INTERNET OF THINGS (IoT)

Connecting Devices • Collecting Data • Building a Smarter World

Live Examples

Smart Home
Smart lights, thermostats, security cameras



Health Care
Wearable devices, remote patient monitoring



Smart Agriculture
Soil sensors, weather monitoring, smart irrigation



Smart Cities
Traffic monitoring, smart parking, waste management



Industrial IoT
Equipment monitoring, predictive maintenance



Smart Transport
GPS tracking, vehicle diagnostics



Wearables
Fitness tracking, health alerts













SENSORS
NETWORKS
CLOUD
APPLICATIONS





Presented By
K. Vineet Kumar
UG CS Lecture

TYPES OF SENSORS

CLASSIFICATION OF SENSORS

1. PASSIVE & ACTIVE

PASSIVE

Cannot independently sense the input.

Examples: Accelerometer, Soil moisture, Water level, Temperature sensors.



ACTIVE

Independently sense the input.

Examples: Radar, Sounder, Laser altimeter sensors.

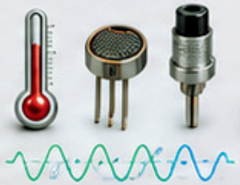


2. ANALOG & DIGITAL

ANALOG

Output is a continuous function of input.

Examples: Temperature sensor, LDR, Analog pressure sensor, Analog Hall effect.



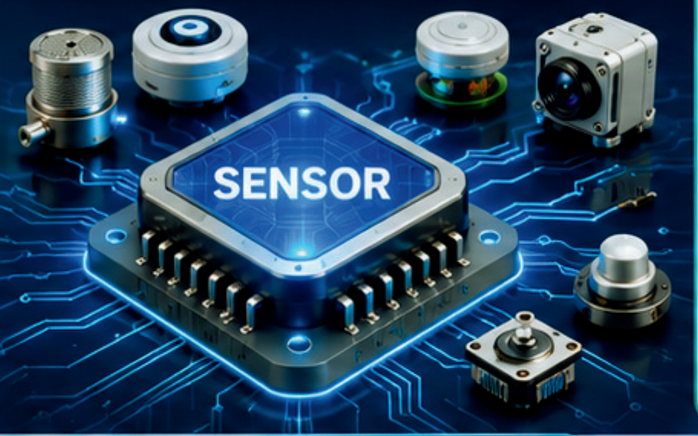
DIGITAL

Output is binary in nature. Overcomes the disadvantages of analog sensors.

Examples: PIR sensor, Digital temperature sensor (DS1620).



Smart Sensors, Smarter World



3. SCALAR & VECTOR

SCALAR

Detects input based only on magnitude. Not affected by direction.

Examples: Temperature, Gas, Strain, Color, Smoke sensor.



VECTOR

Response depends on magnitude, direction & orientation.

Examples: Accelerometer, Gyroscope, Magnetic field, Motion detector sensors.



TYPES OF SENSORS (WITH EXAMPLES)

1. PROXIMITY SENSOR

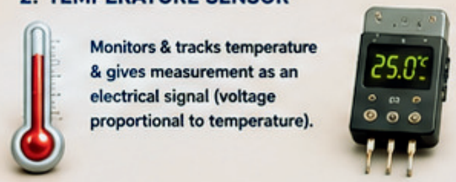
Detects the presence or distance between objects.

Used in elevators, parking lots, automobiles, robotics, etc.



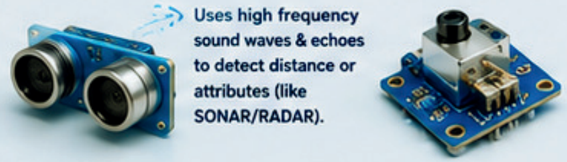
2. TEMPERATURE SENSOR

Monitors & tracks temperature & gives measurement as an electrical signal (voltage proportional to temperature).



3. ULTRASONIC SENSOR

Uses high frequency sound waves & echoes to detect distance or attributes (like SONAR/RADAR).



4. VIBRATION SENSOR

Detects mechanical vibrations and converts into electrical signals.

- (a) Displacement
- (b) Velocity
- (c) Acceleration



5. TILT SENSOR

Detects the tilting or inclination of an object.



6. LEVEL SENSOR

Measures the level of substances like water, oil, slurries, dust particles, solids, etc.



7. TOUCH SENSOR

Detects physical touch on a surface.

Used in touchscreens, trackpads, etc.



8. SOIL MOISTURE SENSOR

Measures the volumetric water content in soil.

Used in agriculture & irrigation.



9. GYROSCOPIC SENSOR

Measures & maintains the orientation and angular velocity of an object.



10. RADIATION SENSOR

Detects alpha, beta, or gamma particles.

Used in surveys & sample counting.



11. POSITION SENSOR

Senses the position of valves, doors, throttles, etc. Supplies signals to control or display devices.



12. GAS SENSOR

Detects & measures the concentration of different gases in the atmosphere or environment.



13. TORQUE SENSOR

Measures the rotating torque. Used to measure the speed of rotation.



14. OPTICAL SENSOR

Detects light waves (UV, visible, IR). Used in smartphones, robotics, Blu-ray players, etc.



15. IMAGE SENSOR

Used for distance measurement, pattern matching, color checking, structured lighting, motion capture & more. Used in 3D imaging, medical, security, automotive, etc.



WHY SENSORS MATTER?

Sense the real world → Convert to signals → Process intelligently → Enable smart actions → Create a better future

APPLICATIONS

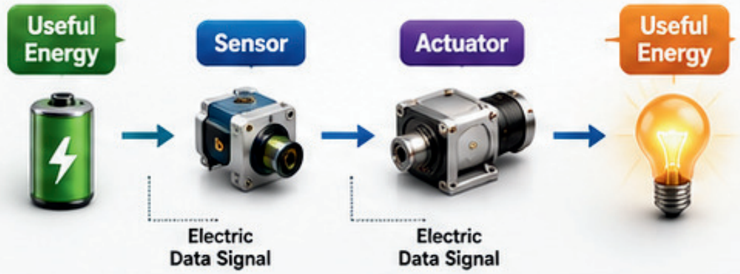
AUTOMOTIVE · INDUSTRY · HEALTHCARE · AGRICULTURE · AEROSPACE · SMART DEVICES

TYPES OF ACTUATORS

Convert Energy into Motion, Powering the World

An actuator is an important mechanical device in many industries. It can convert electrical or fluid energy into mechanical motion to move or control various mechanisms or ssystems in industry.

HOW AN ACTUATOR WORKS



In IoT, an actuator is responsible for the physical movement of an object.

It can be a device that can move things and is powered using battery, electric or manually generated energy.

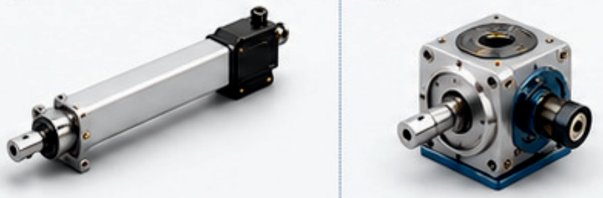
ACTUATORS USED IN

1. STEPPER MOTORS
2. ELECTRIC MOTORS
3. HYDRAULIC CYLINDER



I. BASED ON MOTION

1. LINEAR ACTUATORS
2. ROTARY ACTUATORS



a. HYDRAULIC ACTUATORS
b. PNEUMATIC ACTUATORS
c. MECHANICAL ACTUATORS
d. ELECTRO ACTUATORS



ELECTRO ACTUATORS TYPES

i. ELECTRO MECHANICAL
ii. ELECTRO HYDRAULIC
iii. ELECTRO MAGNETIC



II. BASED ON SOURCE OF ENERGY

a. SUPER COILED POLYMER



TYPES OF ACTUATORS (BY TECHNOLOGY)

1 ELECTRIC ACTUATORS

Use electrical energy to produce mechanical motion using motors, solenoids or servo mechanisms.

Examples:
- Electric motors
- Servo motors
- Solenoids



2 PNEUMATIC ACTUATORS

Utilize compressed air to create linear or rotary motion, used in industrial automation.

Examples:
- Pneumatic cylinders
- Air motors



3 HYDRAULIC ACTUATORS

Employ pressurized fluid (hydraulic oil) to produce motion, suitable for high force and speed applications.

Examples:
- Hydraulic cylinders
- Hydraulic motors



4 THERMAL ACTUATORS

Use temperature changes to trigger movement, often using materials like thermal wax.

Example:
- Used in HVAC systems



5 MAGNETIC ACTUATORS

Leverage magnetic forces to create motion (moving magnet, moving coil, moving iron).

Examples:
- Used in medical devices (MRI machines)
- Precision manufacturing



6 PIEZOELECTRIC ACTUATORS

Use the piezoelectric effect, converting electrical energy into precise physical displacement.

Example:
- Micro-positioning and control



7 MECHANICAL ACTUATORS

Convert one type of musion (e.g., rotary) into another (e.g., linear) using mechanisms like gears, rails or chains.

Examples:
- Gears
- Lead screw
- Rack & pinion



REAL WORLD APPLICATIONS



> "Actuators are the muscles of modern technology, powering motion and building a smarter world."



RFID
Radio Frequency Identification

Working Principle of RFID & Working of RFID System

Identify • Track • Manage



Smart Tags... Smarter Tomorrow...

1 Working Principle of RFID

RFID uses radio waves to identify and track objects. A tag (with a microchip and antenna) stores data. When it enters the range of a reader, the reader's radio waves power the tag (in passive tags) and retrieve the stored data.

- **Radio Frequency** (Wireless communication)
- **Microchip** (stores unique ID/data)
- **Antenna** (sends & receives signals)
- **Tag** (attached to object)

2 Working of RFID System

1. Tag Detection
Reader sends radio waves. Tag receives energy and responds with its ID/data.

2. Data Transmission
Tag sends the stored information back to the reader using radio waves.

3. Data Processing
Reader decodes the data and sends it to the host system (computer/server).

4. Action
The system uses the data for identification, tracking or automation.



RFID Tag (with chip & antenna)

RFID Reader (sends & receives signals)

Host System (computer / server)

3 Live Examples of RFID

1. Retail & Inventory
Automatic product tracking at store checkout.
Example: Walmart uses RFID for faster billing and inventory management.



2. Access Control
Door entry with RFID cards.
Example: Office buildings, colleges, and smart homes.



3. Library Management
Book tracking and issue/return.
Example: University and public libraries.



4. Vehicle Tracking
Track vehicles and manage fleet.
Example: Toll collection, logistics and transport companies.



5. Healthcare
Track patients and medical equipment.
Example: Hospital asset management and patient wristbands.



RFID Tag Types



Real Life Applications



Presented By
K.Vineet Kumar | UG CS Lecture

RFID – Small Tags... Big Impact !

Arduino IDE and Board Types

Code → Upload → Build → Create

Small Code... Big Ideas...

1 Setting Up the IDE

1. Download and Install

Go to the official Arduino website and download the IDE for your operating system.



- Windows
- macOS
- Linux

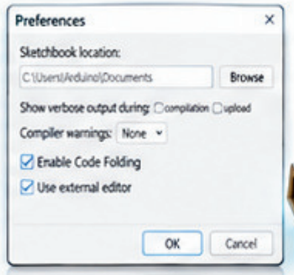
2. Open the IDE

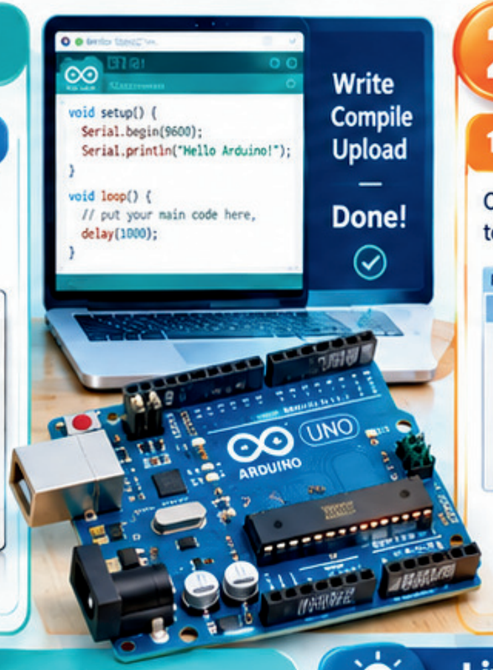
After installation, open the Arduino IDE from your desktop or start menu.



3. Preferences

Set the board, port and other preferences from File > Preferences.

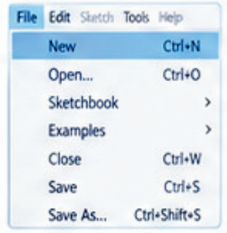




2 Creating and Uploading Sketches

1. New Sketch

Click File > New to create a new sketch.



New Sketch

2. Coding

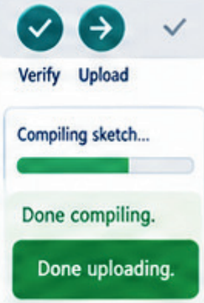
Write your Arduino code in the editor.

```
void setup() {
  pinMode(13, OUT
  Pinvadelapi.setrun);
  Serial.begin(9600);
  Serial.println("LED ON");
}

void loop() {
  digitalWrite(13, HIGH);
  delay(1000);
  digitalWrite(13, LOW);
  delay(1000);
}
```

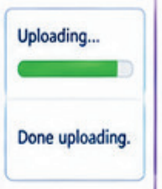
3. Verify and Upload

Click the ✓ button to verify and the → button to upload.



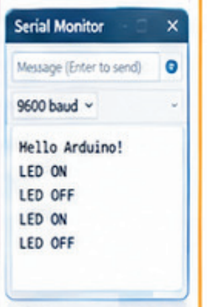
4. Upload

The code is uploaded to the board.



5. Serial Monitor

Open the Serial Monitor to see output from your board.



Arduino Board Types

Different boards for different needs



Arduino Uno (R3)

Beginner Friendly (ATmega328P)



Arduino Nano

Compact Size (ATmega328P)



Arduino Mega

More I/O Pins (ATmega2560)



Arduino Leonardo

USB HID Support (ATmega32u4)



Arduino ESP32

Wi-Fi + Bluetooth (Dual Core)



Arduino Nano 33 IoT

IoT & Sensors (ARM Cortex-M0+)

Live Examples / Real-World Applications

See how Arduino works in real life!

1. Smart Home (LED Control)



Control home lights using Arduino and a mobile app.

2. Temperature Monitoring



Read temperature and humidity using DHT11 sensor.

3. Ultrasonic Obstacle Avoidance



Detect obstacles using ultrasonic sensor.

4. Automated Plant Watering



Water plants automatically using soil moisture sensor.

Learn | Build | Innovate

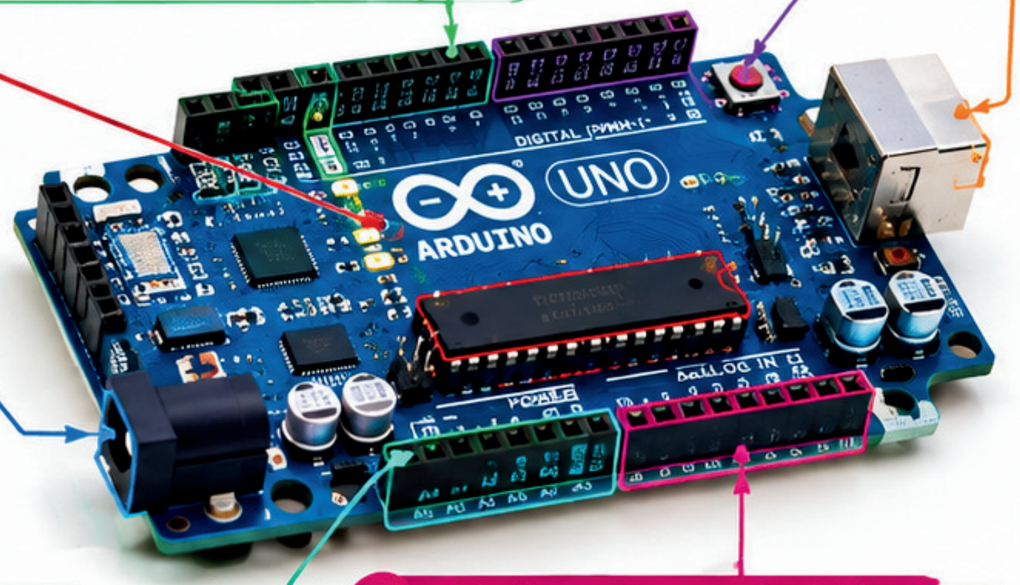
Technology Today... A Smarter Tomorrow...

Presented By
K.Vineet Kumar
UG CS Lecture

Basic Components of an Arduino Board

Small Board | Big Possibilities | Build • Learn • Create

From Ideas to Real Projects



1 Microcontroller

The heart of the board. It processes the code and controls all operations.

(ATmega328P)



Example:
Reads sensor data and controls an LED based on the input.



7 LEDs (Power, TX/RX, Pin 13)

- **Power LED** – Shows board is ON
- **TX/RX LEDs** – Shows data transmission
- **Pin 13 LED** – Built-in LED for testing



6 Reset Button

Resets the microcontroller and restarts the program.



2 USB Connection

Used to upload code and provide 5V power.



Example:
Connects to your computer to upload the sketch.

3 Power Jack

Accepts an external 7–12V DC supply.



Example:
Used to power the board when USB is not available.

5 Analog Input Pins

Read analog values (0–5V).



Example:
Reads the position of a potentiometer (0–1023).

4 Digital Input/Output Pins (I/O)

Read digital signals or control devices.



Example:
Connects to an LED, button, or relay module.

Live Examples

1. Blink LED (Pin 13)



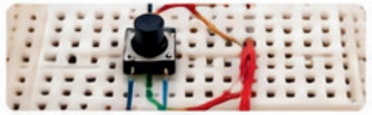
Uses digital pin 13 to blink the built-in LED.

2. Read Potentiometer (Analog)



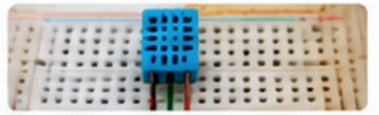
Reads analog value (0–1023) from a potentiometer.

3. Button Input (Digital)



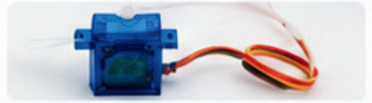
Reads button press and turns on an LED.

4. Temperature Sensor (Analog)



Reads temperature & humidity using a DHT11 sensor.

5. Servo Motor Control



Controls a servo motor using PWM.

6. LCD Display



Displays text on a 16x2 LCD using digital pins.

ARDUINO | Learn Hardware | Write Code | Build Projects | Explore Innovation

Presented By
K.Vineet Kumar
UG CS Lecture

Think Build Grow

Raspberry Pi Development Kit

Small Board | Big Possibilities | Build • Learn • Create

Turn Ideas into Reality!

1 Processor



- Quad-core ARM Cortex-A72
- 1.5 GHz (Raspberry Pi 4)
- Handles all operations of the board.

Example: Runs Python programs, web servers, and AI applications.

2 HDMI



- Connects to monitor/TV
- Supports HD video & audio (up to 4K)

Example: Display desktop, videos, or run a media player.

7 Ethernet



- 10/100/1000 Mbps
- Stable network connection

Example: Connect to the internet or local network.

3 GPIO Ports



- 40 GPIO pins
- Digital I/O
- I2C, SPI, UART, PWM

Example: Connect sensors, LEDs, motors, and other hardware.

6 Reset Button



- Reboots the system
- Useful for troubleshooting

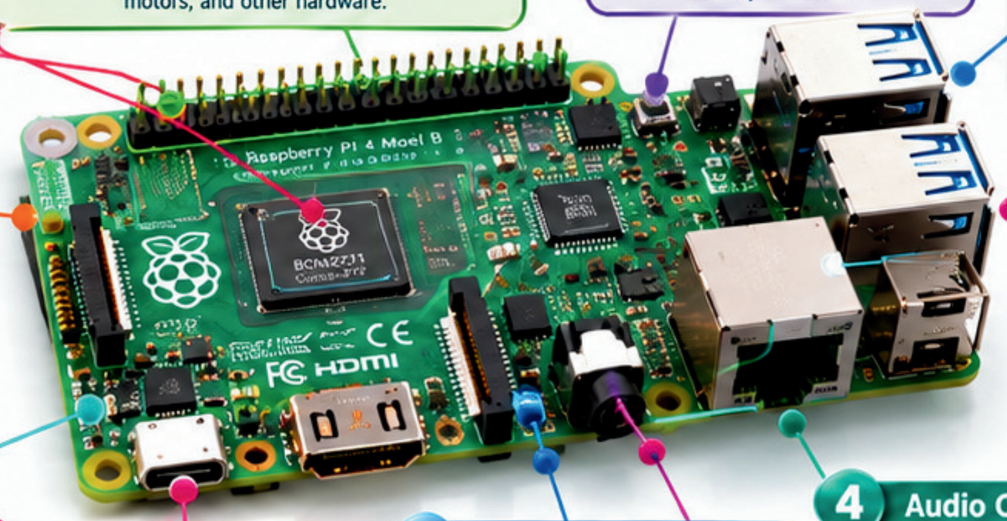
Example: Press to restart if the system freezes.

5 USB Ports



- 2 × USB 3.0
- 2 × USB 2.0
- Connect keyboard, mouse, storage, etc.

Example: Use a USB keyboard, mouse or a Wi-Fi adapter.



Live Examples

- Home Media Center (YouTube, Netflix)
- Smart Security Camera
- Weather Station
- Robotics & Automation
- Web Server
- AI & Machine Learning
- IoT Projects

8 Power Supply



- 5V via USB-C
- 3A recommended

Example: Power the board with a 5V adapter or power bank.

9 Camera Module



- 5 MP (standard)
- 8 MP (higher models)
- CSI interface

Example: Capture photos, record videos, face recognition.

4 Audio Output



- 3.5 mm jack
- HDMI (digital audio)

Example: Play music, YouTube or system sounds.

10 Display



- Touchscreen display
- DSI interface
- Various sizes (7", 5", etc.)

Example: Build a touchscreen interface or control panel.

Real Project Examples



Media Center
Stream videos, music & more



Monitor your home remotely



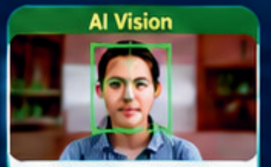
Track temperature & humidity



Build & control with GPIO



Host your own website



Detect faces & objects

Presented By
K.Vineet Kumar
UG CS Lecture

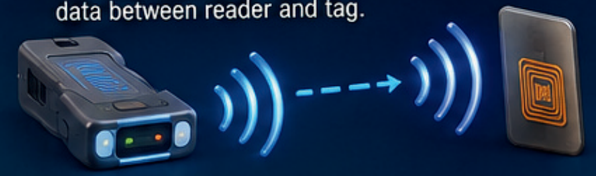
RFID

Principles & Components

PRINCIPLES OF RFID

1 Radio Waves

RFID systems use radio frequency waves to transfer data between reader and tag.



2 Active & Passive Tags

Active Tags

Have battery, longer range, used for tracking large items.





Passive Tags

No battery, use reader's energy, smaller, cheaper, shorter range.

3 Frequency Bands

LF	HF	UHF
125–134 kHz	13.56 MHz	300 MHz – 3 GHz
Short range (up to 10 cm) Animal tracking Access control	Range up to 1 meter Smart cards Library systems	Range up to 12+ meters Supply carin Inventory tracking

4 Data Encoding

Tags store data in a specific format like unique ID, product info, etc. Data can be read or written.





REAL-WORLD EXAMPLES



COMPONENTS OF RFID SYSTEMS

1 RFID Tags

Small devices with microchip and antenna. Store information and communicate with reader.



2 RFID Readers

Emit radio waves to communicate with tags and convert signals into digital data.



3 Antenna

Facilitates the transmission and reception of radio waves. Affects read range and performance.



4 Middleware / Software

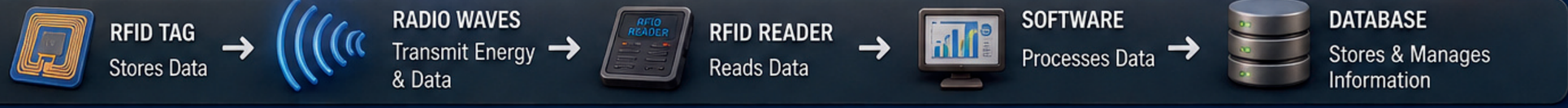
Processes data collected by readers, manages inventory, tracks assets, and integrates with other systems.



5 Database

Stores RFID data for analysis and reporting. Can be part of ERP or standalone system.





Wireless Sensor Networks

From Early Ideas to a Smarter, Connected Future

Small Sensors... Big Impact...

1. Early Developments (1960s – 1990s)

1 Initial Concepts
- Idea of distributed sensing and wireless communication.

Live Example:
Military surveillance systems (e.g., early sensor networks for battlefield monitoring in the 1980s–70s).



2 Advancements in Sensor Technology
- Smaller, cheaper, more accurate sensors.

Live Example:
Weather monitoring sensors, temperature and humidity sensors in research labs (1980s–90s).



3 Networking Technologies
- Development of wireless protocols and ad-hoc networks.

Live Example:
Early wireless standards like Bluetooth (1994) and IEEE 802.11 (Wi-Fi).



2. Formalization and Research (1990s)

4 First WSNs
- Low-power, multi-hop sensor networks.

Live Example:
UC Berkeley's WSN testbed (TinyOS project, 2000).



5 Protocols and Standards
- MAC, routing, and industry standards.

Live Example:
ZigBee (2003), IEEE 802.15.4 for low-power communication.

ZigBee®

IEEE 802.15.4



3. Commercialization and Expansion (2000s)

6 Commercial Products
- Ready-to-use sensor nodes and platforms.

Live Example:
Moteiv (MicaZ), Crossbow, TelosB (popular WSN hardware).



7 Research and Development
- Scalable, energy-efficient, long-range networks.

Live Example:
Self-organizing networks, LEACH protocol, data aggregation techniques.



4. Integration with IoT (2010s)

8 Internet of Things (IoT)
- WSNs become part of larger IoT ecosystems.

Live Example:
Smart home sensors, industrial IoT (IIoT) platforms, connected devices.



9 Smart Cities and Industry 4.0
- Large-scale, real-time applications.

Live Example:
Smart traffic systems, factory monitoring, predictive maintenance.



5. Current Trends and Future Directions (2020s and Beyond)

10 Edge Computing
- Local processing, reduced latency.

Live Example:
Edge gateways in smart factories, real-time video analytics.



11 AI and Machine Learning
- Smarter data analysis & prediction.

Live Example:
Anomaly detection in environmental monitoring.



12 Security and Privacy
- Data protection and secure communication.

Live Example:
Encryption, authentication, blockchain-based WSNs.



12 Sustainability
- Energy-efficient and eco-friendly designs.

Live Example:
Solar-powered sensor nodes, green IoT solutions.



Presented By **K.Vineet Kumar** | UG CS Lecture

Connect • Monitor • Build a Safer & Smarter Tomorrow

IoT Internet of Things

Connect • Collect • Analyze • Automate • Make Life Smarter

Smarter Devices
Better Decisions
A Connected World

1 Sensors Collect Data

Sensors detect changes in the environment and collect data (temperature, motion, humidity, light, location, etc.).



Real Life Examples:

- Smart farm sensors monitor soil moisture and weather.
- Motion sensors in smart homes.
- Wearable fitness trackers collect health data.

2 Data Transmission

The collected data is sent over a network (Wi-Fi, Bluetooth, LoRa, 4G/5G, etc.) to the cloud or a local server.



Real Life Examples:

- Smart devices send data via Wi-Fi or cellular networks.
- LoRa used for long-range communication (e.g., agriculture).

3 Data Processing and Analysiss

The data is stored, processed and analyzed using cloud platforms and AI/ML to find patterns, trends or conditions.



Real Life Examples:

- Weather prediction from sensor data.
- AI detects unusual activity (like a fire or intrusion).
- Predictive maintenance in industries.

4 Actions and Automation

Based on the analysis, the system takes automatic actions or triggers devices (e.g., turn on/off, open/close, adjust settings).



Real Life Examples:

- Smart sprinkler turns on/off automatically.
- Smart lights adjust based on motion or time.
- Industrial machines adjust settings automatically.

5 User Interface

Users can monitor and control the system through a mobile app, web dashboard or voice assistant.



Real Life Examples:

- Check farm conditions on a mobile app.
- Monitor home appliances via web dashboard.
- Use voice assistant (Alexa/Google) to control devices.

Smart Homes | Smart Agriculture | Smart Cities | Industrial Automation | Healthcare | Logistics | Security

Presented by K.Vineet Kumar

Wireless Sensor Networks (WSNs)

Small Sensors... Big Impact...

Sense → Collect → Transmit → A Smarter Tomorrow

1 Distributed Sensing

Multiple sensor nodes are deployed across a region to collect data from different locations and send it to a central base station.

Live Examples:

- Forest fire detection
- Environmental monitoring
- Agricultural field monitoring



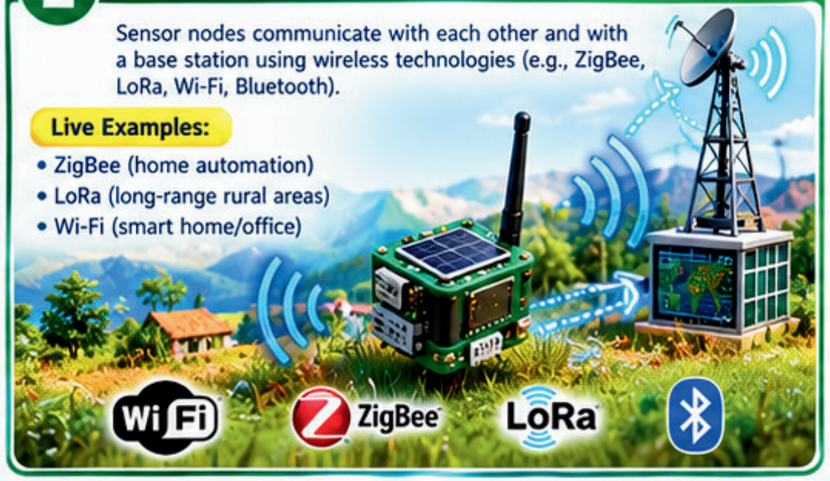
Example: Sensors in a forest detect temperature rise and send data to the base station.

2 Wireless Communication

Sensor nodes communicate with each other and with a base station using wireless technologies (e.g., ZigBee, LoRa, Wi-Fi, Bluetooth).

Live Examples:

- ZigBee (home automation)
- LoRa (long-range rural areas)
- Wi-Fi (smart home/office)



3 Low-Cost and Resource-Constrained

WSN nodes are inexpensive and have limited processing power, memory and battery life.

Live Examples:

- Smart agriculture (low-cost soil sensors)
- Home security (battery-powered nodes)
- Industrial monitoring (cost-effective sensors)



- Low Cost
- Low Power
- Limited Memory
- Small Size

4 Energy Efficiency

Designed to consume minimal power and extend battery life using low-power protocols and sleep modes.

Live Examples:

- Solar-powered sensors
- Sleep mode operation
- Energy harvesting (vibration, solar)



Solar Power | Low Power Protocols | Sleep Mode | Energy Harvesting → Longer Battery Life

5 Scalability and Flexibility

WSNs can grow from a few nodes to thousands, and can be easily adapted to different environments and applications.

Live Examples:

- Smart city deployments
- Industrial automation
- Healthcare monitoring



a small farm → to a smart city → to a large industry

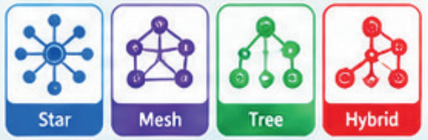


6 Network Topology

Nodes can be organized in different topologies like star, mesh, tree or hybrid for reliable communication and better coverage.

Live Examples:

- Star topology (home automation)
- Mesh topology (disaster recovery)
- Tree topology (agriculture)





Wireless Sensor Networks (WSNs)

The Node, Connecting Nodes, Networking Nodes

Small Nodes... Big Connections... Smart Networks...

1 Nodes in Wireless Sensor Networks

Nodes are the building blocks of a WSN. They sense, process and communicate data.

Sensor Nodes

- Collect data from the environment
- Measure temperature, humidity, light, pressure, etc.
- Low power, low cost

Live Example:
Soil moisture sensor node in smart agriculture.



Temperature
Humidity
Light
Soil Moisture

Actuator Nodes

- Perform actions based on data
- Control devices and systems
- Used in automation and monitoring

Live Example:
Smart irrigation system (opens/closes water valve).



Valve Control
Motor Control
Light Control
Alarm Trigger

Gateway Nodes

- Connect WSN to external networks
- Bridge with Internet / cloud
- Higher processing power

Live Example:
Smart city gateway sending data to cloud server.



Wi-Fi
LoRa
4G/5G
Cloud

2 Connecting Nodes

Nodes communicate with each other using protocols, routing and network topologies.

Communication Protocols

- Define how nodes exchange data
- Ensure reliable and efficient communication

ZigBee
LoRa
Bluetooth
Wi Fi

Live Example:
ZigBee, LoRa, Bluetooth (used in real WSNs).



Routing

- Finds the best path for data transmission
- Handles dynamic changes and failures

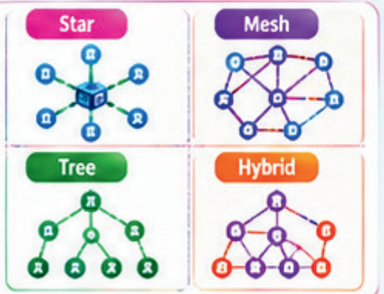
Live Example:
RPL (IPv6 Routing Protocol) in low-power networks.



Network Topologies

- Define how nodes are organized
- Impact coverage, reliability and performance

Live Example:
Star, Mesh, Tree, Hybrid (used in different applications).



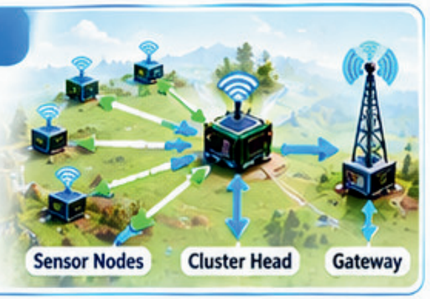
3 Networking Nodes

Networking nodes manage data, save energy and ensure scalability of the WSN.

Data Aggregation

- Combines data from multiple nodes
- Reduces redundancy and traffic

Live Example:
Temperature data from many sensors aggregated at a cluster head.



Energy Management

- Extends node battery life
- Uses sleep modes and efficient routing

Live Example:
Duty cycling and energy-aware routing in solar-powered nodes.



Energy Harvesting
Sleep Mode
Low Power
Battery Saving

Scalability

- Handles more nodes and larger areas
- Maintains performance and reliability

Live Example:
Large-scale forest monitoring with thousands of nodes.



Presented By **K.Vineet Kumar** | UG CS Lecture

Connect | *Monitor* | *Analyze* | *Build a Smarter Tomorrow*

S.V.K.P & Dr. K.S. RAJU ARTS & SCIENCE COLLEGE (A)

Department of Computer Science

IoT

IEEE 802.15.4 | Zigbee | HART | NFC

Connecting Devices • Enabling Smart Solutions

IEEE 802.15.4

A low-power, low-data rate standard for Wireless Personal Area Networks (WPANs).



- Low-power, low-data rate
- Wireless Personal Area Network
- PHY and MAC layers
- Frequency bands (2.4 GHz, 915 MHz, 868 MHz)
- Network topologies
- Device types

IEEE 802.15.4 Architecture

Application Layer

MAC Layer

PHY Layer

Applications in IoT

- Smart Homes
- Industrial Automation
- Wearables
- Agriculture
- Healthcare

Frequency Bands

868 MHz (Europe)	915 MHz (US)	2.4 GHz (Global)

Zigbee

Low-power, reliable and secure communication for IoT devices.

Zigbee Network Components and Structure

- **Zigbee Coordinator (ZC)** – Manages the network
- **Zigbee Router (ZR)** – Extends the network
- **Zigbee End Device (ZED)** – Simple, low power device

Zigbee Network Topologies



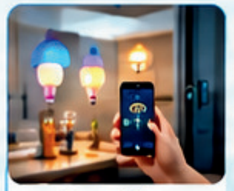
Zigbee Protocol Stack

- Application Layer (APL)
- Network Layer (NWK)
- Medium Access Control (MAC)
- Physical Layer (PHY)

Zigbee in IoT Applications

- Smart Homes
- Industrial Automation
- Healthcare
- Agriculture

Zigbee Live Examples



Smart Home (Lights, Thermostat)



Industrial Automation (Sensors, Control)



Healthcare (Health Monitoring)

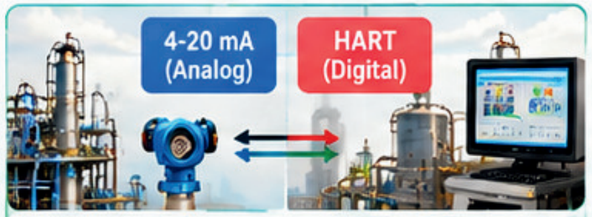


Agriculture (Soil Sensors)

HART — Highway Addressable Remote Transducer

HART

- Open Standard
- Hybrid Communication
- How it works
- Digital Communication over Analog Wiring
- Master-Slave Architecture
- Two Simultaneous Communication Channels



Benefits of using HART in IoT

- Leveraging Existing Infrastructure
- Enhanced Device Information
- Improved Operational Efficiency
- Cost-Effective Solution

HART Live Example



Monitoring temperature & pressure in a chemical plant using HART

NFC (Near Field Communication)

Short-range wireless technology for simple and secure data exchange.

Types of NFC

Passive NFC Devices

- NFC tags
- Smart posters
- Access cards



Active NFC Devices

- Smartphones
- NFC readers
- Wearables



Advantages of NFC for IoT

- Short-range communication (1–10 cm)
- Low power consumption
- No need for batteries
- Ease of use
- Cost-effective implementation
- Bidirectional communication

NFC Live Examples



Mobile Payments (Google Pay, Apple Pay)



Access Control (Smart Locks, ID Cards)



Asset Tracking (NFC Tags)



Wearables (Health, Fitness)

S.V.K.P & Dr. K.S. RAJU ARTS & SCIENCE COLLEGE (A)

Department of Computer Science



Z-Wave | BLE (Bluetooth Low Energy)

Smart Connectivity • Low Power • A Connected Future

Z-WAVE

Reliable. Secure. Smart Home.

Z-Wave is a wireless communication protocol designed for home automation, security and IoT devices. It operates on low power and provides reliable mesh networking.



Z-Wave Protocol Stack

5. Application Layer
4. Network Layer
3. Transport Layer
2. MAC Layer
1. PHY Layer

Z-Wave Components

- **Controllers** (Primary / Secondary)
- **Slave nodes** (End devices)
- **Home ID** (Identifies the network)
- **Node ID** (Unique device address)
- **Routing table** (Finds best path for data)

Characteristics of Z-Wave

- Low power consumption
- Reliable mesh networking
- Secure (AES-128 encryption)
- Interoperability
- Easy installation
- Supports smart home devices

Applicaterions of Z-Wave

- Smart Homes
- Security Systems
- Lighting Control
- Energy Management
- Building Automation

BLE

Bluetooth Low Energy

BLE is a low-power wireless technology that enables short-range communication between devices. It is widely used in IoT for its low energy consumption and simplicity.



Key Characteristics of BLE in IoT

- **Low Power Consumption** (extends battery life)
- **Short-Range Communication** (10–100 meters)
- **Data Transmission** (low data rate, high efficiency)
- **Cost-Effectiveness** (affordable implementation)
- **Compatibility** (wide device support)

BLE's Role in IoT

- **Wearable Devices** (fitness trackers, smart watches)
- **Smart Home Automation** (lights, locks, thermostats)
- **Industrial IoT** (sensors, monitoring)
- **Healthcare** (patient monitoring)
- **Retail** (beacons, payment systems)

BLE in IoT Applications

- **Asset Tracking** (people, vehicles, assets)
- **Fitness and Health Devices** (heart rate, steps, sleep)
- **Smart Environments** (office, campus, city)
- **Industrial and Commercial IoT** (equipment, inventory, logistics)

Z-Wave in Real Life – Examples



Smart Home Automation (Connected Lights, Locks)



Home Security (Sensors & Alarms)



Smart Thermostat (Temperature Control)



Smart Plug (ON/OFF Devices)



Smart Blinds (Automated Control)

BLE – Live Examples



Fitness Tracker (Heart Rate, Steps)



Smart Home (Lights, Locks)



BLE Beacon (Indoor Navigation)



Asset Tracking (Inventory Management)



Healthcare (Health Monitoring)

Small Devices... Big Impact... Smarter Tomorrow !

Presented By **K.Vineet Kumar** UG CS Lecture

Smart Home | IoT | BLE | Z-Wave

BACnet & Modbus

Building Smarter, Connected Systems

Presented By
K.Vineet Kumar
UG CS Lecture

BACnet
Building Automation and Control Networks

ASHRAE BACnet™

Focus
Building automation & management systems (BAS)

Architecture
Client-Server / Peer-to-Peer

Transmission
IP (Ethernet), MS/TP, ARCNET, Wi-Fi, etc.

Features
- Object-oriented
- Multi-vendor interoperability
- Real-time data exchange
- Supports large-scale systems



VS



Modbus
Industrial Communication Protocol

Modbus

What is Modbus protocol and how does it work?
Master-slave communication, simple, reliable, widely used in industrial devices.

Variants of Modbus protocol
- Modbus RTU
- Modbus TCP
- Modbus ASCII
- Modbus Plus

In an IoT context
Connects industrial devices to IoT platforms for monitoring and control.

HOW DOES BACNET WORK?

Device → BACnet Object → Protocol (BACnet) → Other Devices

Real World Examples

BACnet Examples


Smart Buildings (Hotel, Hospital, Office)


HVAC Control (Heating & Cooling)


Lighting Automation (Smart Lighting)

Modbus Examples


Factory Automation (PLC + Sensors)


Energy Monitoring (Power Meters)


Weather Station (Sensors to Cloud)

Modbus Key Differences

Aspect	BACnet	Modbus
Complexity	Higher	Lower
Use Cases	Building automation, HVAC, smart buildings	Industrial automation, SCADA, IoT
Architecture	Client-Server / P2P	Master-Slave

BACnet in IoT Context

- Smart buildings
- Smart cities
- Energy management
- Industrial IoT

BACnet Gateways



- BACnet to Modbus
- BACnet to IP
- BACnet to Cloud

Modbus Architecture



Use Cases

- Industrial automation
- SCADA systems
- Energy & power systems
- IoT device integration

IPv6 Key Benefits of IPv6 for IoT

More Addresses | Stronger Security | Smarter Networks | A Connected Future

IPv6 Powers the Next Generation of IoT

1 Large Address Space

IPv6 provides **128-bit addresses**, allowing an almost unlimited number of devices to connect.

Live Example:
A smart city with millions of sensors, cameras and devices can get a unique IP address each.



2 Enhanced Security

IPv6 has built-in security features like **IPsec**, encryption and better authentication mechanisms.

Live Example:
Secure communication in smart homes, healthcare devices and industrial IoT using IPsec.



3 Support for 6LoWPAN

IPv6 works over low-power, low-bandwidth networks using 6LoWPAN (IPv6 over Low-Power **Wireless Personal Area Networks**).

Live Example:
Smart sensors in agriculture, wearables and home automation using 6LoWPAN.





Presented By
K.Vineet Kumar
UG CS Lecture

4 Improved Routing and Packet Handling

IPv6 simplifies routing, supports **larger packets** and reduces **fragmentation**, improving efficiency.

Live Example:
Faster and smoother data flow in smart vehicles and industrial networks with fewer delays.



5 Efficient Multicast Communication

IPv6 supports native multicast, allowing data to be sent to multiple devices at once.

Live Example:
Live video streaming to multiple security cameras or smart TVs in a building.



6 Auto-Configuration

Devices can configure their own IPv6 addresses automatically (without manual setup).

Live Example:
New smart devices (like bulbs, thermostats or sensors) join the network instantly.



7 Scalability and Interoperability

IPv6 scales for future growth and works across different networks and devices.

Live Example:
Global IoT systems, from home devices to **enterprise and cloud**, working together seamlessly.



6LoWPAN

(Low-Power Wireless Personal Area Networks)

IPv6 for Low-Power Devices | Efficient | Secure | Connected

Small Devices... Big Connections... Powered by IPv6



Key Features of 6LoWPAN

1. **IPv6 Integration**
   Brings IPv6 connectivity to low-power devices and constrained networks.
2. **Low Power and Low Data Rate**
   Designed for battery-powered devices with low bandwidth requirements.
3. **Mesh Networking**
   Allows multi-hop communication for extended coverage.
4. **Header Compression and Fragmentation**
   Reduces packet size and handles small MTU efficiently.
5. **Open Standards**
   Based on IEEE 802.15.4 and IETF 6LoWPAN standards.
6. **Security**
   Supports encryption and secure communication for IoT devices.



6LoWPAN Protocol Stack and Operation

- Application Layer (CoAP, MQTT, HTTP)
- Transport Layer (UDP / TCP)
- IPv6 Layer (IPv6)
- 6LoWPAN Adaptation Layer (Header Compression, Fragmentation)
- MAC Layer (IEEE 802.15.4)
- Physical Layer (IEEE 802.15.4)

Operation Flow

1. Data from sensor is created at the application layer.
2. IPv6 packets are compressed and fragmented (if needed) at 6LoWPAN layer.
3. Sent over low-power IEEE 802.15.4 network.
4. Reaches the gateway/router and then the Internet/Cloud.

Application Areas



- Smart lights • Smart locks
- Thermostats • Security cameras



- Machine monitoring • Asset tracking
- Predictive maintenance



- Smart meters • Energy monitoring
- Grid management



- Environmental monitoring • Agriculture
- Wearables • Smart cities

Live Examples



Smart Thermostat
USES 6LoWPAN + IPv6 for home automation.



Industrial Sensor
Monitors temperature & pressure in factories.



Smart Energy Meter
Sends data to utility via 6LoWPAN network.



Smart Agriculture
Soil moisture sensor in farm fields.



Wearable Device
Health monitoring with low power 6LoWPAN.



Smart City Sensor
Air quality & environment monitoring.

Why 6LoWPAN?

- Enables IPv6 in constrained devices
- Saves energy and extends battery life
- Supports large-scale IoT networks
- Interoperable with global standards
- Provides secure and reliable communication



From Sensors to Smart Cities 6LoWPAN Makes IoT Possible!

Presented By **K.Vineet Kumar** UG CS Lecture

Real World Signals...

Analog and Digital Sensors

...to Smart Decisions!

Sense ▸ Measure ▸ Connect ▸ Build a Smart World

Analog Sensors

Produce a **continuous** electrical signal that varies in proportion to the physical quantity being measured.

Output

- Continuous signal (voltage/current)
- Analog waveform (e.g., 0 – 5V, 4 – 20mA)
- Needs an ADC (Analog to Digital Converter)
- for microcontrollers



Analog Output (Continuous)

Characteristics

- Continuous range of values
- Higher noise and signal variation
- Simple and low cost
- Requires calibration and conditioning

Example:
Temperature sensor (LM35) gives analog voltage proportional to temperature.



Analog Output (0 – 1V)



Applications

- Environmental monitoring
- Industrial process control
- Medical devices
- Agriculture and weather stations



Digital Sensors

Produce discrete digital signals (usually HIGH/LOW or I²C/SPI) that represent the measured value.

Output

- **Digital signal** (HIGH/LOW or data bits)
- I²C, SPI, UART, 1-Wire, etc.
- Directly read by microcontrollers
- (no ADC required)

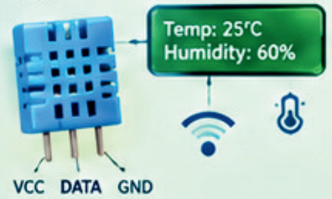


Digital Output (0 or 1)

Characteristics

- Discrete values (binary)
- High noise immunity
- More accurate and stable
- Supports complex data (multiple parameters)

Example:
DHT11 digital temperature and humidity sensor provides data via digital signal (single wire).



Applications

- Smart homes and automation
- Health monitoring
- Robotics and control systems
- Security and surveillance

Live Examples of Analog Sensors



Temperature Sensor (LM35)

Analog output: 0 – 1V



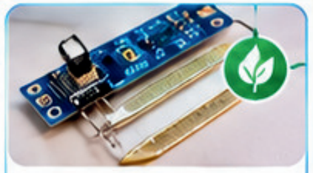
Pressure Sensor (0 – 5V)

Analog output: 0 – 5V



Light Sensor (LDR)

Analog output: variable resistance/voltage



Soil Moisture Sensor

Analog output: 0 – 3.3V

Live Examples of Digital Sensors



DHT11 (Temperature & Humidity)

Digital output: I²C / Single wire



MPU6050 (Accelerometer & Gyroscope)

Digital output: I²C



PIR Motion Sensor

Digital output: HIGH / LOW



HC-SR04 (Ultrasonic Distance)

Digital output: TRIG / ECHO (PWM)

Better Sensors...

Presented By **K.Vineet Kumar** UG CS Lecture

Smarter IoT...

Cloud Based Platform AWS IoT

Connect • Collect • Analyze • Build a Smarter Future

Internet of Things Powered by AWS

Key Features of AWS IoT Cloud Platform

Device Connectivity
Connect millions of devices securely using MQTT, HTTP and WebSockets.

Device Shadowing
Maintain a virtual copy of device state for offline and online access.

Rules Engine
Process device data in real-time and trigger actions (e.g., SNS, Lambda, S3).

Security and Authentication
Use certificates, IAM, and encryption for devices and data.

Scalability
Handle millions of devices and billions of messages with ease.



Use Cases of AWS IoT

Smart Home Applications
- Smart lights, thermostat, security cameras
- Voice-controlled devices

Industrial IoT
- Predictive maintenance
- Real-time monitoring

Healthcare Solutions
- Remote patient monitoring
- Wearable health devices

Smart Cities
- Smart traffic systems
- Environmental monitoring

Live Examples

Smart Home – AWS IoT



Control home devices from anywhere using AWS IoT Core.

Factory Monitoring – AWS IoT



Monitor machines and get alerts for maintenance.

Healthcare – AWS IoT



Track patient vitals and send real-time data to the cloud.

Smart City – AWS IoT



Manage traffic, air quality and public safety with connected devices.

Getting Started with AWS IoT

1. **Create an AWS Account**
   Sign up at aws.amazon.com
2. **Set Up AWS IoT Core**
   Go to AWS IoT Core and create a new thing (device) and certificate.
3. **Set Up AWS IoT Core**
   Configure policies, rules and connect your device using MQTT/HTTPS.
4. **Explore Additional Resources**
   Use AWS documentation, tutorials and sample code to build your IoT solution.

Blynk IoT Cloud Based Platform

Build • Connect • Control • Monitor • Automate

Your IoT Ideas Made Simple

Overview of Blynk IoT Platform

Blynk is a powerful, easy-to-use IoT platform that helps you build and manage connected devices, create custom apps, and control your hardware from anywhere.

- Cloud-based platform
- No complex infrastructure
- Works with multiple devices
- Simple drag & drop interface

Launch and Evolution

2014
Blynk 1.0
(Initial Launch)

2018
Blynk 2.0
(New Architecture)

Today
Blynk IoT
(Advanced Features)



Key Features of Blynk

- **Device Management** — Add, organize and manage your devices.
- **User Roles and Permissions** — Control access for teams and users.
- **Data Handling** — Store and process device data.
- **Automations** — Create smart workflows and rules.
- **Datastreams** — Read and write data in real-time.
- **Integration and Connectivity** — Connect with third-party services and APIs.
- **Events** — Trigger actions based on device events.
- **Firmware Updates** — Update device firmware over the air (OTA).

Core Components

- **Blynk App** — Mobile app to control and monitor devices.
- **Blynk Console** — Manage devices, data and users from web.
- **Blynk Cloud** — Reliable and secure cloud infrastructure.

Supported Protocols

MQTT · HTTP/HTTPS · CoAP · BLE · Wi-Fi · LoRa

Scalability and Security

Scalable Architecture
- Handle thousands of devices
- Global infrastructure
- High availability

Data Security
- End-to-end encryption
- Secure authentication
- GDPR compliant

Conclusion

- Easy to use and beginner friendly
- Powerful features for real-world IoT projects
- Secure, scalable and reliable
- Perfect for hobbyists, businesses and industries

Blynk Makes IoT Simple!

Live Examples

1. Smart Home Automation



- Control lights, fans, security
- Monitor temperature & humidity
- Get notifications on your phone

2. Agricultural Monitoring



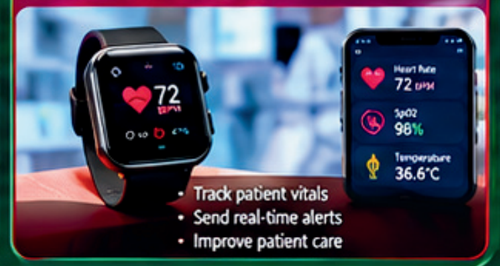
- Monitor soil moisture, temperature
- Control irrigation system
- Improve crop yield

3. Industrial IoT



- Monitor machines in real-time
- Predict maintenance
- Reduce downtime

4. Healthcare Solution



- Track patient vitals
- Send real-time alerts
- Improve patient care

5. Smart City



- Monitor air quality & traffic
- Smart lighting & waste management
- Build sustainable cities

Blynk *Build Smarter, Live Smarter!*

Presented By **K.Vineet Kumar** UG CS Lecture

Privacy and Security Issues in IoT

Connected Devices • Collected Data • Real Risks • Safer Tomorrow

Protect Your Data Secure Your Future



Privacy Challenges in IoT



1. Data Collection

IoT devices collect sensitive personal and behavioral data (e.g., location, health, usage patterns).

Live Example:
Smartwatches track heart rate, location and sleep data, which can reveal your lifestyle.



2. Informed Consent

Users often don't fully understand what data is collected or how it will be used.

Live Example:
Many apps get location access without clear explanation of data usage.



3. Transparency Issues

Hidden data practices and unclear privacy policies make it hard to know how data is used.

Live Example:
Smart home brands may share your data with third parties without clear notice.



4. De-identification Risks

Even "anonymous" data can be re-identified using other information.

Live Example:
Location + time data can reveal your home, work and daily routine, even if your name is removed.

5. Vendor Practices

Google, amazon, mi

Different manufacturers have varying privacy standards and data sharing policies.

Live Example:
Some smart device vendors may share usage data for advertising or research purposes.



Your Privacy + Your Data = Your Right

Be Aware • Stay Safe • Use IoT Smart

Security Risks in IoT



6. Weak Authentication

- Default passwords and weak login methods make devices easy to hack.

Live Example:
Many IP cameras use default passwords like "admin/admin" which attackers exploit.



7. Legacy Devices

Older devices may not receive security updates, leaving known vulnerabilities.

Live Example:
Old smart TVs or routers continue to run outdated firmware with known flaws.



8. Rogue Devices

Unauthorized devices can join the network and steal or manipulate data.

Live Example:
A fake smart device can be added to your Wi-Fi and capture network traffic.



9. Interoperability Issues

Different devices and platforms may not work securely together, creating gaps.

Live Example:
A smart home device from one brand may connect through an insecure API with another platform.



10. Data Breaches

Stolen data can lead to identity theft, financial loss or device control.

Live Example:
In 2023, a major IoT platform was breached, exposing millions of user records and device data.

Live Examples



1. Smart Home Camera

Collects video & location data
→ Privacy risk if hacked.



2. Health Monitoring Wearable

Tracks health data
→ Sensitive personal information.



3. Smart Car

Shares location & driving behavior
→ Can be misused by third parties.



4. Smart Factory

Connected machines
→ Vulnerable to rogue devices or cyber attacks.



5. Smart City

Massive data collection
→ Privacy and surveillance concerns.



6. Home Wi-Fi Router

Default credentials
→ Easy entry for attackers.



7. Smart Meter

Energy usage data
→ Can reveal daily routines.



8. Industrial Sensor

Unsecured device
→ Can be used for data theft or system disruption.

Smarter Devices | Stronger Security | A Safer Connected World

Presented By **K.Vineet Kumar** UG CS Lecture

S.V.K.P & Dr. K.S. RAJU ARTS & SCIENCE COLLEGE (A)
Department of Computer Science

LoRa, RPL & REST in IoT

Low Power • Reliable Networks • Connected Future

Presented By
K.Vineet Kumar
UG CS Lecture

LoRa

Long Range
Low Power
Wide Area



Key Features of LoRa Protocol for IoT

- Physical Layer (LoRa Modulation)
- LoRaWAN Protocol (MAC and Network Layers)
- Low Power Consumption
- Security
- Device Classes
- Data Rate and Payload

LoRaWAN

- Application Layer
- Network Layer
- MAC Layer
- Physical Layer

Typical Use Cases in IoT

- Smart Cities
- Agriculture
- Industrial IoT
- Environmental Monitoring
- Supply Chain & Logistics

Real-World Example



LoRaWAN soil moisture sensor in agriculture sends data from fields to a gateway over long distance with very low power.

Sensor → Gateway → Cloud

RPL

Routing Protocol for Low-Power and Lossy Networks



Key Features and Benefits of RPL

- Multipoint-to-point traffic
- Scalability
- Adaptability
- Energy efficiency
- Quality of Service (QoS)
- Security

Real-World Applications of RPL for LLNs

- Industrial Automation
- Agriculture
- Smart Buildings
- Healthcare
- Traffic Management
- Disaster Response

DODAG Messages

DIS (DODAG Information Solicitation)
Used by a node to request DODAG information.

DIO (DODAG Information Object)
Provides DODAG information (rank, parent, etc.).

DAO (Destination Advertisement Object)
Used to advertise routes to the root.

Live Example



In a smart agriculture field, RPL helps low-power sensor nodes route data efficiently to the gateway even with unreliable links.

REST

Representational State Transfer



Client-Server Model



Stateless Communication

- Each request is independent
- No session is maintained
- Server does not store client state

Standard HTTP Methods

GET	Retrieve data
POST	Create data
PUT	Update data
DELETE	Remove data
PATCH	Partial update

Advantages of REST for IoT

- Simplicity and Familiarity
- Scalability
- Flexibility in Data Handling
- Interoperability
- Ubiquity

Use Cases in IoT

- Device Management
- Data Collection and Monitoring
- Cloud Integration

Limitations of REST in IoT

- Synchronous and Unidirectional
- Statelessness Overhead
- Not Ideal for Constrained Devices

Live Example



A smart home device uses REST API to get current temperature data from the cloud.



S.V.K.P & Dr. K.S. RAJU
ARTS & SCIENCE COLLEGE (A)
Department of Computer Science

AMQP, CoAP, MQTT & Edge Connectivity

Protocols for a Connected IoT World

Lightweight • Reliable • Secure • Scalable • Smart

Presented By
K.Vineet Kumar
UG CS Lecture

AMQP (Advanced Message Queuing Protocol)

AMQP in IoT Context

- Enables reliable message exchange between IoT devices, gateways and cloud.
- Used for device-to-cloud and device-to-device communication.

Purpose

- Reliable message delivery
- Decouples applications and devices
- Supports asynchronous communication

Architecture

IoT Device → AMQP Broker (Message Queue) → Application (Cloud/Server)

Reliability and Delivery Guarantees

- At-least-once / At-most-once / Exactly-once

Security

- TLS/SSL, SASL, Authentication & Authorization

Extensibility

- Supports custom plugins and protocols

Live Example: Industrial sensors send data to cloud using AMQP for reliable processing.

CoAP (Constrained Application Protocol)

Key Features of CoAP

- Lightweight and Efficient
- Based on UDP
- RESTful Architecture
- Support for Multicast
- Reliability and Congestion Control
- Security (DTLS)

Use Cases in IoT

- Smart Homes
- Industrial Automation
- Smart Energy
- Building Automation

Live Example: Smart thermostat in a home uses CoAP to communicate with the gateway.



MQTT (Message Queuing Telemetry Transport)

Key Features of MQTT for IoT

- Lightweight and Efficient
- Publish/Subscribe Architecture
- Quality of Service (QoS) Levels (0,1,2)
- Persistent Sessions and Last Will Messages
- Security (TLS, Username/Password)
- Scalability

How MQTT Works in IoT

1. **Connection Establishment** (Client connects to Broker)
2. **Publish and Subscribe** (Topics and Messages)
3. **Message Delivery** (QoS ensures delivery)
4. **Session Management** (Persistent sessions, Last Will)

Live Example: Weather station publishes temperature data to MQTT broker.

Sensor → MQTT Broker → Cloud

Edge Connectivity Overview

Sensors (Devices) → Edge Device (Processing) → IoT Gateway (Protocol Translation) → Cloud (Analytics)

Edge Devices

- Raspberry Pi, Industrial PC, Smart Gateway, Edge Sensors

IoT Gateways

- Protocol conversion, Data filtering, Local processing, Security

Benefits of Edge Computing

- Low latency
- Reduced bandwidth
- Real-time analytics
- Improved reliability
- Better security

Live Example: Smart factory uses edge gateway to process sensor data locally and send only important data to the cloud.



Common Protocols Used for Edge Connectivity

Application Layer Protocols

1. MQTT (Message Queuing Telemetry Transport)
2. CoAP (Constrained Application Protocol)
3. HTTP/HTTPS

Network and Transport Layer Protocols

1. TCP (Transmission Control Protocol)
2. UDP (User Datagram Protocol)
3. IPv4/IPv6
4. 6LoWPAN

Data Link and Physical Layer Protocols

1. IEEE 802.15.4
2. Zigbee
3. Bluetooth Low Energy (BLE)
4. LPWAN (Low Power Wide Area Network)

Specialized Protocols and Interfaces

1. Modbus
2. OPC UA
3. BACnet
4. Allen Bradley

Role of Protocols in Edge Connectivity

- Enable device communication
- Support real-time data processing
- Ensure interoperability
- Provide security and reliability

Real-World Examples



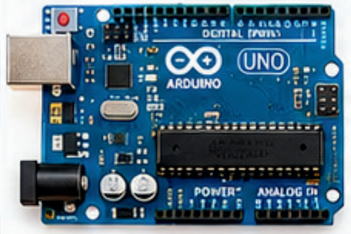
Connected Devices
Smarter Decisions
A Better Tomorrow

Arduino UNO Architecture & IDE Setup, Software, Libraries & LED Interfacing

Build | Program | Connect | Create IoT Solutions



1. Arduino Uno Architecture



- ATmega328P Microcontroller
- Digital Pins (0-13)
- Analog Pins (A0-A5)
- PWM Pins (~3, ~5, ~6, ~9, ~10, ~11)
- USB Interface
- Power Jack
- ICSP Header

2. ATmega328P Microcontroller



- 8-bit AVR
- 32 KB Flash
- 2 KB SRAM
- 1 KB EEPROM
- 16 MHz

3. Input/Output Pins

- **Digital Pins 0 – 13** (Input / Output)
- **Analog Pins A0 – A5** (Analog Input)
- **PWM Pins ~3, ~5, ~6, ~9, ~10, ~11** (Pulse Width Modulation)

4. Communication & Connectivity

- **USB** (Programming & Power)
- **Power Jack** (7–12V DC)
- **ICSP Header** (Bootloader / Programming)

5. Role in IoT

- Sensor Data Collection
- Data Processing & Control
- Cloud Communication
- Low Cost & Easy to Use

6. Setting up the IDE

ARDUINO IDE

1. Download Arduino IDE
2. Install
3. Connect Board
4. Launch IDE
5. Select Board & Port
6. Upload Sketch
7. Troubleshooting

7. Writing Arduino Software

LED Blink Example

```
void setup() {
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);
  delay(1000);
  digitaalWrite(13, LOW);
  delay(1000);
}
```

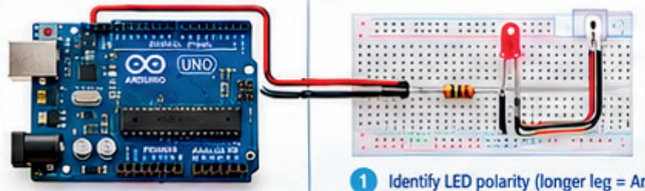
- Simple
- Easy to Learn
- Builds Logic
- Works on Real Hardware

8. Arduino Libraries

- ArduinoIoTCloud
- Arduino_ConnectionHandler
- WiFiNINA
- Ethernet
- PubSubClient
- SoftwareSerial
- LiquidCrystal

- Simplified Development
- Hardware Support
- Code Reusability
- Community-Driven
- IDE Integration

9. Interfacing Arduino with LED



Components:

- Arduino Board
- LED
- Resistor (220Ω)
- Breadboard
- Jumper Wires

1. Identify LED polarity (longer leg = Anode, shorter leg = Cathode)
2. Connect resistor (220Ω) in series with LED
3. Connect cathode to GND

Live Examples

Smart Home LED Control
Control LED from anywhere using IoT (Mobile / Web).

Temperature Sensor → Cloud
Read temperature, send to cloud, view on dashboard.

IoT Device Monitoring
Monitor sensor data, get alerts, smart decision making.

Presented By
K.Vineet Kumar
UG CS Lecture

Implementation of IoT with Arduino

Sensors → Data → Cloud → Smart Applications

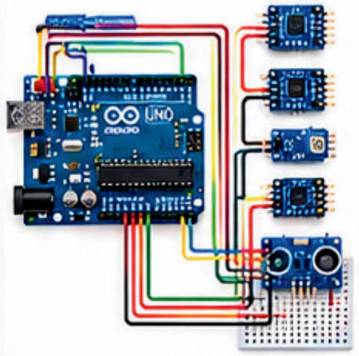
Build
Connect
Innovate

1 Hardware Setup

Choose Sensors

- DHT11 (Temperature & Humidity)
- HC-SR04 (Ultrasonic) (Ultrasonic Distance)
- LDR (Light Sensor)
- Soil Moisture (Soil Health)

Connect Sensors



Wi-Fi Module (Optional)



ESP8266

- Wi-Fi Connectivity
- 2.4 GHz
- Low Power
- Easy Integration

Live Example:

✓ Read Temperature & Humidity and send to cloud via Wi-Fi

2 Coding



Example Code:

```
void setup() {
  Serial.begin(9600);
  dht.begin();
  WiFi.begin(ssid, pass);
  Blynk.begin(auth, ssid, pass);
}
void loop() {
  float t = dht.readTemperature();
  float h = dht.readHumidity();
  Blynk.virtualWrite(V0, t);
  Blynk.virtualWrite(V1, h);
  delay(2000);
```

Libraries

- DHT.h (DHT Sensor)
- ESP8266WiFi.h (Wi-Fi)
- BlynkSimpleEsp8266.h (Blynk)

Code Structure

- setup() → Initialize
- loop() → Repeatedly run

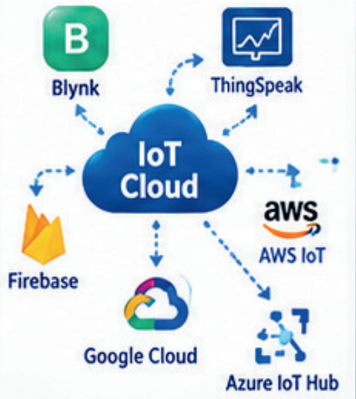
Data Acquisition

→ Read sensor values

Data Transmission

→ Send to cloud / app

3 Cloud Integration (Optional)



IoT Platforms

- Blynk
- ThingSpeak
- AWS IoT
- Firebase
- Google Cloud
- Azure IoT Hub

Data Visualization



Connecting and using various IoT Cloud Based Platforms



5 Steps for Connecting and Using IoT Cloud Platforms

1 Platform Selection

Blynk
ThingSpeak
aws

Choose the right platform for your needs

2 Device Registration

- Create account
- Add device
- Get authentication details

3 Establishing Secure Connections

- Use SSL/TLS
- Get tokens/keys
- Secure Wi-Fi setup

4 Data Management

- Store data
- Organize data
- Handle real-time updates

5 Application Development

- Build dashboard
- Add features
- Integrate with app

6 Monitoring & Management

- Track data
- Set alerts
- Manage devices

5 IoT Cloud Based Platform - Blynk

Blynk

Launch and Evolution

- 2014 – Launch
- IoT for Everyone
- Growth & Global Adoption

Core Components

Blynk App | Blynk Console | Blynk Cloud

Key Features of Blynk

- Device Management
- Data Handling
- Datastreams
- Events
- User Roles and Permissions
- Automations
- Integration and Connectivity

Supported Protocols

- Wi-Fi
- Bluetooth
- MQTT
- HTTP
- CoAP

Firmware Updates

- OTA Updates
- Easy Deployment
- Remote Management

Scalabliity and Security

- Scalable Architecture
- Data Security
- Encrypted Communication

Live Examples









Presented By
K.Vineet Kumar
UG CS Lecture

IoT Cloud Based Platforms

ThingSpeak | AWS IoT

Connect → Collect → Analyze → Build Smart Solutions

Smart Solutions Better Tomorrow

ThingSpeak
IoT Analytics Platform

Visualize • Analyze • Act



MATLAB

1. Platform Purpose
- Data Handling
- Data Collection
- Visualization

2. Key Features of ThingSpeak
- MATLAB Integration
- REST API
- Channel Management

MATLAB

3. Getting Started with ThingSpeak
1. Account Creation
2. Channel Setup
3. Integration with Other Platforms

4. Integration with Other Platforms
The Things Network

Device Compatibility

Limitations and Considerations

Free Tier
- Limited channels
- Data retention
- Rate limits

Custom Cloud Solutions
- More control
- Higher cost
- Scalability



Live Example – ThingSpeak
- ✓ Collect temperature & humidity data
- ✓ Store in ThingSpeak channel
- ✓ View live charts & alerts

28.6°C | 65%

AWS IoT
Secure ◆ Scalable ◆ Reliable

1. Key Features of AWS IoT
- **Device Connectivity** — Connect and manage devices
- **Device Shadowing** — Maintain device state
- **Rules Engine** — Route data to AWS services
- **Security and Authentication** — IAM, certificates, encryption
- **Scalability** — From few devices to millions

2. Use Cases of AWS IoT
- Smart Home Applications
- Industrial IoT
- Healthcare Solutions
- Smart Cities

3. Getting Started with AWS IoT
1. Create an AWS Account
2. Set Up AWS IoT Core
3. Utilize AWS SDKs
4. Explore Additional Resources

AWS IoT Architecture (Live Example)

Device → AWS IoT Core → Rules Engine → DynamoDB →
- Real-time data
- Alerts
- Dashboards
- Mobile Access

Live Examples

ThingSpeak – Smart Agriculture
- ✓ Soil moisture
- ✓ Temperature
- ✓ Humidity
- ✓ Live chart on ThingSpeak

ThingSpeak – Weather Monitoring
- ✓ Wind speed
- ✓ Temperature
- ✓ Humidity
- ✓ Real-time data

AWS IoT – Smart Home
- ✓ Lights control
- ✓ Temperature
- ✓ Security
- ✓ Mobile app

AWS IoT – Industrial Monitoring
- ✓ Machine data
- ✓ Predictive maintenance
- ✓ Real-time alerts

AWS IoT – Healthcare
- ✓ Patient monitoring
- ✓ Live health data
- ✓ Alerts & notifications

Real Devices | Real Data | Real World Solutions

From Sensors to the Cloud... Build Smarter with IoT!

Presented By
K.Vineet Kumar
UG CS Lecture

S.V.K.P & Dr. K.S. RAJU ARTS & SCIENCE COLLEGE (A)

Department of Computer Science

Technology Connects The World

Google Cloud IoT Core



Overview of Google Cloud IoT Core

Google Cloud IoT Core is a managed service that helps you securely connect, manage and ingest data from IoT devices at scale.

Key Features

- **Device Management** — Register, monitor and manage devices at scale.
- **Data Ingestion** — Collect and process data from millions of devices.

Use Cases

- Asset Tracking
- Remote Monitoring
- Smart Homes

Live Examples



Fleet Tracking (GPS devices)



Factory Monitoring (Sensors)



Smart Home Devices (Smart thermostats, cameras)

Alternatives to Google Cloud IoT Core

- AWS IoT
- Microsoft Azure IoT
- IBM Watson IoT
- ThingsBoard

Cloud Computing

1. Service Models

IaaS — Infrastructure as a Service (e.g., AWS EC2, Google Compute Engine)

PaaS — Platform as a Service (e.g., Google App Engine, Heroku)

SaaS — Software as a Service (e.g., Gmail, Google Workspace, Salesforce)

2. Deployment Models

- **Public Cloud** (e.g., AWS, Azure, Google Cloud)
- **Private Cloud** (e.g., on-premises)
- **Hybrid Cloud** (e.g., combination of both)

3. Benefits

- Cost Efficiency
- Scalability
- Accessibility
- Disaster Recovery

! Challenges

- Security
- Downtime
- Vendor Lock-in

Live Example



Using Google Drive (SaaS), we can store, access and share files from anywhere.

Fog Computing

Key Features

- Proximity to Data Sources
- Reduced Latency
- Bandwidth Efficiency
- Scalability
- Enhanced Security and Privacy

Applications of Fog Computing

- Smart Cities
- Healthcare
- Industrial IoT

Live Examples



Smart Traffic System (Edge cameras + Fog nodes)



Remote Patient Monitoring (Local gateway + Fog node)



Factory Automation (Sensors + Fog node)



Presented By **K.Vineet Kumar** UG CS Lecture

S.V.K.P & Dr. K.S. RAJU ARTS & SCIENCE COLLEGE (A)

Department of Computer Science

Secure Connected Smart Future

Privacy and Security Issues in IoT

Privacy Challenges in IoT

1. Data Collection

IoT devices collect huge amounts of personal and sensitive data (e.g., location, health, behavior).

Live Example:
Fitness tracker collects your heart rate, steps, location and sleep data.

2. Informed Consent

Users often don't fully understand or agree to how their data is used.

Live Example:
Smart TV collects viewing habits without clear user consent.

3. Transparency Issues

It's not always clear how data is shared, stored or used.

Live Example:
Smart home devices may share data with third-party companies without notice.

4. De-identification Risks

Even anonymized data can sometimes be re-identified.

Live Example:
Location data from ride-sharing apps can reveal your identity by tracking your daily routes.

5. Vendor Practices

Different vendors have different privacy policies and data usage rules.

Live Example:
A smart device brand may share your data for targeted ads or marketing without your knowledge.



Security Risks in IoT

1. Weak Authentication

Poor passwords or lack of multi-factor authentication can be easily exploited.

Live Example:
Default admin password on a router gives attackers full access.

2. Legacy Devices

Older devices may not receive security updates and remain vulnerable.

Live Example:
Old IP cameras without patches can be hacked remotely.

3. Rogue Devices

Unauthorized devices can join the network and cause harm.

Live Example:
Hacker adds a fake device to the network and steals data.

4. Interoperability Issues

Different devices and platforms may not work securely together.

Live Example:
Smart home device from one brand may not support the security protocol of another.

5. Data Breaches

Attackers can steal, modify or destroy data stored in IoT systems.

Live Example:
Hacker accesses cloud data from insecure IoT device and leaks personal information.

Live Examples – Real World IoT Scenarios



Smart Home
Privacy risk: Voice & video recordings, home activity data.



Healthcare
Privacy risk: Health records, location, personal data.



Smart Cities
Privacy risk: Movement patterns, facial recognition data.



Industrial IoT
Security risk: Device tampering, production data theft.



Connected Vehicles
Privacy risk: Location tracking, driving behavior.



Wearables
Privacy risk: Heart rate, fitness and location data.

Be Aware
Be Safe
Stay Secure

Presented By **K.Vineet Kumar** UG CS Lecture

LOAD BALANCING

From Single Server to Multiple Servers

Distribute the Load | Improve Performance | Avoid Downtime

Load Balancing is the process of distributing incoming network traffic across multiple servers to ensure **better performance, high availability** and **avoid any single point of failure.**

1. IPL FINAL MATCH

Live Example



During IPL final, millions of fans try to watch live, so the traffic is distributed across multiple servers to handle the huge load smoothly.

2. D-MART CASH COUNTERS

Real Time Example



In D-Mart, when the supervisor asks customers to go to different counters, **the load is distributed across multiple counters for faster service.**

3. INTER RESULTS

Live Example



During Inter results, lakhs of students access the website at the same time. If only a single server is used, it can get overloaded and go down.

Single Server

- Overloaded
- Slow Response
- May Go Down

Multiple Servers

- Handles Huge Traffic
- Fast Response
- High Availability

Single Server (Without Load Balancing)



Huge Traffic
(Thousands / Lakhs of Users)



- Server Overloaded
- Slow Performance
- Downtime / Service Failure

One Server = One Point of Failure

Load Balancing

Multiple Servers (With Load Balancing)



- Balanced Load
- Faster Response
- High Availability
- No Single Point of Failure

Presented By **K.Vineet Kumar** UG CS Lecture


Thank You
Your Support and Attention Mean a Lot!
LEARN
PRACTICE
IMPROVE
SUCCEED
Keep Learning Keep Growing
Presented by
K.Vineet Kumar