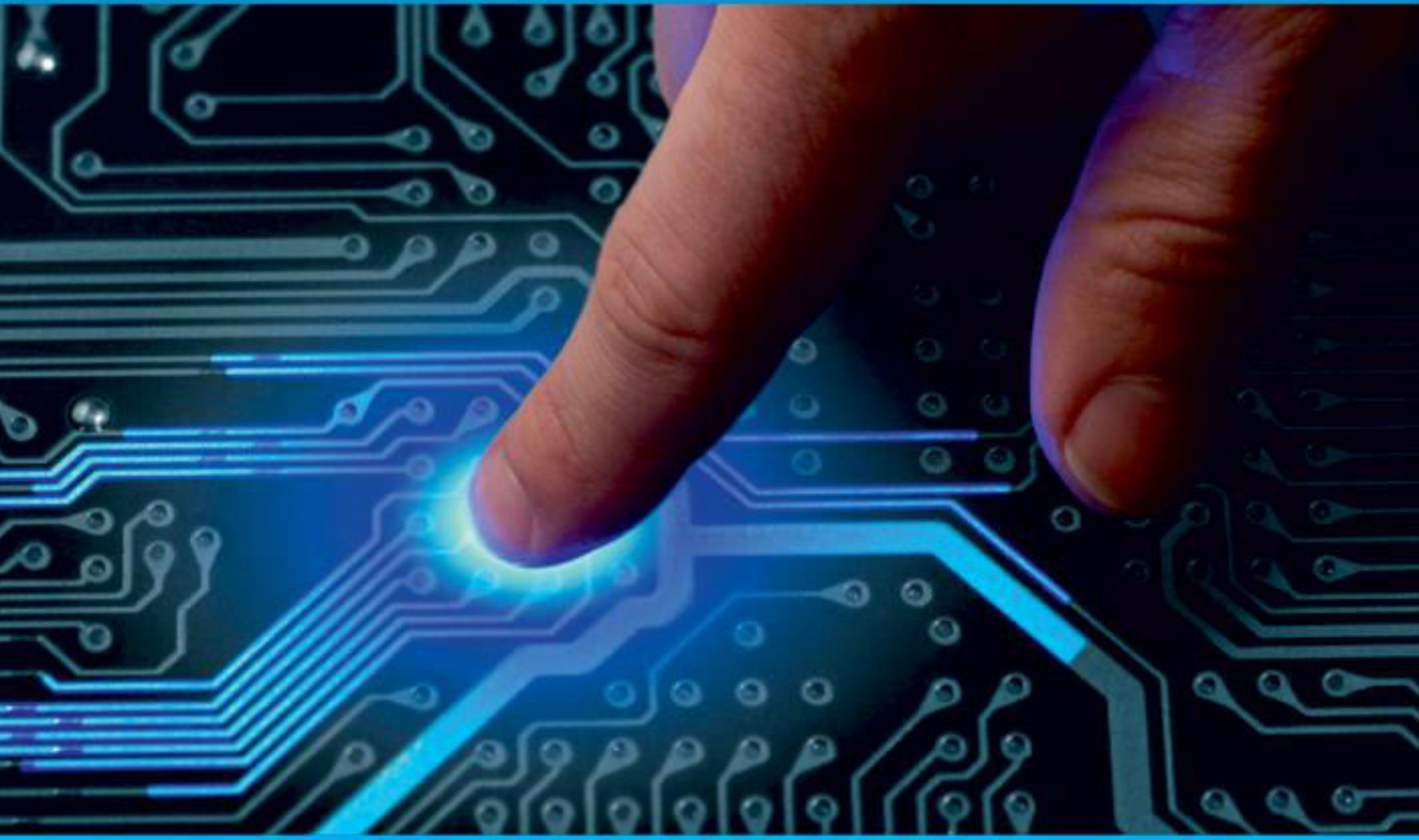
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 12, Issue 4, April 2024

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.379

A Study on Internet of Things

G. Sivakumar[1], R. Arthi[2], Kirubakaran Rangasamy[3]

Assistant Professor, Gnanamani College of Engineering, Rasipuram, India[1]

Assistant Professor, Department of Cyber Security, Muthayammal Engineering College, India [2]

Project Manager, Strive Technology, Chennai, India [3]

ABSTRACT: The paper titled "Internet of Things" provides a comprehensive overview of the Internet of Things (IoT), emphasizing its evolution, importance, applications, and various architectural layers. The authors delve into the physical and logical design of IoT, highlighting computing methods, analytics, and the significance of IoT in sensing, connectivity, data processing, and user interfaces. The document explores IoT protocols, microcontrollers, and popular hardware like Raspberry Pi, showcasing their applications in IoT development. Additionally, the paper addresses power management, form factors, cost considerations, library installations, and the role of programming languages in IoT projects. The authors discuss security challenges, common attacks, and strategies for securing IoT devices. Practical guidance on troubleshooting common issues, identifying potential attack areas, and the importance of documentation is also provided.

KEYWORDS: Internet of Things, IoT evolution, IoT applications, IoT architecture, physical design, logical design, computing methods, analytics, IoT protocols, microcontrollers, Raspberry Pi, power management, form factor, cost considerations, library installation, programming languages, security, troubleshooting, documentation.

I.INTRODUCTION

Internet: Inter connectivity-For global connection
Things: Embedded system devices-sensors, actuators, RFID tags, QR codes and so many.
IoT, or the Internet of Things, network of interconnected computing devices which are embedded in everyday objects that are connected to the internet , enabling them to send and receive data.

Evolution of IOT

Why Is Internet of Things (IoT) so important?

□ In recent years, the Internet of Things (IoT) has emerged as a crucial 21st-century technology.
□ The global count of interconnected IoT devices has surged by 16%, reaching 16.7 billion.
□ Projections indicate a substantial increase to 30 billion.
□ Enterprise spending on IoT experienced notable growth, expanding by 21.5% in 2022 and reaching a significant milestone of $201 billion.

Important Applications of IoT

- Manufacturing and Industry 4.0
- Precision Agriculture in the realm of Agriculture
- Smart Cities
- Retail
- Healthcare
- Logistics and Transportation
- Smart Homes, and Building Automation
- Environmental Monitoring
- Energy and Utilities

are all key domains influencing and shaped by technological advancements.

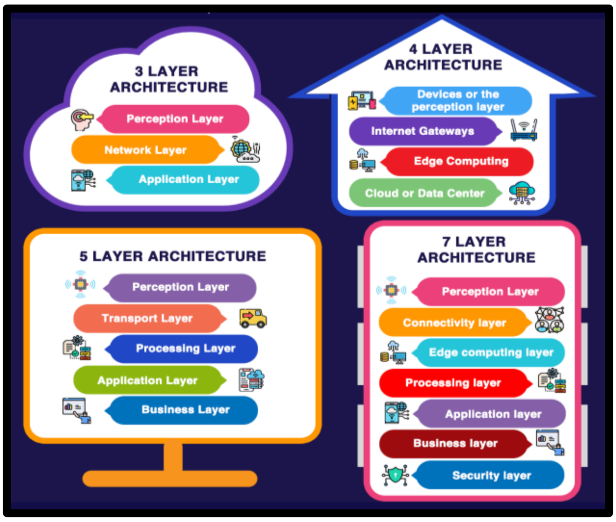
Layer of IOT

The intricacy of IoT solution frameworks may pose a significant hurdle to the widespread acceptance of IoT technology.

The four types of models

3 layer architecture
4 layer architecture
5 layer architecture
7 layer architecture

○ The three-layer architecture provides a fundamental representation of what an IoT network should resemble.
○ The four-layer architecture presents a depiction of the technical framework of an IoT solution.
○ The 5-layer architecture presents an **extended high-level overview** of how IoT solutions work
○ The seven-layer architecture for the Internet of Things provides a comprehensive framework for implementing IoT solutions.

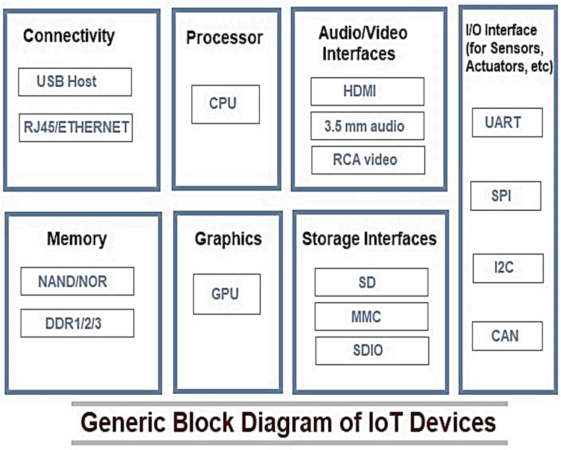


Physical Design of IoT

The term "Things" in IoT typically denotes devices with distinct identities capable of remote sensing, actuating, and monitoring.

- IoT devices can exchange data directly or indirectly with other connected devices and applications.

- They may collect and process data locally, transmit it to centralized servers or cloud-based back-ends, or execute tasks both locally and within the IoT infrastructure, adapting to temporal and spatial constraints.

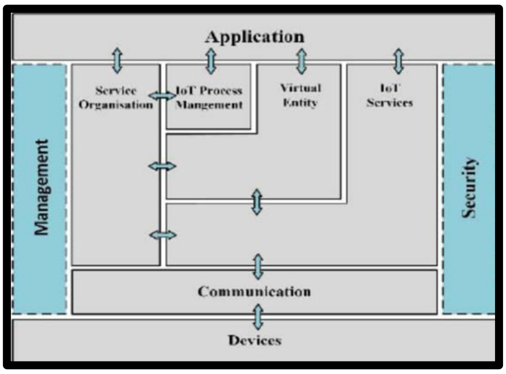


Logical Design of IoT

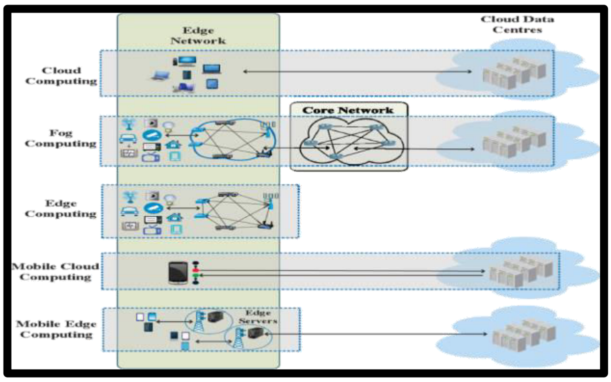
- The logical design of an IoT system involves an abstract representation of entities and processes, avoiding low-level implementation specifics.

- It delineates essential components, data flow, interactions, and processes constituting the IoT solution without delving into technical details.

- Functional blocks within an IoT system contribute to identification, sensing, actuation, communication, and management capabilities.



Computing Methods



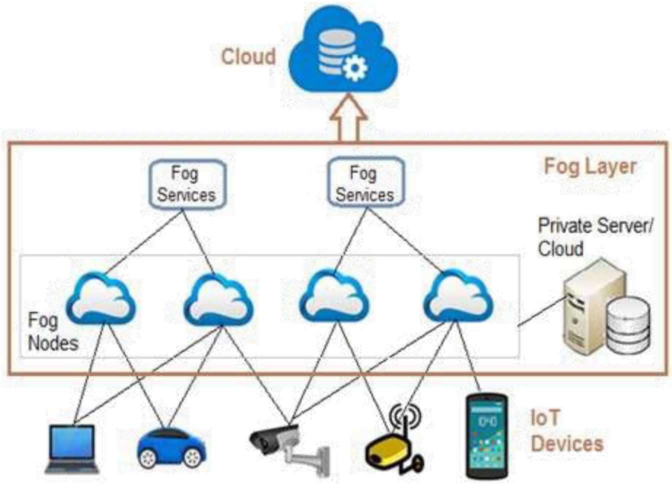
1. **IoT and Data**: IoT devices connect to the internet, generating a lot of data.
2. **Cloud's Role**: The cloud helps transfer data to central servers over the internet.
3. **Centralized Access**: Cloud systems make data and programs easily accessible from one place.
4. **Data Handling**: IoT stores real-time and historical data in the cloud.
5. **Cloud Categories**:
    1. **Platform-as-a-Service (PaaS)**: No need for equipment or software maintenance to build cloud applications.
    2. **Software as a Service (SaaS)**: Access applications via web browsers.
    3. **Infrastructure as a service (IaaS)**: Provides storage, servers, networks, and data processing.
    4. **Public Cloud**: Public network access.
    5. **Private Cloud**: Restricted, often for organizations.

Hybrid Cloud: Integrates both private and public cloud components, enhancing flexibility in computing environments.

II. FOG COMPUTING

Fog computing is like having a helper in your neighborhood for your smart devices.

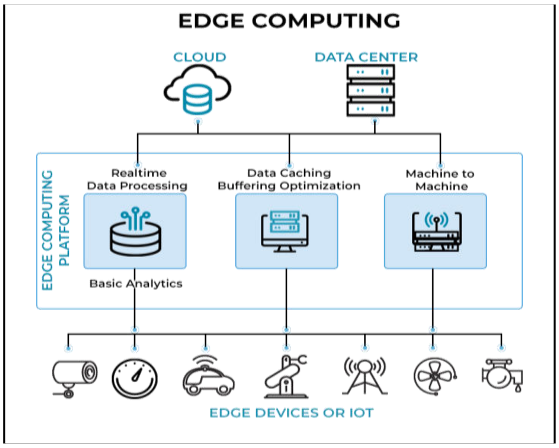
1. **Quick Decisions**: Fog computing helps your smart devices make decisions faster because it processes data nearby instead of sending it far away.
2. **Saves Internet Space**: It also saves internet space because it only sends important data to the faraway cloud, not everything.
3. **Keeps Secrets**: Fog computing can keep your data closer to home, making it more private and secure.
4. **Always Works**: Even if your internet connection is bad, fog computing can still help your smart devices work.
5. **Grows with You**: As you add more smart devices, fog computing can easily grow to help them all.



Fog Computing Architecture

Edge Computing

- Edge computing in IoT means processing data closer to where it's generated, like on IoT devices or nearby servers, instead of sending everything to the cloud.
- This helps make quick decisions, especially in things like self-driving cars and factories.
- It saves internet bandwidth and keeps sensitive data safer.
- Edge computing can grow as needed and even work when not connected to the internet.
- Sometimes, it teams up with cloud computing for the best of both worlds.



IOT Analytics

IoT Data Analytics involves examining data derived from Internet-connected devices or sensors, aiming to extract insights, recognize patterns, and facilitate well-informed decision-making.

How Does IoT Data Analytics Work?

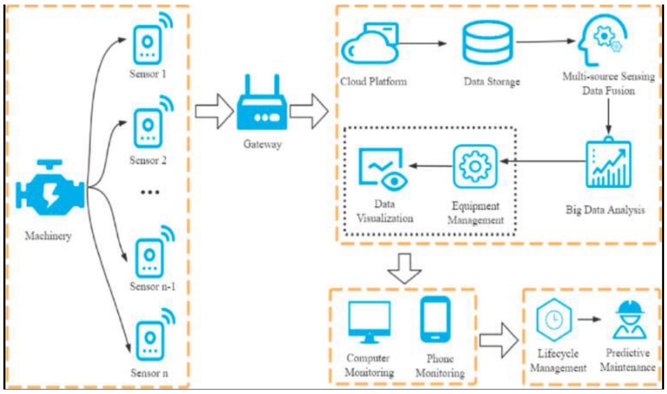
Data Collection: IoT devices have sensors that gather data like temperature, humidity, location, and energy use. This data goes to a cloud system for analysis.

Data Storage: All the data from IoT devices is stored in a big database made for lots of different data types.

Data Processing: After collecting and storing data, we use smart tools and math to find patterns and important stuff in the data.

Data Visualization: The insights we find are shown in easy pictures like graphs and charts. This helps us explain what we learned to others.

Actionable Insights: What we learn from IoT data helps us do better at work. For example, we can predict when things might break, save energy, or make better products.



Purpose of IOT

□ For sensing and collecting the data – sensors & devices
□ Connectivity
□ Data Processing and Analytics
□ User Interfaces and Applications

III. CHARACTERISTICS OF LOT

Dynamic Global network & Self-Adapting : Adapt the changes w.r.t changing context
Self Configuring : Eg. Obtaining the most recent software updates automatically, without the need for manual intervention.
Interoperable Communication Protocols : Communicate through various protocols
Unique Identity : Distinct Identity, such as a unique IP address or a URI, is seamlessly integrated into the information network. This integration enables communication and data exchange with other devices, facilitating specific analyses.
Integrated into Information Network :This allows to communicate and exchange data with other devices to perform certain analysis.

Planning for Your IoT Device

Define the purpose
Clarity about the purpose will serve as a guiding factor for your decision-making process throughout the development phase.

Identify the target audience
Understanding your target audience enables you to customize features and functionalities to align with their specific requirements.

Outline the features
Outline the features will help you the actions that you want the device to perform, and the overall user experience you want to deliver.

Consider scalability
A scalable design ensures that your device can adapt and expand seamlessly in response to changing demands, promoting long-term viability.

Hardware and software requirements
Take into account factors such as power consumption, connectivity options, data storage needs, and processing capabilities.

User interface and interaction
Developing an intuitive and user-friendly interface is crucial for ensuring a gratifying user experience.

Data security and privacy
Implement encryption, authentication, and access control mechanisms to protect sensitive information and maintain user trust.

Choosing the Hardware for Your IoT Device

Cost
Assess the cost-effectiveness of various alternatives and find a equilibrium between quality and affordability.

Sensors
Select the appropriate sensors to collect the necessary data

Connectivity
Examine the available connectivity options, including Wi-Fi, Bluetooth, Zigbee, or cellular networks, and select the one that aligns most effectively with your device's requirements.

Microcontroller/Processor
Select a microcontroller or processor that provides enough processing power, memory, and input/output (I/O) capabilities to meet the requirements of your device.

Power Supply
Consider how your IoT device will be powered. Will it rely on a battery, USB connection, or a traditional power outlet?

Form Factor
Consider the intended use and environment of your device and choose a form factor that is suitable and practical.

Sensors and actuators
Sensors and actuators work together in the Internet of Things (IoT) to capture data and take physical actions. Sensors can detect and measure physical phenomena such as heat, pressure, sight, hearing, touch, taste, and smell. They can also be connected to a network to share data with other devices. An actuator transforms electrical signals into tangible actions, such as force and motion.

IoT Protocols

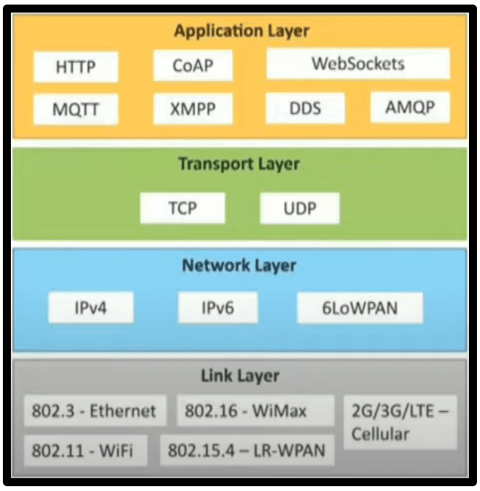
IoT protocols facilitate the transmission of commands and data among a network of devices, which are typically governed by sensors or physical attributes such as motion, temperature, or vibration. Network protocols play a crucial role in ensuring dependable data transfer across various layers, including the application, transport, network, and link layers.

Application Layer Protocols
Establishing Physical Connections between Devices.
Protocols: HTTP, XMPP, WebSocket, DDS, MQTT, AMQP.

Transport Layer Protocols
Retrieve services from the network layer and deliver them to the application layer.
Notable Protocol: TCP/IP for point-to-point communication.

Network Layer
The transfer of data from one host to another, situated in distinct networks.
Protocols: IPv4 and IPv6, used for host identification and data packet transmission.

Link Layer
Controls data transmission over the physical layer, including device signalling and packet encoding.
Significance: Determines how data packets are sent and received on the network.



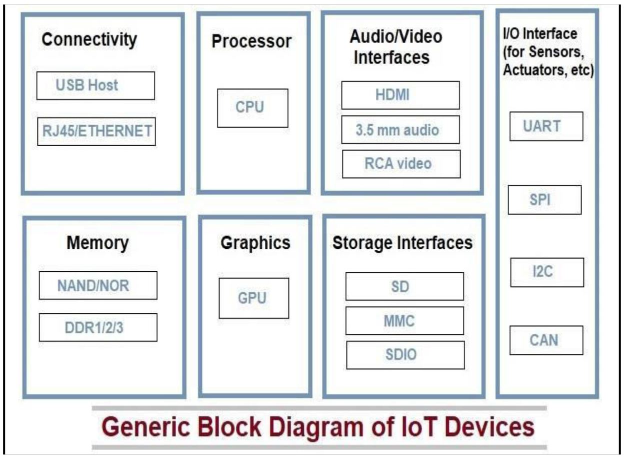
Microcontroller

Microcontrollers are small, low-power devices that are used in IoT projects. They are responsible for executing the computer's instructions

- Small computer for controlling electronic devices.
- Ideal for IoT due to low cost and power efficiency.
- Contains processor, memory, and I/O components.
- Processor executes instructions, memory stores data/programs.
- I/O facilitates communication with sensors/tools.

Components of a Microcontroller:

- Processor for executing instructions.
- Memory for data and program storage.
- Input/Output parts for external communication.



- **Programming Microcontrollers**
  - Use languages like C/C++.
  - Program via USB or serial port connection.

- **IoT Connectivity**
  - Microcontroller connects to the internet.
  - Engineered to accommodate one or multiple network protocols, such as WiFi, Bluetooth, cellular networks like 2G/3G, or even RFID.
  - **IoT Control and Communication**
  - Microcontroller can control devices.
  - Communicates with other devices in the IoT network.

Selecting the optimal microcontroller for your IoT application involves careful consideration of various factors.
Determine Your Application's Requirements:
Define the sensors, actuators, and communication protocols needed.
Consider memory, processing power, and environmental factors.

Consider the MCU Architecture:
Choose from 8, 16, or 32-bit architectures based on project complexity and power/memory needs.

Look at Available MCUs:
Compare MCUs to find one that matches your application's requirements.

Consider Communication Protocols:
Ensure the MCU supports the required wireless, Bluetooth, cellular, or Zigbee protocols.

Evaluate Development Tools and Support:
Check for available development tools and community support for troubleshooting.

Test and Evaluate:
Conduct performance tests to determine the best MCU fit for your IoT application

Raspberry Pi
Raspberry pi is a series of small single-board computers (SBCs) approximately 7 cm x 5.5cm sized developed by the Raspberry pi foundation, based in the U.K.

- Can be used for many tasks that your computer does, like games, word processing, spreadsheets and also to play HD video.
- It gets plugged in a TV or monitor and provides a fully functional computer capability.
- The main purpose of designing the raspberry pi board is, to encourage learning, experimentation and innovation for school level students
- The Raspberry Pi was introduced in 2012, and since its inception, there have been numerous iterations and variations released.
- It also provides a set of general purpose input/output pins allowing you to control electronic components for physical computing and explore the Internet of Things (IOT).
- Numerous generations of Raspberry Pi have been introduced, spanning from Pi 1 to Pi 4.
- There is generally a model A and model B.

The Model A serves as a more economical variant, typically featuring reduced RAM and dual cores, along with functionalities like USB and Ethernet.

Raspberry Pi – Applications in IOT
IoT applications based on Raspberry Pi involve the fusion of AI and machine learning for analyzing IoT data. The future of AI technology lies in processing vast amounts of IoT-generated data, both on the edge and in the cloud, utilizing diverse algorithms and frameworks.

Power Management in IoT Embedded Systems
Efficient power management is essential in designing and implementing IoT embedded systems. Developing a robust power management framework for these intricate technologies can markedly diminish total power consumption, prolong the lifespan of your device, cut down costs, and improve overall performance. With the escalating computing power in IoT devices, it becomes imperative to integrate power management best practices into all embedded software systems.

How to power the IOT Devices
Determine Power Requirements:
Start by understanding the power requirements of your IoT device. This includes the voltage and current it needs to operate. Review the device's datasheet or specifications to find this information.

Consider Power Sources:
Identify the available power sources where your IoT device will be deployed.

Common power sources for IoT devices include:
Battery: If your device is portable or needs to operate in remote areas, consider using batteries. Determine the type and capacity of batteries required based on your device's power consumption and expected operational duration.

Mains Power (AC): If your device is stationary and located near an electrical outlet, you can use mains power. Ensure that the voltage and frequency of the local power grid match your device's requirements.

Solar Panels: In outdoor or remote applications, you may use solar panels to charge batteries or power your device directly.

Harvesting Energy: Some IoT devices can harvest energy from the environment, such as kinetic energy (e.g., from motion), thermal energy (e.g., from temperature differences), or radio frequency (RF) energy.

Power over Ethernet (PoE): If your device has network connectivity, PoE can be a convenient option for providing both data and power over an Ethernet cable.

Calculate Power Consumption:

Estimate or measure the power consumption of your IoT device under different operating modes (e.g., active, sleep, standby). This information is crucial for battery sizing and energy-efficient design.

Select Voltage Regulators:

Depending on your power source and device requirements, you may need voltage regulators to provide a stable and consistent voltage to your IoT device. Step-down (buck) converters are commonly used to reduce voltage, while step-up (boost) converters can increase voltage if needed.

Plan for Energy Efficiency:

Design your IoT device to be energy-efficient. Implement low-power modes and strategies such as duty cycling, sleep modes, and sensor data sampling optimization to reduce power consumption and extend battery life.

Consider Redundancy and Backup:

In critical applications, consider adding redundancy or backup power sources to ensure continuous operation in case of primary power source failure.

Safety and Environmental Factors:

Ensure that your power supply design adheres to safety standards and regulations relevant to your application. Consider environmental conditions such as temperature, humidity, and exposure to dust or water when selecting power components.

Prototyping and Testing:

Prototype your IoT device's power supply system and conduct thorough testing to validate its performance and energy efficiency under real-world conditions.

Monitoring and Maintenance:

Implement remote monitoring and management features in your IoT device to keep track of its power status and perform maintenance or replacements as needed.

Optimize for Longevity:

Design your IoT device with a focus on long-term reliability and energy efficiency to minimize maintenance and replacement costs over time.

Form Factor

Form factor in IoT (Internet of Things) refers to the physical size, shape, and packaging of IoT devices or modules. The choice of form factor is a crucial design consideration as it directly impacts how and where the IoT device can be deployed, its aesthetics, and its ability to blend seamlessly into its environment. Different IoT applications may require varying form factors to meet specific requirements.

Why Form Factor Matters:

Form factor significantly impacts how IoT devices are deployed, their usability, and their integration into various environments.

Common IoT Form Factors:

There are several common form factors for IoT devices:

Enclosures: Traditional boxes or casings made of various materials.
Wearables: Devices designed to be worn, such as smartwatches and fitness trackers.
Miniaturized Modules: Compact surface-mount components or system-on-modules (SoMs).
Sensor Nodes: Compact modules or devices equipped with various sensors.
Plug-and-Play Devices: Devices that can be directly plugged into outlets or USB ports.
Outdoor and Industrial Enclosures: Designed to withstand harsh environments.
Custom Enclosures: Tailored to specific IoT applications.

Factors for Choosing the Right Form Factor:

When selecting a form factor for an IoT device, consider factors like the device's intended use case, the environment it will operate in, its power requirements, and its overall appearance.

Real-World Examples:

To illustrate the importance of form factor, think about the difference between a wall-mounted smart thermostat and a wearable fitness tracker. The form factor directly affects how these devices are used and interacted with.

Form factor is a critical aspect of IoT device design. It impacts not only the physical appearance but also the device's functionality, deployment options, and user experience. Choosing the right form factor is essential for successful IoT solutions.

Cost Considerations in IoT

Budget is an important consideration when choosing hardware components. Assess the cost-effectiveness of various alternatives and find a balance between quality and affordability. Conduct thorough research on different suppliers, comparing prices to ensure optimal value for your investment.

Components of IoT Costs:

IoT project costs typically consist of:

Hardware costs (sensors, devices, modules).
Connectivity costs (data plans, communication protocols).
Software development costs (programming, apps, platforms).
Maintenance and support costs.
Deployment and infrastructure costs.

Balancing Act:

Achieving cost-effectiveness while maintaining quality and functionality is a delicate balance in IoT projects.

Cost Factors to Consider:

When managing IoT costs, consider factors such as:
Scale: The number of devices or sensors deployed.
Data Volume: The amount of data generated and transmitted.
Device Lifespan: Replacement and maintenance costs.
Connectivity Options: Choosing the right network technology.
Security Measures: Costs associated with data protection.
Software Licensing: Fees for using IoT platforms or software.

Cost-Saving Strategies:

Implement cost-saving measures like:
Energy Efficiency: Design devices to minimize power consumption.
Data Optimization: Transmit and process data efficiently.
Open Source Solutions: Leverage open-source software and hardware.
Scalability: Plan for future growth and scalability.

ROI Calculation:

Calculate the return on investment (ROI) to assess the long-term cost-effectiveness of your IoT project.
Managing costs effectively is essential for the success and sustainability of IoT projects. A well-balanced approach ensures that cost considerations align with project goals and desired outcomes.

Library Installation Methods for IoT Development

To install required libraries in IoT development, you typically use your chosen Integrated Development Environment (IDE) or platform-specific package managers. Depending on the functionalities you aim to incorporate into your IoT device, the utilization of external libraries or frameworks may be necessary. These libraries provide pre-written code and functionalities to simplify your IoT project development.

Configure the hardware

Select and Assemble Hardware:

Choose the appropriate microcontroller or development board for your project.

Connect necessary sensors, actuators, and other hardware components to the microcontroller following the datasheets and pinout diagrams.

Ensure proper power supply connections and voltage levels for all components.

Set Up Connectivity:

Configure network connectivity based on your IoT device's requirements.

For wired connections, set up Ethernet or other communication protocols.

For wireless connections, configure Wi-Fi, Bluetooth, cellular, or other wireless modules.

Implement Sensors and Actuators:

Write or install the necessary code or drivers to interface with sensors (e.g., temperature sensors, motion detectors).

Implement control logic for actuators (e.g., motors, relays).

Ensure sensors and actuators are correctly connected and calibrated.

Security and Authentication:

Implement security measures such as encryption and authentication protocols to protect data and device access.

Generate and manage secure authentication keys or certificates.

Data Handling and Processing:

Develop code to capture data from sensors.

Process and format data for transmission or storage.

Implement data filtering and validation as needed.

Connect to the Cloud or Backend:

Configure your IoT device to connect to cloud platforms or backend servers.

Set up communication protocols (e.g., MQTT, HTTP, CoAP) for data transmission.

Power Management:

Implement power-saving mechanisms to optimize energy consumption, especially for battery-powered IoT devices.

Set up sleep modes and power management features when applicable.

Testing and Debugging:

Test your IoT device thoroughly to ensure all hardware components are functioning as expected.

Employ debugging tools and techniques to pinpoint and address issues effectively.

Firmware and Over-the-Air (OTA) Updates:

Implement OTA update functionality if needed, allowing you to remotely update device firmware.

Ensure secure and reliable OTA update processes.

Documentation:

Create documentation detailing the hardware setup, pin configurations, and any specific hardware considerations.

Document the overall architecture and connections for future reference and troubleshooting.

Setting Up Version Control for IoT Projects

Choose the Right Version Control System (VCS):

Opt for a VCS like Git, which is versatile and widely adopted for IoT projects.

Install Git:

Download and install Git from the official website (https://git-scm.com/).

Create a Local Git Repository:

Open your project directory in a terminal or command prompt.

Run git init to initialize a local Git repository.

Add Project Files:

Use git add filename to stage files for tracking.

To add all files, run git add ..

Make Commits:

Commit changes with git commit -m "Your descriptive message" to save a snapshot of your project.

Set Up a Remote Repository:
Establish a remote repository on platforms such as GitHub, GitLab, or Bitbucket.
Link Local and Remote Repositories:
Run git remote add origin repository-url to connect your local repository to the remote.

Push Your Code:
Use git push -u origin master to upload your code to the remote repository.
Collaboration and Branching:
Invite collaborators to the remote repository.
Create branches for feature development or bug fixes.
Merge changes using pull requests or merge requests.
Regularly Pull Updates:
Keep your local repository up to date with git pull origin master.
Issue Tracking and Documentation:
Use issue tracking tools provided by the VCS platform.
Maintain documentation on your project's VCS workflow.

Learn the programming language

- IoT (Internet of Things) is not a programming language itself, but rather a concept that involves connecting physical objects and devices to the internet to collect and exchange data.
- In IoT development, you typically use a combination of programming languages and technologies to create IoT applications and solutions.
- Gain proficiency in the programming language employed for your IoT device.
- Whether it’s C++, Python, JavaScript, or another language, understanding the syntax, functions, and libraries will enable you to write efficient and effective code.

Explore the documentation

- Refer to the documentation provided by the hardware manufacturer, libraries, and frameworks you’re using.
- The documentation comprises valuable information regarding functionalities, APIs, and usage examples, serving as a useful resource to navigate through the development process.
- Explore IoT hardware documentation for microcontrollers like Arduino, Raspberry Pi, and ESP8266 on their official websites.
- Find MQTT and CoAP protocol specifications on their respective websites for IoT communication.
- Access IoT platform and cloud service documentation for AWS IoT, Azure IoT, Google Cloud IoT, and IBM Watson IoT on their respective websites.
- Check out software framework and library documentation for your chosen programming languages and tools.
- Prioritize IoT security by following recommended guidelines from security organizations.
- Join IoT communities, forums, and tutorial websites like GitHub, Stack Overflow, and Hackster.io for valuable resources.
- Refer to sensor and actuator datasheets for technical details and pin configurations.
- Explore IoT standards and documentation from organizations like the IoT Consortium and the Open Connectivity Foundation.

Securing the IoT Device
Secure the Device
IoT security is an extensive concept encompassing strategies, tools, procedures, systems, and techniques employed to safeguard all elements within the Internet of Things. To guarantee the availability, integrity, and confidentiality of IoT ecosystems, protection measures must be applied to physical components, applications, data, and network connections.

Most Common Security Attacks
Spoofing is a cybersecurity term that refers to when a cybercriminal disguises their identity as a trusted source.
Radio jamming refers to the intentional disruption, blocking, or interference with wireless communications.
Node capture is a serious attack on wireless sensor networks (WSNs). In a node capture attack, an intruder physically seizes a node and removes confidential data from it. The attacker can then deploy the node to perform various operations on the network, including:
Compromising the entire network
Transforming data

Cloning and redeploying malicious nodes

A node outage is when a device in a network is unresponsive. This can cause a variety of problems, including making many devices unreachable. A node outage can be caused by a fault or planned maintenance.
Selective forwarding is a type of security attack that can occur in wireless sensor networks (WSNs). In this attack scenario, a malicious node within the network selectively forwards certain data packets to the base station while intentionally dropping others.

A Sybil attack is a type of online security threat where a single entity creates multiple identities to gain control of a network. The attacker uses these identities to influence the network's reputation system and transactions.
The Hello Flood attack is a network layer attack that targets routing protocols. In this attack, an illegal node in the network continuously floods hello requests to legitimate nodes. This breaks security and can cause a Denial of Service (DoS) attack.

Data tampering is the deliberate act of modifying (destroying, manipulating, or editing) data through unauthorized channels.

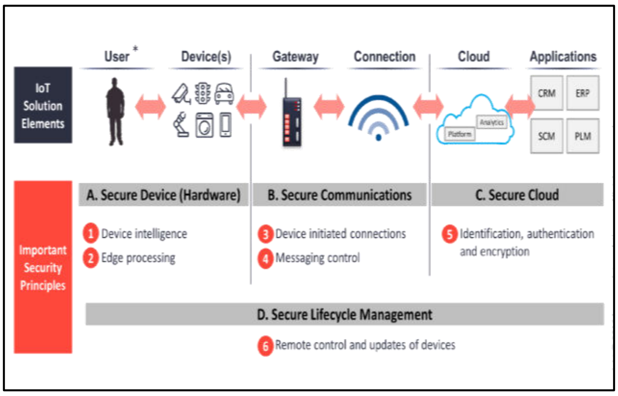
Unauthorized access is when someone gains access to a computer system, network, software, or data without permission. This can include using someone else's account or other methods to gain access.
Keyloggers, also known as keystroke loggers, are tools designed to record the keystrokes made by a person on a device.

Injection is a method employed by attackers to send data to an application in a manner that alters the interpretation of commands sent to an interpreter.
Session hijacking, or TCP session hijacking, involves clandestinely acquiring the session ID to take control of a web user session, posing as the authorized user.

III. CONCLUSION

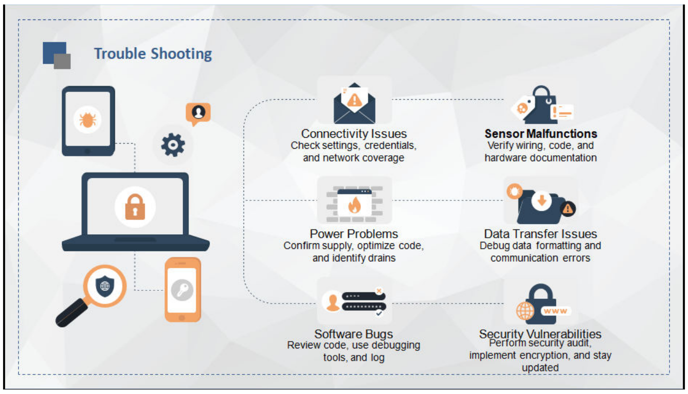
□ Invest in a Security Solution
□ Improve Physical Security
□ Enable MFA Whenever Possible
□ Disable Unused Features
□ Strengthen Settings on Your Devices
□ Change the Default Passwords of Your Devices
□ Keep Your Devices Up to Date
□ Implement Network Segmentation
□ Secure Your Router

Identify the Potential Attack Areas


IoT Solution Elements
User *
Device(s)
Gateway
Connection
Cloud
Applications
CRM
ERP
SCM
PLM
Platform
Analytics
Important Security Principles
A. Secure Device (Hardware)
1 Device intelligence
2 Edge processing
B. Secure Communications
3 Device initiated connections
4 Messaging control
C. Secure Cloud
5 Identification, authentication and encryption
D. Secure Lifecycle Management
6 Remote control and updates of devices

Troubleshooting Common Issues

Troubleshooting common problems is an essential skill to ensure your device runs smoothly and delivers the desired functionality.



Connectivity Issues
Check settings, credentials, and network coverage

Sensor Malfunctions
Verify wiring, code, and hardware documentation

Power Problems
Confirm supply, optimize code, and identify drains

Data Transfer Issues
Debug data formatting and communication errors

Software Bugs
Review code, use debugging tools, and log

Security Vulnerabilities
Perform security audit, implement encryption, and stay updated

REFERENCES

[1] M. H. Miraz, M. Ali, P. S. Excell, and R. Picking, “A Review on Internet of Things (IoT), Internet of Everything (IoE) and Internet of Nano Things (IoNT)”, in 2015 Internet Technologies and Applications (ITA), pp. 219– 224, Sep. 2015, DOI: 10.1109/ITechA.2015.7317398.

[2] P. J. Ryan and R. B. Watson, “Research Challenges for the Internet of Things: What Role Can OR Play?,” Systems, vol. 5, no. 1, pp. 1–34, 2017.

[3] M. Miraz, M. Ali, P. Excell, and R. Picking, “Internet of Nano-Things, Things and Everything: Future Growth Trends”, Future Internet, vol. 10, no. 8, p. 68, 2018, DOI: 10.3390/fi10080068.

[4] E. Borgia, D. G. Gomes, B. Lagesse, R. Lea, and D. Puccinelli, “Special issue on" Internet of Things: Research challenges and Solutions".,” Computer Communications, vol. 89, no. 90, pp. 1–4, 2016.

[5] K. K. Patel, S. M. Patel, et al., “Internet of things IOT: definition, characteristics, architecture, enabling technologies, application future challenges,” International journal of engineering science and computing, vol. 6, no. 5, pp. 6122–6131, 2016.

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

☎ 9940 572 462 ☎ 6381 907 438 ✉ ijircce@gmail.com

www.ijircce.com

Scan to save the contact details