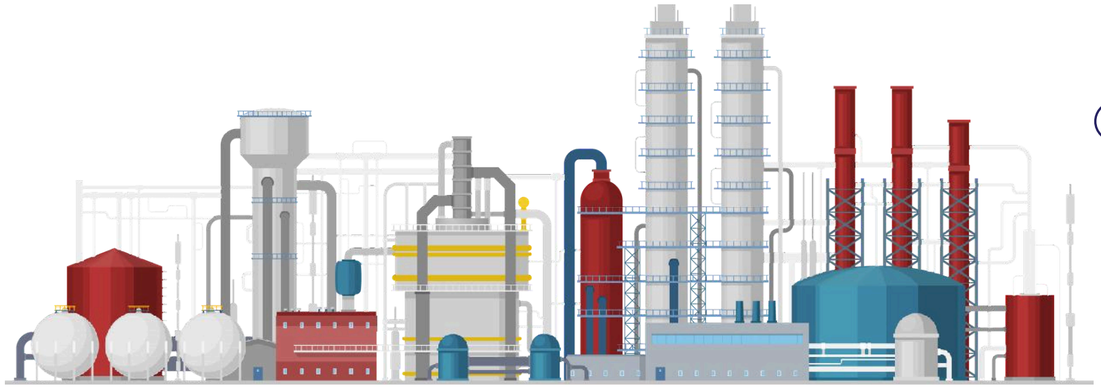
- Machines and devices communicate through the Internet.
- Real-time data enables predictive maintenance and smart decisions.
- Production systems become flexible, efficient, and self-optimizing.

Example:

A smart factory uses IoT sensors to monitor machine performance, and AI predicts when maintenance is needed — reducing downtime.

07





Industrial 4.0

Autonomous Robot

Big Data Analytics

Supply Chain

Cloud Computing

Cyber Security

Autonomous Robot

Artificial Intelligence

New Business Models

Additive Manufacturing

Industrial IoT

Simulation & Augmented Reality

11 Pillar of Technology Advancement

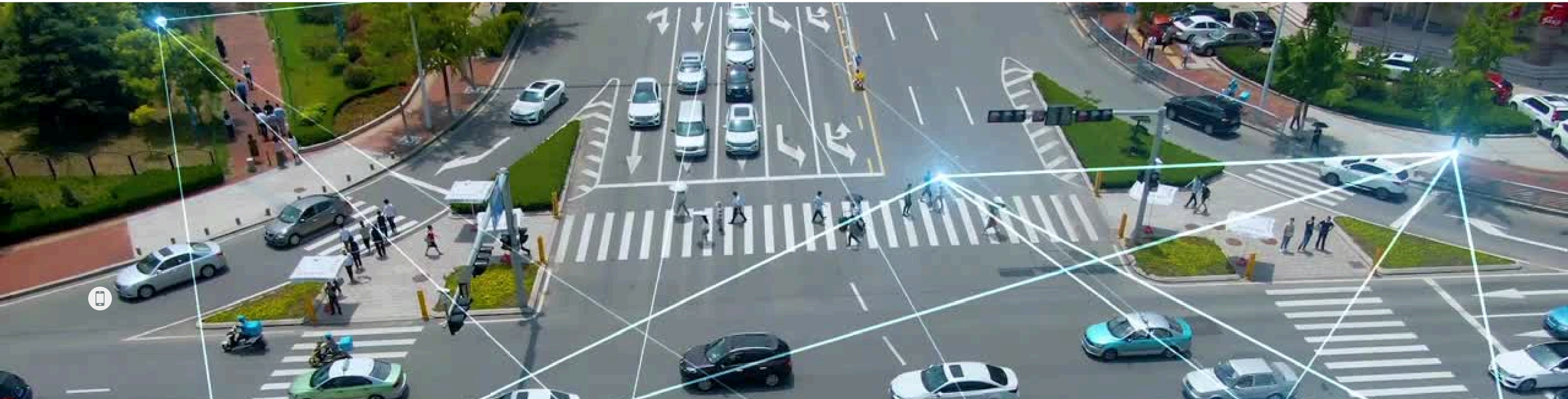
Dr. Nur Hayati Mohd Yusoff

08

Introduction to Internet of Things

The **Internet of Things (IoT)** refers to a concept where **physical devices such as sensors, machines, and smart gadgets** are connected **to the Internet to communicate, collect, and exchange data automatically.**

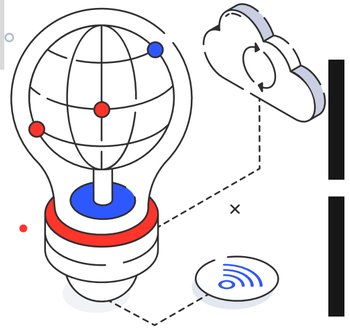
The main purpose of IoT is **to improve efficiency, save time, and enhance daily life through automation and real-time data analysis.**



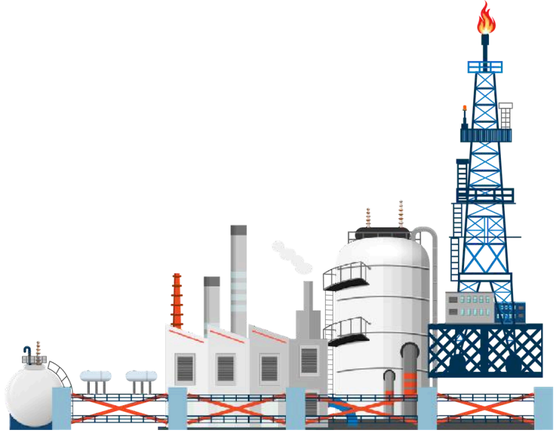
Internet of Things Trends

...2030

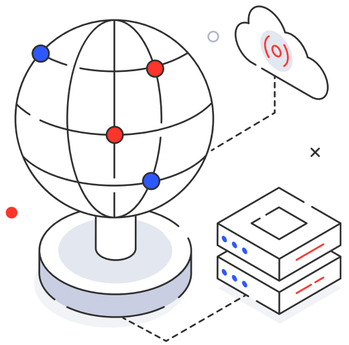
05	$2	30+	15+	60
Billion	Trillion	Million	Billion	Trillion
Connected People	Revenue Opportunity	Apps	Embedded and Intelligent System	Gbs of Data



IoT in Industry 4.0

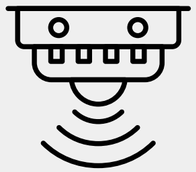


The IoT is one of the core technologies driving Industry 4.0. In Industry 4.0, IoT enables smart factories, intelligent automation, and data-driven decision-making, forming the foundation of cyber-physical systems that integrate the digital and physical worlds.



IoT in Industry 4.0

Connectivity

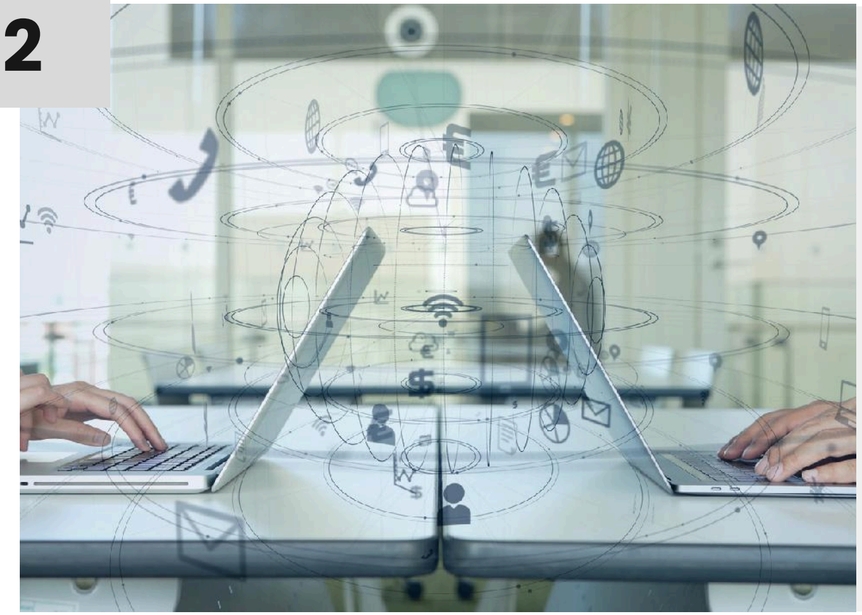


Data Collection

Data Transmission

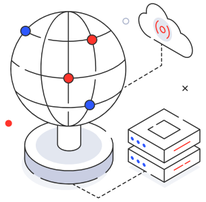
Data Processing and Analytics

Automation and Control



Connectivity

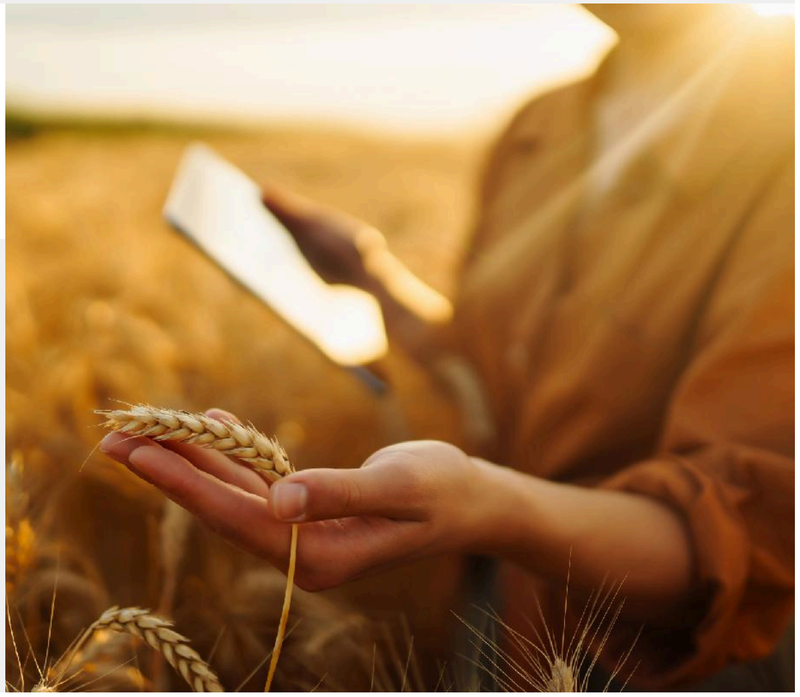
- The most fundamental concept of IoT is interconnectivity.
- Devices, machines, and systems communicate using wireless technologies such as Wi-Fi, Bluetooth, Zigbee, LoRa, and 5G.
- Connectivity allows machines to exchange data seamlessly across industrial networks or the cloud.
- Example: A temperature sensor connected to Wi-Fi sends live data to a cloud dashboard for monitoring.



IoT in Industry 4.0

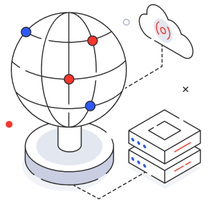
Data Collection

- IoT relies on sensors and actuators to collect real-time information from the physical environment.
- Sensors detect temperature, humidity, pressure, light, gas, or motion, while actuators perform actions like turning on a fan or opening a valve.
- Example: In a smart greenhouse, sensors collect humidity and soil moisture data to automate irrigation.

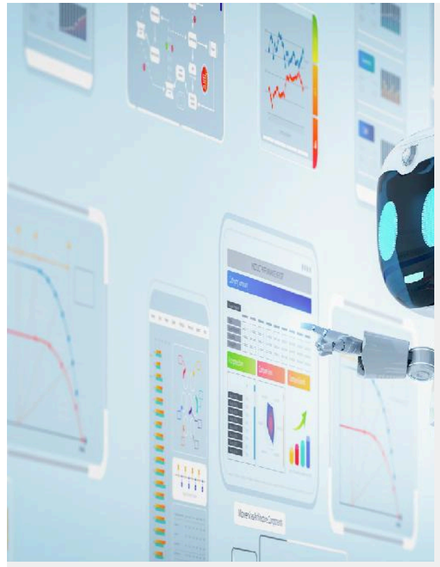


Data Transmission

- Once collected, data must be transmitted to processing systems via communication networks.
- Transmission can occur through local networks (LAN) or cloud-based IoT platforms.
- Common Protocols: MQTT, HTTP, CoAP, and LoRaWAN.
- Example: An ESP32 sends sensor data to AWS IoT Core using MQTT protocol.



IoT in Industry 4.0

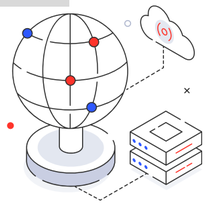


Data Processing and Analytics

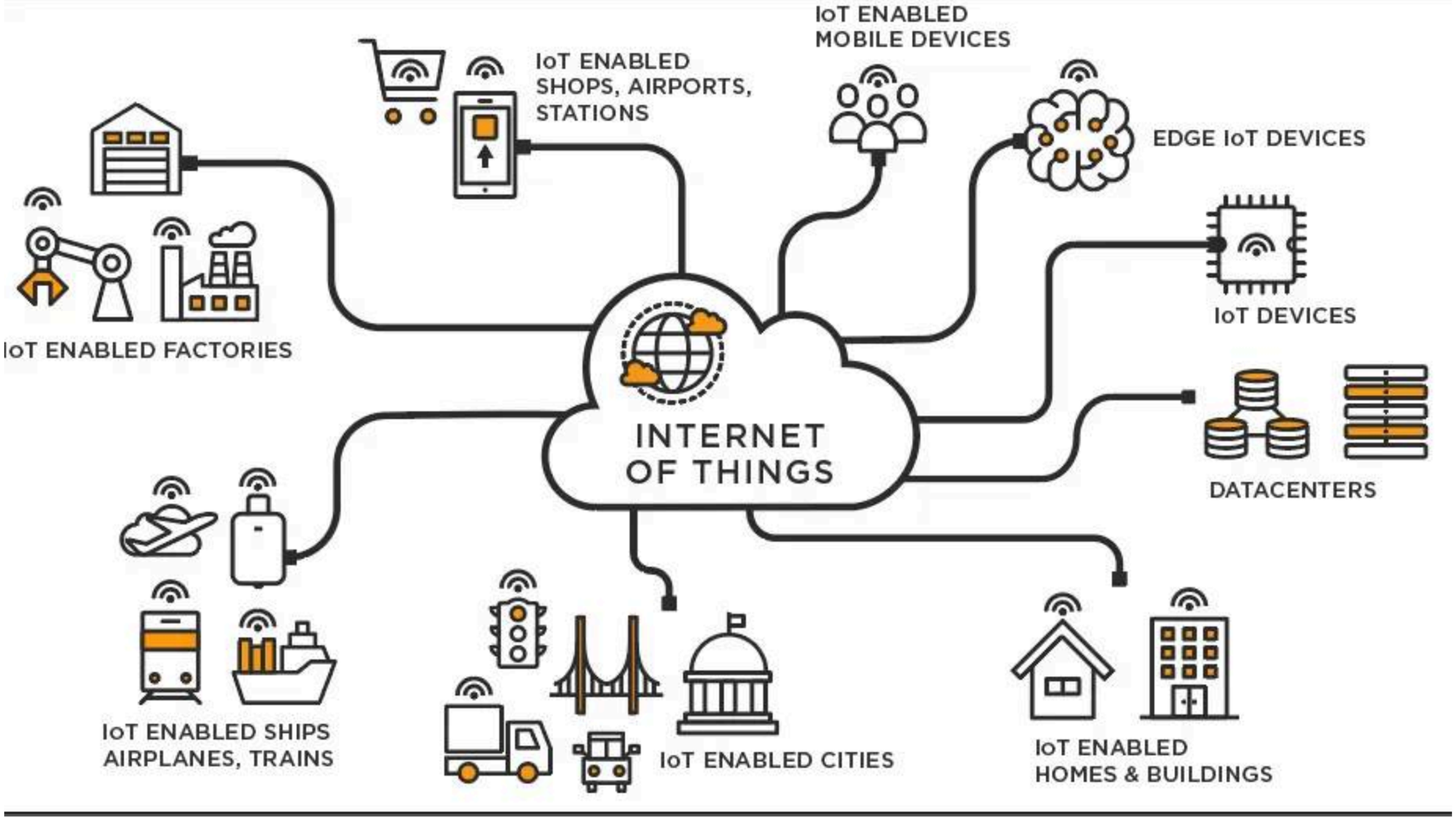
- The core strength of IoT in Industry 4.0 lies in data analysis.
- This transforms raw data into actionable intelligence.
- Example: Predictive maintenance – AI analyzes vibration data from machines to detect potential failures before they happen.

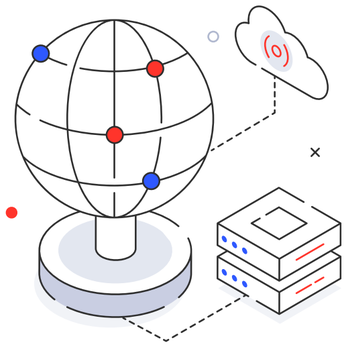
Automation and Control

- Based on data analysis, IoT systems can make automatic decisions without human intervention.
- It improves efficiency, safety, and accuracy in industrial operations.
- Example: When a sensor detects high temperature, an IoT system automatically switches on the cooling fan.



IoT in Industry 4.0





IoT in Industry 4.0: Issues and Challenges

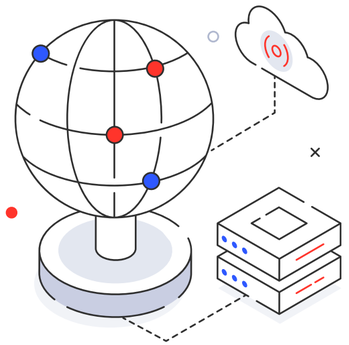
Security

IoT devices often collect sensitive data (e.g., personal, industrial, or environmental information) and are connected to networks that can be exploited by cyber attackers. Many IoT devices lack strong security features due to limited processing power and cost constraints.

Challenges:

- Data Breaches: Unsecured communication can lead to data theft or manipulation.
- Weak Authentication: Default passwords and lack of encryption make IoT systems vulnerable.
- Botnet Attacks: Compromised IoT devices can be used for distributed denial-of-service (DDoS) attacks.
- Privacy Concerns: Continuous data collection raises issues of user consent and surveillance.

Example:
In a smart home, if the Wi-Fi camera or smart lock is hacked, it could expose personal data or grant unauthorized access.



IoT in Industry 4.0: Issues and Challenges

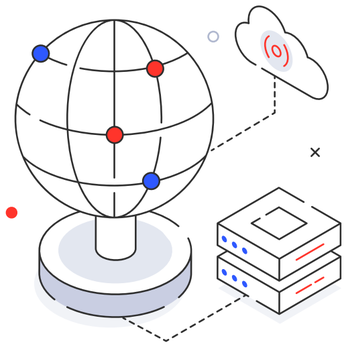
Cost

Implementing IoT requires investment in sensors, devices, communication networks, data storage, and analytics systems.

Challenges:

- High Initial Setup Cost: Deploying large-scale IoT networks in industries or smart cities involves expensive hardware and infrastructure.
- Maintenance Costs: Continuous updates, replacements, and technical support increase operational costs.
- Scalability: Expanding IoT systems to cover more devices or locations can significantly raise costs.

Example:
A manufacturing company may find it costly to equip all machines with smart sensors and maintain cloud-based monitoring services.



IoT in Industry 4.0: Issues and Challenges

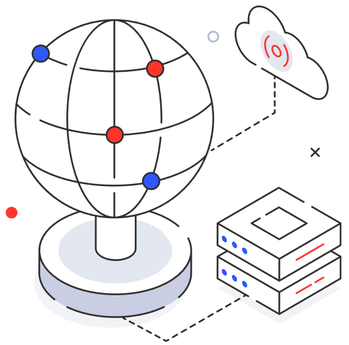
Reliability and Hardware

IoT systems depend on continuous connectivity and reliable hardware components. Any failure can disrupt operations or cause inaccurate data collection.

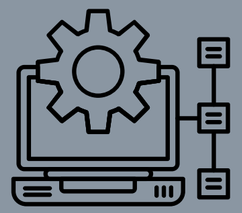
Challenges:

- Device Durability: Sensors and actuators may fail in harsh environments due to heat, moisture, or dust.
- Network Downtime: Unstable internet or power supply can cause interruptions in real-time monitoring.
- Battery Life: Many IoT devices are battery-powered and require efficient power management.
- Data Accuracy: Faulty sensors can lead to incorrect decision-making in automation systems.

Example:
In agriculture IoT systems, unreliable soil moisture sensors may lead to overwatering or crop damage.



IoT in Industry 4.0: Issues and Challenges



Integration with Current Technology

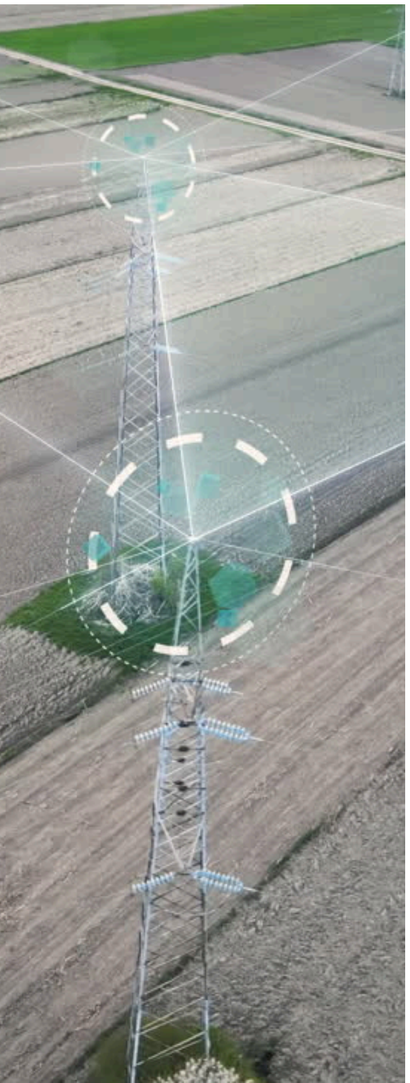
IoT must be integrated with existing systems such as legacy machines, ERP software, and industrial networks.

Challenges:

- Compatibility Issues: Legacy equipment may not support modern IoT protocols.
- Standardization: Different vendors use various communication standards, causing interoperability problems.
- Data Management: Combining IoT data with traditional IT systems requires robust middleware and analytics tools.
- Technical Expertise: Skilled professionals are needed to integrate and manage hybrid systems.

Example:
Integrating IoT-based monitoring into an old factory's control system can be difficult if the machines do not support digital interfaces.

Internet of Things History



1989
The first IoT device is created

1993
The first online webcam is used at Cambridge

1994
Steve Mann creates WearCam

1999
Kevin Ashton coins the term Internet of Things

2008
The number of connected devices overtakes the number of people in the world and IoT is 'born'

2007
First iPhone is launched

2005
UN publishes its first report on the Internet of Things

2000
LG announces the first smart refrigerator

2009
The original Fitbit activity tracker is released

2011
IIoT comes into being

2014
Seoul becomes the world's first smart city

2015
IoT goes Mobile with smartphones

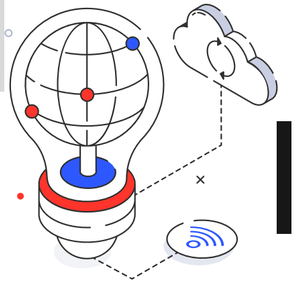
2022
World Economic Forum names IoT as one of the three most impactful technological advancements

2021
More than 10 billion active IoT devices active

2020
The number of IoT device connections increased more than 50% of the active connected devices

2016
AWS IoT core is launched

BYTEBEAM

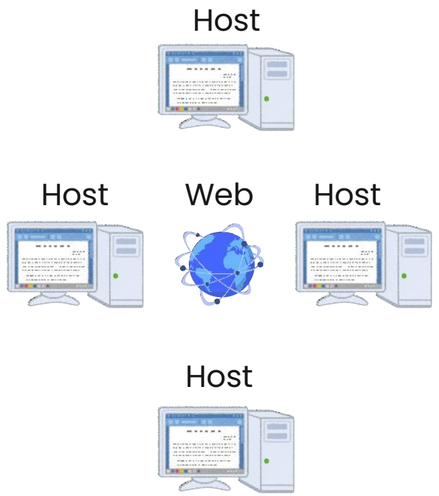


Internet of Things Revolution

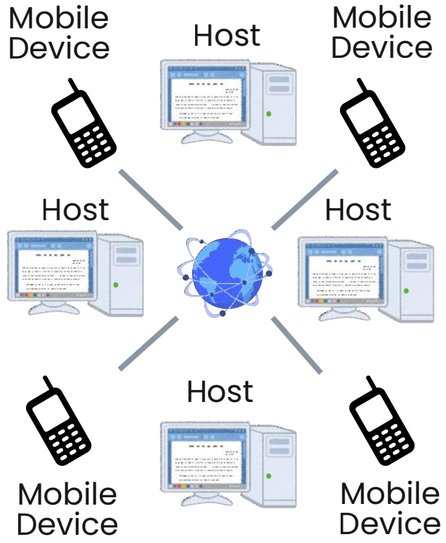
The evolution of internet connectivity started with simple computer networks for data sharing. It then grew into the Internet, connecting computers worldwide. With the rise of mobile technology, people could access the internet anywhere through smartphones. Later, both people and devices became connected, enabling easy communication and data exchange. Today, in the Internet of Things (IoT) era, not only people but also everyday objects like cars, homes, and machines are connected, making life smarter and more efficient.

Network

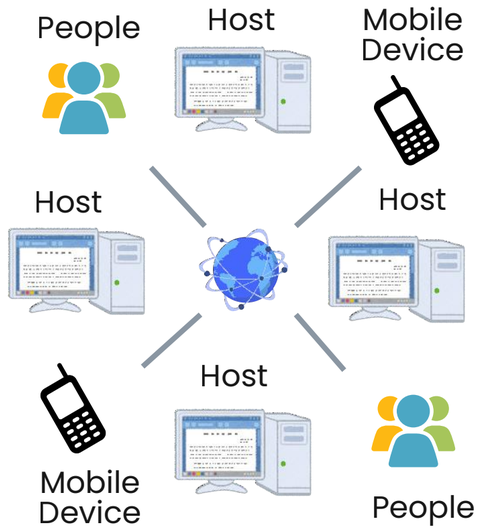
The Internet



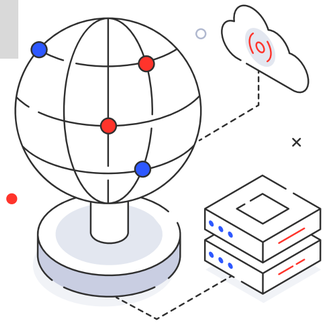
Mobile-Internet



Mobiles + People + PCs



Internet of Things

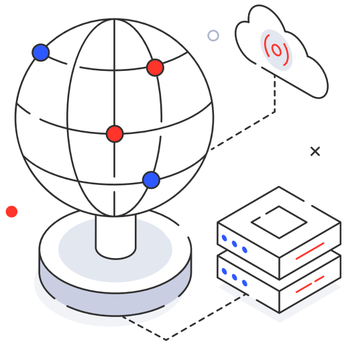


IoT in Sustainable Development Goals (SDG)

The Sustainable Development Goals (SDGs) are a global framework introduced by the United Nations (UN/UNESCO) in 2015. They consist of 17 key goals designed to make the world more prosperous, inclusive, and sustainable by the year 2030.

The SDGs emphasize the balance between economic growth, social inclusion, and environmental protection, ensuring that technological advancement benefits both current and future generations.

The IoT is a transformative technology that connects physical devices to the internet, enabling real-time monitoring, intelligent decision-making, and automation. It plays a significant role in achieving SDGs by fostering innovation, sustainability, and inclusivity across multiple sectors.



IoT in SDG 3: Good Health and Well-being

IoT enables smarter healthcare systems through remote patient monitoring, wearable health devices, and smart hospital management. Devices such as smartwatches, glucose monitors, and heart-rate sensors collect and transmit patient data to healthcare providers in real-time.

Impact:

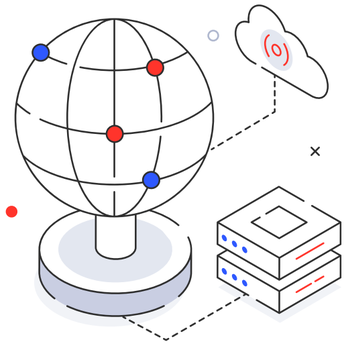
- Improves access to healthcare, especially in rural or underserved areas.
- Enables early disease detection and continuous monitoring.
- Reduces hospital readmissions and operational costs.

Examples:

Wearable Health Devices: Monitor vital signs and alert healthcare providers in emergencies.

Remote Patient Monitoring: IoT enables telemedicine and real-time health data transmission.

Smart Hospitals: Connected systems improve medical resource management.



IoT in SDG 4: Quality Education

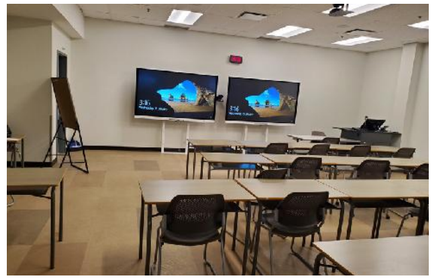
4 QUALITY EDUCATION

IoT supports digital transformation in education by enabling smart classrooms, interactive learning tools, and remote learning systems. Connected devices enhance engagement and provide personalized learning experiences.

Impact:

- Increases access to education for students in remote areas.
- Creates safe and efficient learning environments through smart attendance and monitoring systems.
- Promotes digital literacy and prepares students for future technology-driven jobs.

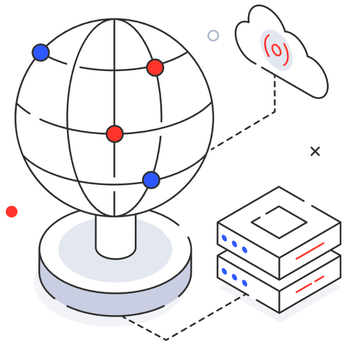
Examples:



Smart Classrooms:
IoT-enabled devices help teachers monitor attendance, classroom environment, and student engagement.

Campus Safety:
IoT-based security systems ensure safe learning spaces for students.

Remote Learning:
IoT-powered wearables and devices assist in tracking students' participation in online classes.



IoT in SDG 6: Clean Water and Sanitation

IoT supports sustainable water management by connecting sensors, pipelines, and monitoring systems to ensure clean and safe water access. It enables real-time tracking of water quality, leakage detection, and efficient irrigation control for both urban and rural areas.

Impact:

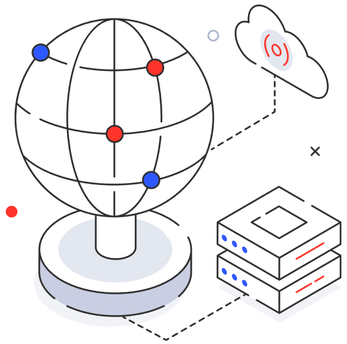
- Ensures safe and clean water through continuous quality monitoring.
- Reduces water wastage by detecting leaks and optimizing distribution.
- Improves agricultural productivity through smart irrigation systems.

Examples:

Water Quality Sensors:
Detect contaminants and ensure safe drinking water.

Smart Irrigation:
IoT regulates water usage in agriculture.

Leak Detection Systems:
Prevent water loss in distribution networks.



IoT in SDG 7: Affordable and Clean Energy

7 AFFORDABLE AND CLEAN ENERGY

IoT plays a crucial role in energy management and renewable energy systems. Smart meters and sensors monitor power usage, detect faults, and optimize energy distribution across grids.

Impact:

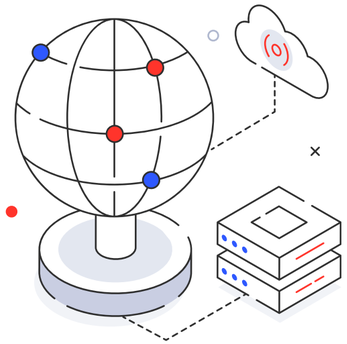
- Reduces energy waste and carbon emissions.
- Encourages efficient energy consumption in homes and industries.
- Promotes the adoption of solar and wind energy through smart monitoring.

Examples:

Smart Grids:
IoT sensors manage electricity distribution, detect outages, and balance energy loads.

Energy Monitoring:
Smart meters and connected appliances optimize energy usage in homes and industries.

Solar Energy Systems:
IoT tracks solar panel performance and predicts maintenance needs.



IoT in SDG 9: Industry, Innovation, and Infrastructure

9 INDUSTRY, INNOVATION AND INFRASTRUCTURE

IoT drives industrial digitalization by connecting machines, tools, and control systems to improve production efficiency and safety. It supports predictive maintenance, process automation, and smart logistics.

Impact:

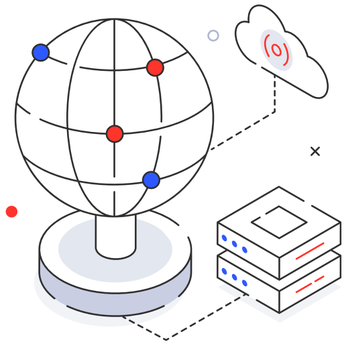
- Enhances industrial productivity and competitiveness.
- Reduces downtime and maintenance costs.
- Creates innovation opportunities in manufacturing and supply chains.

Examples:

Smart Manufacturing: Sensors collect data on machine performance to prevent downtime.

Logistics and Supply Chain: Real-time tracking of goods enhances efficiency and transparency.

Smart Infrastructure: IoT assists in monitoring bridges, roads, and public utilities for preventive maintenance.



IoT in SDG 11: Sustainable Cities and Communities

IoT enables the development of smart cities by integrating data from transportation, waste management, energy, and safety systems to improve quality of life.

Impact:

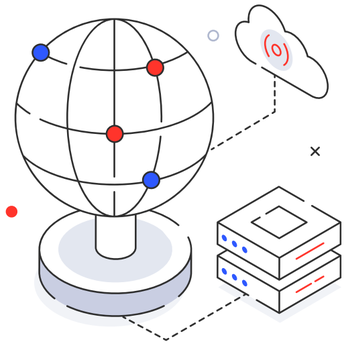
- Reduces traffic congestion and air pollution.
- Enhances public safety through smart surveillance and lighting systems.
- Promotes efficient waste management and water usage.

Examples:

Smart Traffic Systems: Reduce congestion and air pollution through sensor-based monitoring.

Waste Management: IoT bins monitor waste levels for efficient collection.

Environmental Monitoring: Air and water quality sensors help city planners maintain clean environments.



IoT in SDG 13: Climate Action

13 CLIMATE ACTION

IoT provides valuable environmental data to help governments and organizations monitor and respond to climate change. Sensors track temperature, humidity, carbon emissions, and natural disasters.

Impact:

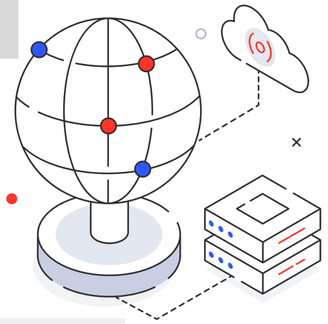
- Improves climate prediction and disaster preparedness.
- Enables better resource management in agriculture and forestry.
- Supports sustainable environmental policies through accurate data.

Examples:

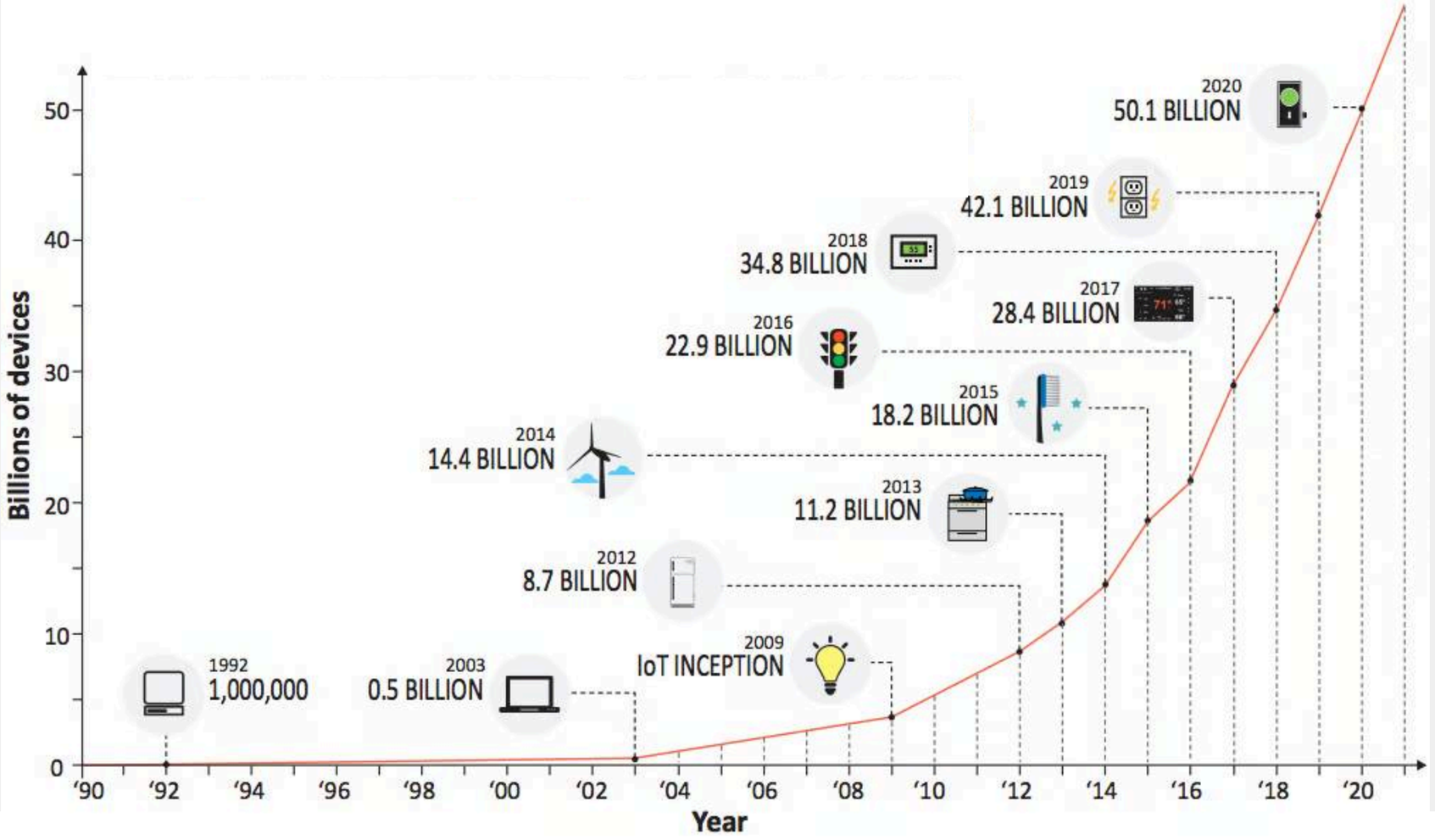
Weather Stations: IoT sensors collect temperature, humidity, and rainfall data for early disaster warnings.

Agriculture Monitoring: IoT enables precision farming to reduce water and fertilizer waste.

Carbon Footprint Tracking: Smart devices help monitor and reduce industrial emissions.



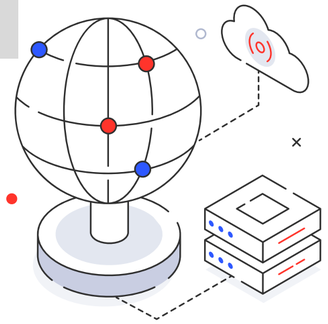
IoT Connected Possibility



The estimation shows that by 2020, the world would have over 50 billion connected devices. This projection highlights the rapid expansion of IoT technology, connecting not only people but also machines, vehicles, and smart devices globally.

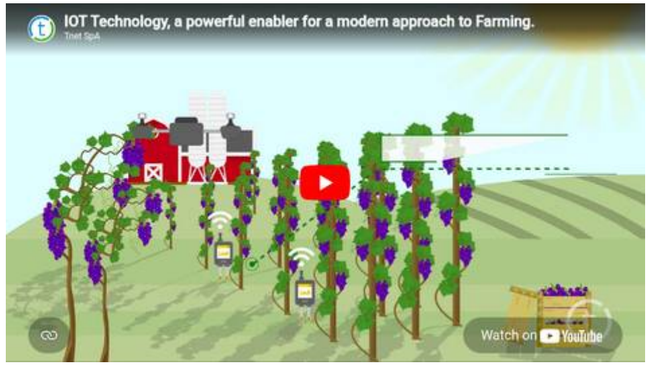
DOI:10.48550/arXiv.2104.06768

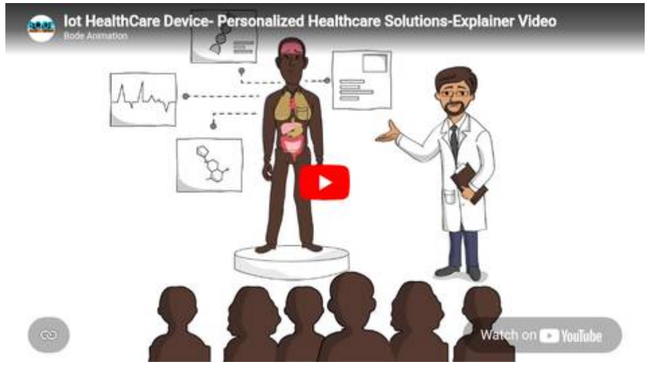
10.3390/s22239271

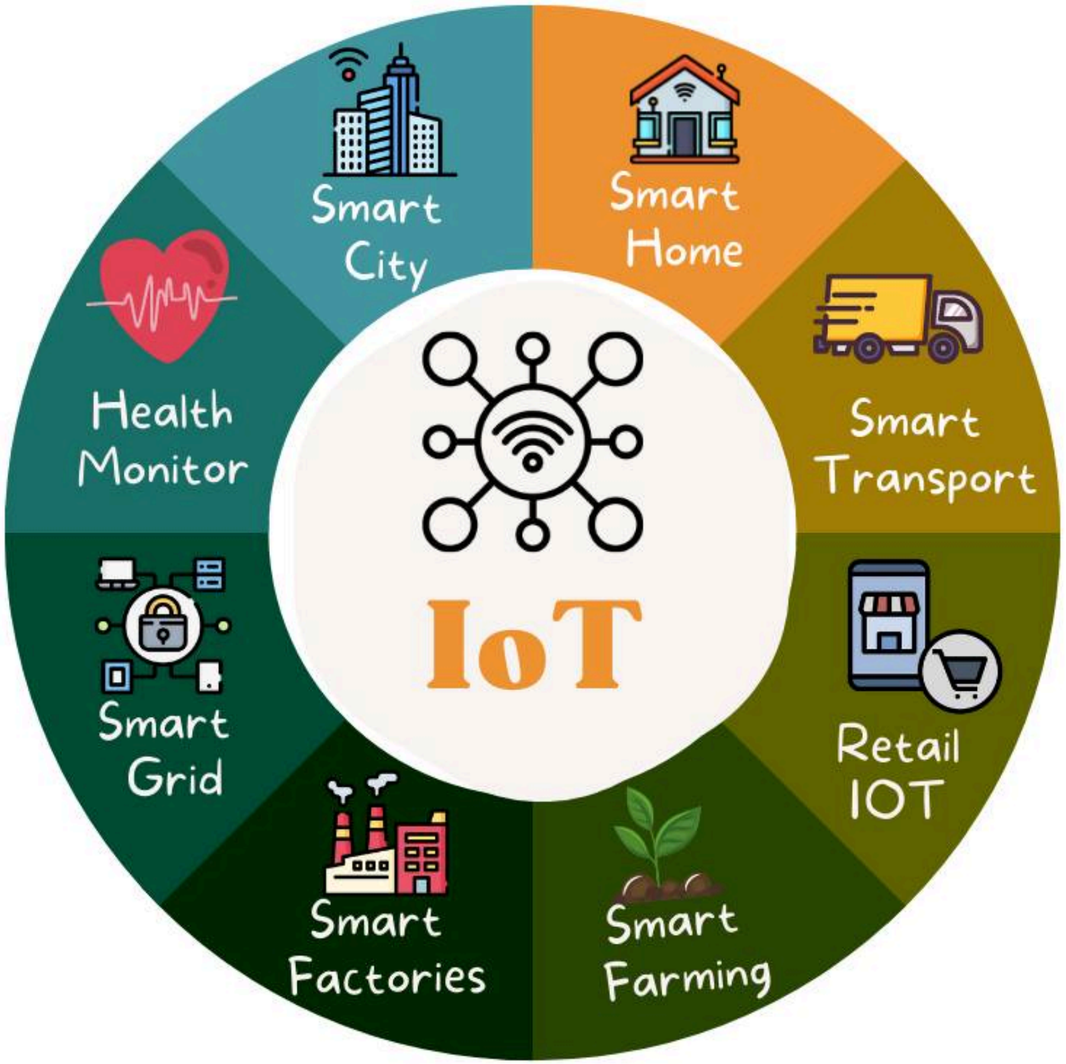


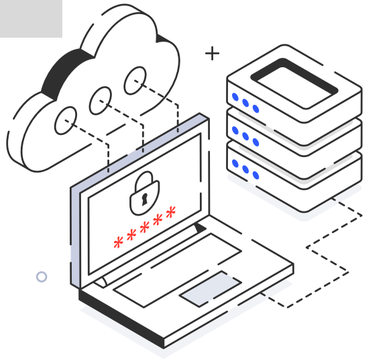
Internet of Things Applications

The IoT is applied in many fields such as smart cities, smart homes, transportation, healthcare, agriculture, industry, energy, and retail. It helps connect devices and systems to make operations smarter and more efficient.

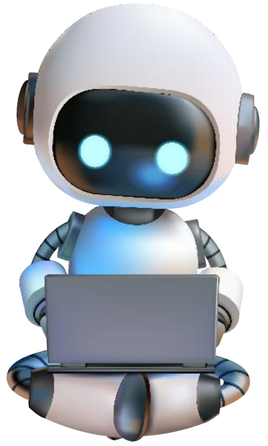








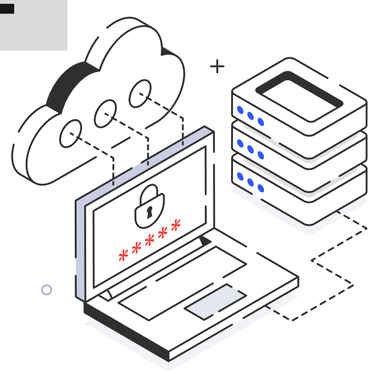
Internet of Things Architecture



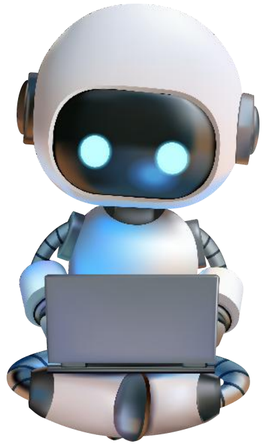
The IoT architecture defines how various components such as sensors, networks, data storage, and applications interact with each other. It provides the structural design that allows devices to communicate, exchange data, and perform automated actions effectively.

Service-Oriented Architecture (SOA-based)

Application Programming Interface (API-based)

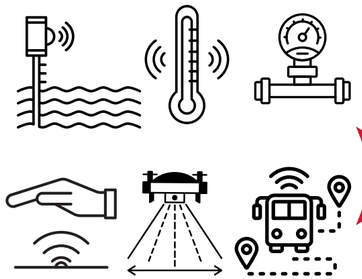
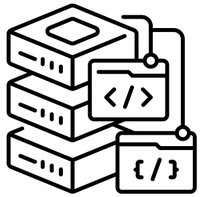


IoT Architecture: SOA-Based

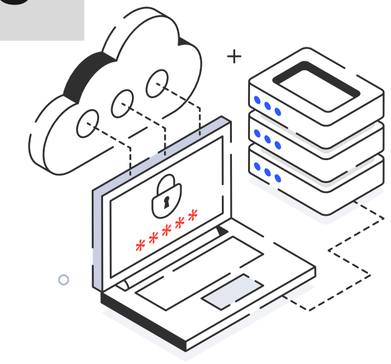


Service-Oriented Architecture (SOA) is an approach where all IoT components are designed as independent services that communicate with one another over a network. Each service performs a specific function—such as data collection, data processing, or device control—and can be reused or integrated into larger systems.

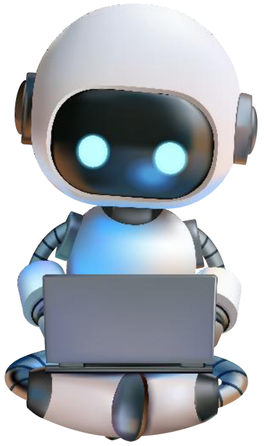
SOA focuses on interoperability, reusability, and modularity. This means different IoT devices and systems (even from different manufacturers) can work together through standardized service interfaces.

Device/Sensing Layer	Network Layer	Service Layer	Application Layer
			
Includes sensors and actuators that collect or act on data.	Transmits data between devices and cloud services.	Defines and manages services (e.g., “temperature reading service,” “lighting control service”).	Uses the services to build user applications (e.g., smart home app, healthcare monitoring).

SOA-Based Architecture



IoT Architecture: SOA-Based



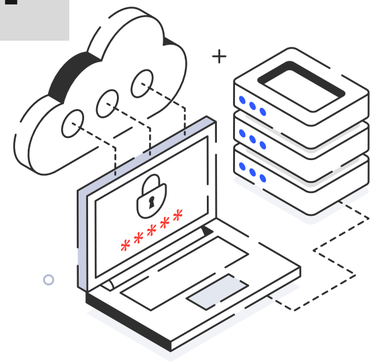
Example:
In a smart agriculture system, one service collects soil moisture data, another service controls irrigation, and a third analyzes weather conditions. These services communicate through a unified service layer, allowing flexible control and easy integration with other systems.

Advantages:

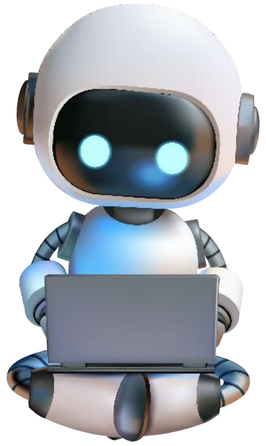
- Promotes scalability and reusability of services.
- Allows interoperability among heterogeneous devices.
- Simplifies system maintenance and upgrades.

Limitation:

- More complex to design and manage due to the need for multiple service definitions and interfaces.



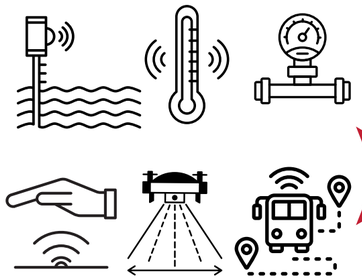
IoT Architecture: API-Based



API-based architecture uses APIs (Application Programming Interfaces) to enable communication between IoT devices, cloud platforms, and applications. APIs act as "bridges" that allow different systems to exchange data and functions securely and efficiently.

In this model, IoT devices send and receive data through RESTful APIs, MQTT APIs, or WebSocket APIs—making integration with cloud services, mobile apps, and databases much simpler.

Devices and Sensors



Collect data from the environment.

Gateway/Edge Node

Transmits data to the cloud through APIs.

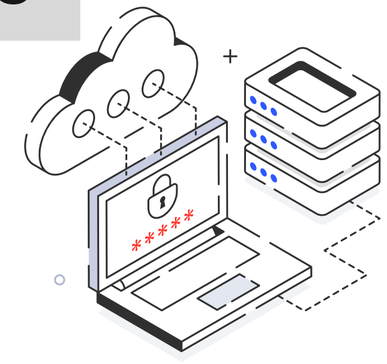
Cloud Platform

Stores, analyzes, and processes data (e.g., AWS IoT Core, Google Cloud IoT).

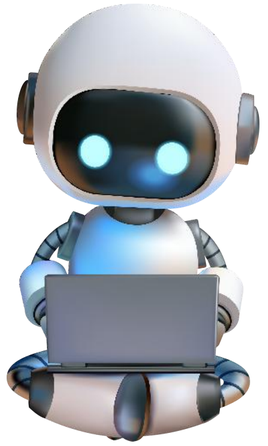
Application Interface

Provides API endpoints for developers to build dashboards, alerts, or mobile apps.

API-Based Architecture



IoT Architecture: API-Based



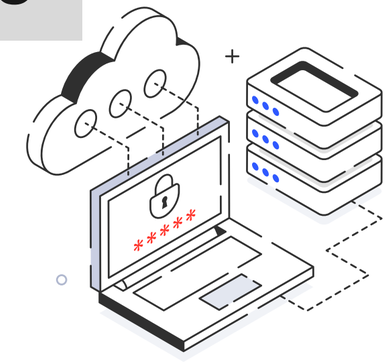
Example:
In a smart home system, sensors send temperature and motion data to the cloud using REST APIs. The user's mobile app retrieves this data through the same API to display real-time conditions and control devices remotely.

Advantages:

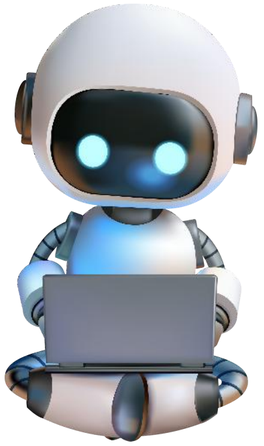
- Provides real-time communication and easy integration with third-party applications.
- Supports cross-platform development (mobile, web, or cloud).
- Easier for developers to extend or modify applications.
-

Limitation:

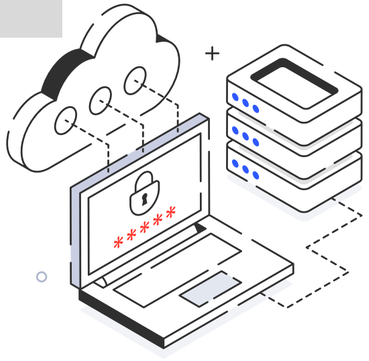
- Security must be properly managed since APIs expose data endpoints.
- High dependency on internet connectivity and cloud performance.



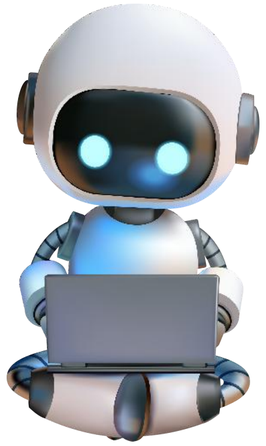
IoT Architecture: API-Based vs SOA-Based



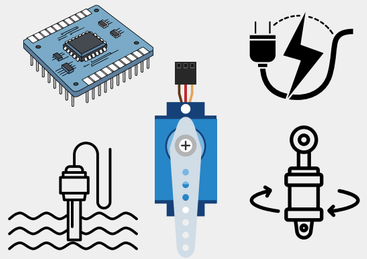
Aspect	SOA-Based Architecture	API-Based Architecture
Core Concept	Based on modular and reusable services	Based on communication through APIs
Communication	Service-to-service communication	Application-to-application or device-to-cloud
Integration	Complex but supports large-scale systems	Easier and faster integration
Flexibility	Highly flexible, scalable, and interoperable	Flexible and developer-friendly
Example Use Case	Smart city or industrial IoT platform	Smart home or mobile-controlled IoT app



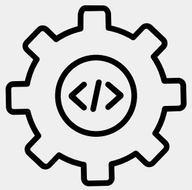
Internet of Things Technology



The IoT relies on a combination of hardware, software, connectivity, and data technologies that work together to collect, transmit, process, and analyze information from the physical world. Each component plays a crucial role in enabling IoT devices to function intelligently and autonomously.



Hardware:
Hardware forms the physical foundation of IoT systems. It includes all the devices, sensors, and electronic components that interact with the physical environment to collect and respond to data.



Software:
Software provides the intelligence and control of IoT systems. It processes sensor data, runs algorithms, and allows communication between devices and cloud services.

Connectivity and Network:
Connectivity enables IoT devices to communicate and share data with other devices, gateways, and cloud servers. It is the link that connects the physical world to the digital world.

Data and Cloud Platform:
IoT generates large amounts of data that must be stored, analyzed, and visualized. Cloud platforms and analytics systems handle this process efficiently.

Security and Privacy:
Security technologies protect IoT systems from unauthorized access, data breaches, and cyberattacks.


4IR Explained in 3 Minutes! (The Future is HERE)
Fourth Industrial Revolution
AI
FOURTH
INDUSTPAL
REVOLUTION
IN 3 MINUTES
EXPLAINED
Watch on YouTube

Summary

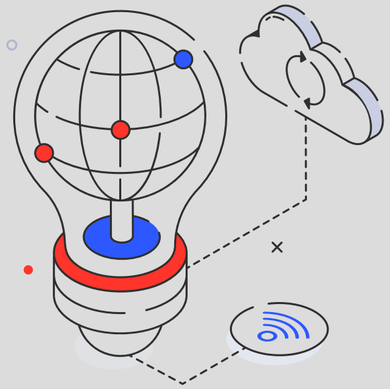
The Internet of Things (IoT) connects physical devices with digital systems to enable automation, data exchange, and intelligent decision-making.

It plays a key role in transforming industries under Industry 4.0, integrating technologies such as sensors, networks, cloud computing, and data analytics.

IoT concepts provide the foundation for understanding how devices, communication, and platforms work together to create smart environments that improve efficiency, productivity, and sustainability.

Chapter 2

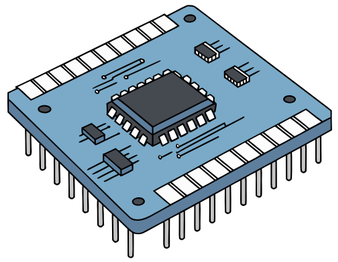
INTERNET OF THINGS DEVICES



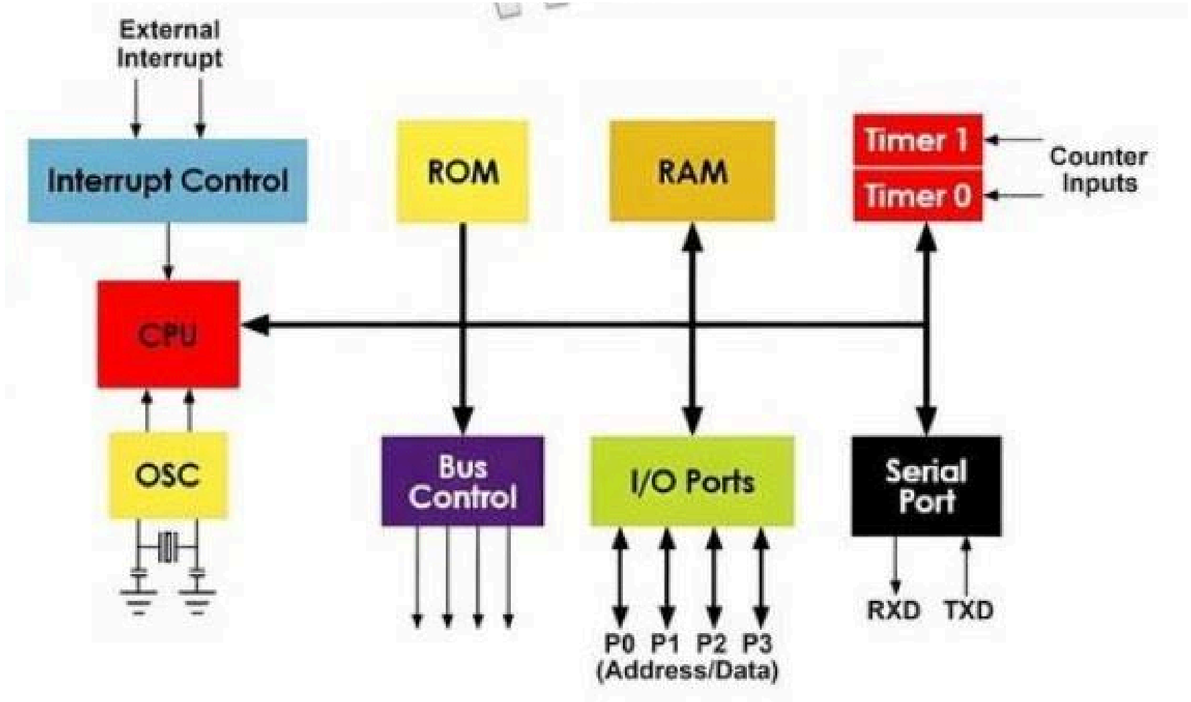
Introduction to IoT Devices

IoT devices are physical objects embedded with sensors, actuators, microcontrollers, and communication modules that enable them to collect, process, and exchange data through the internet. These devices act as the building blocks of IoT systems, bridging the physical and digital worlds to support automation, monitoring, and intelligent decision-making.

An IoT device can be as simple as a temperature sensor or as complex as a smart home security camera. Regardless of complexity, all IoT devices share the same goal there are to sense, communicate, and respond intelligently.

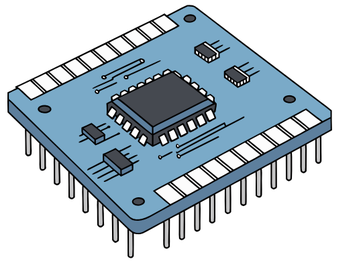


Microcontroller (MCU) in Internet of Things

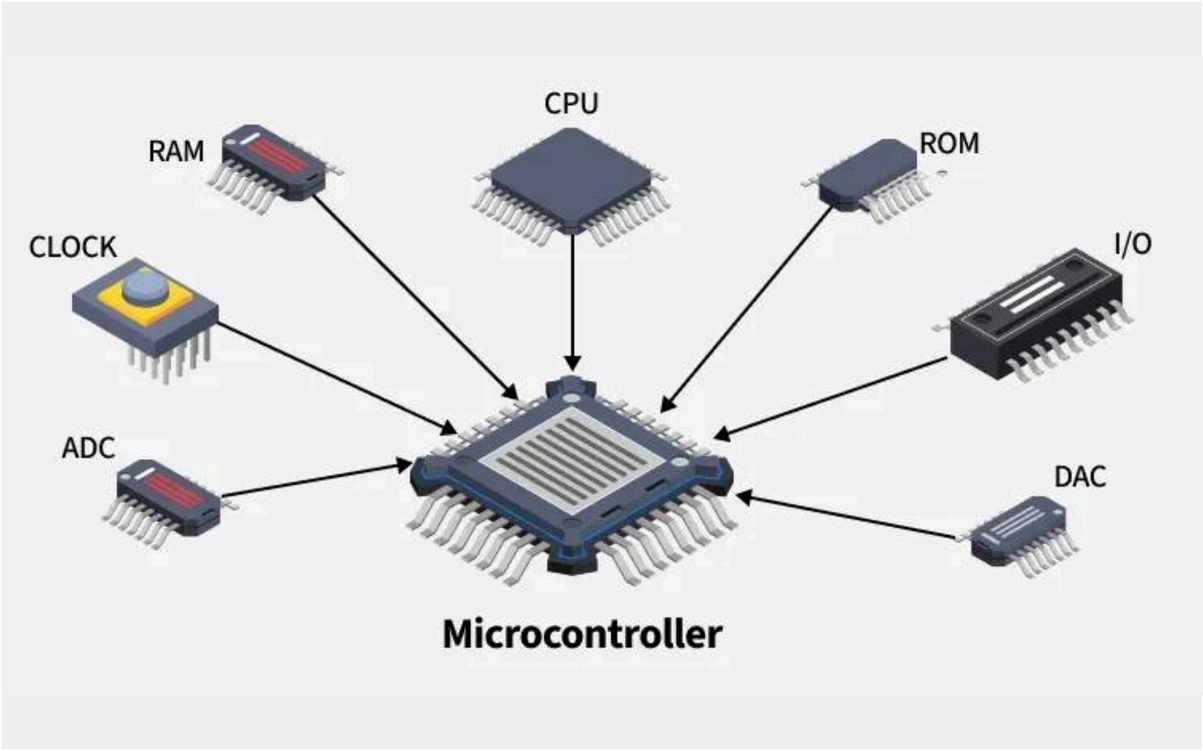


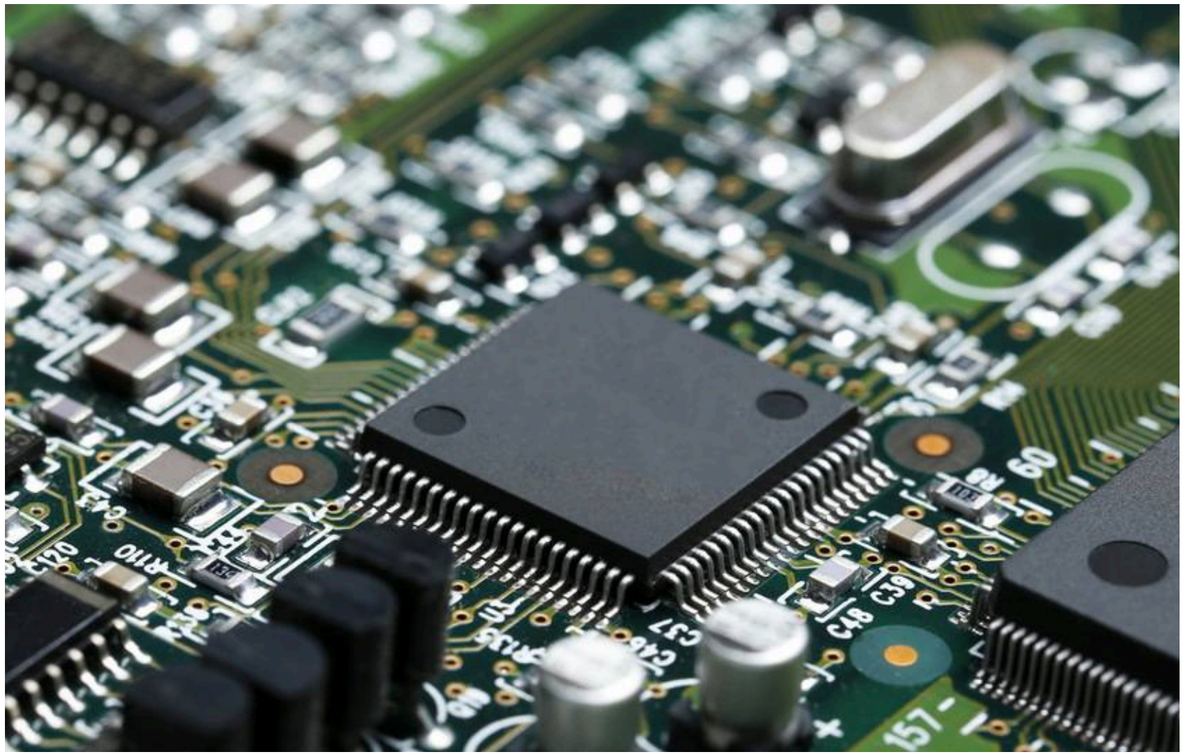
A Microcontroller Unit (MCU) is the central component or “brain” of an IoT embedded system. It is a small, low-cost computer built on a single chip that contains a processor (CPU), memory (RAM and ROM), and input/output (I/O) ports. These components allow the MCU to collect data from sensors, process it, and control actuators or other devices automatically.

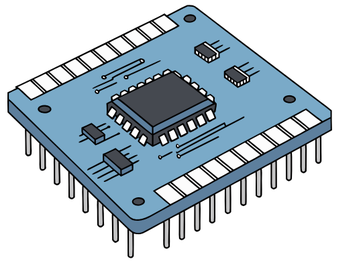
In an IoT system, the MCU acts as the main controller that manages all operations — it reads input signals from sensors, makes decisions based on programmed logic, and sends output commands to actuators or cloud platforms. Many modern MCUs are equipped with built-in communication modules such as Wi-Fi, Bluetooth, or LoRa, enabling wireless data transfer to the internet or IoT platforms.



Microcontroller (MCU) in Internet of Things







Microcontroller (MCU) in Internet of Things

1970s – The Beginning

Intel 8048, 1KB ROM, foundational for embedded control.

1980s – Rise of 8-bit MCUs

Intel 8051 and PIC series, simple and cost-effective for various applications.

1990s – Integration and Versatility

Introduction of flash memory and 16-bit MCUs such as MSP430, offering better performance.

2000s – 32-bit Revolution

ARM Cortex-M MCU introduced greater speed, efficiency, and lower power consumption.
2010s – IoT and Connectivity

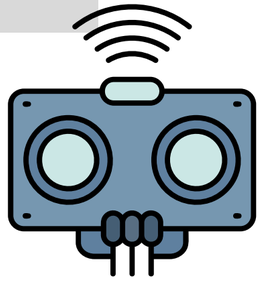
2010s – IoT and Connectivity

MCUs with built-in Wi-Fi, Bluetooth (BLE), and LoRa such as ESP8266, ESP32, and STM32 enabled IoT development.

2020s – Smart, Secure, and AI-ready

Microcontrollers with AI at the edge, enhanced hardware security, cryptographic features, and multicore processing for advanced embedded systems

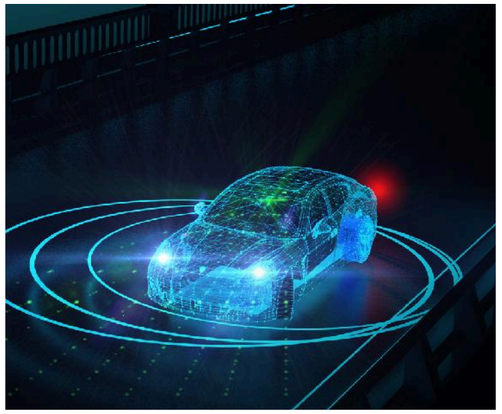
Figure: Evolution of MCU Technology

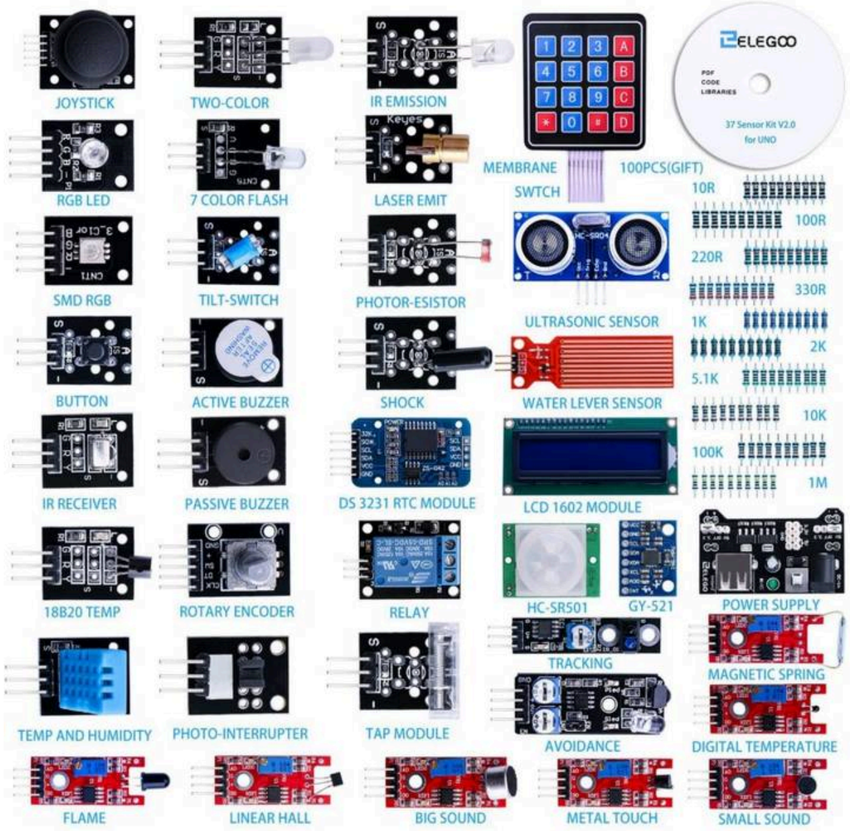


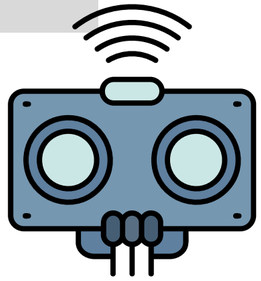
Sensor in Internet of Things

In the Internet of Things (IoT), sensors act as the “things” that allow devices to sense, detect, and collect data from their surrounding environment. They are one of the most essential components in any IoT system because they serve as the main input source for data collection. Without sensors, IoT devices would not be able to interact intelligently with the physical world.

A sensor is a device that converts physical quantities—such as temperature, light, humidity, gas, motion, or pressure—into electrical signals that can be read and processed by a microcontroller (MCU). These signals are then analyzed, stored, or transmitted to a cloud platform for monitoring and decision-making.







Functions of Sensors in IoT

Data Detection

Sensors detect changes in the environment, such as heat, sound, movement, or air quality.

Data Conversion

The detected physical quantity is converted into an electrical signal that can be interpreted by an MCU.

Data Transmission

The sensor sends the processed data to the microcontroller or IoT platform through wired or wireless connections.

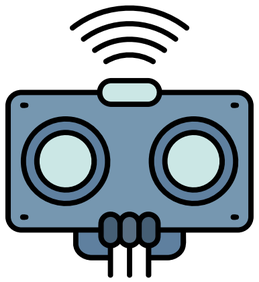
Real-Time Monitoring

Sensors allow continuous, real-time observation of environmental conditions.

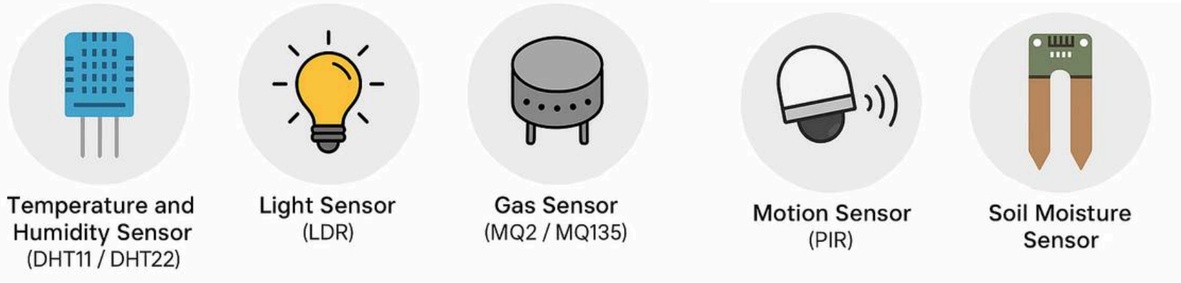
Automation Trigger

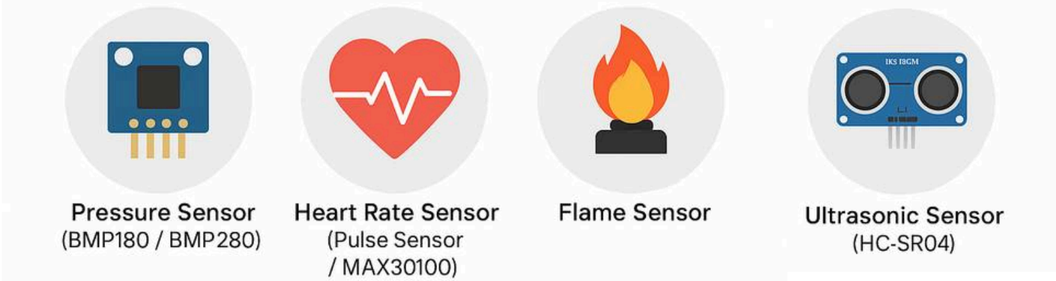
Based on sensor data, the system can automatically perform actions (e.g., turn on lights when it’s dark or start a fan when temperature rises).

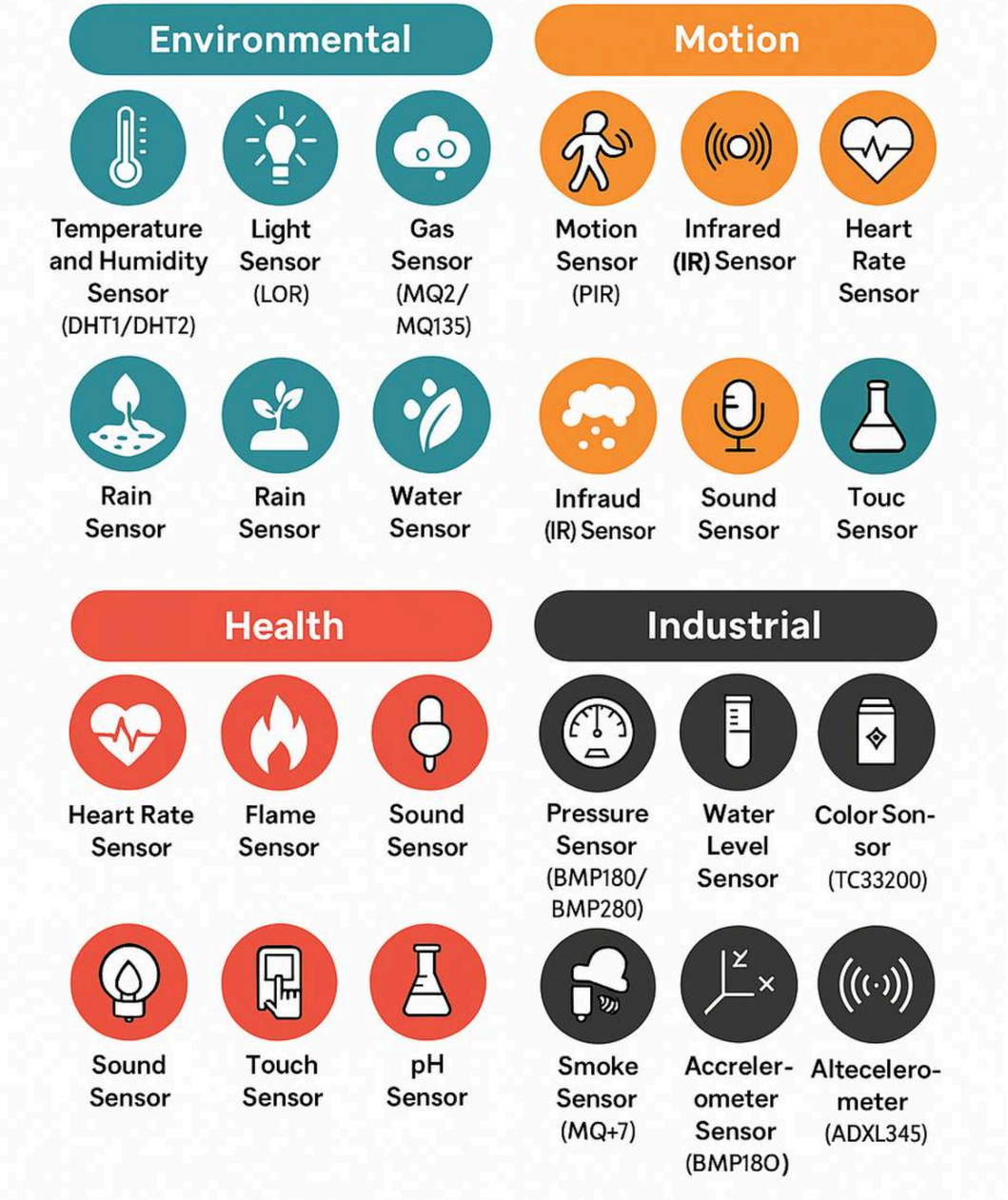
10.3390/s22239271

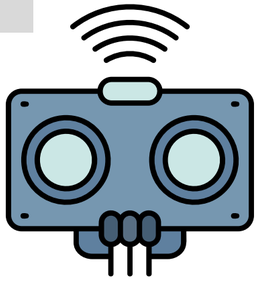


Example of Sensors in IoT









Categories of Sensors in IoT

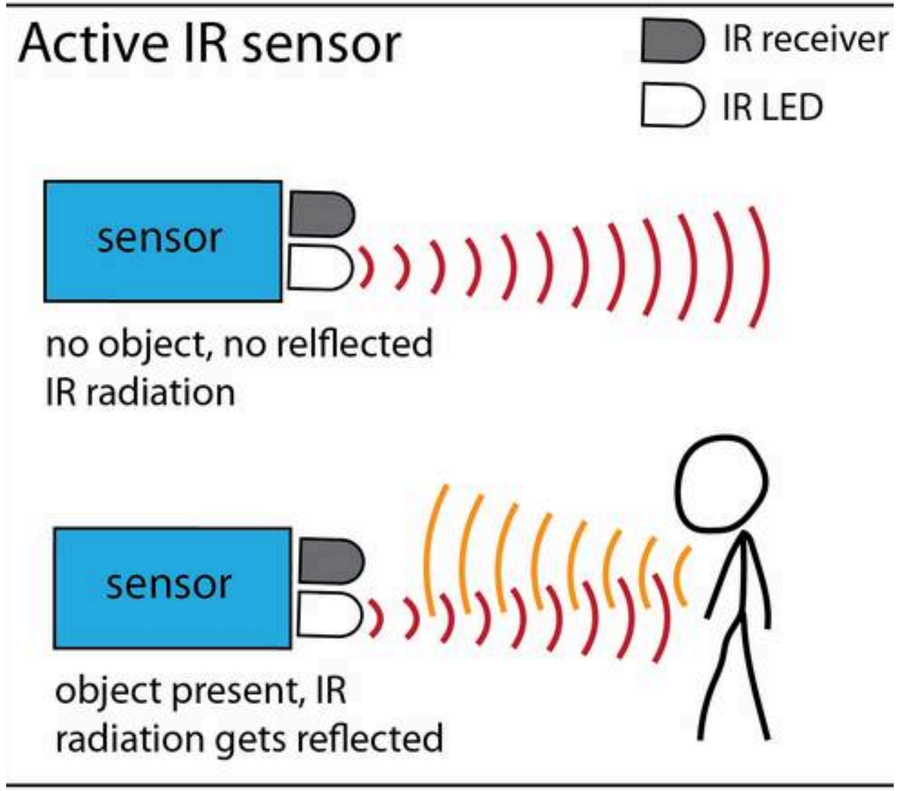
Active Sensor

Active Sensors

Active sensors are sensors that require an external power source to operate.

They emit energy or a signal (such as light, sound, or electromagnetic waves) toward the target and then measure the response or reflection of that energy to detect physical changes in the environment.

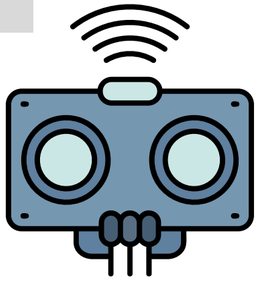
In other words, active sensors send out their own signal and analyze how it interacts with the surroundings.



How They Work:

- The sensor emits energy (e.g., light, radio, or sound wave).
- The emitted signal interacts with the target (object, surface, or material).
- The sensor detects the reflected or returned signal.
- The system processes the data to calculate distance, speed, position, or other information.

DOI:10.48550/arXiv.1805.00881



Categories of Sensors in IoT

Active Sensor

Active Sensors

Advantages:

- Can operate in dark or poor environmental conditions.
- Provides high accuracy and faster response time.
- Can measure both distance and velocity.
- Suitable for automation and industrial applications.

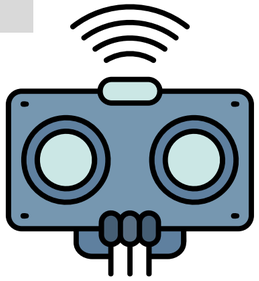
Disadvantages:

- Requires external power source (higher energy use).
- More expensive and complex than passive sensors.
- May cause interference with nearby electronic systems.

Sensor Type	Working Principle	Application Example
Ultrasonic Sensor (HC-SR04)	Emits ultrasonic waves and measures echo time to calculate distance.	Smart parking, obstacle detection.
LIDAR (Light Detection and Ranging)	Sends laser pulses and measures reflection time to map surfaces.	Autonomous vehicles, 3D mapping.
Infrared (IR) Proximity Sensor	Emits infrared light and detects reflection from objects.	Automatic doors, motion sensors.
Radar Sensor	Sends radio waves and measures Doppler shift or reflection.	Vehicle speed detection, weather radar.
Active RFID	Transmits signal using internal power to communicate with reader.	Asset tracking, logistics.

Examplae: In a Smart Parking System, an ultrasonic active sensor emits sound waves to detect the presence of a vehicle in a parking spot. The reflected signal determines whether the space is empty or occupied, and the data is sent to an IoT dashboard in real time.

DOI:10.48550/arXiv.1805.00881



Categories of Sensors in IoT

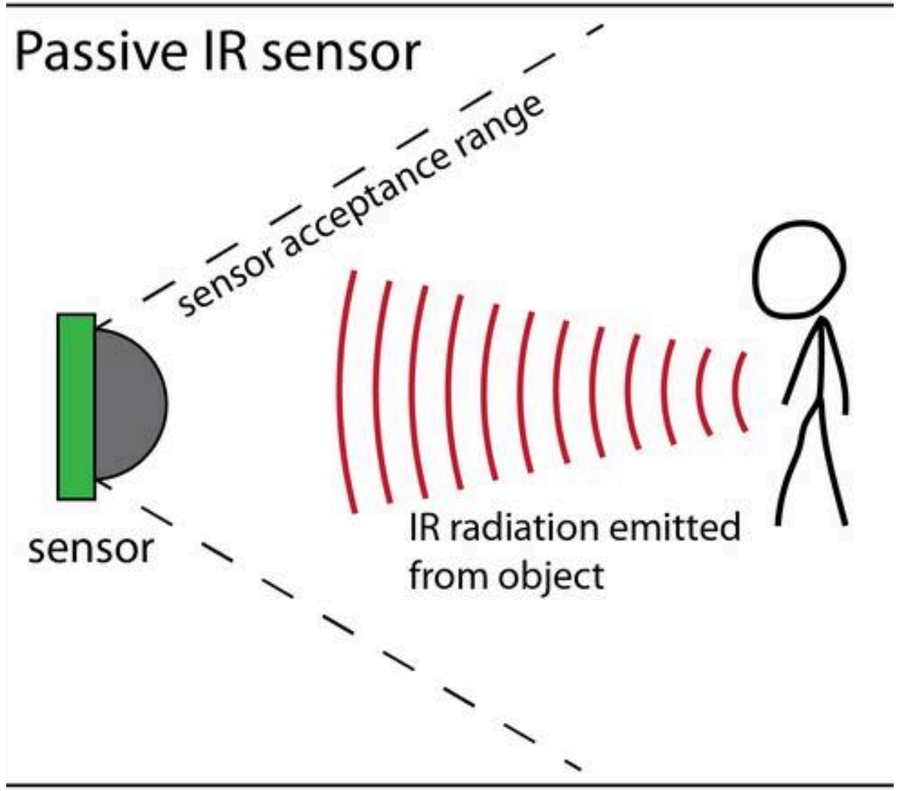
Passive Sensor

Passive Sensors

Passive sensors are sensors that do not require any external power source to operate.

Instead of sending out their own signal, they detect or measure energy that is naturally emitted, reflected, or generated by the object or environment being observed.

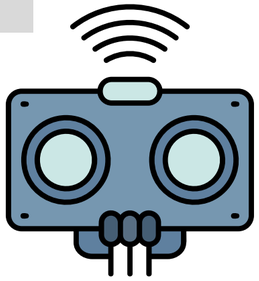
In short, passive sensors receive signals only, without actively sending anything out.



How They Work:

- The environment or target naturally emits or reflects energy (such as heat, light, or vibration).
- The passive sensor detects this energy.
- The sensor converts it into an electrical signal.
- The microcontroller or processor interprets the signal to measure a physical quantity (e.g., temperature, light intensity, sound level).

DOI:10.48550/arXiv.1805.00881



Categories of Sensors in IoT

Passive Sensor

Passive Sensors

Advantages:

- Does not need external power for signal generation.
- Simple design and low maintenance.
- Low cost and energy-efficient.
- Reliable for continuous environmental monitoring.

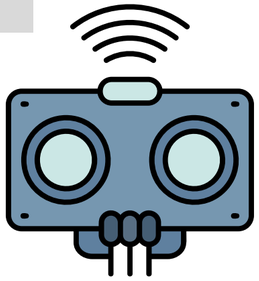
Disadvantages:

- Cannot work effectively in total darkness or no-signal environments.
- Output signal may be weak and needs amplification.
- Slower response and less accurate than active sensors in some applications.
-

Sensor Type	Working Principle	Application Example
Thermocouple	Generates voltage based on temperature difference between two metals.	Temperature monitoring.
Photodiode / LDR (Light Dependent Resistor)	Changes resistance according to light intensity.	Smart lighting, solar tracking.
Microphone	Converts sound waves into electrical signals.	Voice recognition, sound monitoring.
PIR (Passive Infrared) Sensor	Detects infrared radiation emitted by human body heat.	Motion detection, security systems.
Accelerometer	Measures change in motion or vibration using internal sensing elements.	Smartphones, fitness trackers.

Example: In a smart street lighting system, a passive LDR (Light Dependent Resistor) detects the ambient light level. When the light decreases at night, the LDR resistance increases, signaling the controller to automatically turn on the streetlights.

DOI:10.48550/arXiv.1805.00881



Types of Sensor in IoT

Position Sensors

Detect the position, movement, or displacement of an object.
Applications:

- Robotics arm movement detection.
- Vehicle tracking and navigation.
- Door or valve position sensing.

Force and Pressure Sensors

Measure mechanical force or pressure exerted on a surface.
Applications:

- Tire pressure monitoring systems.
- Industrial process control.
- Weather stations and altimeters.

Temperature and Humidity Sensors

Measure temperature and moisture content in the air.
Applications:

- Smart homes (HVAC systems).
- Agriculture (greenhouse monitoring).
- Weather and environmental monitoring.

Biosensors

Detect biological or chemical reactions and convert them into electrical signals.
Applications:

- Healthcare and wearable devices.
- Medical diagnostics.
- Environmental water quality monitoring.

Velocity and Acceleration Sensors

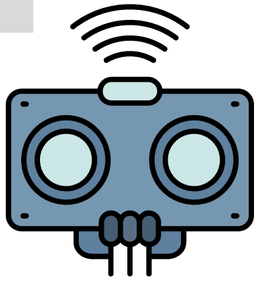
Measure speed (velocity) and change of motion (acceleration) of objects.
Applications:

- Vehicle motion detection.
- Fitness tracking and smartphones (step counting).
- Robotics and drones (orientation control).

Light Sensors

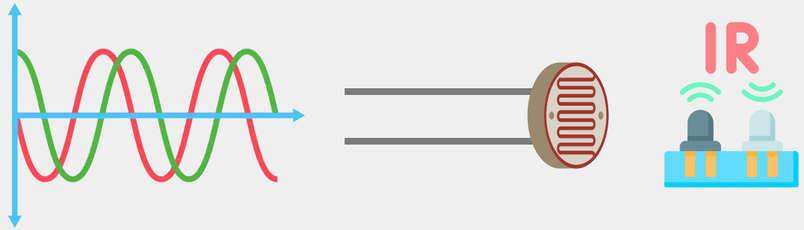
Measure intensity or brightness of light in the environment.
Applications:

- Automatic street lighting systems.
- Smart indoor lighting control.
- Solar energy tracking and power management.



Digital and Analogue Sensor in IoT

Based on the type of output signal they produce, sensors are classified into two main categories: Analog sensors and Digital sensors.



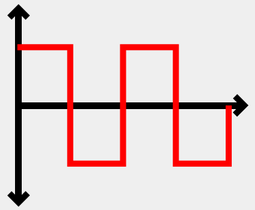
Analog sensors produce a continuous output signal that varies in proportion to the physical quantity being measured.

Analog sensors provide gradual and detailed information but require an Analog-to-Digital Converter (ADC) to be read by most microcontrollers.

The voltage or current changes smoothly over time, representing the actual value of the measured parameter.

Example:

- An LDR (Light Dependent Resistor) changes resistance continuously as light intensity varies.



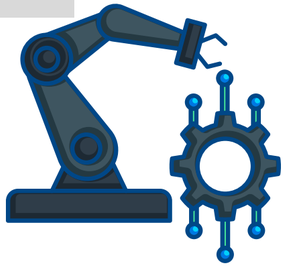
Digital sensors produce a discrete or binary output — either HIGH/LOW (1/0) or a stream of digital data.

Digital sensors give ready-to-use, noise-free signals that can be read directly by microcontrollers without extra conversion.

They often contain internal circuits that process the analog signal and convert it into a digital format automatically.

Example:

- A PIR motion sensor gives a digital output (1 when motion is detected, 0 when idle).



Actuators in Internet of Things

In an IoT system, actuators serve as output devices that carry out actions based on signals received from the microcontroller (MCU). They are responsible for converting electrical signals into physical movement or mechanical actions. This allows IoT systems to not only sense and process data but also respond to it automatically.

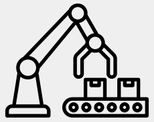
Actuators play a key role in automation. When a sensor detects a change in the environment, the MCU processes this information and sends a command to the actuator to perform a specific task. For example, when a temperature sensor detects high heat, the MCU sends a signal to the actuator to turn on a fan.

Functions of Actuator:

Converts electrical energy into mechanical or physical action.

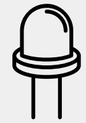
Executes commands received from sensors and MCU.

Enables automation and real-time response.



Acts as the “hands” of an IoT system.

Common Types of Actuators:



LED – Produces light output.

Relay – Acts as an electronic switch.

Buzzer - Produces sound as an alert.

Solenoid Valve – Controls liquid or air flow.

Motor (DC, Servo, Stepper) – Creates motion or rotation.

1965
Thomson **Performance Pak** electromechanical actuators are developed

1967
The first actuators for use in **garden tractors** and **farm equipment** are released

1974
First line of actuators with parallel motors and both acme and ball screw drive is released

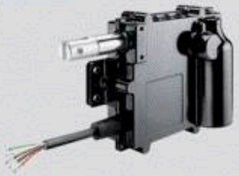
1982
The **Electrak line of actuators** is released

1987
Electrak 205 and the first line of **MCS controls** are released

2007
Electrak Pro series released

2012
WhisperTrak series released

2013
Electrak Throttle is released



2013
Max Jac heavy duty actuator released

2016
Electrak HD is released

Hydraulic

Shafer RV Series

Biffi Morin Water Hydraulic Scotch Yoke

Bettis & Biffi Hydraulic Scotch Yoke & Linear Actuators

Electric

Bettis RTS Intelligent Control & Fail-Safe

Bettis XTE3000 Intelligent Multiturn

Bettis EHO Spring-Return

Pneumatic

Bettis G-Series Scotch Yoke

Biffi Morin Stainless Steel Scotch Yoke

Bettis Rack & Pinion Aluminum

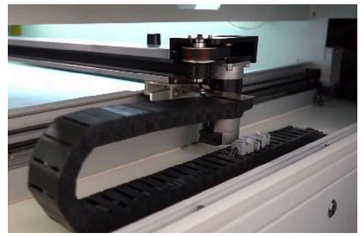
OVATION USERS' GROUP

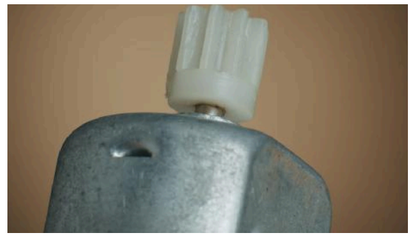
ALL TYPES OF AUTOMATION & MECHANICAL VALVES @ONE PLACE

FLOW-TECH ENTERPRISE

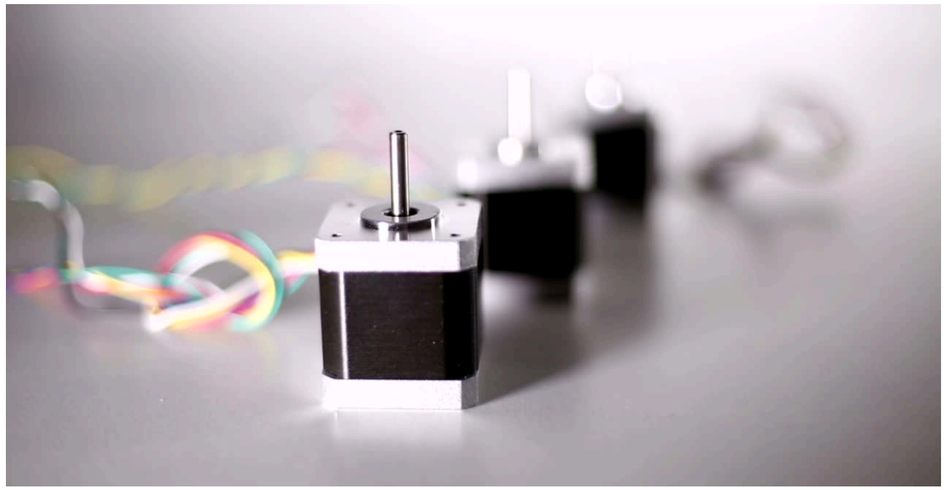
192/13, Dr.Nanjappa Road,
King's Complex, Adjacent to
'Church of Christ the King',
Coimbatore - 641018 (T.N)

Motor in Internet of Things





A motor is one of the most important actuators used in the Internet of Things (IoT). It converts electrical energy into mechanical motion, allowing IoT systems to perform physical actions such as rotation, movement, or mechanical control. Motors are widely used in smart devices, automation systems, and robotics, making them an essential part of IoT applications.



Functions of Motor in IoT:

Receives control signals from the MCU

Converts electrical signals into mechanical rotation or movement.

Performs automation tasks such as opening, moving, or turning components.

Types of Motors:

DC Motor – Provides continuous rotation; used in fans, pumps, and wheels.

Servo Motor – Offers precise angular movement; used in robotic arms or automatic doors.

Stepper Motor – Moves in small, fixed steps; used in 3D printers and positioning systems.

Stepper Motor

Disk Motor

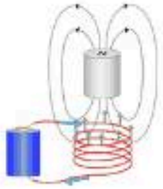
Servo Motor

Synchronous Motor

Repulsion Motor

Hysteresis Motor

Reluctance Motor



Homopolar Motor

Shaded-pole Motor

Linear DC Motor

AC Induction Motor

DC Brushed Motor

DC Brushless Motor

In-wheel Motor

Series Wound DC Motor

Out-runner Motor

Capacitor Motor

Torque Motor

Magnetic Levitation Motor

Compound Wound DC Motor

Three-phase Induction Motor

Single-phase Induction Motor

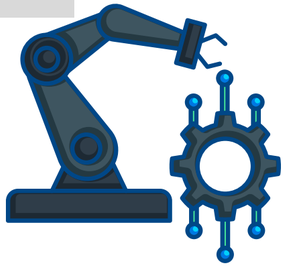
Permanent Magnet DC Motor

Universal Motor

Hydraulic Motor

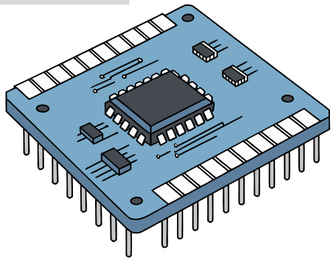
In-runner Motor

Piezoelectric Motor



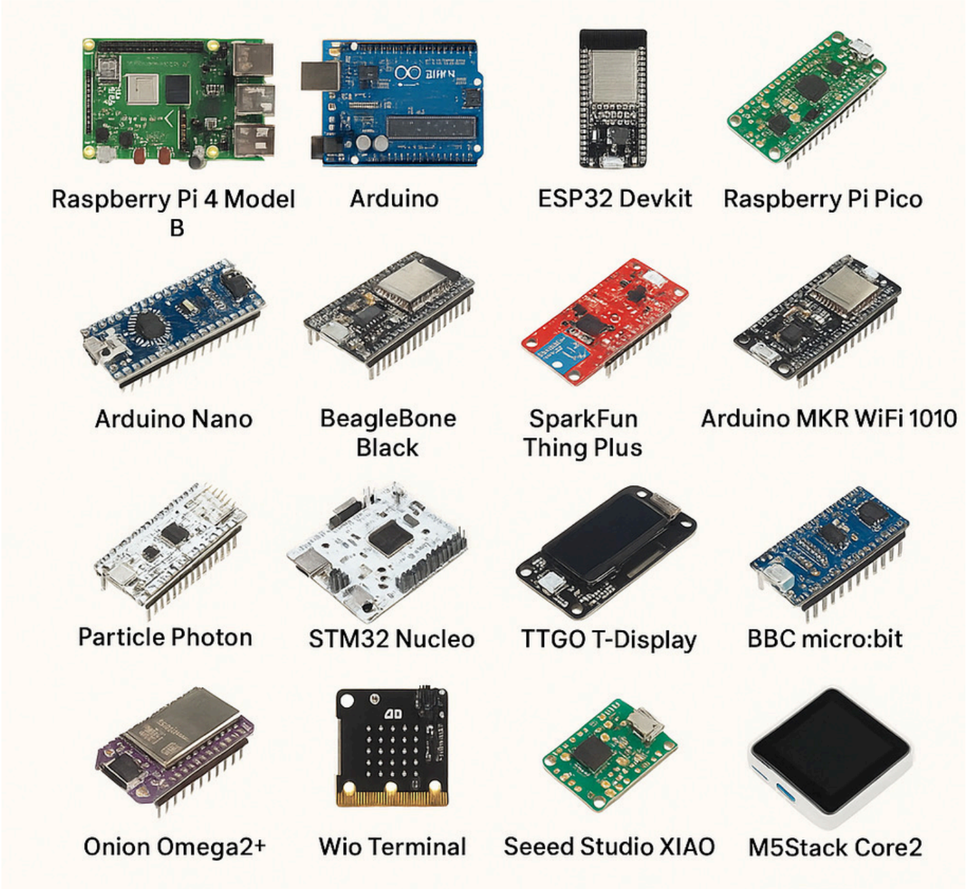
Difference Between Motor and Actuator

Aspect	Motor	Actuator
Definition	A motor is a device that converts electrical energy into mechanical energy (rotation or motion).	An actuator is a device that converts a control signal (electrical, hydraulic, or pneumatic) into physical action such as movement, pressure, or opening/closing.
Main Function	Produces continuous rotational movement, such as spinning a fan or wheel.	Performs a specific action, such as pushing, pulling, opening, or turning something.
Control	Usually controlled directly by a power supply or speed controller.	Controlled by a microcontroller or automation system based on data from sensors.
Type of Motion	Typically provides rotational motion.	Can provide rotational or linear motion, depending on the actuator type.
Examples	DC motor, Stepper motor, Servo motor.	Servo motor, Solenoid, Relay, Hydraulic actuator, Pneumatic actuator.
Role in IoT	Acts as a component within an actuator to produce motion.	Serves as the main output device that performs actions based on system commands.



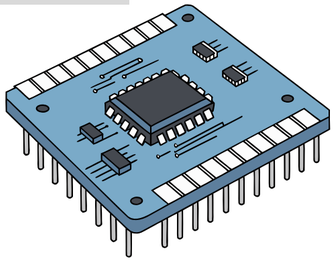
Internet of Things Development Board

IoT development boards combine microcontrollers with built-in sensors, communication modules, and I/O pins to simplify prototyping.



Board Name	Wi-Fi	BLE	OS Support	Best For
ESP32 DevKit			Arduino / IDF	General IoT
Raspberry Pi 4			Linux	AI / Multimedia
Arduino Uno R4			Arduino	Learning basics
NodeMCU ESP8266			Arduino	Budget IoT projects
Pico W			MicroPython	Sensor & cloud experiments
Arduino Nano 33 IoT			Arduino	Wearables
XIAO ESP32-C3			Arduino	TinyML
BeagleBone Black			Linux	Industrial IoT
Feather ESP32			Arduino	Wearables, quick prototyping
SparkFun Thing Plus			Arduino	Firebase / Cloud projects

Comparison Table: IoT Development Boards



Internet of Things Development Board

When choosing an IoT board, there are several important factors to consider.

Ease of use

Boards that are simple to program and set up are ideal for beginners who have limited experience in coding or electronics.

Connectivity

Boards with built-in Wi-Fi or Bluetooth save time and effort since they can connect easily to the internet or other devices.

Community support

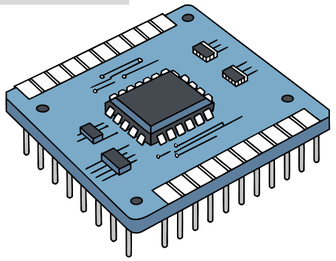
A large online community provides tutorials, examples, and troubleshooting help, making learning easier.

Cost and Availability

Budget-friendly boards are preferred especially when working on multiple projects or learning experiments.

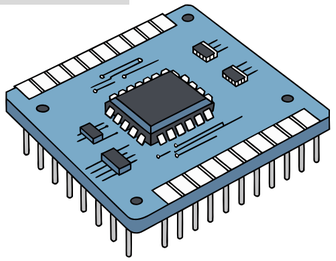
Expandable

Boards that support sensors, cloud integration, or AI features are more versatile and useful for advanced IoT applications.

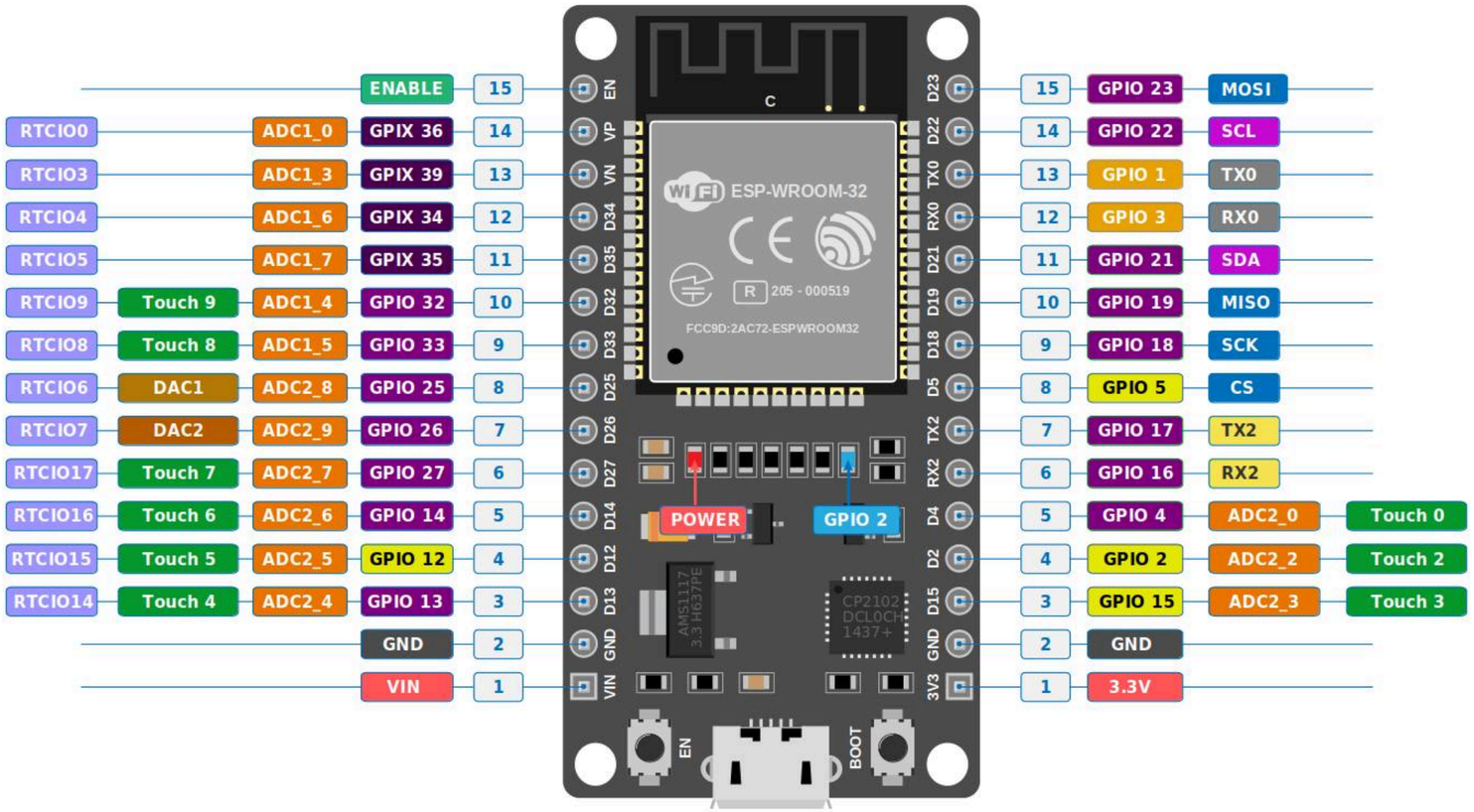


Internet of Things Development Board

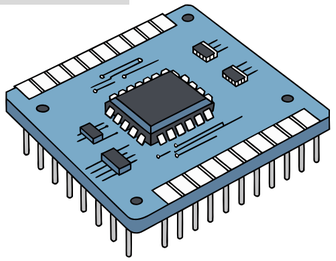
Specification Aspect	Typical Values / Options	Example Modules	Remarks
Ports (Pin Layout & Logic Level Signal)	3.3V logic (ESP8266, ESP32, STM32) 5V logic (Arduino Uno, Mega) GPIO, Analog, PWM, UART, SPI, I²C ports	Arduino Uno (5V), ESP32 (3.3V)	Ensure voltage matching between devices; use level shifter when mixing 3.3V and 5V.
MCU Memory, Speed, Power Use & Security	Speed: 16–240 MHz Flash: 32 KB–4 MB Power: 3.3V–5V operation Security: Encryption, secure boot	Arduino Uno (16 MHz, 32 KB Flash) ESP8266 (80 MHz, 4 MB Flash) ESP32 (240 MHz, 4 MB Flash, secure boot)	Higher speed = faster processing; more memory = handles complex tasks; low power = better for battery use.
Connectivity Types	Wireless: Wi-Fi, Bluetooth/BLE, LoRa, GSM/4G/5G Wired: Ethernet, USB, UART	ESP8266 (Wi-Fi) ESP32 (Wi-Fi + BLE) SIM800L (GSM) Raspberry Pi (Ethernet/Wi-Fi)	Choose based on range, bandwidth, and power requirements.
Cost & Support	Low-cost: ESP8266 ($3–6) Mid-range: ESP32 ($5–9) High-end: Raspberry Pi ($35+)	Arduino Uno (~$8, strong community) ESP32 (~$6, strong IoT support) STM32 (~$15, industrial use)	Lower cost suits prototyping; strong community means easier learning and troubleshooting.



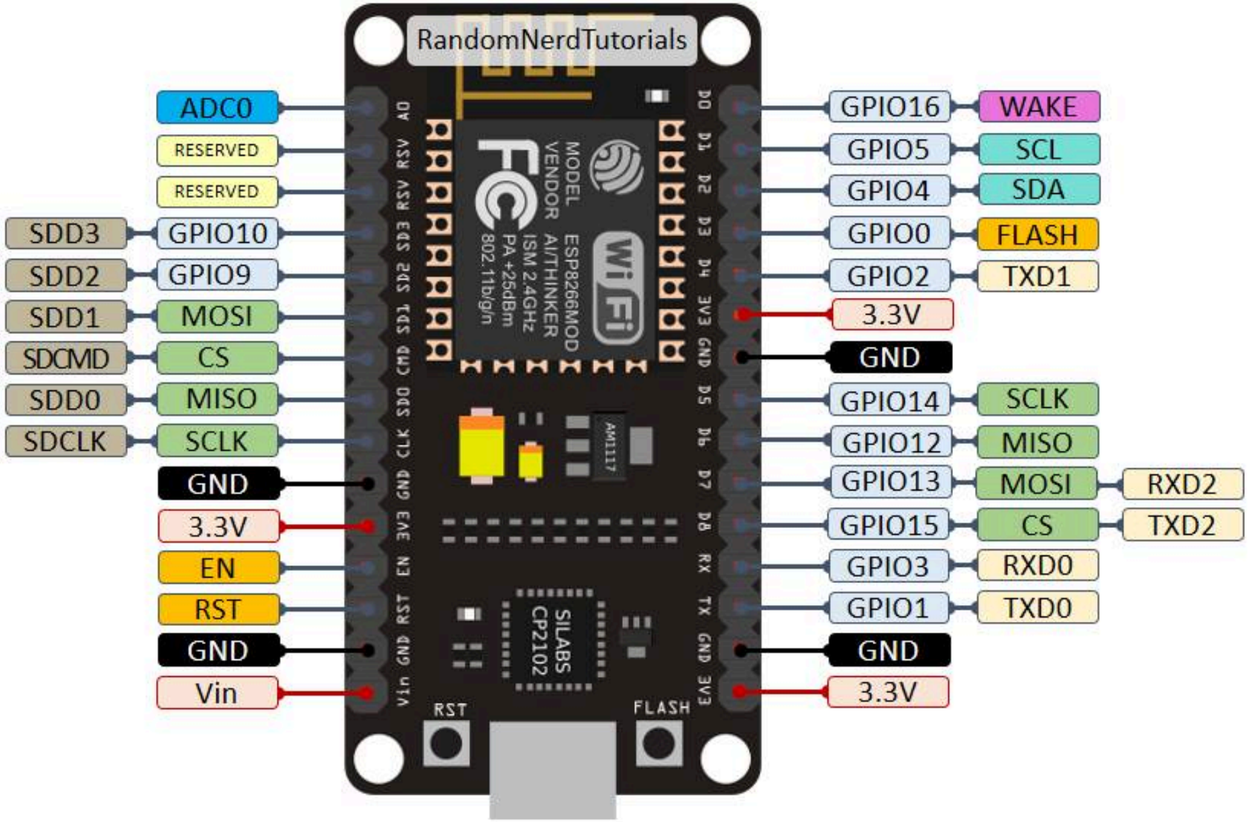
Development Board: ESP32



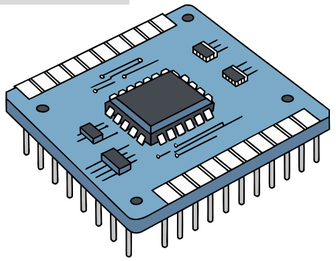
Pin Type	Function	Example Pins
Power	Supply voltage to board and sensors	VIN, 3.3V, GND
Control	Reset or enable the chip	EN
GPIO	General input/output	GPIO0–GPIO39
ADC	Read analog signals	GPIO32–GPIO39
DAC	Generate analog output	GPIO25, GPIO26
Touch	Capacitive touch sensing	GPIO4, GPIO12–GPIO15, GPIO27, GPIO32–GPIO33
UART	Serial communication	GPIO1 (TX), GPIO3 (RX)
SPI	High-speed device communication	GPIO23, GPIO19, GPIO18, GPIO5
I^2C	Sensor communication (two-	GPIO21 (SDA), GPIO22 (SCL)
PWM	Control motor/LED speed	Most GPIO pins



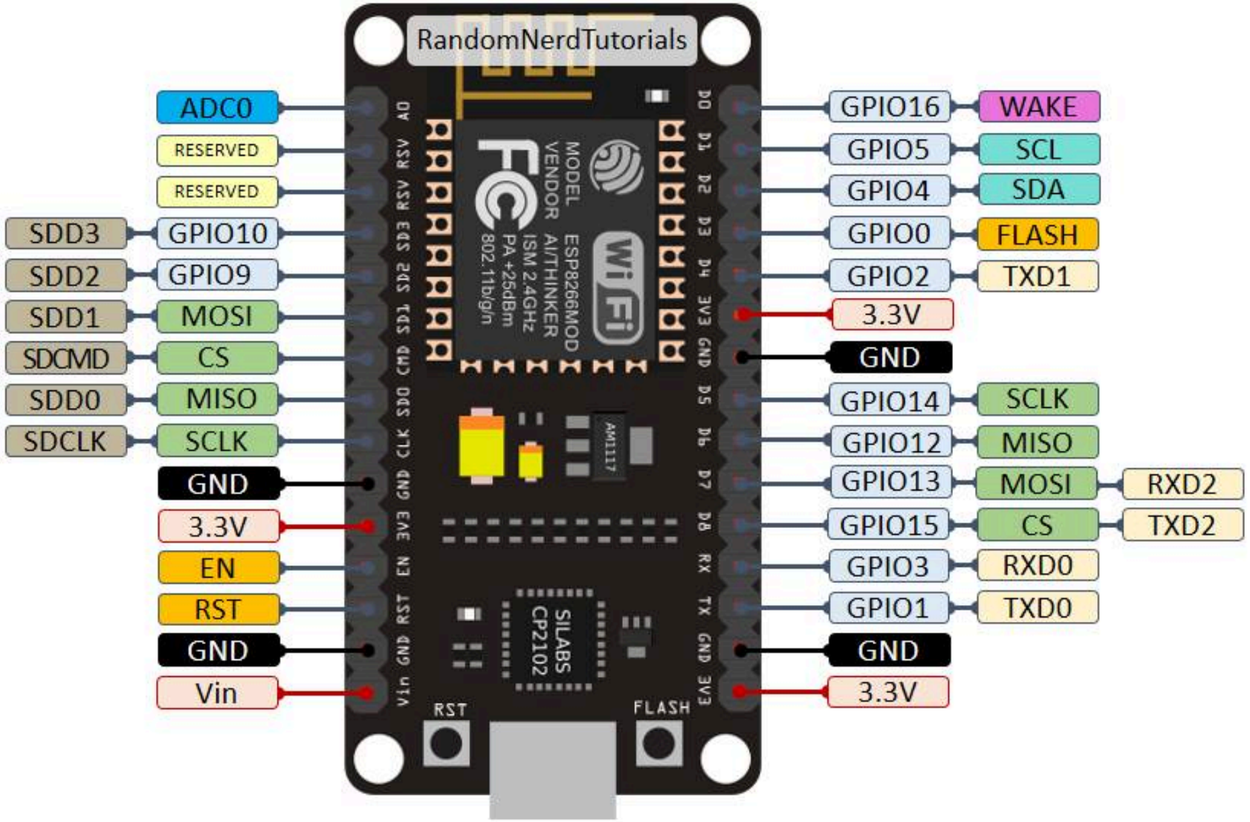
Development Board: ESP8266



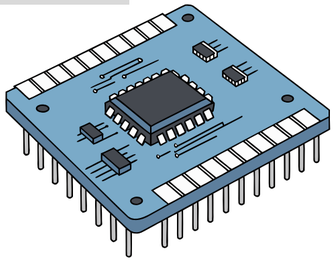
Pin Category	Function	Example Pins	Description
Power	Power supply to and from the board	3.3V, Vin, GND	Provides voltage or ground to components
Control	Reset and enable operations	EN, RST	Used to restart or enable the chip
GPIO	Input/output for sensors and actuators	GPIO0–GPIO16	Controls external devices
ADC	Analog input	ADC0	Reads analog sensor data (0–1V)
UART	Serial communication	GPIO1 (TX), GPIO3 (RX)	Connects to PC or other serial devices
SPI	High-speed communication	GPIO12–GPIO15	For SD cards, displays, etc.
I²C	Two-wire sensor communication	GPIO4 (SDA), GPIO5 (SCL)	Connects to sensors or LCDs
WAKE	Deep sleep wake-up	GPIO16	Wakes up the ESP8266 from sleep



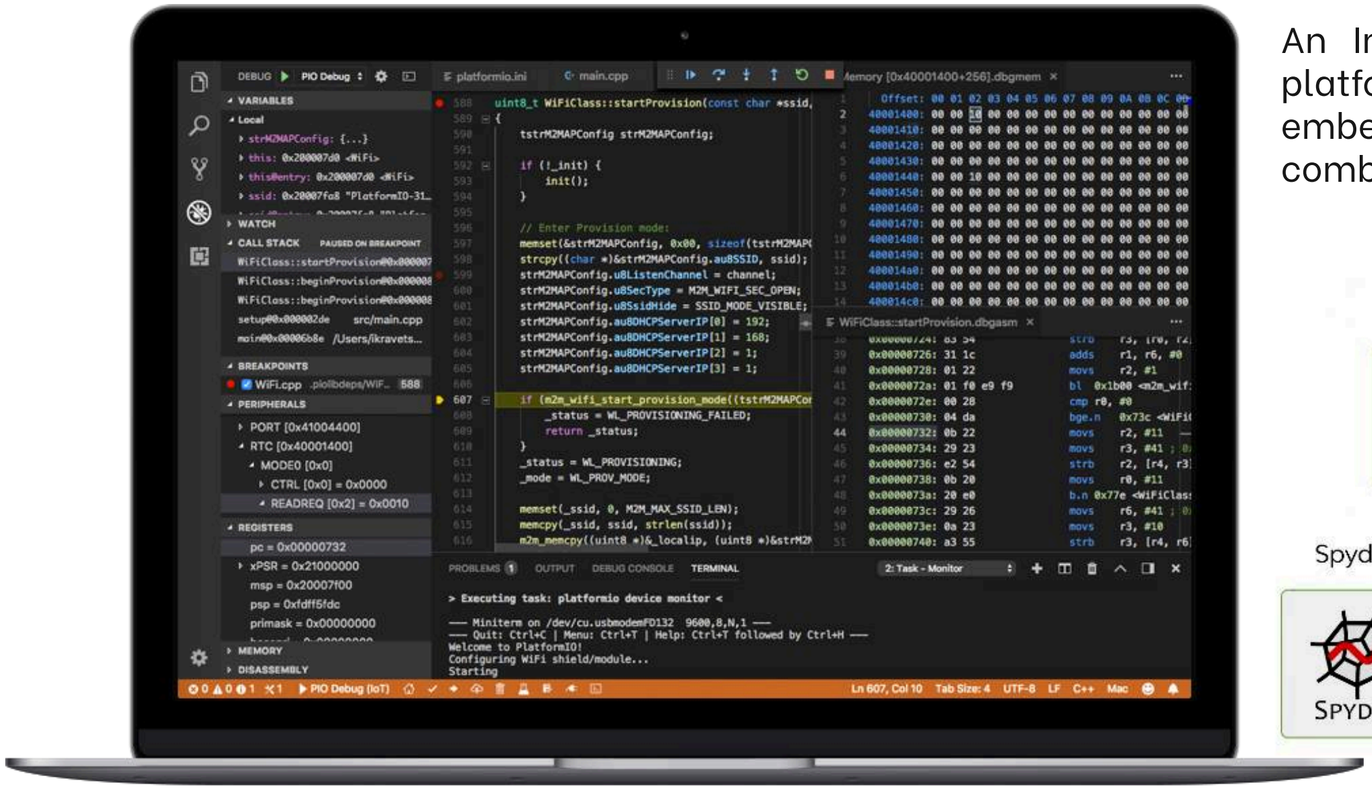
Development Board: ESP8266



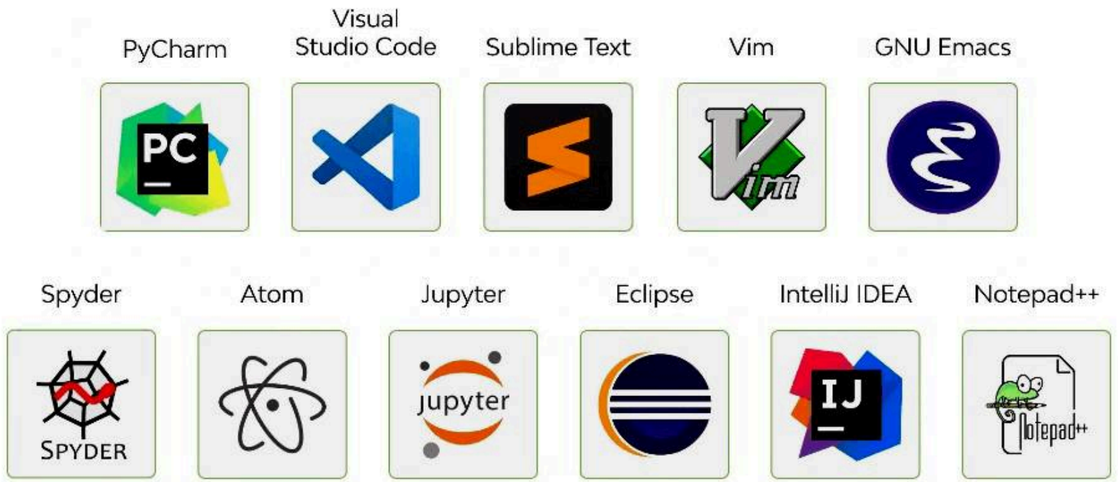
Pin Category	Function	Example Pins	Description
Power	Power supply to and from the board	3.3V, Vin, GND	Provides voltage or ground to components
Control	Reset and enable operations	EN, RST	Used to restart or enable the chip
GPIO	Input/output for sensors and actuators	GPIO0–GPIO16	Controls external devices
ADC	Analog input	ADC0	Reads analog sensor data (0–1V)
UART	Serial communication	GPIO1 (TX), GPIO3 (RX)	Connects to PC or other serial devices
SPI	High-speed communication	GPIO12–GPIO15	For SD cards, displays, etc.
I²C	Two-wire sensor communication	GPIO4 (SDA), GPIO5 (SCL)	Connects to sensors or LCDs
WAKE	Deep sleep wake-up	GPIO16	Wakes up the ESP8266 from sleep

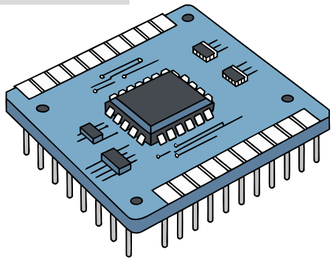


Integrated Development Environment (IDE)



An Integrated Development Environment (IDE) is a software platform used to write, edit, compile, and upload code to embedded system boards. It provides a user-friendly interface that combines all development components in one place.





Integrated Development Environment (IDE)

IDE Name	Supported Boards	Programming Language
Arduino IDE	Arduino, ESP8266, ESP32	C/C++
PlatformIO	Arduino, ESP, STM32, Raspberry Pi Pico	C/C++
MicroPython IDE (Thonny, Mu Editor)	ESP32, Raspberry Pi Pico	Python
Keil uVision	ARM Cortex (STM32, NXP)	C/C++
MPLAB X IDE	PIC Microcontrollers	C/Assembly

Main Functions of an IDE:

1. Code Editor:
   - Allows programmers to write and edit source code (usually in C, C++, or MicroPython).
   - Features include syntax highlighting, auto-completion, and error checking.

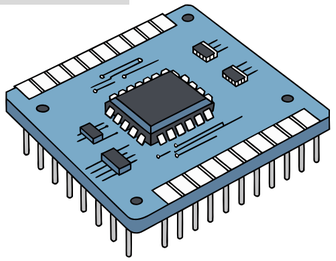
2. Compiler/Assembler:
   - Converts the written source code into machine code (binary format) that can be executed by the microcontroller (MCU).

3. Uploader/Programmer:
   - Transfers the compiled code from the computer to the embedded board through USB, serial, or wireless connection.

4. Debugger:
   - Helps detect and fix errors by showing variable values, memory usage, and system behavior during execution.

IDE: Arduino

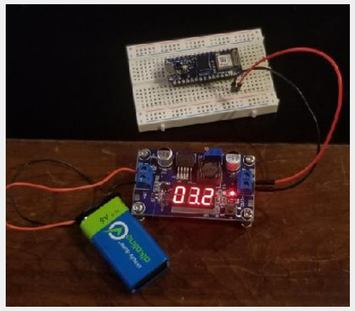
Tool Name	Function / Description	Example Use
Verify / Upload	Compiles (checks for errors) and uploads the program to the microcontroller board.	After coding, click Upload to send the program to an Arduino or ESP board.
Select Board & Port	Lets you choose the correct board type (e.g., Arduino Uno, ESP32) and communication port (COM/USB).	Select ESP32 Dev Board and COM3 before uploading.
Sketchbook	Displays your saved sketches (Arduino programs).	Access previously written codes easily.
Board Manager	Installs or updates support packages for new hardware boards.	Install board package for ESP8266 or ESP32.
Library Manager	Adds libraries to enable sensors, displays, or communication modules.	Install DHT11 or WiFi library.
Debugger	Helps find and fix code errors by running the program step-by-step.	Check why a sensor reading isn't updating.
Search	Finds keywords, variables, or functions inside your code.	Quickly locate Serial.print statements.
Open Serial Monitor	Displays real-time output or data from the microcontroller via serial communication.	View sensor readings printed using Serial.println().
Open Serial Plotter	Graphically visualizes data received from the microcontroller in real time.	Plot temperature or humidity data over time.

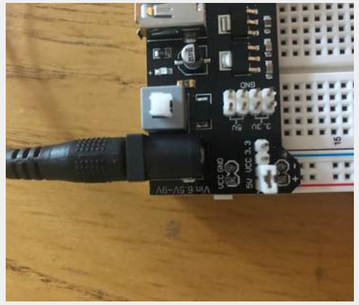


Comunication Module and Power Suply

To transmit data to the cloud or other devices, IoT devices use communication modules that support various wireless protocols:

- Wi-Fi (ESP8266, ESP32) – for home automation
- Bluetooth/BLE – for short-range communication
- LoRa/LoRaWAN – for long-range low-power IoT systems
- ZigBee/Z-Wave – for smart home networks
- Cellular (3G/4G/5G/NB-IoT) – for mobile IoT systems
- These modules ensure the device can send and receive data efficiently in different network environments.





Every IoT device requires a stable power source.

- Common sources:
  - USB or DC power adapter
  - Batteries (AA, Li-ion)
  - Renewable energy (e.g., solar panels for outdoor sensors)
- Power efficiency is crucial in IoT design to extend device lifespan, especially in remote or battery-powered systems.

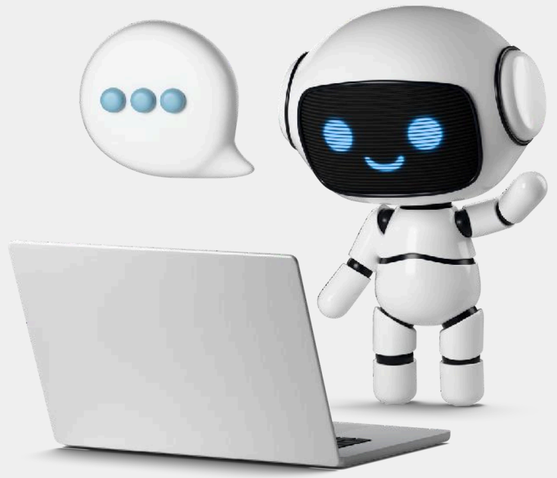
Summary

IoT devices integrate hardware and software to sense, process, and communicate data autonomously.

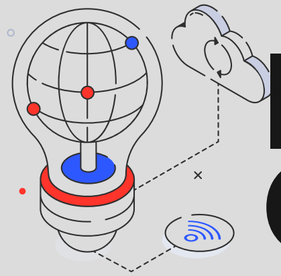
Each component plays a vital role:

- Sensors detect changes in the environment.
- Microcontrollers process and make decisions.
- Actuators and motors perform actions.
- Communication modules enable connectivity.
- IDEs allow programming and development.

Together, these elements create a complete ecosystem that powers modern IoT applications—from smart homes and agriculture to healthcare and industrial automation.



Chapter 3



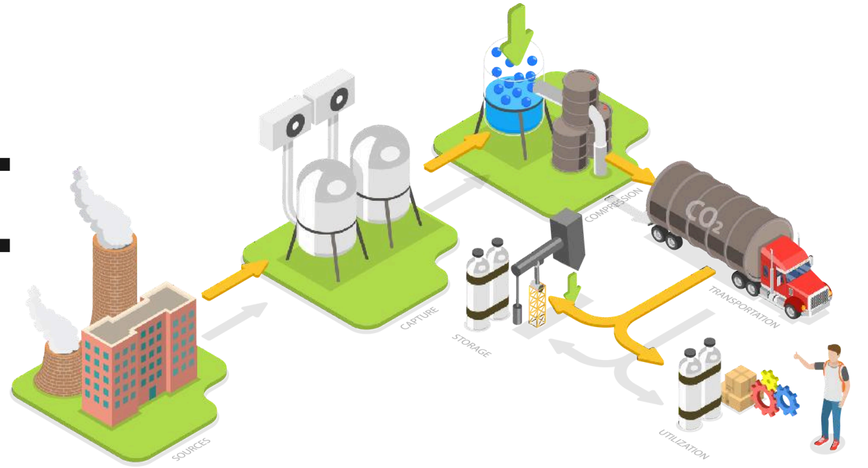
INTERNET OF THINGS CONNECTIVITY & APPLICATION

Introduction to Internet Connectivity

Connectivity is the backbone of the Internet of Things (IoT), enabling devices to communicate, share data, and perform tasks intelligently. This chapter introduces various types of internet connectivity, network and data protocols, and the IoT framework that supports seamless communication between devices, platforms, and cloud services.

DOI:10.48550/arXiv.1805.00881

Types of Internet Connectivity



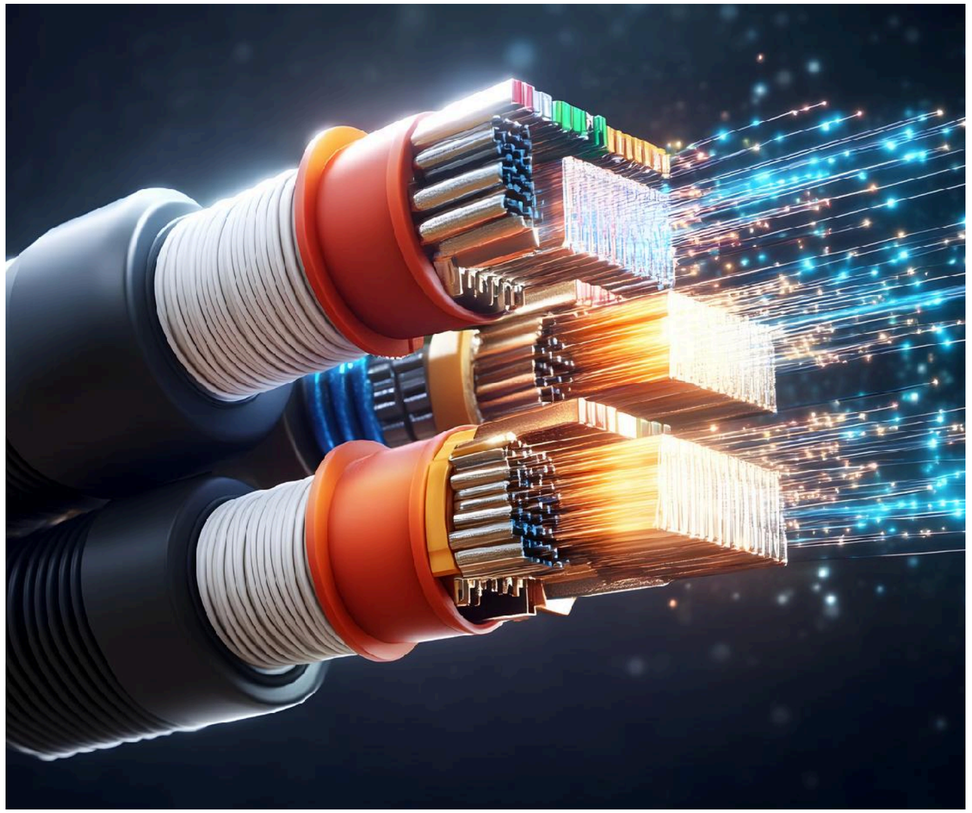
IoT devices communicate through different types of connections to transfer data between sensors, controllers, and the cloud. These connections can be divided into two main categories:

Wired | **Wireless**

DOI:10.48550/arXiv.1805.00831

Types of Internet connectivity: Wired

A wired connection uses physical cables (such as copper or fiber-optic wires) to transmit data and electrical signals between devices.



Type	Description
Ethernet (LAN)	Standard cable-based network connection used in industrial or office environments.
USB (Universal Serial Bus)	Connects microcontrollers to sensors or computers for programming or data transfer.
Serial Communication (UART, SPI, I²C)	Short-distance wired protocols for communication between sensors, MCUs, and modules.

Advantages:

- Stable and reliable communication.
- Higher data transfer rate (speed).
- Less interference from environmental noise.
- More secure as data travels through physical cables.

Disadvantages:

- Limited mobility due to cable connection.
- Higher installation cost and maintenance effort.
- Difficult to expand in large-scale IoT networks.

Types of Internet Connectivity: Wireless

A wireless connection uses radio waves, infrared, or cellular networks to transmit data between IoT devices without physical cables.

Technology	Range	Example IoT Application
Wi-Fi	Short to medium	Smart homes, appliances
Bluetooth / BLE	Short	Wearables, health devices
Zigbee / Z-Wave	Medium	Smart lighting, building automation
LoRa / LoRaWAN	Long	Smart agriculture, environmental monitoring
Cellular (3G, 4G, 5G)	Very long	Smart cities, vehicle tracking
NFC / RFID	Very short	Access control, asset tracking

Advantages:

- Easy installation and flexible deployment.
- Suitable for remote or mobile devices.
- Scalable for large IoT networks.
- Supports cloud-based communication.

Disadvantages:

- Susceptible to interference (walls, signals, weather).
- Higher power consumption (especially Wi-Fi and cellular).
- May require encryption for data security.

Protocol Stack in Internet of Things

A protocol stack is a collection (set) of multiple network protocols that work together in layers to make full communication possible.

Each layer in the stack performs a different role, and data moves from top to bottom (when sending) or bottom to top (when receiving).

Example: IoT Protocol Stack (based on OSI model)	Function
Application Layer HTTP, XMPP, WebSockets, MQTT, CoAP, AMQP, DDS	Communication between applications and cloud servers
Transport Layer TCP, UDP	Controls how data is delivered (connection, reliability)
Network Layer IPv4, IPv6, 6LoWPAN	Handles routing and addressing of data packets
Link/Physical Layer Wi-Fi, Zigbee, Bluetooth, NB-IoT, LoRaWan, NFC, Z-Wave, Sigfox, Celullar	Physical data transmission (wired or wireless)

Types of IoT Protocol

IoT Network Protocol

Protocol	Function	Key Features
Wi-Fi (IEEE 802.11)	Wireless broadband connection	High speed, medium range
Bluetooth & BLE (Bluetooth Low Energy)	Short-range communication	Low power, ideal for wearables
ZigBee (IEEE 802.15.4)	Mesh network for automation	Low power, short range
Z-Wave	Home automation	Up to 100 m range, sub-GHz band
LoRa & LoRaWAN	Long-range, low-power network (LPWAN)	Up to 15 km range, low energy
NFC (Near Field Communication)	Very short-range connection	Less than 10 cm, instant pairing
Cellular (2G–5G, NB-IoT, LTE-M)	Global connectivity through cellular towers	Wide coverage, carrier-dependent
Ethernet	Wired stable connection	Suitable for industrial IoT

IoT Data Protocol

Protocol	Function	Advantages
MQTT (Message Queuing Telemetry Transport)	Publish/subscribe communication	Lightweight, reliable on unstable networks
CoAP (Constrained Application Protocol)	Client-server communication for constrained devices	Lightweight, UDP-based
AMQP (Advanced Message Queuing Protocol)	Enterprise-level messaging	Secure, supports message queues
XMPP (Extensible Messaging and Presence Protocol)	Real-time communication	Can be used for chat or sensor data
M2M / LwM2M (Lightweight Machine-to-Machine)	Device management and data exchange	Standardized by OMA, used in industrial IoT
DDS (Data Distribution Service)	Real-time publish/subscribe system	Used in robotics and mission-critical systems

Network Protocol in IoT

A network protocol is a set of communication rules and standards that define how data is transmitted, received, and processed between IoT devices over a network.

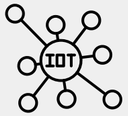
In IoT systems, network protocols allow sensors, actuators, and cloud servers to communicate efficiently and securely, ensuring data from the physical world can be collected, analyzed, and acted upon.

Purpose of IoT Network Protocols:

To establish communication between connected devices.

To format and transfer data correctly between IoT nodes.

To ensure reliability, scalability, and security of IoT communication.



To support different ranges and power requirements (short-range, long-range, low-power).

Network Protocol in IoT

Main Function of IoT Network Protocols:

Define data format – how messages are packaged and sent

Control communication – who sends first, who receives.

Ensure security – by using encryption and authentication.

Enable connectivity – allow wired or wireless connections between devices.

Network Protocol in IoT

IoT Devices

- Energy
- Agriculture
- Smart City
- Healthcare
- Industry
- Energy

Wireless/ Wired

LoRa / ZigBee / Bluetooth /Z-Waze

IoT Gateway

Wireless/ Wired / Celullar/ Ethernet

MQTT / HTTP / HTTPS

The Cloud

Internet

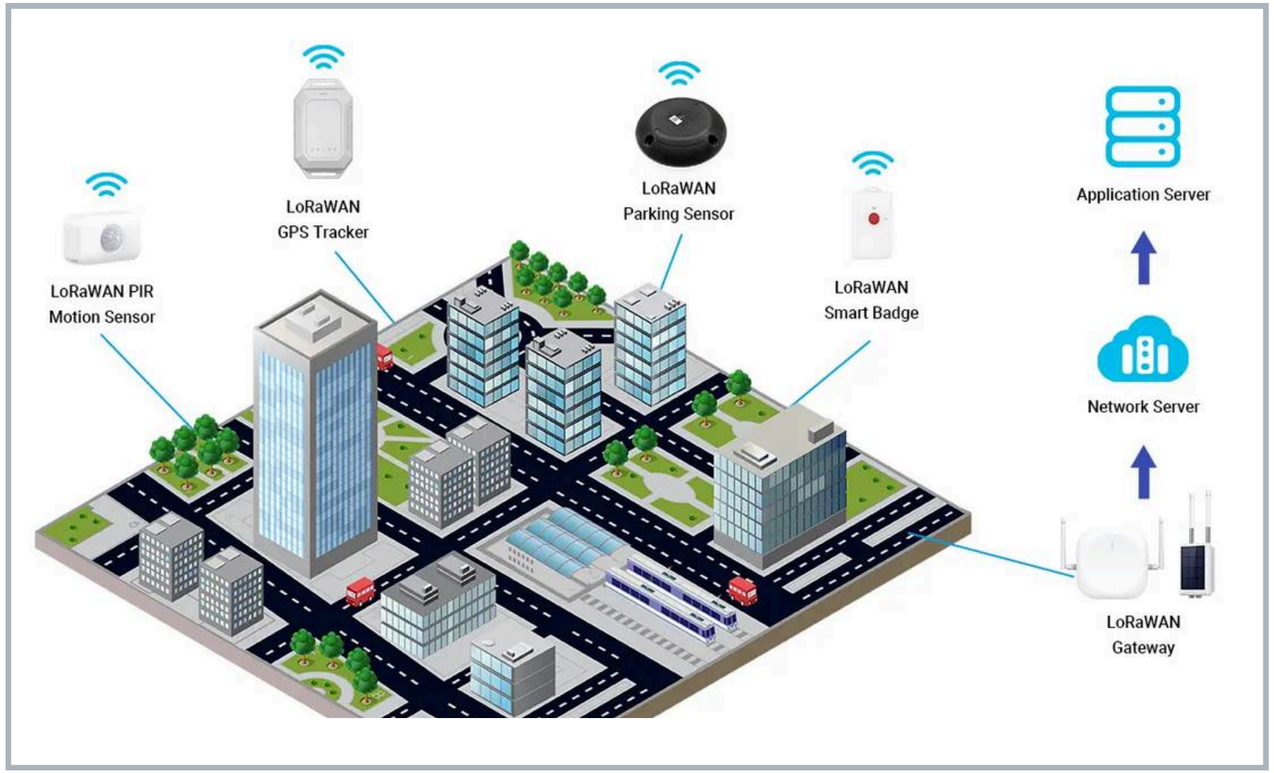
HTTPS / WebSocket

Human Interaction

→ Sends data wirelessly over short or long distance → Uploads sensor data to the Internet or cloud platform → Displays data securely to users via dashboards or apps

Network Protocol in IoT: LoRa and LoRaWAN

LoRa (Long Range) is a wireless modulation technique that enables long-distance, low-power communication between IoT devices. It operates at the physical layer (PHY) and defines how radio signals are transmitted. LoRaWAN (Long Range Wide Area Network) is the network protocol that manages how LoRa devices communicate with gateways and servers.



Feature	Description
Frequency Band	Sub-GHz ISM bands (e.g., 433 MHz, 868 MHz in Europe, 915 MHz in the US, 923 MHz in Malaysia).
Range	2–5 km (urban), up to 15–20 km (rural/open area).
Data Rate	0.3 kbps – 50 kbps (depends on spreading factor).
Power Usage	Very low (ideal for battery-powered sensors).
Topology	Star-of-stars (devices → gateway → network server → application).
Security	End-to-end AES-128 encryption (Network & Application keys).

Network Protocol in IoT: LoRa and LoRaWAN

Example Use

- Smart agriculture sensors sending soil data to a central LoRa gateway.

How its Works in IoT

- **End Devices (Nodes)**
  - Collect data (e.g., temperature, humidity, air quality).
  - Transmit via LoRa radio to the nearest gateway.
- **Gateway**
  - Acts like a bridge between LoRa devices and the internet.
  - Forwards data packets to the network server through Ethernet, Wi-Fi, or cellular.
- **Network Server**
  - Manages devices, security keys, and routing.
  - Removes duplicate packets and handles adaptive data rates.
- **Application Server**
  - Stores and visualizes the data.
  - Allows remote monitoring and control via dashboards.

Advantages

- Long range – Up to 15 km or more in rural areas.
- Low power – Battery life can last 5–10 years.
- Supports many nodes – Thousands of devices per gateway.
- Secure communication with end-to-end encryption.
- Ideal for remote or rural IoT systems (e.g., farms, forests).

Disadvantages

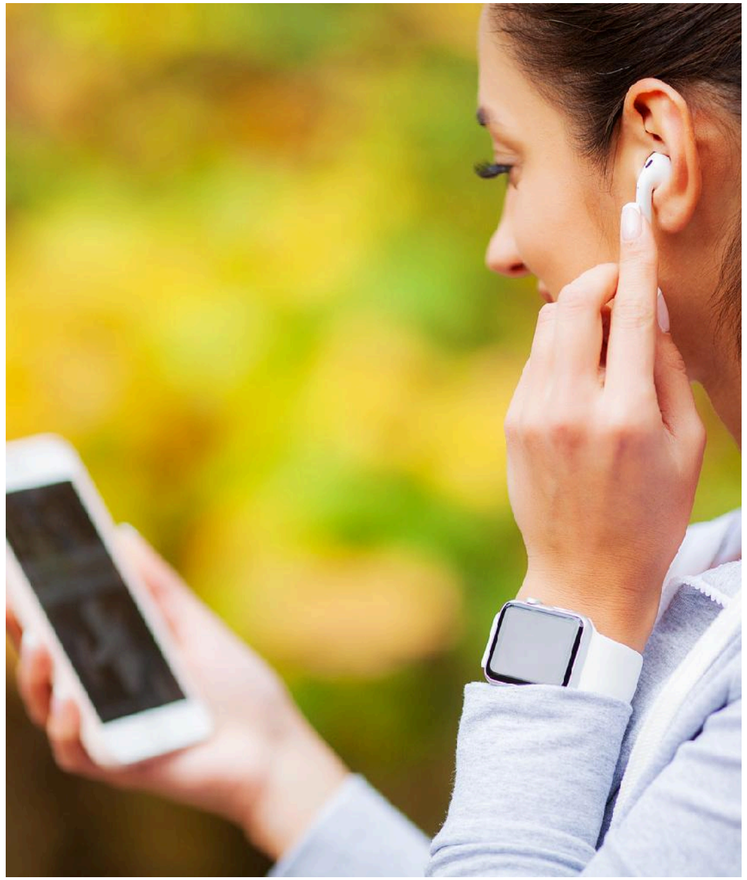
- Low data rate, Not suitable for video or large data.
- Limited downlink – Designed mainly for uplink
- Requires gateway infrastructure.
- Latency – Not suitable for real-time applications.

Network Protocol in IoT: Bluetooth and BLE

Bluetooth is a short-range wireless communication protocol that allows electronic devices to exchange data over short distances (typically 10–100 meters). BLE (Bluetooth Low Energy) is an advanced version of Bluetooth designed specifically for low power consumption, making it ideal for IoT devices and wearables that require long battery life.

Feature	Bluetooth	BLE
Frequency Band	2.4 GHz ISM band	2.4 GHz ISM band
Data Rate	Up to 3 Mbps	Up to 1 Mbps
Range	10–30 meters (up to 100 m with BLE 5.0)	10–100 meters
Power Consumption	High	Very Low (up to 10x less)
Connection Type	Continuous	Event-based / Sleep mode
Topology	Point-to-point	Star, broadcast, or mesh (BLE 5)
Use Case	Audio streaming, file transfer	IoT sensors, wearables, beacons

Network Protocol in IoT: Bluetooth and BLE



How its Works in IoT

- IoT Device (Peripheral) – A sensor or wearable with a Bluetooth/BLE module.
- Central Device – A smartphone, gateway, or computer that collects data.
- Connection Process:
  - Device advertises its presence.
  - Central device scans and connects.
  - Data is exchanged in small packets.
  - BLE device returns to sleep to save power.

Example Use

- Smart home devices (cameras, smart plugs, sensors) using Wi-Fi to send data to cloud dashboards.

Advantages

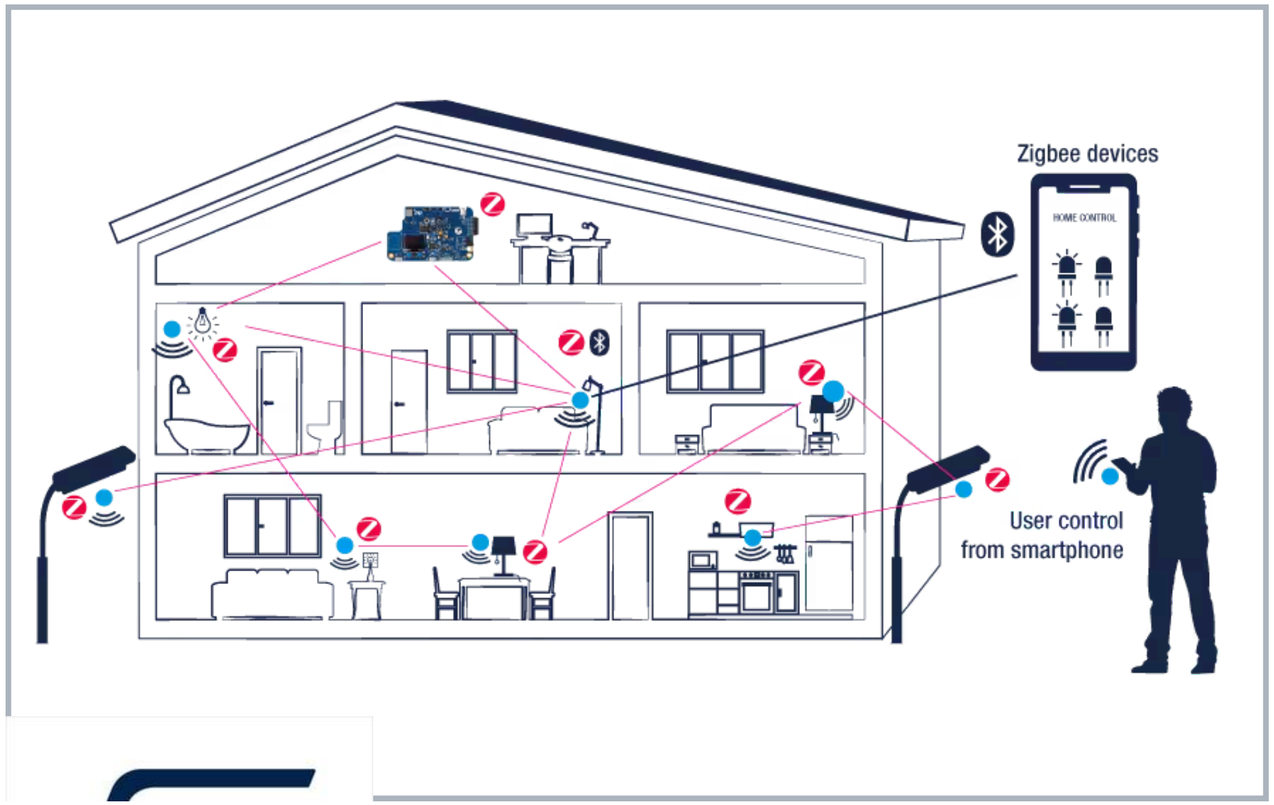
- Very low power → suitable for battery-operated devices (can last years).
- Simple pairing and easy integration with mobile devices.
- Low cost and widely available.
- Supports broadcast mode (advertisements or beaconing).

Disadvantages

- Short range – not ideal for large area coverage.
- Low data rate – not suitable for video or large data transfer.
- Interference – operates in 2.4 GHz, same as Wi-Fi and ZigBee.
- Limited number of connections per node (in BLE).

Network Protocol in IoT: Zigbee

ZigBee is a low-power, wireless communication protocol based on the IEEE 802.15.4 standard.
It is designed for short-range, low-data-rate, and low-cost communication between IoT devices.



Feature	Description
Frequency Band	Operates at 2.4 GHz (globally), 868 MHz (Europe), and 915 MHz (USA).
Data Rate	Up to 250 kbps – suitable for small data packets such as sensor readings.
Range	Typically 10–100 meters indoors (can be extended using mesh networking).
Power Consumption	Very low; suitable for battery-powered IoT devices.
Network Type	Supports Mesh, Star, and Tree topologies.
Number of Devices	Supports up to 65,000 nodes in one network.
Security	AES-128 encryption for secure communication.

Network Protocol in IoT: Zigbee

How its Works in IoT

- Sensors collect data (e.g., temperature, light).
- End devices send data to a router or coordinator using ZigBee.
- The coordinator connects to a gateway that sends data to the cloud platform (e.g., AWS IoT or ThingsBoard).
- The data can then be visualized and controlled remotely through a web dashboard or mobile app.

Example Use

- Smart lighting systems or building automation.

Advantages

- Low power usage → long battery life.
- Reliable mesh networking (self-healing).
- Supports many devices in one network.
- Cost-effective for large sensor networks.

Disadvantages

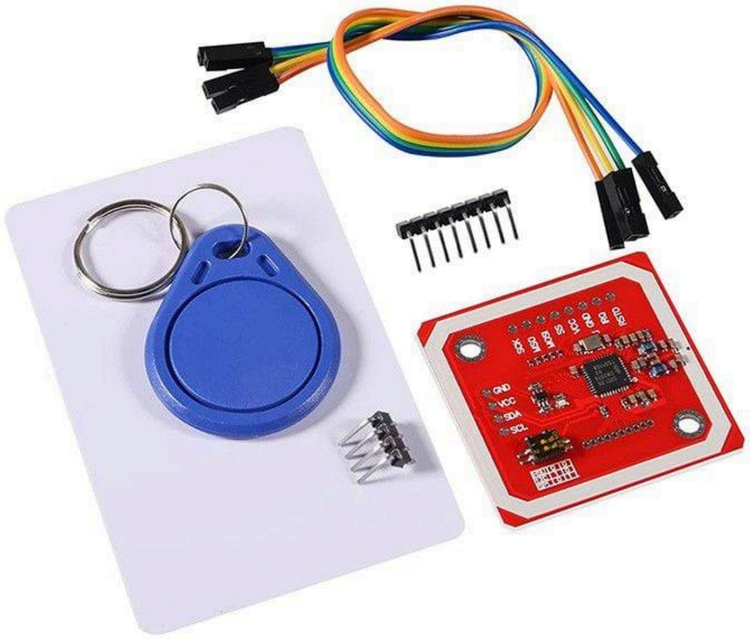
- Low data rate (not suitable for video or large data).
- Shorter range compared to Wi-Fi or LoRa.
- Requires ZigBee-compatible hardware (e.g., XBee modules).

Network Protocol in IoT: NFC

NFC (Near Field Communication) is a short-range wireless communication technology that allows two devices to exchange data when they are placed very close together (typically within 4 cm). It is based on Radio Frequency Identification (RFID) technology and operates at 13.56 MHz. NFC is commonly used for contactless communication, authentication, and data transfer between IoT devices.

Feature	Description
Operating Frequency	13.56 MHz (ISM band)
Communication Range	Up to 4 cm
Data Rate	106 kbps – 424 kbps
Connection Type	Peer-to-peer, reader/writer, or card emulation
Power Consumption	Very low
Security	High – requires close physical proximity
Standard	ISO/IEC 18092, ISO/IEC 14443 (same as RFID standards)

Network Protocol in IoT: NFC



How its Works in IoT

- NFC uses magnetic field induction to enable communication between two devices – one acts as an initiator (active) and the other as a target (passive).

- **NFC Communication Modes:**
  - Reader/Writer Mode – Reads data from or writes data to an NFC tag. Example: Smartphone scanning an NFC tag to read temperature data.
  - **Peer-to-Peer Mode** – Two NFC-enabled devices exchange information directly. Example: Sharing data between two smartphones.
  - **Card Emulation Mode** – NFC device acts as a contactless card. Example: Mobile payment (Google Pay, Apple Pay).

Advantages

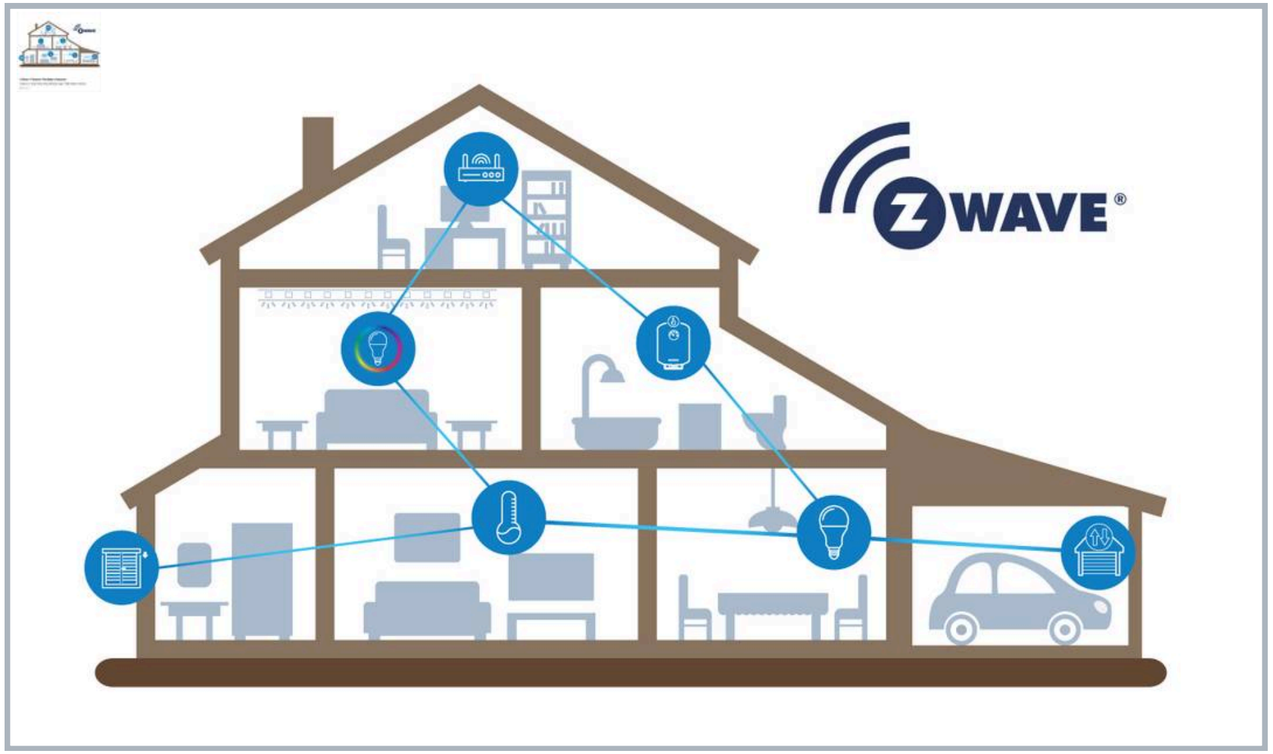
- Easy to use – simply tap or bring close to connect.
- Low power – ideal for passive IoT tags or identification.
- Secure – close-range communication reduces interception risk.
- No pairing required – instant connection.
- Supports contactless payments and access control.

Disadvantages

- Very short range (a few centimeters).
- Low data rate – not suitable for large data transfer.
- Requires physical proximity.
- Limited to short interactions (e.g., tap-and-go).

Network Protocol in IoT: Z-Wave

Z-Wave is a wireless communication protocol specifically designed for home automation and smart IoT systems. It enables devices such as sensors, lights, locks, and thermostats to communicate with each other using low-power radio frequency. It was developed by Zensys (Denmark) and is now managed by the Z-Wave Alliance.



Feature	Description
Frequency Band	Operates on sub-GHz band (around 868 MHz in Europe, 908 MHz in the US) – avoids interference with Wi-Fi (2.4 GHz).
Data Rate	Up to 100 kbps – sufficient for sensor and control data.
Range	30 – 100 meters indoors (longer than ZigBee).
Power Usage	Very low; ideal for battery-powered smart devices.
Network Type	Mesh network – devices relay messages to extend range.
Max Nodes	Supports up to 232 devices per network (newer Z-Wave Long Range supports >4000).
Security	Advanced AES-128 encryption (Z-Wave S2 security framework).

Network Protocol in IoT: Z-Wave

How its Works in IoT

- Controller hub sends commands (e.g., turn on light).
- The signal travels through repeater nodes to reach the target device.
- Devices can send status updates or sensor data back to the controller.
- The controller connects to a gateway or cloud service, allowing remote monitoring via mobile app or dashboard.

Example Use

- Smart lighting, thermostats, and security systems.

Advantages

- Operates on sub-GHz → less interference than Wi-Fi or ZigBee.
- Reliable mesh network for medium
- Strong interoperability – all certified devices work together.
- Low power, stable connection for smart home automation.

Disadvantages

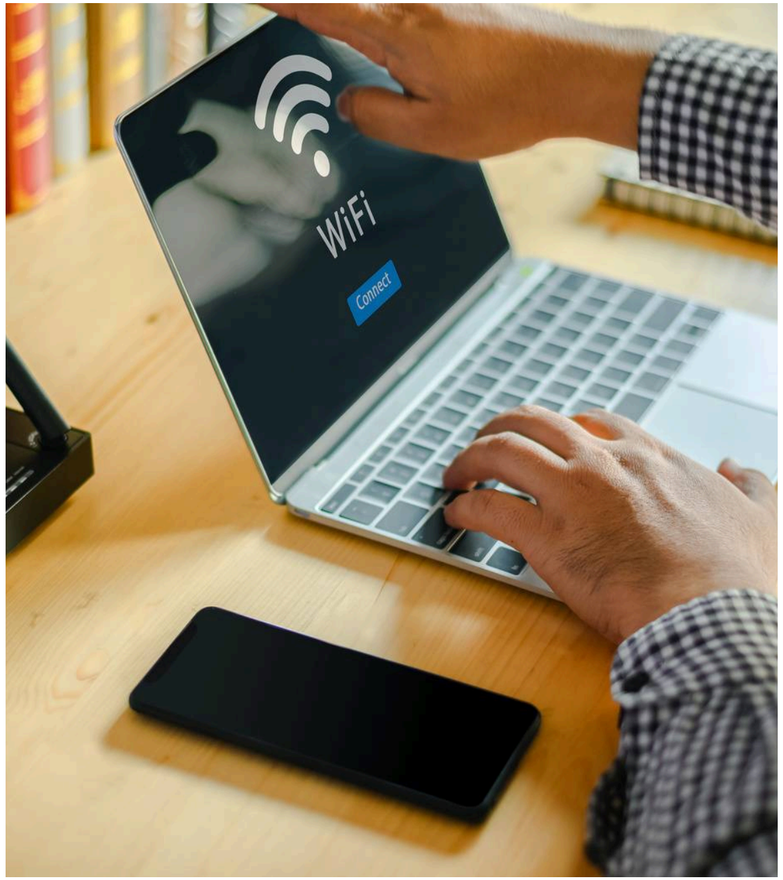
- Limited data rate (not suitable for high-bandwidth data).
- Fewer devices per network (compared to ZigBee).
- Requires Z-Wave-certified hardware (closed ecosystem).

Network Protocol in IoT: Wi-Fi

Wi-Fi (Wireless Fidelity) is a wireless local area network (WLAN) technology based on the IEEE 802.11 standards. It allows IoT devices to connect to the internet or local network using radio waves, typically in the 2.4 GHz and 5 GHz frequency bands. Wi-Fi is one of the most common IoT connectivity methods, especially for devices that require high data rates and direct cloud connectivity.

Feature	Description
Standard	IEEE 802.11 a/b/g/n/ac/ax (latest Wi-Fi 6/6E)
Frequency Band	2.4 GHz, 5 GHz, and 6 GHz (Wi-Fi 6E)
Data Rate	From 11 Mbps (802.11b) up to 9.6 Gbps (Wi-Fi 6)
Range	20–50 meters indoors; 100 meters outdoors
Power Usage	Moderate to high (higher than ZigBee or LoRa)
Network Type	Star topology (devices connect to a router or access point)
Security	WPA2, WPA3 encryption standards

Network Protocol in IoT: Wi-Fi



How Wi-Fi Works in IoT

- IoT devices (e.g., ESP8266, ESP32, Raspberry Pi) connect to a Wi-Fi router or hotspot.
- The router provides access to the local network and internet.
- Data from sensors or devices is sent to cloud platforms (e.g., AWS IoT, Google Cloud IoT, ThingsBoard).
- Users can monitor and control devices remotely via mobile apps or dashboards.

Example Use

- Smart home devices (cameras, smart plugs, sensors) using Wi-Fi to send data to cloud dashboards.

Advantages

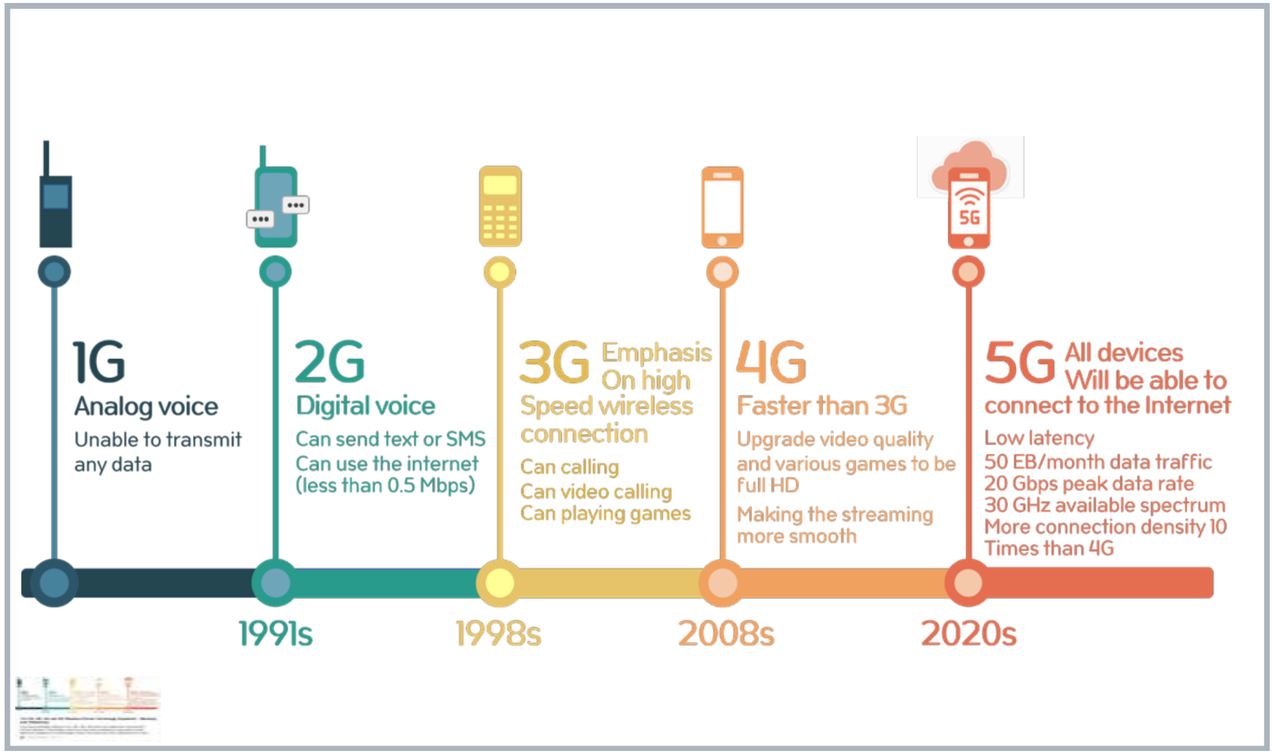
- High data rate – suitable for video streaming, camera sensors, and large data transfer.
- Widespread availability – almost all homes, schools, and offices have Wi-Fi.
- Direct internet connectivity – no need for a gateway in most cases.
- Supports multiple devices simultaneously.

Disadvantages

- High power consumption – not suitable for battery-only devices.
- Limited range compared to LoRa or cellular.
- Interference possible at 2.4 GHz due to many devices using the same band.
- Not ideal for large-scale outdoor deployments.

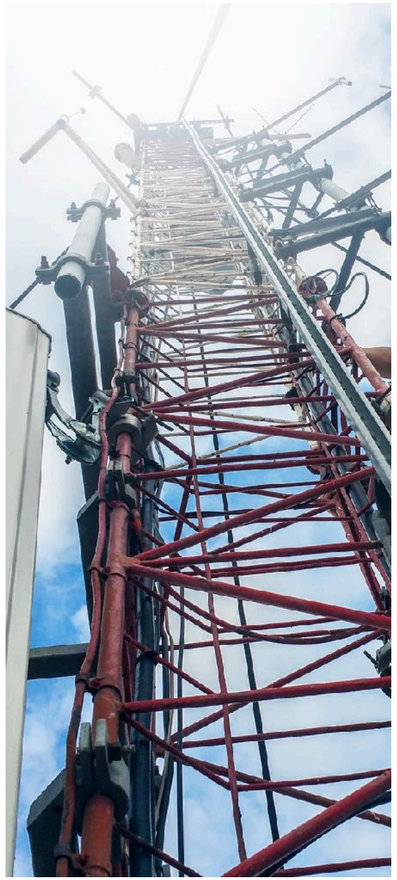
Network Protocol in IoT: Cellular (3G, 4G, 5G, NB-IoT)

Cellular communication refers to wireless network technology that uses mobile operator infrastructure (base stations, towers) to transmit data over long distances. In IoT, cellular networks enable devices to connect directly to the internet via SIM cards or embedded modules, without needing Wi-Fi or gateways.



Feature	Description	IoT Usage
3G (UMTS)	Moderate speed (up to 2 Mbps), supports voice + data	Early IoT (vehicle tracking, telemetry)
4G (LTE)	High speed (up to 100 Mbps), low latency	Smart city, video surveillance, connected cars
5G	Ultra-fast (Gbps), ultra-low latency (<1 ms), supports millions of devices	Industrial IoT, autonomous vehicles, smart factories
NB-IoT (Narrowband IoT)	Low power, long range, low cost	Smart meters, agriculture, environmental monitoring

Network Protocol in IoT: Cellular (3G, 4G, 5G, NB-IoT)



Types of Cellular IoT Technologies

3G (UMTS / HSPA)

- Introduced mobile internet to IoT.
- Moderate data speed, high latency.
- Suitable for basic remote monitoring and vehicle tracking.

4G LTE (Long Term Evolution)

- Provides broadband-like speed and reliable connectivity.
- Enables real-time video, autonomous drones, and smart traffic systems.
- Commonly used modules: SIM7600, EC25, Quectel LTE module.

5G (Fifth Generation)

- Designed for massive IoT and ultra-reliable low-latency communication (URLLC).
- Key advantages:
  - Speeds up to 10 Gbps
  - Latency < 1 ms
  - Connects up to 1 million devices/km²

NB-IoT (Narrowband IoT)

- A low-power wide-area network technology using cellular infrastructure.
- Optimized for IoT devices that send small data packets infrequently.
- Features:
  - Range up to 10 km (urban), 35 km (rural)
  - Battery life up to 10 years
  - Deep indoor penetration (smart buildings)
- Ideal for smart metering, agriculture sensors, water monitoring, etc.

Advantages

- High data rate – suitable for video streaming, camera sensors, and large data transfer.
- Widespread availability – almost all homes, schools, and offices have Wi-Fi.
- Direct internet connectivity – no need for a gateway in most cases.
- Supports multiple devices simultaneously.

Disadvantages

- High power consumption – not suitable for battery-only devices.
- Limited range compared to LoRa or cellular.
- Interference possible at 2.4 GHz due to many devices using the same band.
- Not ideal for large-scale outdoor deployments.

Comparison of IoT Communication Technologies

Feature	LoRa / LoRaWAN	Bluetooth	BLE (Bluetooth Low Energy)	ZigBee	NFC	Z-Wave	Wi-Fi	Cellular (3G / 4G / 5G / NB-IoT)
Operating Frequency	Sub-GHz (433 / 868 / 915 MHz)	2.4 GHz	2.4 GHz	2.4 GHz	13.56 MHz	868 / 915 MHz	2.4 GHz / 5 GHz / 6 GHz	Licensed bands (700 MHz – 3.5 GHz)
Range	2 – 20 km (rural)	10 – 30 m	10 – 100 m	10 – 100 m	≤ 4 cm	30 – 100 m	20 – 100 m	Up to 35 km (NB-IoT) / nationwide
Data Rate	0.3 – 50 kbps	Up to 3 Mbps	Up to 1 Mbps	Up to 250 kbps	106 – 424 kbps	Up to 100 kbps	Up to 9.6 Gbps (Wi-Fi 6)	Up to Gbps (5G) / kbps (NB-IoT)
Power Consumption	Very Low	Moderate	Very Low	Very Low	Very Low	Low	High	Moderate – High (Low for NB-IoT)
Topology	Star-of-stars	Point-to-point	Star / Mesh (BLE 5)	Mesh / Star	Point-to-point	Mesh	Star	Star (via cell tower)
Internet Access	Via gateway	Via smartphone	Via smartphone / hub	Via hub / gateway	Via reader / phone	Via controller / hub	Direct (router → internet)	Direct (SIM / mobile network)
Bandwidth	Narrow	Medium	Narrow	Narrow	Very Narrow	Narrow	Wide	Wide – very wide
Security	AES-128 Encryption	Encryption + Pairing	AES-128 Encryption	AES-128 Encryption	Short-range security	AES-128 Encryption	WPA2 / WPA3	Operator-level encryption
Deployment Cost	Low	Low	Low	Low	Very Low	Moderate	Moderate	High (data plan & module)
Latency	Medium	Low	Low	Medium	Very Low	Medium	Very Low	Very Low (5G) / High (3G)
Scalability	High (> 1000 nodes per gateway)	Low	Moderate	High (up to 65 000 nodes)	Very Low	Medium (≈ 232 devices)	Moderate	Very High (massive IoT)
Mobility Support	Yes (limited)	Limited	Limited	Limited	No	Limited	Yes	Yes (high)
Typical Range Category	Long-range LPWAN	Short-range	Short-range	Short-range	Contact-range	Medium-range	Medium-range	Very long range
Best Use Cases	Smart agriculture, environmental	Audio, file transfer, wearables	Healthcare, beacons, sensors	Smart home, industrial sensors	Access control, payments	Home automation, security	Smart home, CCTV, IoT server	Smart city, transport, industrial IoT
Example Hardware	RFM95, ESP32-LoRa, RAK gateway	HC-05, HC-06	HM-10, ESP32	XBee (ZigBee), CC2530	PN532, NTAG tags	Aeotec, Fibaro Hub	ESP8266, ESP32, Router	SIM7600, SIM7020 (NB-IoT), Quectel 5G
Power Source	Battery / Solar	Battery	Battery	Battery / Mains	Passive (Tag)	Battery / Mains	Mains	Battery or Mains
Suitability for IoT	Excellent for remote, low-power apps	Limited (short range)	Excellent for wearable / low energy apps	Excellent for mesh home networks	Great for identification / authentication	Excellent for smart home control	Excellent for high-bandwidth IoT	Excellent for wide-area mobile IoT

Data Protocol in Internet of Things

In the Internet of Things (IoT), billions of devices communicate and share information across networks. To make this communication reliable, secure, and efficient, IoT systems rely on data protocols, standardized rules that define how data is formatted, transmitted, and received between devices and servers.

A data protocol ensures that all devices regardless of manufacturer or hardware can understand and exchange information consistently.

A data protocol in IoT is a set of communication rules and message formats that enable connected devices, gateways, and cloud platforms to exchange data efficiently and securely.

It specifies:

How data packets are structured (format)

How errors are detected and handled (reliability)

How data is sent and acknowledged (communication process)

How devices connect, authenticate, and manage sessions (security)

Types of Data Protocol in IoT

Protocol	Description	Best Use
MQTT (Message Queuing Telemetry Transport)	Lightweight publish-subscribe protocol designed for low-bandwidth, high-latency networks.	Remote sensing, IoT cloud applications
CoAP (Constrained Application Protocol)	Works like HTTP but optimized for low-power devices; uses UDP instead of TCP.	Smart homes, industrial IoT
AMQP (Advanced Message Queuing Protocol)	Reliable message delivery protocol with strong security and routing features.	Enterprise IoT systems
XMPP (Extensible Messaging and Presence Protocol)	XML-based protocol for real-time messaging and presence detection.	Smart communication systems
M2M and LwM2M (Lightweight Machine-to-	Standard for managing and monitoring IoT devices remotely.	Device management in large IoT networks
HTTP / HTTPS	Web-based communication protocol using TCP; supports encryption (HTTPS).	IoT web APIs, RESTful services

Benefits of Using Data Protocols in IoT

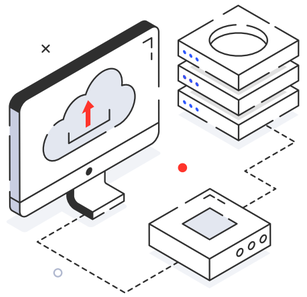
Interoperability:
Enables communication across different devices and platforms.

Efficiency:
Reduces network overhead and power usage.

Reliability:
Ensures data is delivered accurately and on time.

Security:
Protects data integrity and privacy.

Scalability:
Supports large-scale IoT networks.



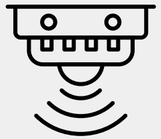
Internet of Things Framework

An IoT Framework is a structured model or architecture that explains how IoT systems are designed, connected, and managed — from data collection at devices to analytics and decision-making in the cloud. It defines layers, functions, and data flow within an IoT ecosystem, ensuring all components (hardware, software, and networks) work together efficiently.

Purpose of the IoT Framework

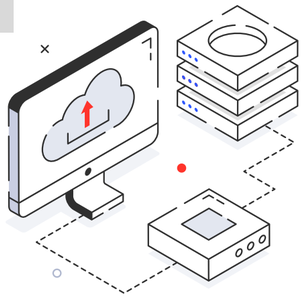
To organize how IoT devices communicate and interact.

To ensure interoperability between different technologies.



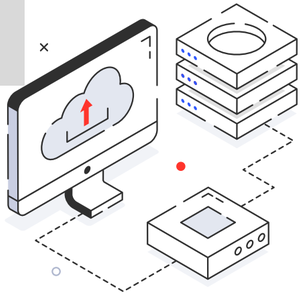
To manage data flow from sensors to cloud applications.

To improve scalability, security, and efficiency in IoT systems.



IoT Framework: Components / Layers

Layer	Function	Example Technology
Business / Decision Layer	Converts analyzed data into useful actions and business insights.	Data analytics, dashboards, AI-based decision systems
IoT Platform / Cloud (Application Layer)	Provides data visualization, control, and management through software or web apps.	AWS IoT Core, Google Cloud IoT, ThingsBoard, Blynk
Middleware Layer (Processing / Gateway Layer)	Acts as the “bridge” – processes, filters, and manages data between devices and applications.	IoT Gateways, Edge Devices, MQTT Broker, CoAP Server
Network Layer (Communication Layer)	Transmits collected data to gateways, servers, or cloud platforms.	Wi-Fi, ZigBee, LoRa, Bluetooth, Cellular (4G/5G), Ethernet
Perception Layer (Device/Sensing Layer)	Detects and collects environmental data (temperature, humidity, light, etc.).	Sensors, RFID, actuators, cameras



Internet of Things Building Block

The IoT building blocks consist of smart things, networks and gateways, middleware, IoT Platform and Cloud. Smart things collect data, networks and gateways transmit it, middleware processes it, and applications present it to users. Together, these components enable seamless connectivity, data management, and intelligent decision-making in IoT systems.

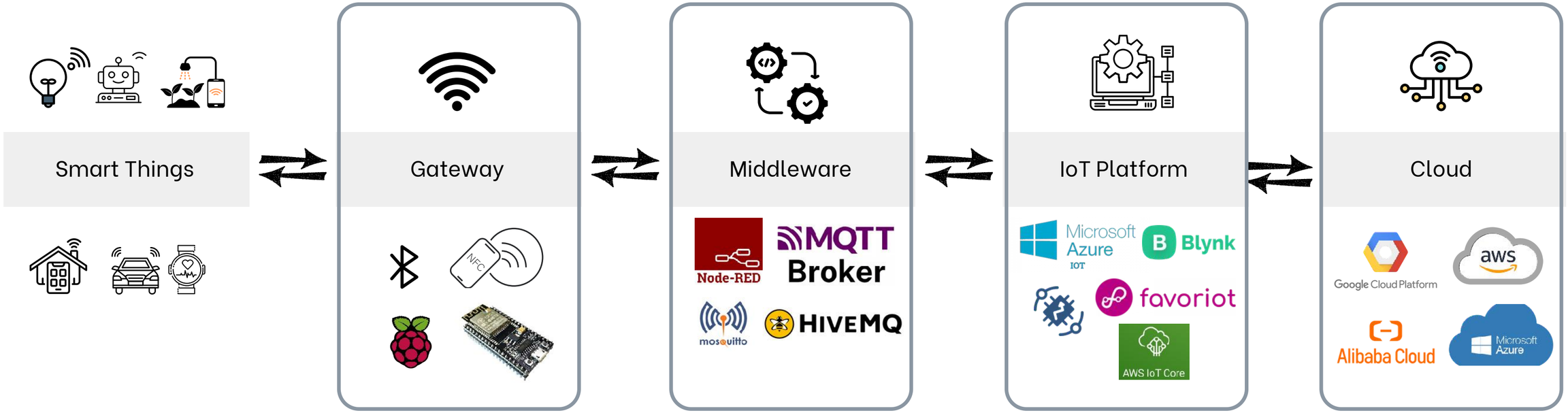
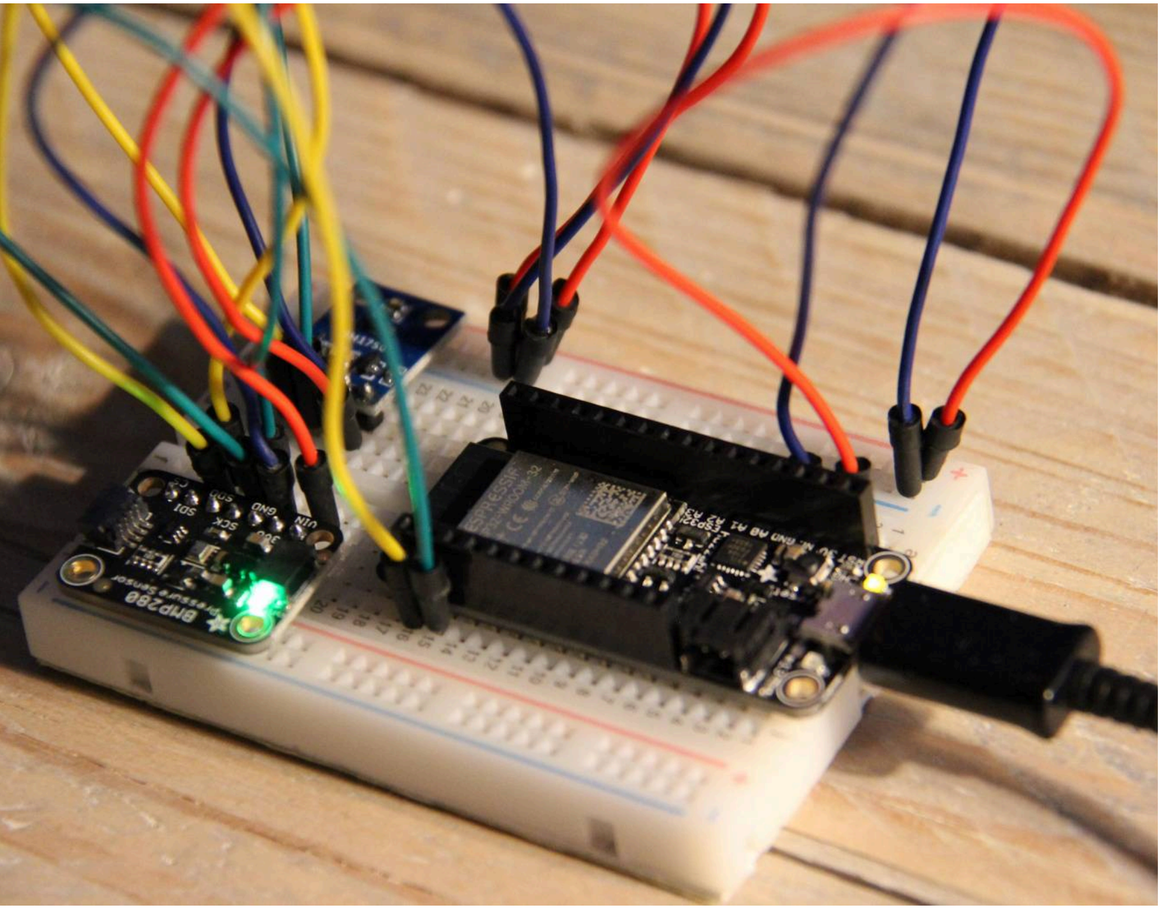


Figure: Diagram of IoT Building Block

IoT Building Block: Gateway



An IoT Gateway is a bridge device that connects sensors or embedded controllers to the internet or cloud platform.

It collects data from local devices, performs initial processing (filtering, aggregation, encryption), and forwards it to the cloud.

Functions

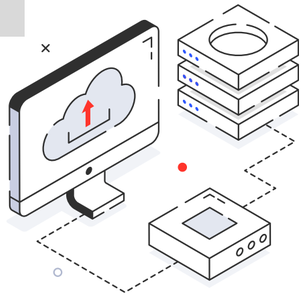
- Acts as a communication bridge between local devices and external networks.
- Converts different communication protocols (e.g., from ZigBee to Wi-Fi, or LoRa to MQTT).
- Performs edge processing – basic analytics near the data source.
- Enhances security by authenticating data before sending to the cloud.

Examples

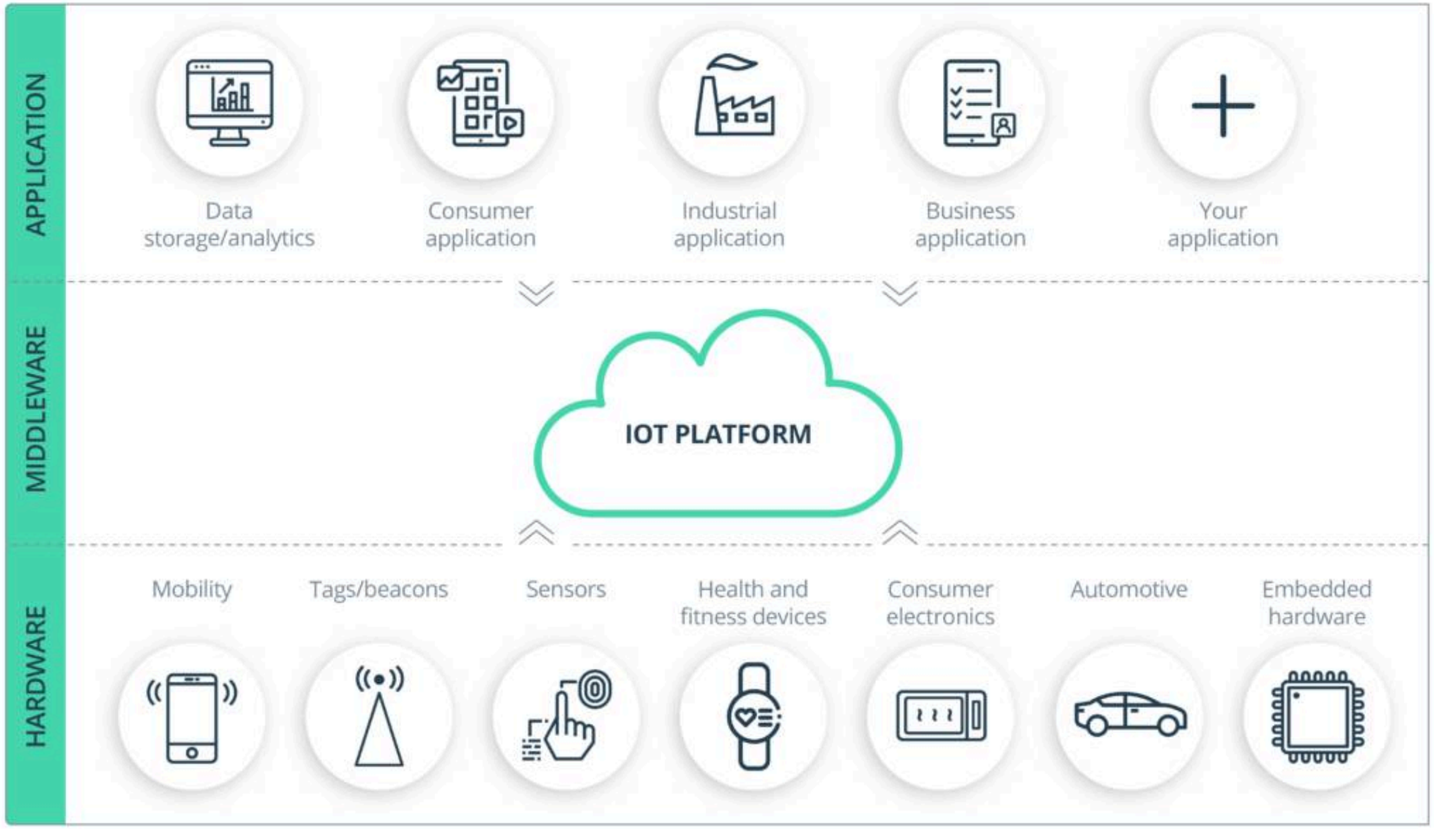
- Raspberry Pi or ESP32 configured as a gateway
- AWS IoT Greengrass (Edge Gateway)
- Cisco IoT Gateway, Dragino LoRa Gateway

Example Use

In a smart farm, multiple LoRa-based sensors send soil data to a LoRa gateway, which then uploads the data to ThingsBoard cloud via Wi-Fi or 4G.



IoT Building Block: Middleware



Middleware is the software layer that manages communication between devices, networks, and applications. It acts as the "brain" of the IoT system, ensuring that data flows smoothly between all components.

Functions

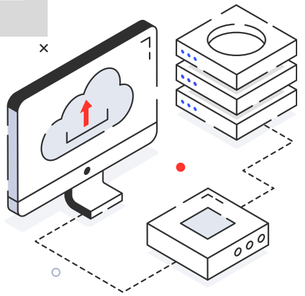
- Handles data collection, filtering, and transformation.
- Provides device management (registration, status, updates).
- Supports data routing and storage.
- Implements messaging protocols like MQTT, CoAP, or AMQP.
- Ensures interoperability between different devices and vendors.

Examples

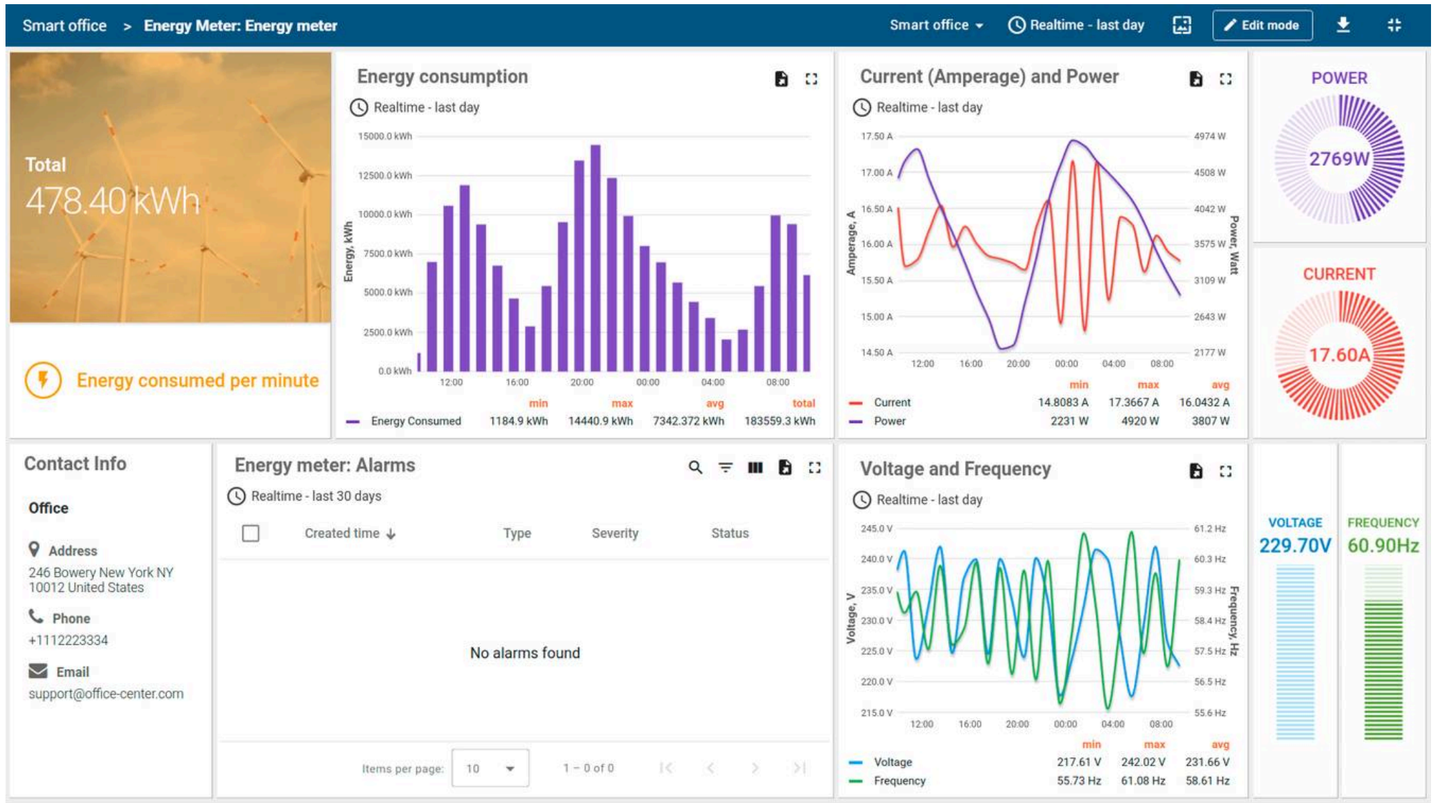
- Node-RED (middleware for data routing and processing)
- MQTT Broker (Mosquitto, HiveMQ)
- CoAP Server
- AWS IoT Core, Google Cloud IoT Core (contain built-in middleware functions)

Example Use

In a smart building, sensor data flows through an MQTT broker (middleware) that filters and directs messages to appropriate applications such as dashboards or alerts.



IoT Building Block: IoT Platform



An IoT Platform is a cloud-based or local system that provides tools for device management, data storage, visualization, and analytics. It is the main interface for users and applications to interact with IoT data.

Functions

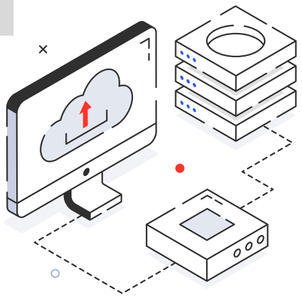
- Stores and manages data received from devices.
- Visualizes sensor readings on dashboards.
- Allows control of devices remotely (e.g., turning on/off an actuator).
- Integrates AI or analytics to generate insights.
- Provides APIs for mobile or web applications.

Examples

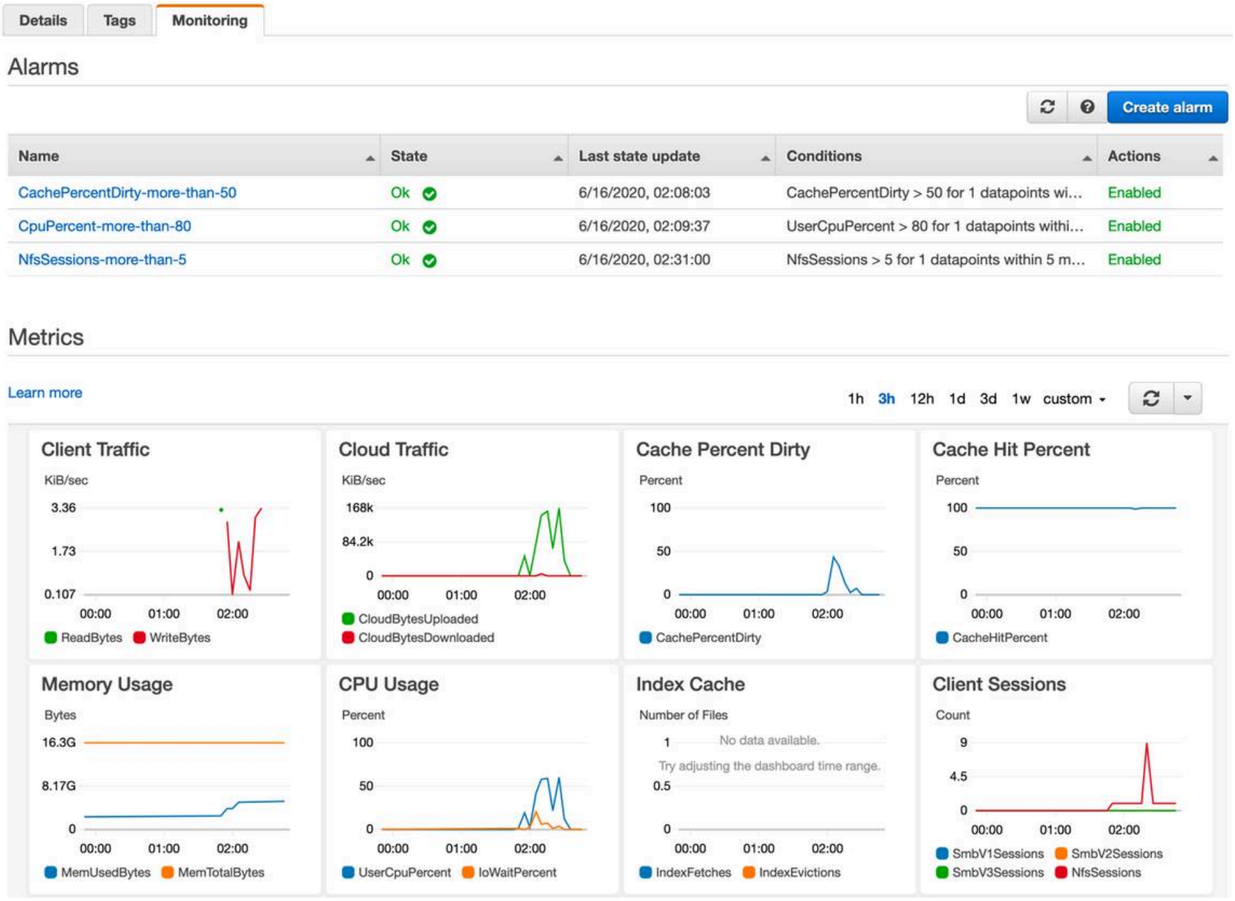
- ThingsBoard, AWS IoT Core, Google Cloud IoT, Microsoft Azure IoT, Blynk, Ubidots

Example Use

A lecturer monitors temperature and humidity data from classroom sensors via ThingsBoard dashboard, receiving alerts when the temperature exceeds 30˚C.



IoT Building Block: Cloud



The Cloud refers to a network of remote servers used to store, process, and analyze IoT data. It provides massive storage capacity, scalability, and accessibility from anywhere via the internet.

Functions

- Stores large amounts of IoT data securely.
- Analyzes real-time and historical data.
- Hosts IoT platforms, machine learning, and AI services.
- Scales automatically with growing numbers of IoT devices.
- Enables remote monitoring and automation through dashboards and mobile apps.

Examples

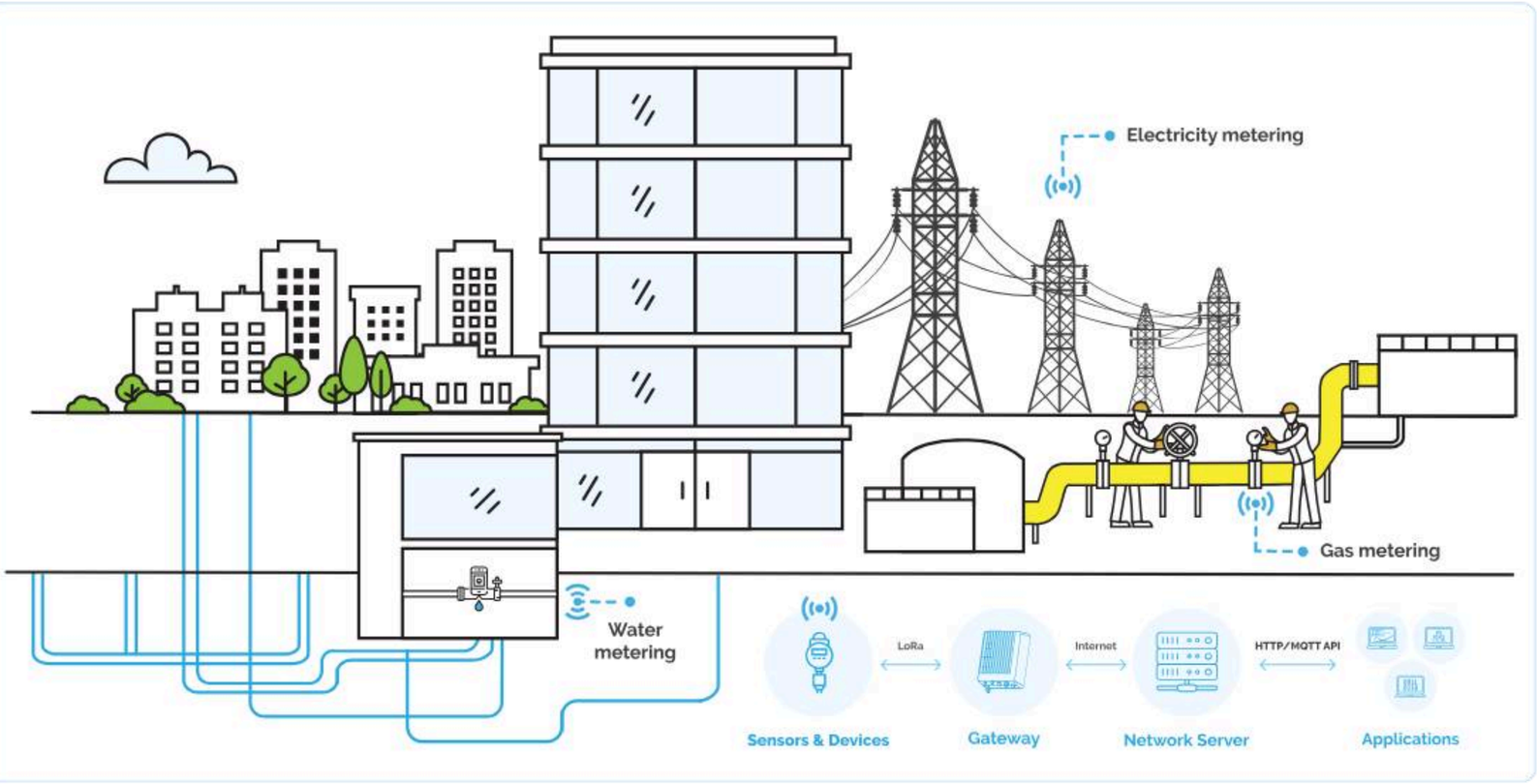
- Amazon Web Services (AWS)
- Microsoft Azure
- Google Cloud Platform (GCP)
- IBM Cloud, Alibaba Cloud

Example Use

Smart energy meters send data to AWS IoT Cloud, where it’s processed and displayed on a dashboard. The system can also trigger automated energy-saving actions.

IoT Application Framework Example: Smart Grid

Class discussion



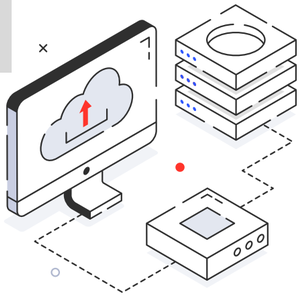
- Observation:
What types of IoT sensors can you identify in this diagram (e.g., water, gas, electricity)?

- Data Flow Understanding:
Explain how data moves from sensors and devices to the applications shown at the right.

- Connectivity:
The diagram mentions LoRa and Internet — why might LoRa be chosen for this IoT system instead of Wi-Fi or Bluetooth?

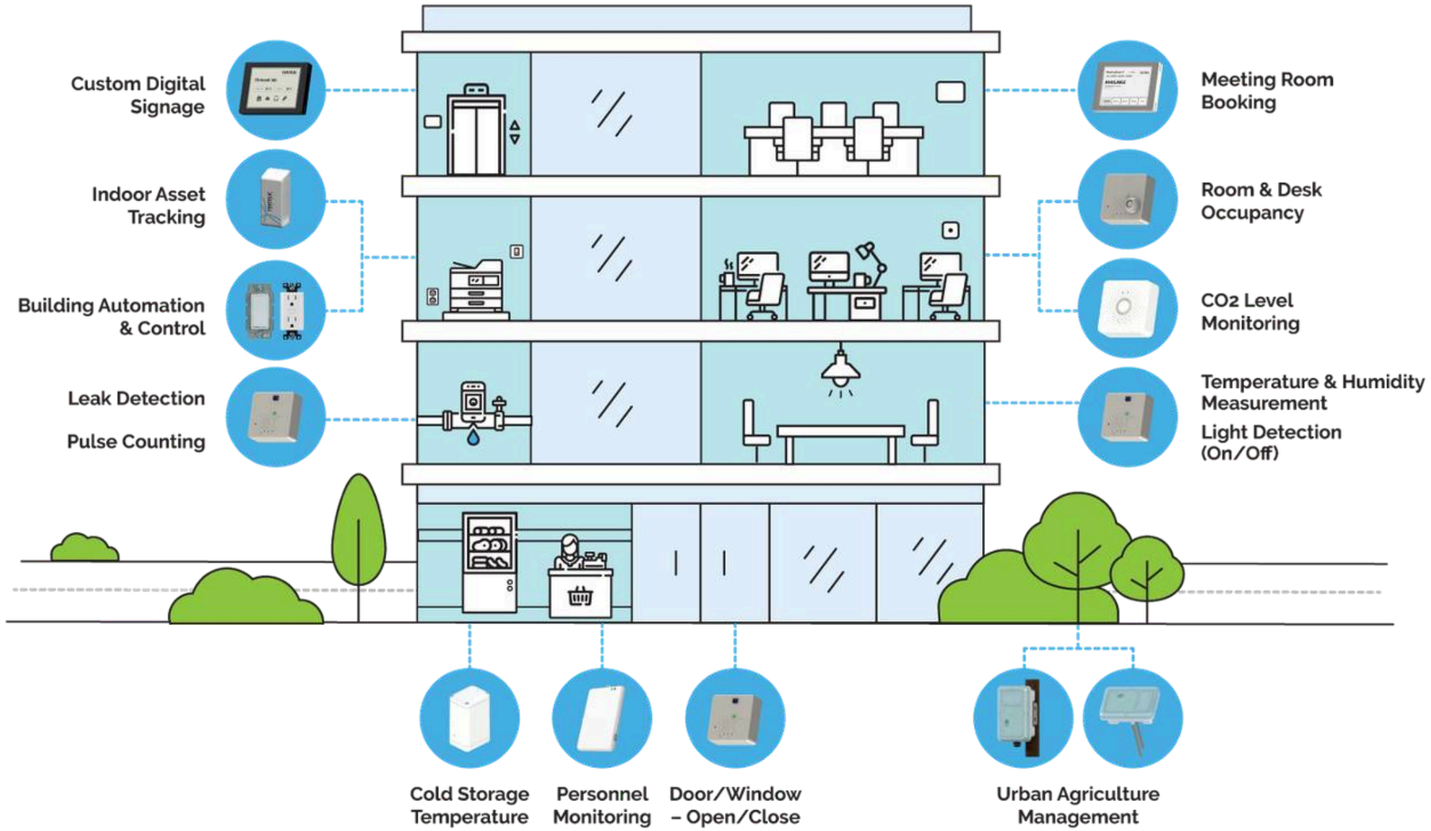
- Practical Application:
How could this IoT system benefit utility companies and consumers in real-time monitoring or billing?

- Extension Question:
If you were to add another smart service (for example, waste management or street lighting), where would it fit in this architecture and why?



IoT Application Framework Example: Building Automation

Class discussion



- Observation:
What types of IoT sensors can you identify in this building (e.g., temperature, CO_2, occupancy, leak detection)?

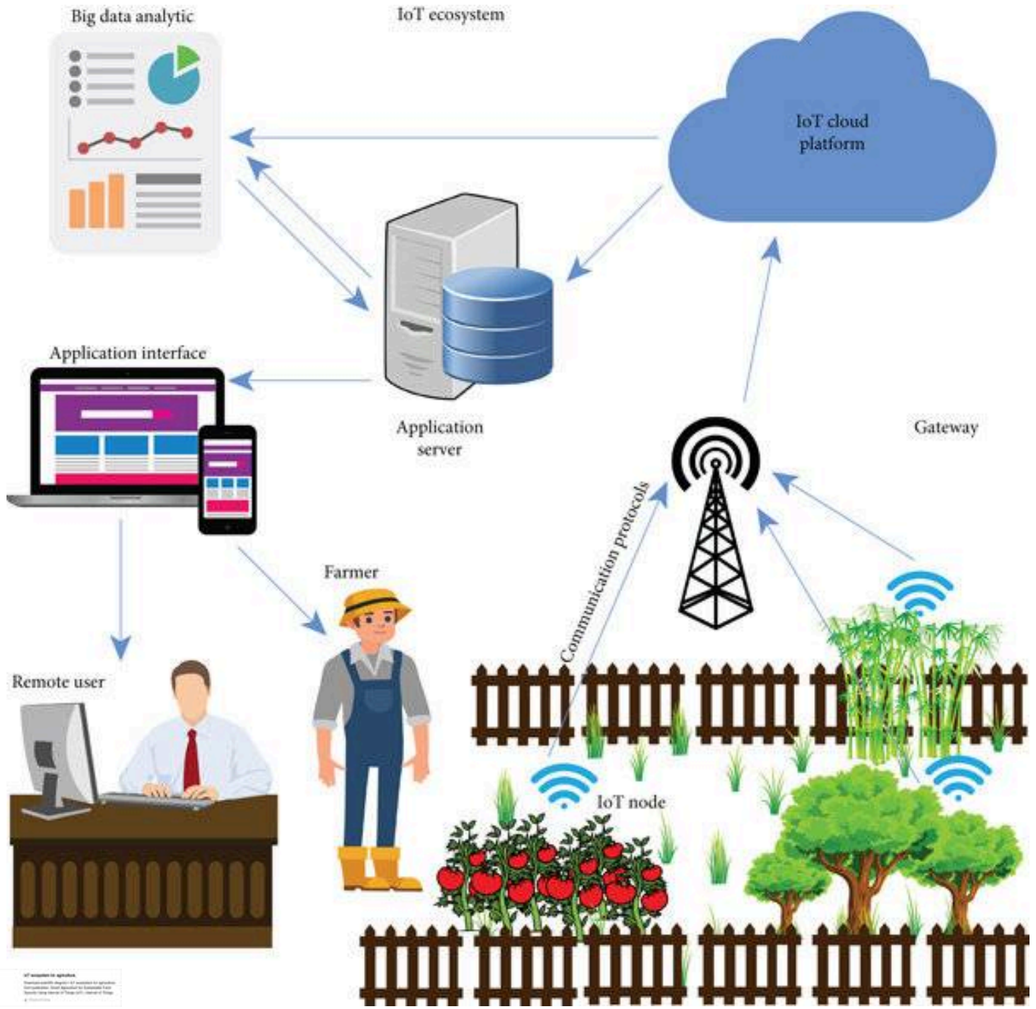
- Data Flow Understanding:
Explain how data from various sensors (like room occupancy or light detection) travels to the cloud or dashboard.

- Connectivity:
What types of communication technologies (e.g., Wi-Fi, ZigBee, Bluetooth) might be used in this smart building, and why?

- Practical Application:
How could building managers use this IoT system to automate daily operations, such as lighting, air conditioning, or meeting room booking?

- Extension Question:
If you were to add another smart feature (e.g., smart parking, air quality monitoring, or elevator maintenance), where would it fit in this architecture and why?

IoT Application Framework Example: Smart Farming



Class discussion

- Observation:

How do IoT nodes help farmers monitor their crops and soil conditions?

- Data Flow Understanding:

Explain how data travels from the sensors in the field (IoT nodes) to the cloud platform and finally to the farmer's device.

- Connectivity:

Which communication technologies (e.g., Wi-Fi, LoRa, Cellular) could be used to connect the farm sensors to the gateway, and why?

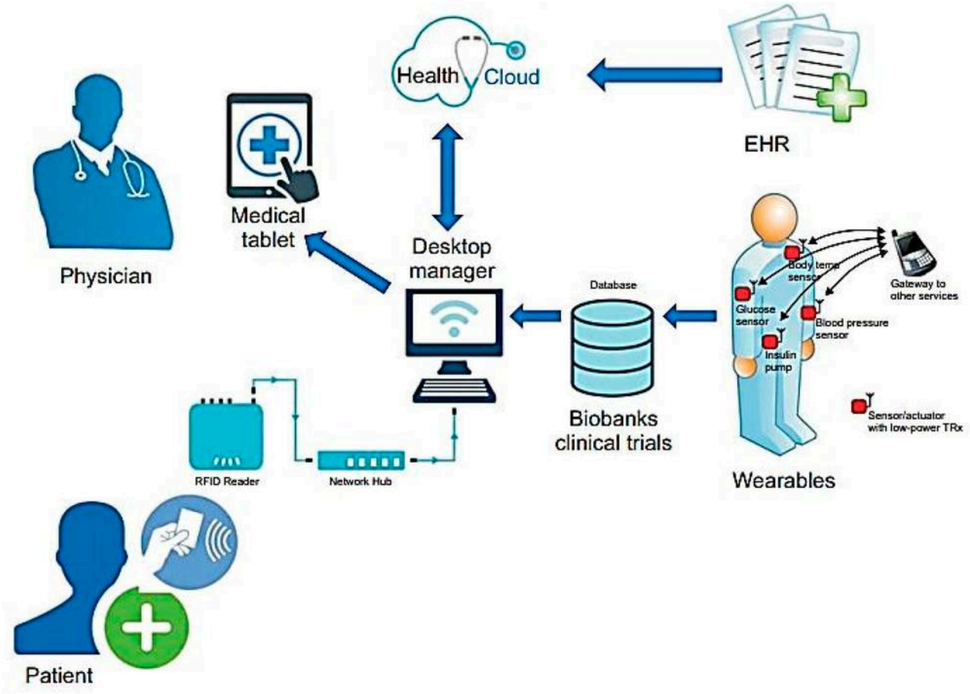
- Practical Application:

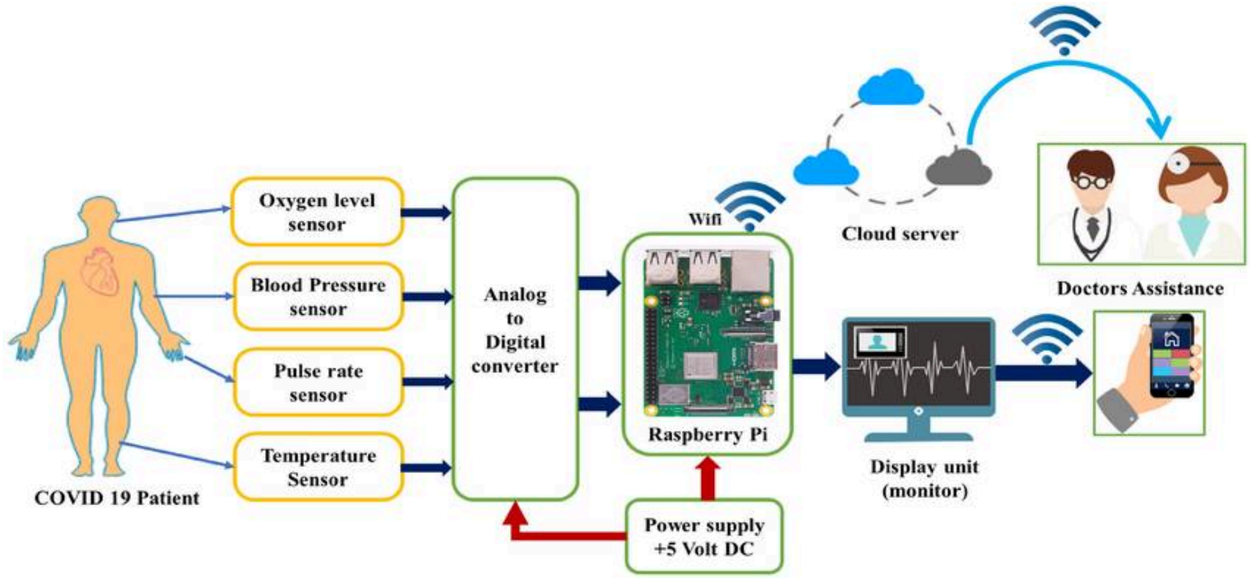
How can farmers use real-time IoT data to improve crop growth and resource management?

- Extension Question:

If you were to expand this IoT agriculture system, what additional sensors or features would you add (e.g., pest detection, weather forecasting), and how would they benefit the farmer?

IoT Application Framework Example: Healthcare





Summary

IoT connectivity enables devices to communicate, exchange, and process data efficiently across networks.
Each element discussed in this chapter plays a vital role in ensuring seamless communication and smart functionality:

- Internet connectivity links IoT devices to networks and cloud platforms.
- Network protocols define how data is transmitted between connected devices.
- Data protocols ensure reliable and secure message exchange.
- IoT framework organizes sensors, gateways, middleware, and cloud into a structured system.
- IoT applications use these frameworks to deliver intelligent automation and real-time data insights.

Together, connectivity, protocols, and frameworks form the foundation of IoT systems that power smart environments in industries, agriculture, healthcare, and daily life.

INTERNET OF THINGS IN FUTURE

The future of the Internet of Things (IoT) promises to be more intelligent, interconnected, and human-centric. As technology continues to evolve, IoT will integrate more deeply with other emerging technologies such as Artificial Intelligence (AI), 5G connectivity, edge computing, and blockchain, creating smarter and more autonomous systems across all industries.

The expansion of AIoT (Artificial Intelligence of Things) will enable devices to not only collect and transmit data but also analyze and make decisions locally, reducing latency and improving efficiency. Meanwhile, 5G networks will accelerate IoT communication speeds, support massive device connectivity, and unlock new applications such as autonomous vehicles, smart manufacturing, and remote surgery.

In addition, sustainable IoT will become a major focus – using energy-efficient sensors, green cloud platforms, and data-driven insights to address environmental challenges and support the UN Sustainable Development Goals (SDGs). As IoT continues to evolve, future professionals must be equipped not only with technical knowledge but also with an ethical and innovative mindset to design systems that improve quality of life while preserving our planet.

Ultimately, the next generation of IoT will move beyond simple connectivity toward intelligent, adaptive, and sustainable ecosystems – where technology seamlessly integrates into daily life, empowering smarter decisions and creating a truly connected world.

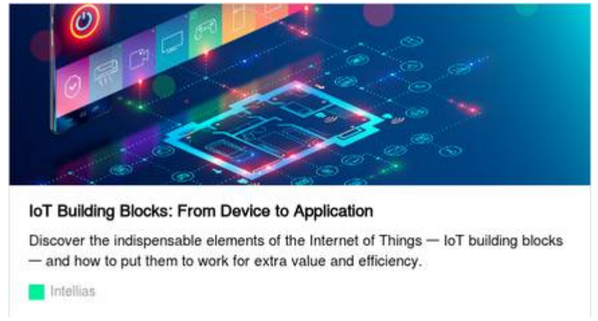
References

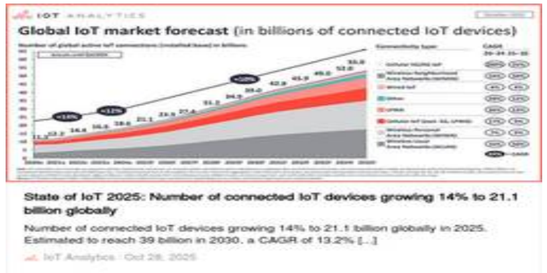
*Some images and reference materials in this book were obtained from online sources and are used for educational purposes only. All visuals and references will be properly updated and credited in the next revision of this publication.

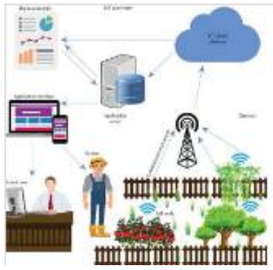
6 IoT Trends Driving Innovation Across Businesses In 2025

This article outlines 6 key IoT trends driving innovation across various industries in 2025, owing to advancements...

Veritis Group Inc / Jun 11, 2025







IoT ecosystem for agriculture.

Download scientific diagram | IoT ecosystem for agriculture. from publication: Smart Agriculture for Sustainable Food Security Using Internet of Things (IoT) | Internet of Things...

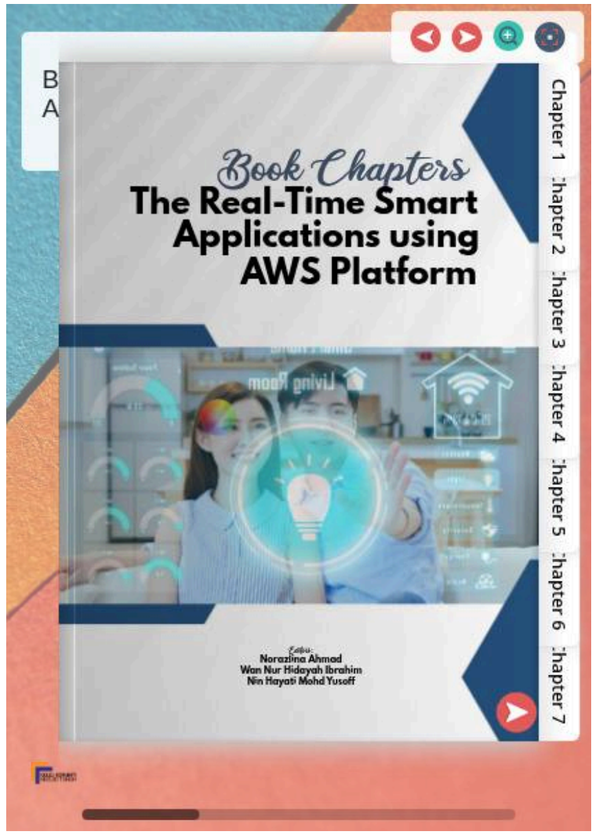
ResearchGate

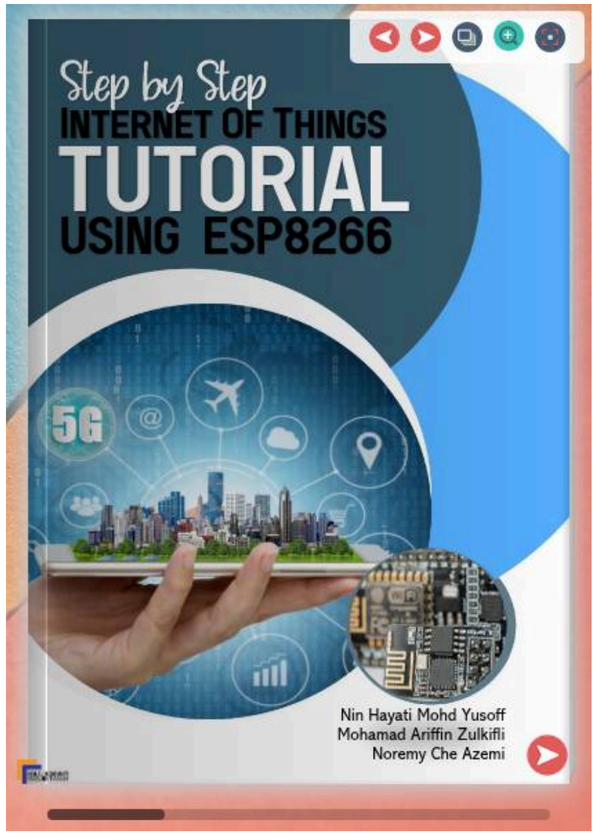
Content is no longer available

https://www.ecot.com.my/industry4_0.html

References

*Some images and reference materials in this book were obtained from online sources and are used for educational purposes only. All visuals and references will be properly updated and credited in the next revision of this publication.





INTERNET OF THINGS FUNDAMENTALS: CONCEPTS, DEVICES, AND APPLICATIONS

Nin Hayati Mohd Yusoff

POLITEKNIK
MALAYSIA
MERLIMAU