Internet Of Things

Christian Rossi

Academic Year 2024-2025

Abstract

The course provides an overview of the four main components of IoT systems: sensors, communication technologies, management platforms, and data processing and storage platforms for sensor data. In the first part, the course covers the characteristics of the hardware components of sensor nodes (microcontrollers/microprocessors, memory, sensors, and communication devices). It then delves into communication technologies used in IoT systems, distinguishing between short-range solutions (ZigBee, 6LoWPAN) and long-range solutions (LoRaWAN, NB-IoT). Finally, the course focuses on application-level protocols for IoT systems (COAP, MQTT) and the analysis of IoT management platforms. The course includes hands-on development activities and is delivered through flipped classroom and/or blended learning formats.

Contents

CHAPTER 1

Introduction

1.1 Internet

Definition (*Internet*)**.** The Internet is a global network that connects various types of networks, enabling communication and data exchange.

Traditionally, the internet was primarily used for fixed, stationary clients accessing well-defined services. However, modern internet usage has shifted significantly with the rise of mobile clients. These mobile devices, often equipped with sensing and actuating capabilities, are no longer just consumers of information and services.

Technological advancements Several breakthroughs have paved the way for the rapid growth of the Internet of Things. The miniaturization of hardware, including CMOS technology, microelectromechanical systems, and advancements in materials and circuits, has enabled the development of compact yet powerful smart devices. At the same time, improvements in energy solutions, such as fuel cells and energy harvesting techniques, have enhanced the efficiency and autonomy of these devices. Increased mobility has further expanded the reach and functionality of Internet of Things applications.

In parallel, communication protocols have evolved to support low-power wireless technologies, ensuring efficient and reliable connectivity. The widespread adoption of cloud computing has also played a crucial role, providing scalable architectures and vast processing power. Additionally, the rise of artificial intelligence, particularly deep learning and generative AI, has unlocked new possibilities for intelligent data analysis, automation, and decision-making within Internet of Things ecosystems.

1.1.1 Internet of Things

Definition (*Internet of Things*)**.** The Internet of Things is a worldwide network of uniquely addressable interconnected objects, based on standard communication.

The Internet of Things is based on:

- *Smart objects*: devices embedded with sensors, actuators, and connectivity
- *Data*: continuous collection and processing of information

- *Pervasiveness*: seamless integration into everyday life
- *Seamless communication*: reliable and efficient interaction between devices, networks, and services

The Internet of Things primarily consists of connected low-cost endpoints, such as consumer devices and everyday smart objects, which focus on accessibility and widespread adoption.

1.1.2 Industrial Internet of Things

Definition (*Industrial Internet of Things*)**.** The Industrial Internet of Things refers to a network of interconnected sensors, instruments, and devices integrated with industrial computing applications, including manufacturing, energy management, and automation.

The Industrial Internet of Things consists of connected industrial assets that are typically medium to high-cost. These devices are more expensive but also more responsive, playing a critical role in industrial automation, manufacturing, and energy management.

Cybersecurity is a central concern in the Industrial Internet of Things, where even minor disruptions can have severe consequences. Unlike consumer Internet of Things, Industrial Internet of Things systems must operate with continuous availability, robustness, and resiliency, ensuring that industrial processes remain uninterrupted.

Industrial Internet of Things environments often coexist with a significant amount of legacy operational technologies such as SCADA systems, Programmable Logic Controllers, and Distributed Control Systems. These legacy systems, designed for reliability rather than cybersecurity, introduce additional challenges in securing industrial networks.

While usability and user experience are critical in consumer Internet of Things, they are not primary concerns in Industrial Internet of Things. Instead, the focus is on system integrity, fault tolerance, and maintaining operational continuity in complex industrial ecosystems.

1.1.3 Building blocks

The Internet of Things endpoints require strong security and reliability to ensure they operate safely and effectively within a network. These devices are not just about connectivity; they depend on a combination of smart objects, reliable connectivity, data collection, and advanced analytics to function properly.

The security of Internet of Things endpoints is critical, as these devices often handle sensitive data and are vulnerable to cyber threats. Ensuring reliability ensures that these devices can perform their tasks without interruption, providing accurate data and seamless communication within the system.

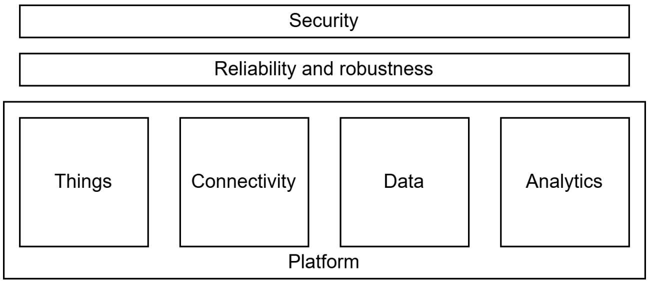

Figure 1.1: Internet of Things building blocks

1.2 Hardware

Definition (*Sensor node*)**.** A sensor node (or mote) is a device with several core capabilities:

- Sensing external phenomena, such as temperature, humidity, or pressure.
- Processing information collected by the sensors.
- Storing the gathered data.
- Communicating with other sensor nodes or external devices.

An actuator performs the following tasks: receiving input signals from control devices, processing and storing information, and acting on the industrial process, executing commands to modify conditions based on the input data.

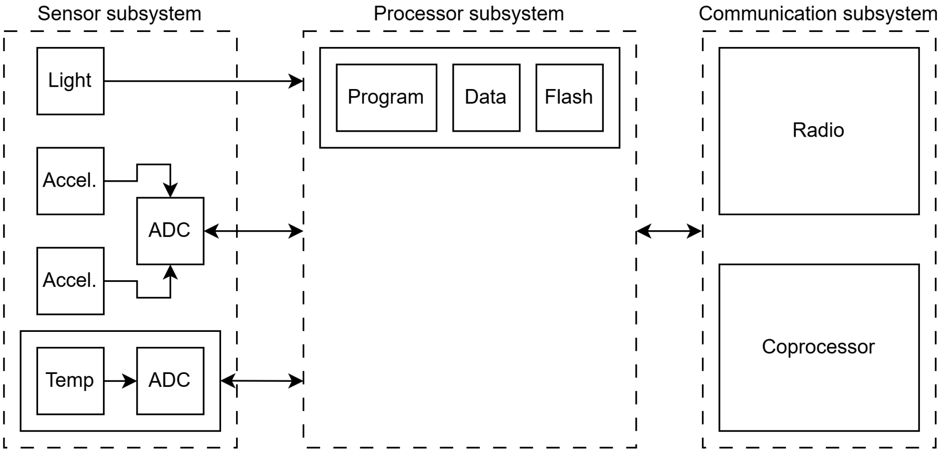

Figure 1.2: Sensor node architecture

1.2.1 Processor

The processor subsystem of a sensor node is often designed based on the SHARC architecture, though there are various alternatives for the individual components. These alternatives offer flexibility in terms of processing power, energy consumption, and other factors critical to specific use cases.

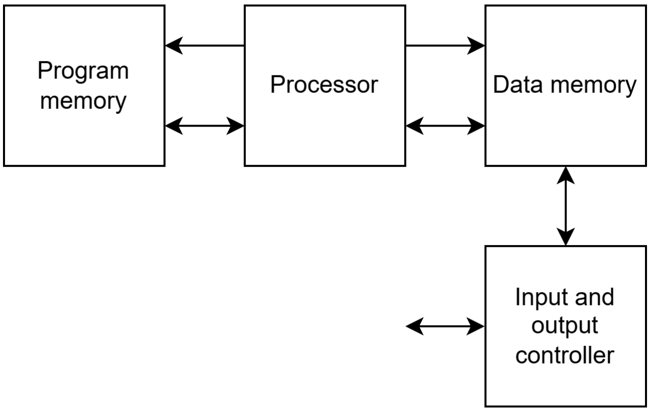

Figure 1.3: Processor architecture

A microcontroller is generally used as the processor in sensor nodes.

Definition (*Microcontroller*)**.** A microcontroller is a single integrated circuit designed for a specific application.

A microcontroller is usually compose by a Central Processing Unit and a clock generator (oscillator with quartz timing crystals). It is usually equipped with RAM, flash memory and an EEPROM. It is connected with a serial BUS, I/O interfaces and analogic and digital converters. While microcontrollers are flexible and low-cost, they can compromise speed in certain use cases.

Definition (*Digital Signal Processor*)**.** A Digital Signal Processor is a specialized microprocessor optimized for processing discrete signals using digital filters.

Digital Signal Processors excel at performing complex mathematical operations with extremely high efficiency. They can process hundreds of millions of samples per second, providing real-time performance. While they are well-suited for data-intensive operations, they are less flexible than microcontrollers.

Definition (*Application Specific Integrated Circuit*)**.** An Application Specific Integrated Circuit is a custom-designed integrated circuit tailored for a specific application.

ASICS offer high speed and can be tailored for specific tasks, but they come with high development costs and limited flexibility once designed.

Definition (*Field Programmable Gate Array*)**.** A Field Programmable Gate Array has a high-level architecture similar to ASICs but allows for some degree of reconfigurability after manufacturing.

Field Programmable Gate Arrays offer high-speed performance, supporting parallel programming, and moderate reconfigurability. However, they are more complex and costly than microcontrollers or Digital Signal Processors.

1.2.2 Sensor

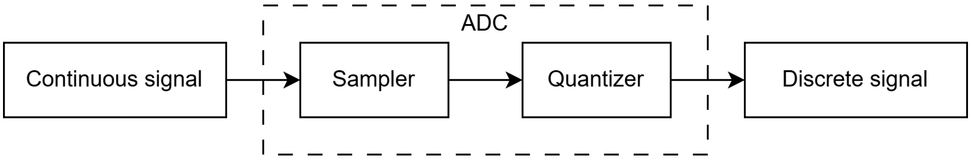

Figure 1.4: Analog to digital converter

Sampling The Nyquist ADC technique involves reading a time-continuous signal at specific points in time. The sampling rate, or bandwidth, is the inverse of the sampling interval:

$$f_s = \frac{1}{T}$$

The key idea is that if the sampling frequency is properly set, the original signal can be losslessly reconstructed from its samples.

Theorem 1.2.1 (Nyquist theorem)**.** *Given the signal bandwith B, we have that the sampling frequency must be chosen as:*

$$f_s = 2B$$

Quantization In quantization, the input voltage V_{in} is approximated by a digital codeword. An ideal quantizer maps input to output with the smallest variation in the input causing a change in the codeword.

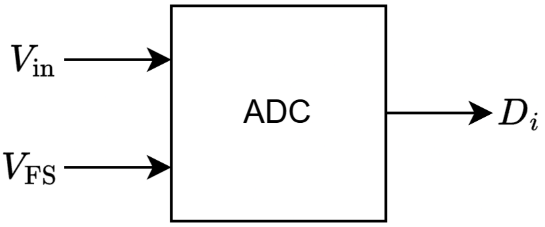

Figure 1.5: Quantization

The resolution refers to the smallest input variation that causes a change in the codeword:

$$\text{LBS} = \frac{V_{\text{FS}}}{2^n}$$

Here, V_{FS} is the full-scale voltage and n is the number of bits of resolution.

Quantization involves discretizing the continuous amplitude of the sampled signal. The quantization error occurs when the output voltage either overestimates or underestimates the input voltage. This error decreases as the resolution increases.

Hardware possibilities The IoT hardware landscape is vast, fragmented, and heterogeneous, with a wide variety of options in terms of CPU types, connectivity, storage, and sensing peripherals. Many end-devices and sensors are now capable of running full operating systems, enabling more complex applications.

There is a clear distinction between the application layer and the hardware control layer, allowing for greater flexibility in how systems are designed and deployed. However, there is also a battle between different types of operating systems in this space, including: commercial RTOS, open-source RTOS, and non-RTOS solutions.

1.2.3 Processor power

The power dissipation of the CPU is due to several factors, including:

$$P_{\text{p}} = P_{\text{dyn}} + P_{\text{sc}} + P_{\text{leak}}$$

Here, P_{dyn} is the power consumed by the work done (dynamic power), P_{sc} is the power lost due to short circuits, and P_{leak} is the power lost due to leakage. The dynamic power consumed during operation is given by:

$$P_{\text{dyn}} = CfV^2$$

Here, C is the capacitance, f is the frequency, and V is the voltage.

Local data processing is crucial in minimizing power consumption, especially in multi-hop networks, where power efficiency is key.

1.2.4 Sensor and actuator power

The absence of cables in wireless sensors and actuators means no wired power or connectivity. This presents unique challenges, especially in terms of energy efficiency, which becomes a must in these systems.

A sensor node typically operates with a limited power source, and its lifetime directly depends on the battery lifetime. The goal is to maximize the energy provided while minimizing the cost, volume, weight, and recharge requirements. However, the problem arises when recharging or battery replacement becomes impractical or too expensive.

There are two main types of batteries used:

- Primary batteries, which are not rechargeable.
- Secondary batteries, which are rechargeable, but only make sense when paired with some form of energy harvesting.

Guidelines To extend battery life, one of the key strategies is to switch off the radio as soon as possible, since radios consume significant power. The power consumption of short-range wireless communication devices remains roughly the same whether the radio is transmitting, receiving, or just idle and listening for potential signals.

Circuit power is primarily dominated by the core components, rather than large amplifiers. The radio must be listening to receive data, even if transmission is infrequent. Listening is often continuous, meaning the total energy consumption is dominated by the power used during idle listening.

Power cycle The power cycle of an IoT device consists of sleep and active states (wake-up/work). During the sleep state, power consumption is minimal, with only essential components running, resulting in some leakage. The average power consumption is then defined as:

$$P_{\text{avg}} = f_{\text{sleep}} P_{\text{sleep}} + f_{\text{wakeup}} P_{\text{wakeup}} + f_{\text{work}} P_{\text{work}}$$

Here, f_{sleep}, f_{wakeup}, and f_{work} are the fractions of time spent in sleep, wake-up, and work states, respectively.

The lifetime of the device is given by:

$$\text{lifetime} = \frac{\text{energy store}}{P_{\text{avg}} - P_{\text{gen}}}$$

Here, P_{gen} is the power generated.

Transmission consumption When data needs to be sent, the device first wakes up and then performs the actual transmission. The total energy consumption for this process is given by:

$$E_{tx} = P_{tx}(T_{wu} + T_{tx}) + P_0 T_{tx}$$

Here, P_{tx} is the power consumed by the transmitter, P_O is the output power of the transmitter,T_{tx} is the time taken to transmit a packet, and T_{wu} is the wake-up time.

Reception consumption When the device needs to receive data, it first wakes up and then performs the reception. The total energy consumption in this case is:

$$E_{tx} = P_{rx}(T_{uw} + T_{rx})$$

Here, P_{rx} is the power consumed by the receiver, T_{rx} is the time taken to receive a packet, and T_{wu} is the wake-up time.

Emitted power The emitted power is often a tunable parameter, and it is generally considered good practice to set it to the lowest value that still allows for reliable reception. The quality of the reception process is typically measured using metrics like:

- *Bit Error Rate*: the fraction of bits that are incorrectly received.
- *Packet Error Rate*: the fraction of packets that are not received correctly.

The relationship between BER and PER, for a packet of length l with independent errors, is given by:

$$\text{PER} = 1 - (1 - \text{BER})^l$$

Both BER and PER are influenced by the level of noise in the transmission and reception channels, which in turn is determined by the transmitted and received power. The Signal-to-Interference-plus-Noise Ratio is a key factor in determining this quality, and is calculated as:

$$\text{SINR} = 10 \log_{10} \left(\frac{P_{\text{recv}}}{N_0 + \sum_{i=1}^{k} I_i} \right)$$

Here, N_0 is the thermal noise (KTB), P_{recv} is the received power, and I_i are the interference contributions from other signals. Given the SINR and the specific modulation of the channel, BER can be computed.

Receiver sensitivity Each receiver is characterized by a sensitivity parameter, which is the minimum input signal power required for the receiver to demodulate the data correctly. Knowing this sensitivity, the required emitted power at the transmitter can be determined by inverting the propagation law of the communication channel.

Sensor power The power consumption due to sensing is highly dependent on the type of sensor used. A rough model for the power consumption of an Analog-to-Digital Converter can be expressed as:

$$P_s \sim f_s 2^n$$

Here, f_s is the sampling frequency, and n is the resolution of the ADC (in bits).

1.2.5 Design guidelines

Avoid full operation all the time. If there's no active task, switch to power-safe modes to preserve battery life.

Use power-aware operating systems that dim displays, enter sleep mode during idle times, and implement power-aware scheduling. Enable radios to forward packets at a lower power level while keeping the rest of the node in a sleep mode. Take advantage of performance-energy trade-offs within the communication subsystem by optimizing neighbor coordination and selecting appropriate modulation schemes.