embitel
delivered with passion

HARDWARE DESIGN BEST PRACTICES

For an IoT Product Developer in You!

A Handbook on IoT Hardware Design

All you need to Know before Kickstarting Your IoT Solution design Journey

For a successful IoT Solution Development, our Embedded Product Development Engineers strongly recommend the following, :

a. Clear understanding of the business requirements

b. Putting this Understanding in Action, by inculcating all the best practices right from the hardware design stage.

But how, you may ask? Especially considering the humongous number and scale of IoT devices, data functions that are associated with an IoT solution.

This IoT Hardware Design Handbook has been created to give you a walk through of all the recommended best practices and how to implement them despite the challenges.

Pro-Tip#1

Before starting the hardware design, ensure that you have clear answers for the following :

- What are the types and number of physical signals that have to be measured?
- How frequently should the signals be collected ?
- What is the required rate at which data is to be exchanged over the network?
- How will the sensors be connected to the cloud - directly or via an intermittent gateway device?
- What is the memory space required to manage all the key functionalities?

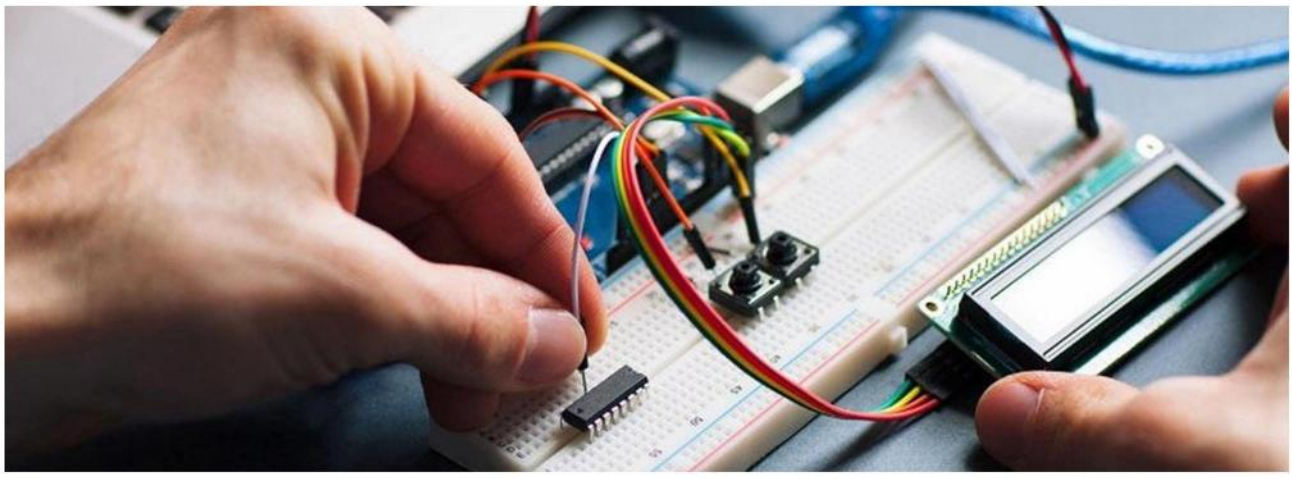


Image Source: Encrypted.com

I. Design Considerations for IoT Sensor Nodes

When we talk about an IoT sensor node, we refer to components like sensor device, microcontroller, power management module , key communication interfaces, and peripherals. While designing the IoT sensor node hardware circuitry, **the following critical design aspects should be duly noted and followed:**

a. IoT Sensor Device:

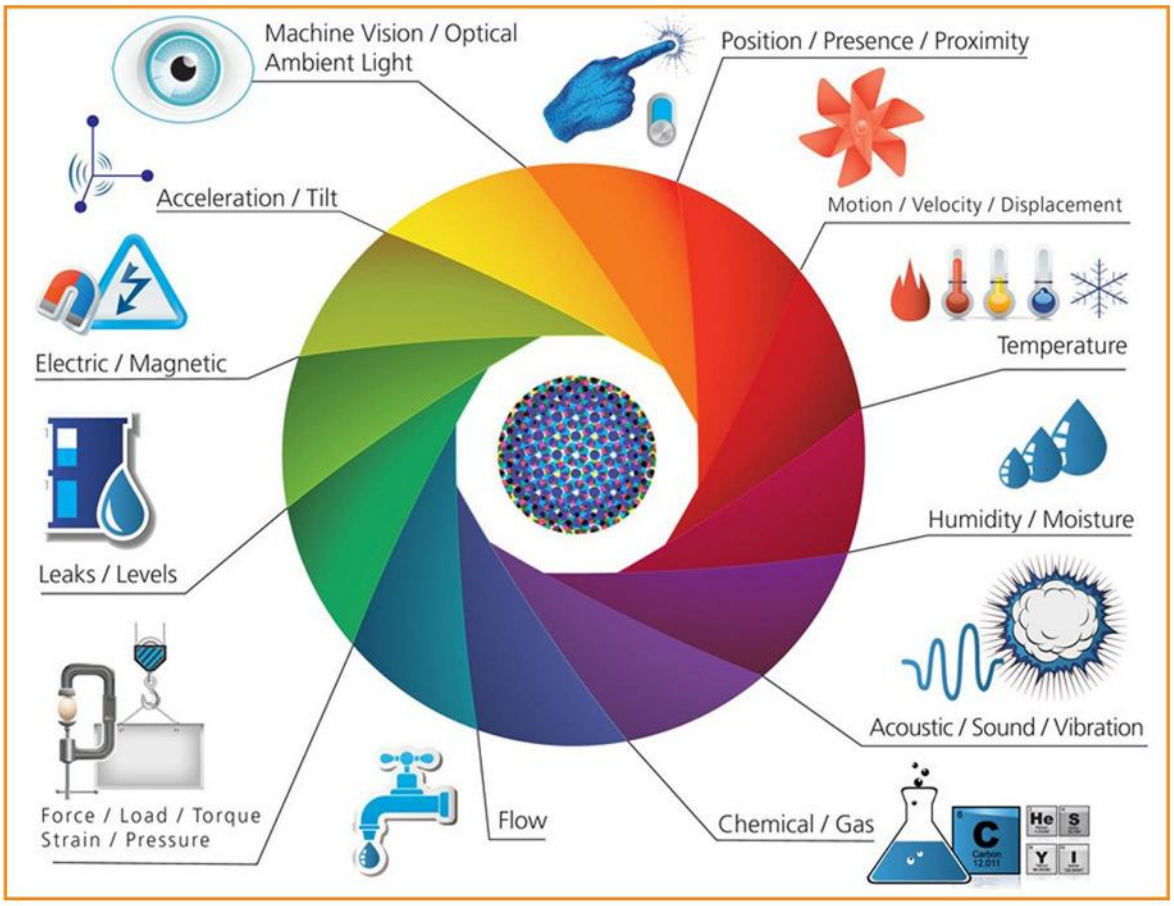


Image: IoT sensors used today are numerous in types ; Source:IoTOne

Evaluation and consequent selection of a suitable IoT sensor will depend on some key factors like:

- IoT Sensor Resolution – The smallest measurement that a sensor can reliably detect.

- Type of IoT Sensor –You may choose analog or digital sensor depending on the project requirement. However, with an analog sensor make sure you use either analogue-to-digital converter (ADC) or Sigma-Delta modulator for signal conditioning & processing.

- IoT Sensor Throughput –Make sure to choose a sensor with sufficient data throughput , which is the measure of speed at which the sensor can transmit data. IoT sensor nodes usually are required to have a throughput ~ 8 to 16 decimals of data in one millionth of second.

b. Microcontroller Hardware Platforms:

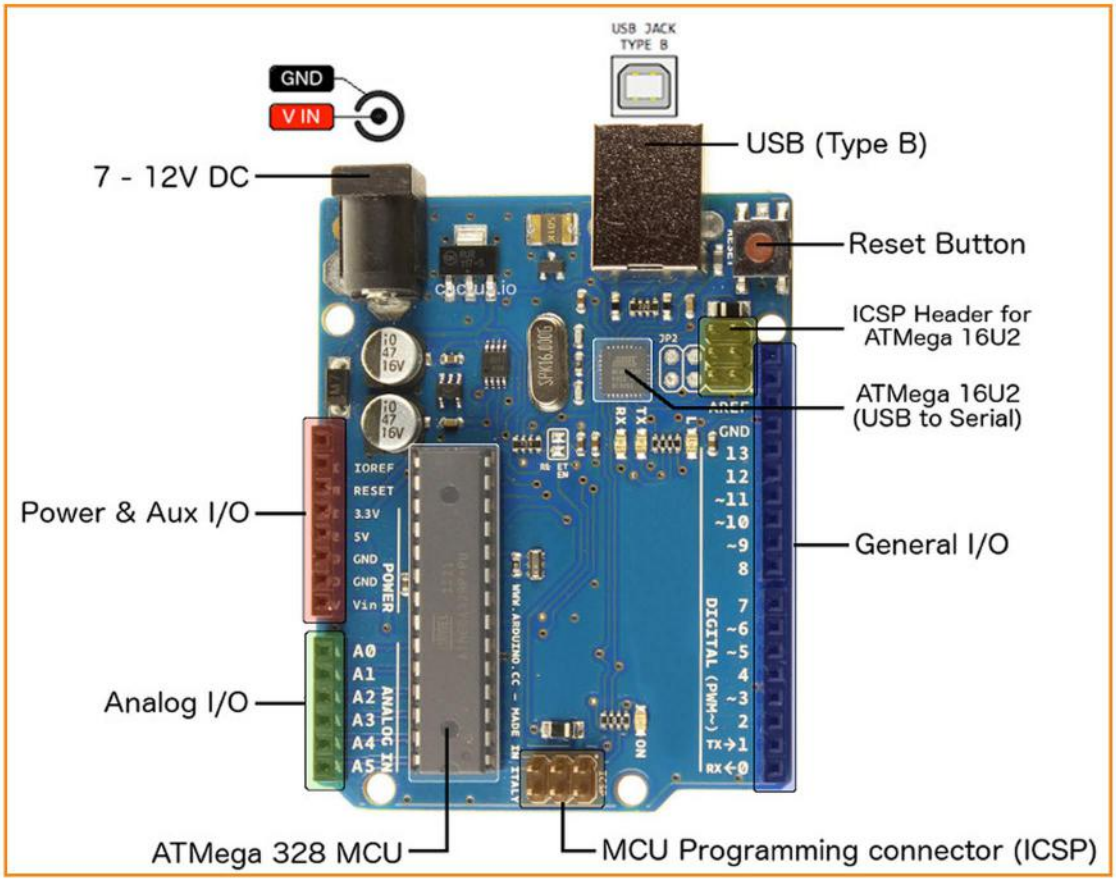


Image: Microcontroller Hardware ; Source Image: pipeworst.hatenablog

Selection of microcontrollers for IoT sensor node hardware design is based on factors such as:

- Integrated Algorithm – The algorithm that the microcontroller uses to process sensor data.
- IoT Data Security – The security mechanism it uses to transmit data securely from sensor nodes to the IoT network.

Pro-Tip#2

Choose a microcontroller with in-built- hardware data encryption engines like Data Encryption Standard (DIS) & Advanced Encryption Standard (AES).

- Power Footprint – Keeping a check on power consumed by microcontrollers can help in optimizing the overall power consumption.

Pro-Tip#3

Use an Ultra-low-power microcontrollers like TI MSP430 or CC2650 .

C. Communication Interfaces:

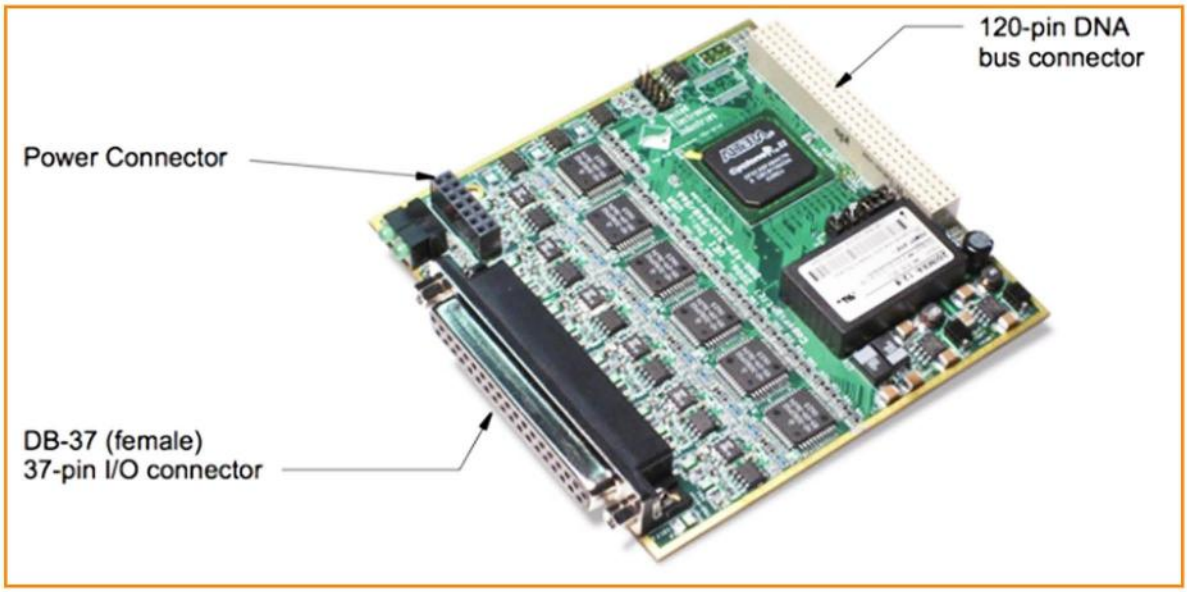


Image: Communication Interface ; Source Image: UEI

For IoT applications, it is advisable to use sensors that are based on low-power wireless communication modems such as NBIoT, LoRA, 4G LTE Cat-M1.

i. Power Management: Power source/ battery capacity of the sensor nodes have a direct relation to their operating life.

Pro-Tip#4

Use low-power batteries, such as CR2032 lithium-ion coin-cell batteries, . They output a constant voltage, are low-cost and have a battery life > 10 years.

ii. Sensor nodes switch between wake-up and sleep modes based on their operating conditions. You can optimize the power consumption by integrating a nano-timer such as TPL5110 with your sensor node circuitry.

II. Hardware Design Considerations for IoT Gateway

Hardware design of an IoT gateway device is closely related to its functional requirements of the IoT application.

You can go for a basic IoT gateway device with core functionalities or alternatively design an edge IoT Gateway device to perform data analysis at the edge.

Let us have a quick look at some critical parameters that can influence your IoT gateway hardware design :

a. Microcontroller OR Microprocessor

Choose microcontroller or a microprocessor for your IoT gateway development, depending on the complexity of functional requirements.

While a microprocessor is useful for advanced functionalities such as data analysis & storage at the edge, a microcontroller is sufficient for small to a medium scale IoT gateway applications.

b. Power Consumption

The complexity of your IoT hardware and the power requirements of other peripheral components will determine your IoT Gateway's power consumption.

C. Type of Processors & OS

If you're deploying an IoT Gateway device with Edge Analytics capabilities, your design should take adequate memory and a multi-core processor into account.

Also, go for an OS with a higher context switching rate and real-time response rate.

d. Memory Footprint

The memory footprint of your IoT gateway device can increase if edge computing is a desired function.

Also, IoT Gateway devices used in time-critical IoT systems are mostly based on real-time controllers and schedulers. This requires data to be stored in RAM.

You can also choose NOR Flash or Flash boot for designing a smart IoT gateway .

e. Communication or Networking Modules

The decision to choose a wired or wireless communication interface to connect the entities within the IoT network, depends on their physical proximity.

In general, the various communication channels commonly used to connect the IoT network components have been summarized in the following table:

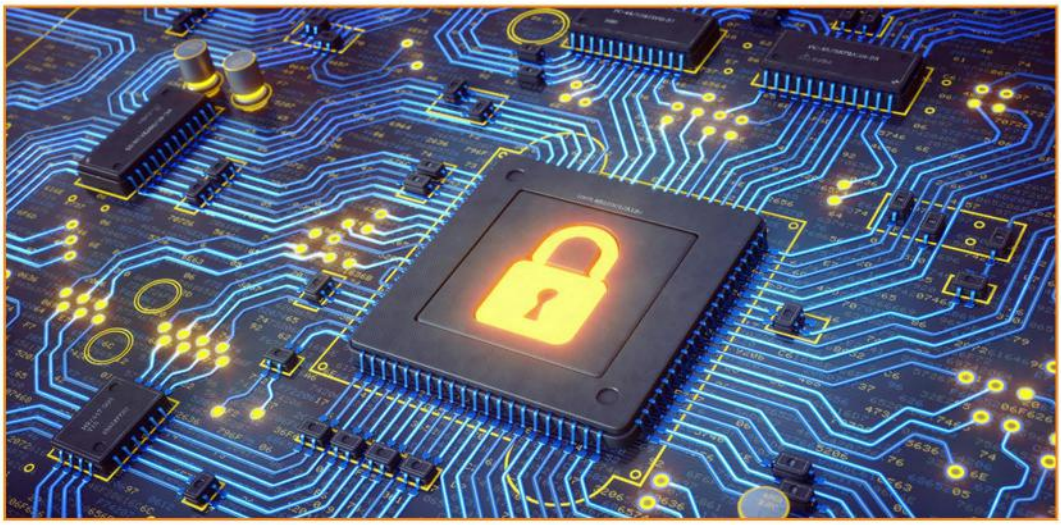
Table 1: Types of wireless communication

Communication Interface	Type of Wireless Communication
Sensor/Device to Gateway	• Cellular(4G), NBIoT, ZigBee, Bluetooth • Applicable for short range communications • Requires minimal hardware installations
Device to Cloud	• WiFi, Cellular(4G), 6LowPan, LoRa • Applicable for short range communications • Requires minimal hardware installations

Table 2: Types of Wired communication Standards

Wired Communication Modes	Features
Fast Ethernet (IEEE 802.3u)	• Offers an average Speed of 100 Mbps • Applicable for long range communications
Gigabit Ethernet(IEEE 802.3z)	• Offers an average Speed of 1000 Mbps • Applicable for long range communications for applications like industrial automation

III. Making IoT Data Security, part of your IoT Hardware Design DNA



To protect the enterprise IoT data against security threats like malware attacks, IP thefts , a foolproof security mechanism must be made an integral part of your IoT hardware design.
Some of the common hardware security mechanisms used today include:

Security Mechanism	Critical Functions
Root Of Trust	• Software Codes hardened into hardware module • Used for data encryption, decryption and creation of digital signatures.
Physically Un-clone-able Function (PUF)	• Generates unique key based on the unique properties of silicon/ semiconductor device. • Used for validating that encryption keys are generated from a secure source
Differential Power Analysis (DPA) Countermeasures	• Hardware devices are secured against DPA Attacks. • Prevents hackers from extracting cryptographic keys from a hardware device

IV. Remote Device Management Through Firmware Over-The-Air (FOTA) Updates

You can enable remote and reliable management of IoT field devices and applications through Firmware Over-The-Air (FOTA) update mechanism.

You can integrate a FOTA module in your IoT Hardware design using either of the following approaches :

a. Custom Build Firmware Over-The-Air Solution : Flexibility to choose Hardware, device interfaces and software to enable Firmware Over-The-Air (FOTA) update. Custom managed scheduling, release, recovery, reporting of firmware images.

b. Off-The-Shelf/Platform Based Firmware Over-The-Air Solution: Managed solution for enabling Firmware Over-The-Air (FOTA).

V. Ensuring Certification and Compliance of your IoT Product

The Product certification and regulatory compliance process should go hand in hand with the IoT hardware design phase.

As an hardware design engineer, you must ensure that every hardware component selected or designed, is compliant with industry recognized regulatory standards and certifications.

Some examples include Restriction of Hazardous Substances (RoHS) Directive, Federal Communications Commission's (FCC) regulations, EMC/EMI compliance .

CONNECT WITH US

INDIA : +91 80 41694200

USA : +1-248-385-2017

GERMANY: +49 152 06 927 221

UK : +49 152 06 927 221

EMAIL : sales@embitel.com

CMMI DEV/3 Exp. 2019-03-30 / Appraisal #26476 · TÜV NORD ISO 9001 · TÜV NORD ISO 27001 · ISO 26262