

Politecnico di Torino

Master Degree course in Electronic Engineering

Master Degree Thesis

Low-Power Firmware Design and Power Consumption Evaluation of a LoRa-Based Node for Critical Control Point Monitoring

Supervisors
Prof. Alessio Carullo
Dott.. Marco Sento

Candidate
Sabina Sarcuni

Academic Year 2025-2026

Acknowledgements

I am sincerely grateful to my professor **Alessio Carullo** and to **Marco Sento** and **Davide Sgró** for their guidance over these months, and for their thoughtful feedback and support in refining and finalizing this work.

Abstract

The evolution of technology plays an important role in the management of risks related to safety, health, and the environment. The current environmental situation is characterized by CO_2 emissions exceeding 38 billion tons per year[1]. According to the World Health Organization, ambient (outdoor) air pollution is estimated to have caused 4.2 million premature deaths worldwide in 2019; almost 99% of the global population is exposed to air pollution levels that put them at increased risk of disease[2]. This evidence highlights the critical role of environmental factors in risk management.

For this reason, it is essential to adopt specific safety and compliance protocols, such as HSE (Health, Safety and Environment) and HACCP (Hazard Analysis and Critical Control Points). Their relevance is closely related to the growing attention to health and sustainability on a global scale. In industrial contexts, the HSE framework enables the management of health, safety, and environmental impacts through the analysis of risk factors and the implementation of mitigation measures. Similarly, the HACCP system introduces a preventive approach based on hazard identification and the monitoring of Critical Control Points (CCPs). These methods are integrated into international standards, forming a multilevel system for effective and continuous safety management.

However, small companies often lack the resources required to monitor all the parameters needed for full compliance with these protocols. To address this limitation, a portable low-power measurement node has been developed at the Laboratorio di Misure e Sensori of Politecnico di Torino. The device is designed to monitor environmental parameters such as temperature, humidity, and gas concentrations (e.g., CO, NO_2, Cl_2), relevant to the identification and monitoring of risk factors, ensuring practical and energy-efficient operation.

The system consists of a main Control Unit (CU) and different interchangeable Measurement Units (MUs). The main unit integrates a LoRa communication module for data transmission, a GNSS module for geolocation, and an STM microcontroller responsible for system coordination.

The thesis focuses on the development of the Control Unit firmware, with particular attention to low-power design strategies to maximize energy efficiency.

The power supply is mainly provided via a DC voltage through a USB-C cable connection, but the device is also equipped with a rechargeable battery for portable applications. When operating on battery power, energy efficiency is crucial to guarantee the longest possible operating time independent of the mains, potentially reaching up to one year of autonomy in a typical use case.

---

[1]Global Carbon Budget: https://globalcarbonbudget.org/fossil-fuel-co2-emissions-hit-record-high-in-2025/

[2]World Health Organization: https://www.who.int/news-room/fact-sheets/detail/ambient-(outdoor)-air-quality-and-health

The main low-power strategy affects LoRa communication between the integrated module and the external server for data transmission. The module must operate in lower power modes, such as Class A and Class B[3], with short receive windows and long latency to avoid wasting energy while waiting for instructions.

The thesis also includes a study of GNSS modules and a characterization of their performance in terms of Time To First Fix (TTFF) in hot, warm, and cold start conditions and power consumption. Different combinations of GNSS modules and antennas were analyzed to identify the best option for the system, representing the best compromise between low power consumption, accurate positioning, and short latency to acquire the first valid coordinates.

[3]Device types in the LoRaWAN protocol, based on different approaches for data exchange, taking into account power consumption and latency

Contents

1 Introduction 5

2 Communication Technologies for Industrial IoT 7
- 2.1 Wired Industrial Architectures . . . 7
  - 2.1.1 Limitations of Wired Architectures . . . 10
- 2.2 Wireless Industrial Architectures . . . 12
  - 2.2.1 Short-Range Wireless Technologies . . . 12
  - 2.2.2 Long-Range Wireless: LPWAN . . . 17
- 2.3 LoRa and LoRaWAN Communication Protocol . . . 20
  - 2.3.1 The LoRa Physical Layer . . . 20
  - 2.3.2 From the MAC Layer Upward . . . 22
  - 2.3.3 Regulatory Constraints in the ISM Band . . . 24

3 Node Design and Low-Power Firmware 27
- 3.1 Measurement Unit . . . 28
- 3.2 Control Unit . . . 28
- 3.3 Low-Power Firmware . . . 29
  - 3.3.1 Power-on Sequence . . . 31
  - 3.3.2 Measurement Unit Discovery . . . 34
  - 3.3.3 Periodic Poll . . . 37
  - 3.3.4 Sleep Policies . . . 38

4 GNSS Module Evaluation 43
- 4.1 The Currently Integrated Module: Teseo-LIV3R . . . 44
- 4.2 Candidate Replacement Modules and Antennas . . . 45
  - 4.2.1 u-blox NEO-6M . . . 45
  - 4.2.2 ATGM336H . . . 45
  - 4.2.3 Antennas and Combinations . . . 45
- 4.3 Methodology . . . 47
  - 4.3.1 Current Consumption Measurement Setup . . . 47
  - 4.3.2 TTFF Measurement Setup . . . 47
  - 4.3.3 Current Uncertainty Estimation . . . 48
  - 4.3.4 Power Consumption Calculation . . . 49
- 4.4 Results . . . 50

4.4.1 Current Consumption 50
4.4.2 Time to First Fix 51
4.4.3 Power Consumption 51

5 Conclusion 53

Bibliography 55

Chapter 1

Introduction

In modern production processes, safety is a critical priority. The increasing complexity of technology introduces new challenges for industrial operations, including chemical and biological risks, accelerated production and distribution cycles, and the rapid digitalization and automation of workflows. These factors significantly affect product quality, worker safety, and consumer health. In this context, post-production quality control is no longer sufficient, as the high speed of manufacturing and distribution can lead to the widespread dissemination of defective products well before any damage assessment can be performed. Consequently, adopting a prevention-based approach is essential to mitigate the risks inherent in modern industrial processes.
Numerous regulations have been established to standardize risk prevention across different areas. Some of the main protocols for industrial applications are outlined below.

HSE (Health, Safety, Environment) Governs the management of workers' health and safety, together with environmental protection [7]. Over time, the concept has evolved into a cross-cutting management model capable of influencing every aspect of corporate life, from organizational policies to individual behavior. It is closely related to the International Standards *UNI EN ISO 45001*, for occupational health and safety management systems [15], and *UNI EN ISO 14001*, which regulates environmental management systems [14].

HACCP (Hazard Analysis and Critical Control Points) Ensures correct food preparation and preservation, preventing contamination by identifying and monitoring specific Critical Control Points (CCPs) throughout the entire production chain. It underpins *Regulation (EC) No 852/2004* [9], together with the *Codex Alimentarius*, a set of international guidelines on the safety and quality of food along production chains [21].

GxP (Good Practices) Regulate the production, distribution, commercialization, and transportation of medications and medical devices [22]. Good Practices guarantee the safety and effectiveness of medications for patients by ensuring compliance with the designated rules. In particular, *GMP (Good Manufacturing Practices)* set out the regulations for production sites and product quality. The *EU GMP Annex 1* states that *"A*

Contamination Control Strategy (CCS) should be implemented across the facility in order to define all critical control points and assess the effectiveness of all the controls (design, procedural, technical and organisational) and monitoring measures employed to manage risks to medicinal product quality and safety" [8]. As in the pharmaceutical industry, it is essential to identify and monitor specific Critical Control Points to comply with regulations and standards.

The three abovementioned protocols protect different fields, but follow the same standard procedure:

1. Risk analysis;
2. Critical Control Point identification;
3. Limit definition;
4. Critical Control Point monitoring;
5. Correction of deviations, where necessary;
6. Data recording and system verification.

Moreover, data recording, especially in the pharmaceutical field, must comply with the *ALCOA+* principle to maintain data integrity and quality: data should be Attributable, Legible, Contemporaneous, Original, and Accurate, as well as Complete, Consistent, Enduring, and Available.
Companies, from small businesses to multinational corporations, must comply with the requirements of the above protocols and regulations. All required data must be safely stored and correctly measured, but manual monitoring is discontinuous, time-consuming, and subject to human error. Paper records are difficult to validate and problematic for data integrity. Moreover, manual monitoring can cause a delayed reaction to deviations due to the significant latency between the anomaly and its human evaluation.
For this reason, automating monitoring procedures represents an excellent alternative, as it provides continuous and precise monitoring, guaranteeing a defined accuracy regardless of human involvement. The automated system can integrate IoT sensors, data acquisition and storage, immediate threshold warnings, and real-time data analysis.
The system described in this thesis, developed at the Laboratorio di Misure e Sensori of Politecnico di Torino, was designed to overcome the abovementioned limitations of manual monitoring through the use of a small, portable monitoring node with interchangeable sensors. It consists of several Measurement Units for data acquisition and a main Control Unit for system coordination, which mainly integrates a long-range communication module for data transmission and a GNSS module for geolocation.
This thesis describes the system's hardware and software components, analyzing the integrated communication technology, the designed low-power firmware, and a final characterization of different low-power alternatives for the GNSS module.

Chapter 2

Communication Technologies for Industrial IoT

Achieving strict compliance with safety, environmental, and quality standards (such as HSE, HACCP, and GxP) requires continuous, real-time monitoring across all critical points in the production chain. To satisfy audit requirements, facilities must deploy dedicated sensor networks to continuously track operational parameters such as temperature, humidity, pressure, and gas levels.

2.1 Wired Industrial Architectures

Industrial communication technologies form the backbone of modern Operational Technology (OT). They allow controllers, field devices, and software systems to exchange data across factory floors and critical infrastructure. *SCADA (Supervisory Control and Data Acquisition)* is a widely adopted architecture for the centralized monitoring of geographically dispersed assets [12]. Supervision and data collection reside at the master level, whereas real-time local control is delegated to *RTUs (Remote Terminal Units)* and *PLCs (Programmable Logic Controllers)* deployed at the field sites [13]. Given their central role in the field layer of this architecture, PLCs and RTUs are examined in more detail in this section, together with the constraints that emerge when their control-oriented design is applied to the fundamentally different requirements of distributed, non-actuating monitoring.

PLC (Programmable Logic Controller) A PLC is a specialized industrial computer designed to execute real-time control logic for manufacturing processes or machinery [19]. It consists of a CPU, a power supply, input/output (I/O) modules (digital/analog), memory, and communication interfaces (RS-485, Ethernet). Its role in the SCADA system is to operate at the field control level in distributed applications such as pipelines, water grids, and power distribution. It reads input signals from sensors and input devices, processes them according to instructions already stored in memory, and sends the corresponding commands to the actuators based on the resulting output values. This is a

closed-loop control methodology, and it is extremely useful for automatically reacting in real time when a problem occurs.

RTU (Remote Terminal Unit) An RTU is a field device that performs data acquisition, local control, and monitoring functions within an industrial automation system [20]. It interfaces directly with sensors and field-level devices, collects and pre-processes their data, and transmits it to a central control system – typically over long-distance communication links – enabling operators to monitor and manage geographically dispersed assets in real time. RTUs collect data from field sensors and forward it to the SCADA master, where it is centralized and made available to operators for real-time monitoring. This division of labor – local acquisition at the RTU, centralized supervision at the SCADA level – is what allows industrial processes spanning geographically dispersed sites to be managed effectively from a single control point.

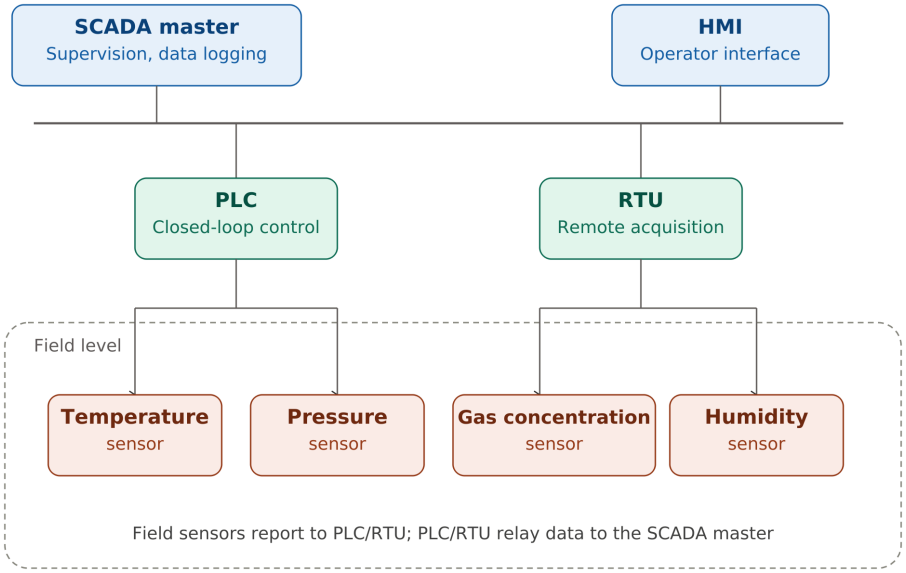


Figure 2.1. SCADA architecture with PLC/RTU field controllers and representative sensors.

While PLCs and RTUs are both core components of industrial automation systems, they are optimized for different tasks. PLCs are designed for local control, typically featuring high I/O density and fast, deterministic scan cycles, and are programmed following the standardized IEC 61131-3 languages (e.g., ladder logic). RTUs, by contrast, are optimized for remote data acquisition and transmission over long-distance communication links, generally at the cost of lower I/O density and reduced control complexity per unit. In short, RTUs are intended to interface with dispersed field devices and relay data over WAN links, whereas PLCs handle real-time control within the boundaries of a single plant.

Feature	PLC	RTU
Primary role	Local, real-time control	Remote data acquisition and transmission
I/O density	High	Comparatively lower
Scan cycle	Fast, deterministic	Not deterministic; polling/event-driven
Programming standard	IEC 61131-3 (e.g., ladder logic)	Vendor-specific; often configuration-based
Communication medium	Local fieldbus / short-range links	Long-distance communication links (WAN)
Typical deployment	Within the boundaries of a single plant	Across geographically dispersed field sites

Table 2.1. Comparison of PLC and RTU characteristics.

This distinction, however, has become less rigid over time: modern PLCs increasingly incorporate remote connectivity features – via Ethernet, cellular modules, or industrial IoT protocols – allowing a degree of remote monitoring and control that was traditionally the domain of RTUs alone.

Beyond the physical distinction between PLCs and RTUs lies a further layer of complexity: the protocol used to exchange data between these devices and the systems that supervise them. Industrial communication protocols are standardized methods that allow field devices – PLCs, sensors, HMIs, drives, robots – and supervisory systems such as SCADA to exchange data accurately and, in most cases, in real time. They form the actual backbone of an automation system, and their design must account for conditions rarely encountered in office or consumer environments: electrical noise from nearby motors and drives, vibration, and, in some sectors, extreme temperatures.
Several protocols have emerged over the past decades, each reflecting a different set of design priorities, and it is worth briefly reviewing the most established ones before turning to their common limitation: all of them, in their traditional form, rely on wired physical media [2].

- **Modbus** is arguably the most widely deployed industrial protocol, largely thanks to its simplicity. Being an open protocol with broad vendor support, it is inexpensive to implement and requires comparatively little integration effort, which explains its popularity in basic data acquisition and monitoring applications, as well as its continued presence in legacy equipment. This simplicity, however, comes at a cost: communication speed is relatively slow, diagnostic capabilities are limited, and the protocol includes no built-in cybersecurity mechanisms.
- **PROFIBUS**, developed primarily around Siemens systems, has served factory

and process automation for decades and remains valued for its reliability and noise immunity in harsh industrial settings, making it well suited to large production lines and remote I/O. Its bandwidth, however, is lower than that offered by Industrial Ethernet alternatives, and both cabling and specialized hardware entail non-negligible installation costs [29].

- **PROFINET** represents the shift toward Industrial Ethernet: it delivers high transmission speeds and deterministic, real-time performance, which makes it suitable for motion control and robotics applications, while also offering more advanced diagnostics than fieldbus-era protocols. This performance comes at a correspondingly higher initial investment and requires networking expertise that is not always available in-house.

- **EtherNet/IP** pursues a similar goal from a different angle, combining standard Ethernet infrastructure with the Common Industrial Protocol to ease integration with enterprise IT systems. Motion synchronization tends to be somewhat less precise than in PROFINET, and, as with any Ethernet-based solution, achieving reliable performance in practice depends heavily on correct network design.

- **OPC UA**, finally, addresses a different layer of the problem altogether. Rather than competing at the field level, it is designed to bridge operational technology and information technology, offering platform independence, structured data models, and – notably, given the security shortcomings observed in Modbus – built-in encryption and authentication. Its processing overhead makes it unsuitable for low-level, real-time machine control, but it has become the reference standard for SCADA, MES, edge computing, and Industrial IoT integration.

It is worth noting, however, that protocol and physical medium are conceptually distinct: the protocols discussed above define how data is structured and exchanged, not necessarily over which medium. OPC UA, for instance, is commonly deployed over standard Ethernet regardless of the underlying network topology, and several of these protocols have since been adapted to run over wireless infrastructures. The association with wired media described here reflects their historical origin and still-dominant deployment base in industrial settings, rather than an intrinsic limitation of the protocols themselves.

2.1.1 Limitations of Wired Architectures

While PLCs and RTUs provide the determinism and robustness required by closed-loop industrial control, the wired infrastructure they rely on imposes a set of structural limitations that become increasingly significant once the objective shifts toward distributed monitoring. These limitations can be grouped into five main categories.

- **Cost and installation**: The cost per measurement point is dominated not by the sensor itself, but by the cabling infrastructure required to connect it: shielded cable, cable trays and conduits, junction boxes, and specialized labor for installation and commissioning. Critically, this cost scales with the physical distance between

Protocol	Speed / bandwidth	Installation cost	Diagnostics	Built-in security	Typical use case
Modbus	Relatively slow	Low	Limited	None	Basic data acquisition; legacy equipment
PROFIBUS	Lower than Industrial Ethernet	Non-negligible (cabling, specialized hardware)	Moderate	Not built-in	Large production lines; remote I/O
PROFINET	High-speed, deterministic	Higher initial investment	Advanced	Not built-in	Motion control; robotics
EtherNet/IP	High (standard Ethernet)	Moderate; depends on network design	Moderate	Not built-in	Enterprise IT integration; Rockwell ecosystems
OPC UA	Lower (higher processing overhead)	Moderate (software-level)	Rich, structured data models	Built-in encryption and authentication	SCADA, MES, edge computing, IIoT integration

Table 2.2. Comparison of industrial communication protocols.

the measurement point and the control system, rather than with the volume or complexity of the data being transmitted – meaning that a single, infrequently sampled variable can be as expensive to wire as a high-frequency one, simply by virtue of its location [29].

- **Scalability**: Adding a new measurement point requires either an available I/O channel on an existing card or the installation of an additional expansion module, which in turn depends on rack capacity planned at design time. As a result, system capacity must be over-provisioned in advance to accommodate future growth, and scalability proceeds in discrete steps rather than continuously – expanding beyond the originally planned capacity often requires a non-trivial hardware upgrade rather than the simple addition of a device.
- **Rigidity**: Relocating a sensor in a wired architecture means rerouting cabling, which penalizes reconfigurable processes and temporary monitoring campaigns, and

makes it effectively impossible to instrument mobile or rotating assets, vehicles, or containers in transit.

- **Barrier to adoption for SMEs**: Beyond the direct cost of cabling, wired architectures typically entail a significant initial capital expenditure, reliance on proprietary toolchains and software licensing, and the need for specialized in-house competence or the involvement of a system integrator – factors that can place this technology out of reach for small and medium enterprises.

Taken together, these limitations show that wired architectures, while well suited to fixed, high-density control applications, are poorly matched to the requirements of distributed, low-frequency monitoring – particularly in environments where cabling itself constitutes a liability rather than a mere cost item. The following section examines wireless alternatives, beginning with short-range technologies commonly adopted in industrial settings, before turning to long-range LPWAN solutions capable of addressing the coverage and deployment constraints identified above.

2.2 Wireless Industrial Architectures

While macro-economic analyses by the FAO and ITU (2020) identify digital infrastructure gaps and high initial capital expenditure (CapEx) as the primary barriers to Industry 4.0 adoption for Small and Medium Enterprises (SMEs) in the agrifood sector [10], automation engineering literature (e.g., MDPI Sensors, 2021) shows that, in operational environments, this hurdle stems directly from the cost of physical cabling, trenching, and conduit routing, which can account for up to 60–70% of total deployment budgets [18]. Consequently, the adoption of low-power, long-range wireless networks emerges as the key enabling solution for making compliance with safety and quality standards economically viable for SMEs.

2.2.1 Short-Range Wireless Technologies

Bluetooth and Bluetooth Mesh

Bluetooth. Bluetooth is a short-range wireless technology standardized under IEEE 802.15, originally developed for point-to-point and point-to-multipoint communication between personal devices [24]. Its low-power variant, **Bluetooth Low Energy (BLE)**, was introduced with Bluetooth v4.0 in 2010, specifically to address applications with strict power constraints, trading raw throughput for significantly reduced energy consumption. Bluetooth 4.2, released in 2014, extended this further by introducing features aimed explicitly at the Internet of Things, including faster data transfer through larger Bluetooth Smart packet capacity [24]. In its classic form, however, Bluetooth follows a *centralized* communication model in which a single "master" device manages the connection with one or more "slave" devices – a topology adequate for simple pairing scenarios but poorly suited to large-scale device networks, since the failure of the master disrupts the entire connection [33].

Bluetooth Mesh. Bluetooth Mesh is built on top of BLE and extends its native communication model from point-to-point or one-to-many to a *many-to-many* pattern, using a mesh topology in which messages are relayed from node to node through managed flooding rather than single-path routing [3]. A device becomes part of the network only after a mandatory process called **provisioning**: a privileged node, the *Provisioner*, authenticates the device, establishes a secure channel using Elliptic-Curve Diffie-Hellman (ECDH) key exchange, and distributes the network credentials (Network Key) in encrypted form [3]. Only after provisioning can a node send or receive mesh messages. The internal architecture is organized around the concepts of **Node**, **Element**, and **Model**: a node represents a physical device, an element is an addressable entity within it, and a model defines its actual functionality (e.g., the on/off switch of a lamp or the reading function of a thermometer) [3].

Bluetooth Mesh enables a robust, integrated, extended-coverage network, allowing multiple devices to be supervised and controlled more efficiently than with classic point-to-point Bluetooth. The theoretical maximum number of connected devices is up to 65,535, and the network is compatible with any BLE-enabled device [3]. Security is layered through three key types (Network Key, Application Key, and Device Key), combined with two-level encryption [3]. In a two-month field deployment reported by Zheng et al. [33], a Bluetooth Mesh-based IoT control system achieved an average command-to-execution latency of about 120 ms, with an error rate below 1%; its decentralized topology also allowed the network to keep functioning even when individual nodes disconnected due to unplanned events – a direct advantage over the centralized structure of classic Bluetooth, where the loss of the single "master" node disrupts the entire system [33].

This coverage improvement, however, operates at the network level, resulting from multi-hop relaying rather than from any increase in the range of a single radio link, which remains bounded by standard BLE limits of typically tens of meters per hop – Bluetooth Mesh extends *network* coverage, not *link-level* range.

Wi-Fi and Wi-Fi Aware

Wi-Fi. Wi-Fi, standardized under the IEEE 802.11 family, is the dominant wireless local area network (WLAN) technology and can also be organized into wireless mesh networks (WMNs) composed of gateways, mesh routers, and clients [24]. Compared with short-range personal-area technologies such as Bluetooth or ZigBee, Wi-Fi offers substantially higher transmission range, data rate, and throughput. This performance, however, comes at the cost of significantly higher power consumption, which has traditionally made conventional Wi-Fi a poor fit for battery-operated sensor nodes: unlike low-power alternatives, a typical Wi-Fi device can operate for only a matter of days without a wired power source, limiting its applicability wherever devices must run unattended for extended periods [33].

Wi-Fi Aware. Wi-Fi Aware, standardized by the Wi-Fi Alliance under the name Neighbor Awareness Networking (NAN), is a low-power service discovery mechanism that runs on top of standard Wi-Fi and follows a "discover before connect" paradigm: devices

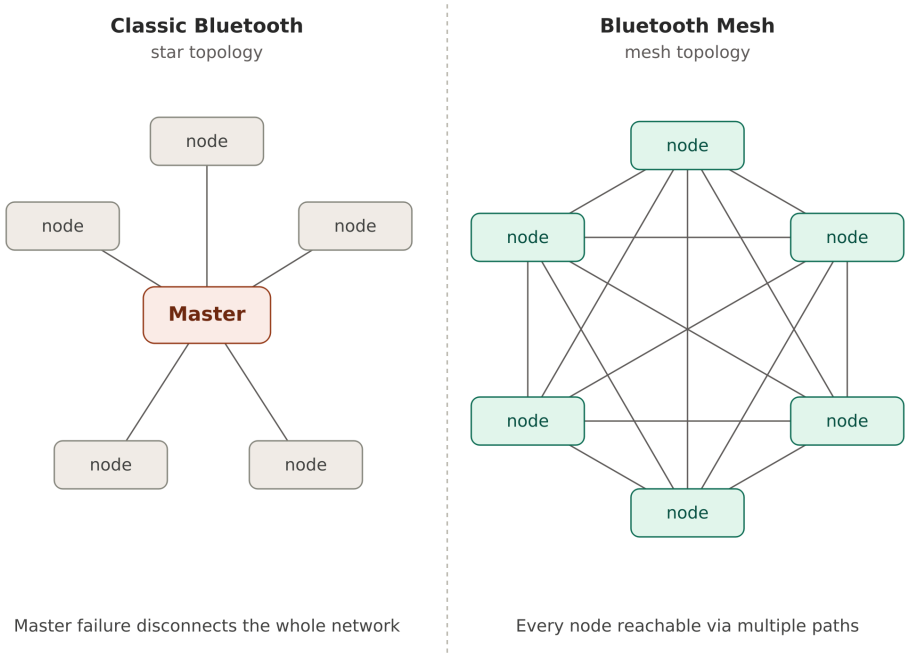


Figure 2.2. Comparison between the star topology of classic Bluetooth and the mesh topology of Bluetooth Mesh.

discover each other's services without first establishing a conventional Wi-Fi connection [23]. Its topology is cluster-based, coordinated by a device elected as the *Anchor Master*. Each device wakes up periodically during a shared **Discovery Window (DW)**, lasting 16 ms every 512 ms, to exchange three types of frames: synchronization beacons, discovery beacons, and service discovery frames [23].
Wi-Fi Aware addresses the core connectivity requirements of IoT applications [23]:

- **Low power consumption**, since the active window for synchronization and discovery is brief relative to the sleep interval;
- **Small memory footprint**, as the mechanism operates without a persistent connection and does not require storing peer device information;
- **Favorable communication range**, since devices operate at the 802.11n/ac layer, which offers better range and data rates than short-range alternatives;
- **Interoperability**, ensured by the Wi-Fi Alliance's certification program [32];
- **Infrastructure reuse**, leveraging existing Wi-Fi infrastructure already ubiquitous in mobile devices.

Conventional Wi-Fi remains a comparatively poor fit for IoT applications due to its higher power requirements. Wi-Fi Aware addresses this by offering a low-power, small-memory-footprint, *medium-range* connectivity solution for IoT platforms – notably *medium*, rather

than long, range, which places it alongside the other short/medium-range technologies discussed in this section rather than among the long-range LPWAN solutions covered later in this chapter [23].

ZigBee

Topology and routing. ZigBee operates above the physical and MAC layers defined by IEEE 802.15.4. The standard defines two topologies: *star* and *mesh* (also referred to as peer-to-peer) [27]. In the mesh topology, a PAN coordinator initiates the network, but the communication paradigm is decentralized: any device can communicate directly with any other device within its radio range, and messages are routed over multiple hops using a protocol similar to Ad hoc On-Demand Distance Vector (AODV) [27]. Route discovery proceeds through the broadcast of Route Request (RREQ) packets and a unicast Route Reply (RREP), with the destination selecting the lowest-cost path based on link quality indication (LQI) [27].

Advantages. The advantages of the mesh topology include robust multi-hop communication, greater network flexibility, and lower latency [27]. The transmission time between nodes is 15 ms. Theoretical network capacity reaches approximately 65,535 devices, owing to the 16-bit network address [27]. Security is based on 128-bit AES, providing four basic services – authentication, message integrity, message confidentiality, and replay protection – coordinated through a **Trust Center** that authenticates new devices and distributes network keys [27].

Coverage and capacity limitations. The disadvantages of the mesh topology include the inability to form superframes, costly route discovery, and the memory overhead required for routing tables [27]. More significantly for this thesis, coverage and capacity are shown to trade off directly against each other: assuming a 100-meter communication range per hop and a total network scan time of 20 s, the mesh network reaches a maximum capacity of 224 nodes, corresponding to a total coverage area of roughly $1.96\,\mathrm{km}^2$ [27]. The analysis further shows that as more nodes are located toward the edge of the network topology rather than near its center, the scan time increases and the effective network capacity decreases – meaning coverage and capacity compete for the same limited scanning-time budget [27].

A further practical limitation, noted in the WirelessHART literature, is that ZigBee, unlike WirelessHART, does not support frequency hopping in its base form: a ZigBee network operates on a single, static channel throughout its entire lifetime, making it more susceptible to noise and interference in harsh industrial radio environments [17]. In response, the ZigBee Alliance released **ZigBee PRO** in October 2007, adding enhanced security features and a "frequency agility" mechanism that allows the network to switch channels in response to degraded link quality – a lighter-weight alternative to true frequency hopping, which was not adopted in order to preserve full compliance with the IEEE 802.15.4 MAC layer [17].

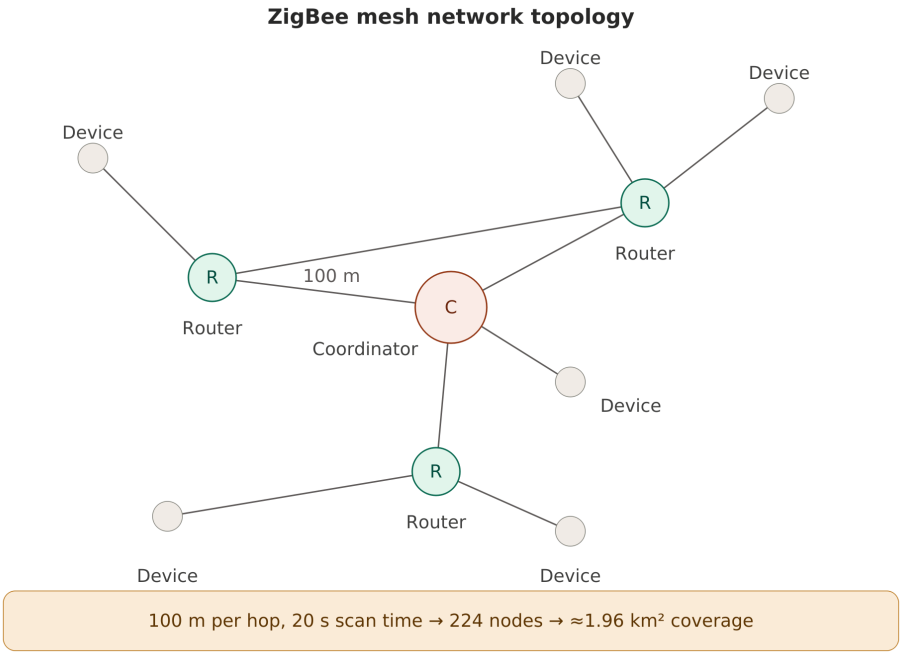


Figure 2.3. ZigBee mesh network topology.

WirelessHART

Architecture. WirelessHART is the wireless interface to the HART field communication protocol, introduced with HART Version 7 in September 2007 and designed specifically for process automation [17]. A WirelessHART network consists of *network devices*, each capable of acting as a source, sink, or router, and supports both star topology (direct connection to the gateway) and mesh topology (multi-hop connections) [17]. The **gateway** bridges the wireless network to the host plant, while a single **network manager** per network is responsible for configuration, routing, and link scheduling; a **security manager** works alongside it to generate session keys and prevent intrusion [17].

Physical and MAC layer. At the physical layer, WirelessHART is based on IEEE 802.15.4-2006, using O-QPSK modulation at 2.4 GHz with data rates of up to 250 kbps [17]. It employs both direct-sequence spread spectrum (DSSS), to resist jamming, and – unlike ZigBee – frequency-hopping spread spectrum (FHSS), which is effective against narrowband interference such as multipath fading [17]. At the MAC layer, WirelessHART uses time division multiple access (TDMA) with 10 ms time slots organized into superframes to ensure contention-free transmission [17].

Advantages. The standard's principal advantage is that it inherits the simplicity and robustness of the wired HART protocol, with an estimated 24 million devices already

in operation, combined with a self-organizing mesh network that requires little or no additional training for plant personnel [17]. Security follows a multi-layered approach covering authentication, integrity, and encryption, with the security level configurable to the needs of the plant. The combination of FHSS, DSSS, retransmission mechanisms, and spatial path diversity through the mesh network ensures reliable transmission even in demanding industrial radio environments [17].

Coverage and energy trade-offs. The literature is explicit about a direct trade-off between topology, coverage, and power consumption: a direct star connection to the gateway can reduce transmission latency, but at the cost of higher power consumption, whereas in the mesh topology latency can be minimized through smart routing, though energy efficiency then depends heavily on the design of the network manager, since routing and link-scheduling decisions directly determine how quickly a device's battery is drained – particularly when that device is repeatedly used to relay data on behalf of other nodes [17]. One energy-aware routing algorithm reviewed in the literature demonstrated a 44% increase in network lifetime compared with routing based purely on minimum energy cost, by occasionally selecting suboptimal paths to avoid rapidly depleting the nodes closest to the gateway [17]. It is also explicitly noted that implementing WirelessHART is more complex than its wired counterpart, requiring careful design of every system aspect to ensure long-term network stability [17].

Summary Comparison

Technology	Radio basis	Reported coverage	Max nodes	Power
Bluetooth Mesh	BLE	Multi-hop (no figure reported)	∼65,535	Low
Wi-Fi Aware (NAN)	802.11n/ac	"Medium range" (qualitative)	–	Higher than BLE/ZigBee
ZigBee Mesh	IEEE 802.15.4	∼10 m/hop; ∼1.96 km^2 @ 224 nodes	∼65,535 (theor.)	Very low
WirelessHART	IEEE 802.15.4 + TDMA/FHSS	Depends on routing/topology	–	Depends on network manager

Table 2.3. Comparison of short-range wireless technologies for industrial monitoring.

2.2.2 Long-Range Wireless: LPWAN

The short-range technologies discussed so far – Bluetooth Mesh, Wi-Fi Aware, ZigBee, and WirelessHART – share a common structural limitation: extending their coverage beyond a few tens or hundreds of meters requires either a dense mesh of relaying nodes

or an equally dense grid of access points, both of which increase deployment cost, infrastructure complexity, and, in the case of mesh relaying, energy consumption at the nodes closest to the gateway. Low-Power Wide-Area Network (LPWAN) technologies address this limitation directly by trading data rate for range: operating in sub-GHz frequency bands and using narrowband or spread-spectrum modulation schemes, LPWAN devices can reach distances measured in kilometers rather than meters, typically at data rates below 100 kbps, while maintaining battery lifetimes on the order of several years to a decade [16]. This combination of long range, low power, and low deployment cost is what makes LPWAN technologies particularly well suited to distributed, low-frequency monitoring applications such as the one addressed in this thesis – precisely the profile identified in Section 2.1.1 as poorly served by wired infrastructure and, as shown above, only partially addressed by short-range wireless alternatives.

This section reviews three representative LPWAN technologies – LTE-M, NB-IoT, and Sigfox – highlighting their respective advantages and limitations, before dedicating the following section to LoRaWAN, the technology adopted in this work.

LTE-M

LTE-M (LTE Cat-M1, also known as eMTC – enhanced Machine Type Communication) is a cellular LPWAN standard built as part of the 4G LTE specification, designed for low-power devices that connect directly to existing mobile network infrastructure [25]. Unlike the unlicensed-spectrum technologies discussed later in this section, LTE-M operates entirely within *licensed* cellular spectrum, inheriting the coverage footprint, reliability guarantees, and mobility support (including handover between cells) of the underlying LTE network.

This reliance on carrier infrastructure is simultaneously its main advantage and its main limitation. On one hand, it provides carrier-grade reliability and, unlike unlicensed technologies, is not exposed to duty-cycle restrictions or interference from other unlicensed-band users. On the other hand, LTE-M devices depend entirely on the deployment footprint and pricing policies of mobile network operators: connectivity requires a subscription and recurring cost, and coverage is only as good as the carrier's own network – a significant drawback in rural or remote deployments, where cellular infrastructure may be sparse or absent altogether. This dependency also introduces a degree of vendor lock-in that is largely avoided by technologies operating in unlicensed bands, such as LoRaWAN.

NB-IoT

Narrowband IoT (NB-IoT) is a licensed-spectrum LPWAN standard developed by 3GPP, designed for applications with limited data-rate requirements [16]. Rather than operating within the full LTE carrier bandwidth, NB-IoT is confined to a narrow 200 kHz channel, which can be deployed within existing GSM spectrum or carved out of LTE base station resources [16]. An NB-IoT network is composed of terminal devices, base stations, a core network, a cloud platform, and a business center, and can support upward of 52,000 connections per cell [16].

NB-IoT's principal strengths lie in its quality of service and latency: because it operates on managed, licensed spectrum, it avoids the contention and interference issues inherent to unlicensed bands, and battery lifetimes of up to ten years are reported despite its cellular-grade signaling overhead [16]. It has been adopted in applications such as smart parking and livestock tracking, where carrier-grade reliability and static or near-static device placement are acceptable trade-offs [16]. However, as with LTE-M, this reliability comes at a structural cost: NB-IoT requires *licensed* spectrum operated by mobile network carriers, meaning that, unlike Sigfox or LoRaWAN, it is neither free to deploy nor operable as a private, carrier-independent network. Coverage is also comparatively more limited than that of the unlicensed alternatives – reported urban coverage figures are on the order of 1 km, substantially lower than either LoRaWAN or Sigfox in the same environment – and handover cannot be performed while the device is in connected mode, with cell reselection available only in idle mode [16].

Sigfox

Sigfox is a proprietary LPWAN technology operating in unlicensed sub-GHz ISM bands (868 MHz in Europe, 915 MHz in North America, 433 MHz in Asia), using an Ultra-Narrowband (UNB) approach: signals are transmitted in an extremely narrow band – on the order of 100 Hz – using DBPSK or GFSK modulation, at a maximum data rate of only 100 bps [16]. Rather than devices connecting to a specific base station, Sigfox follows a star topology in which uplink messages are broadcast and can be received by any base station within range, with all network and computing complexity managed centrally in the Sigfox cloud rather than on the end device [16]. This architectural choice significantly reduces the energy consumption and complexity required of the end node.

Sigfox's principal advantage is coverage combined with extremely low power consumption: two of the sources reviewed here report rural coverage of up to 30–50 km, among the widest of any LPWAN technology considered [11], while a third reports a substantially lower figure of 10 km [16], illustrating how strongly these numbers depend on test conditions, terrain, and deployment configuration rather than reflecting a single standardized value. Its limitations, however, are severe from a data standpoint: payloads are capped at only 12 bytes uplink and 8 bytes downlink, and devices are limited to 140 uplink messages and 4 downlink messages per day, making Sigfox suitable only for extremely sparse, low-frequency telemetry [16]. Security is comparatively weaker by default, with no mandatory end-to-end encryption – an optional AES-128 layer is available at the air interface, but application-level security must be implemented separately [11]. Critically, Sigfox is a subscription-based, single-operator network: unlike LoRaWAN, a private, self-hosted deployment is not possible, and long-term availability depends entirely on the commercial continuity of Sigfox as a company, introducing a form of technological and economic lock-in analogous to that of the cellular LPWAN options discussed above, despite operating in unlicensed spectrum.

Summary

Table 2.2.2 summarizes the main trade-offs among the three technologies discussed in this section [11, 16, 25]. A recurring theme emerges: LTE-M and NB-IoT trade cost and deployment independence for carrier-grade reliability and QoS, while Sigfox trades data capacity and security features for exceptionally low power consumption and long range, at essentially no infrastructure cost to the network operator, but at the cost of dependency on a single commercial provider. None of the three, however, allows for a genuinely private, self-owned network infrastructure with full control over the data path from sensor to server – a requirement that motivates the choice of LoRaWAN, discussed in the following section.

Feature	**LTE-M**	**NB-IoT**	**Sigfox**
Spectrum	Licensed (LTE)	Licensed (LTE/GSM)	Unlicensed ISM
Max. data rate	Higher than NB-IoT/Sigfox (Mbps-class)	Up to 250 kbps	100 bps
Reported range	Carrier-dependent	∼1 km urban / up to 10–25 km rural	10 km urban / 30–50 km rural
Battery life	Multi-year	Up to 10 years	Up to 10 years
Network ownership	Carrier-operated	Carrier-operated	Subscription, single operator
Cost model	Subscription	Subscription	Subscription
Private network	Not possible	Not possible	Not possible
Security	LTE-grade encryption	LTE-grade encryption, mutual authentication	Optional AES-128, no default end-to-end encryption

Table 2.4. Comparison of LTE-M, NB-IoT, and Sigfox.

2.3 LoRa and LoRaWAN Communication Protocol

2.3.1 The LoRa Physical Layer

LoRa (Long Range) is a proprietary spread-spectrum modulation technique, developed and patented by Semtech, derived from Chirp Spread Spectrum (CSS) [1]. Rather than

encoding data as a fixed-frequency carrier, CSS encodes each symbol as a linear frequency sweep, or *chirp*, across the channel bandwidth; the direction and starting point of the sweep carry the information. This encoding gives LoRa its two defining properties: strong robustness to noise and multipath interference, since a chirp's information is distributed across the entire sweep rather than concentrated at a single frequency and can be recovered from the correlation pattern even well below the noise floor, and a large usable communication range, since this same processing gain allows the receiver to demodulate signals at extremely low signal-to-noise ratios [1].

The physical layer exposes three parameters that jointly determine the trade-off between range, data rate, and energy consumption: the **spreading factor** (SF, ranging from 7 to 12), the **bandwidth** (typically 125, 250, or 500 kHz), and the **coding rate** (a forward-error-correction overhead, expressed as a fraction such as 4/5) [1]. The spreading factor is the parameter with the most direct practical consequence: each increment doubles the time needed to transmit a symbol, and therefore the time-on-air of a given payload, while proportionally improving receiver sensitivity and, consequently, achievable range. A device transmitting at SF7 completes its transmission quickly and consumes correspondingly little energy per packet, but requires a stronger received signal to be decoded correctly; a device transmitting at SF12 can be decoded at a far lower signal level – and hence reach much greater distances or penetrate more obstacles – but occupies the channel, and drains its battery, for substantially longer per packet [1].

This trade-off has a direct and compounding effect on energy consumption. A higher spreading factor does not only extend the time-on-air of each packet: by operating closer to the receiver's sensitivity threshold, it also increases the packet error rate, which in turn raises the expected number of retransmissions needed for reliable delivery [1]. The energy cost of a transmission therefore scales with spreading factor along two reinforcing paths at once – a longer time-on-air, and a higher probability that this longer transmission will need to be repeated – which is precisely the mechanism that motivates the Adaptive Data Rate (ADR) feature described in the following subsection, as well as the more general set of low-power firmware strategies developed in Chapter 3.

In practice, this range/energy trade-off translates into communication distances that clearly exceed those of the short-range technologies discussed earlier in this chapter, at data rates within the LPWAN band, typically below 50 kbps. It is worth noting that LoRa, in this narrow physical-layer sense, can in principle be used point-to-point without any of the network-layer machinery described below; it is only once a LoRa radio is placed within a LoRaWAN network that join procedures, device classes, and the regulatory duty-cycle accounting discussed further on come into play.

Typical LoRaWAN deployments in the EU868 band report communication distances on the order of 2–5 km in urban or obstructed environments, extending to 10–15 km or more in open rural terrain [5], figures that place LoRaWAN well beyond the short-range technologies discussed earlier in this chapter and broadly in line with the LPWAN alternatives reviewed in Section 2.2.2.

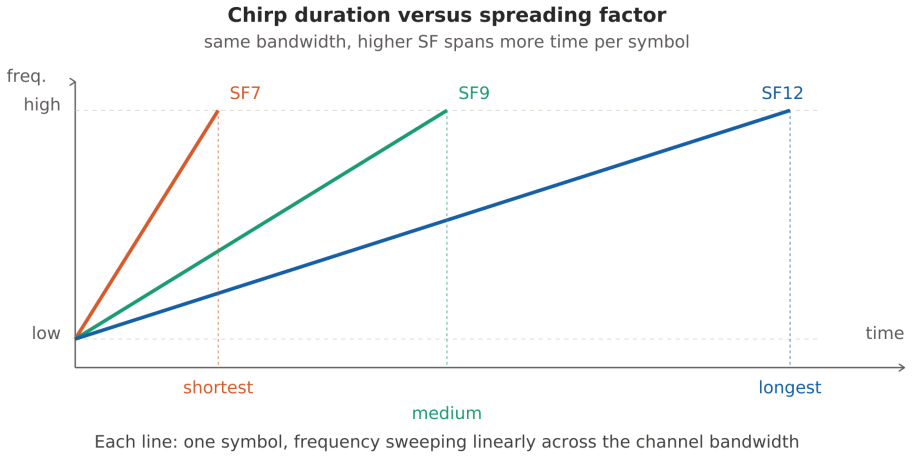


Figure 2.4. Chirp duration versus spreading factor in LoRa modulation.

2.3.2 From the MAC Layer Upward

LoRa refers strictly to the physical layer: a proprietary spread-spectrum modulation technique derived from Chirp Spread Spectrum (CSS), developed by Semtech, that provides the long range and interference robustness discussed in the previous section. LoRaWAN, by contrast, is the open MAC-layer protocol and network architecture standardized by the LoRa Alliance, defining how devices join a network, how transmissions are scheduled, secured, and acknowledged, and how data flows from the end device to the application [30]. The distinction matters because LoRa alone does not define a network: it is LoRaWAN that turns a set of long-range radios into an actual internet-connected system.

A LoRaWAN network follows a *star-of-stars* topology [1]. End devices do not communicate with one another and are not associated with a single base station; instead, each uplink transmission is broadcast and can be received simultaneously by every gateway within range. Gateways perform no processing of their own: they simply forward every received packet, together with metadata such as received signal strength and time of arrival, to a centralized **Network Server** over a backhaul connection (Ethernet, cellular, or Wi-Fi) [1]. The Network Server is responsible for de-duplicating packets received by multiple gateways, managing device sessions, enforcing the network's MAC-layer behavior (including duty-cycle accounting and the Adaptive Data Rate mechanism), and routing decrypted application payloads to an **Application Server** [30]. A separate **Join Server** handles device authentication during network entry. This separation of concerns – radio reception at the gateway, session and protocol logic at the Network Server, application logic at the Application Server – is what allows a single physical gateway deployment to serve many independent applications simultaneously.

Device activation follows the Over-The-Air Activation (OTAA) procedure, the method recommended by the current specification [1]. A device wishing to join transmits a `join-request` message containing its application identifier (`JoinEUI`), its unique device

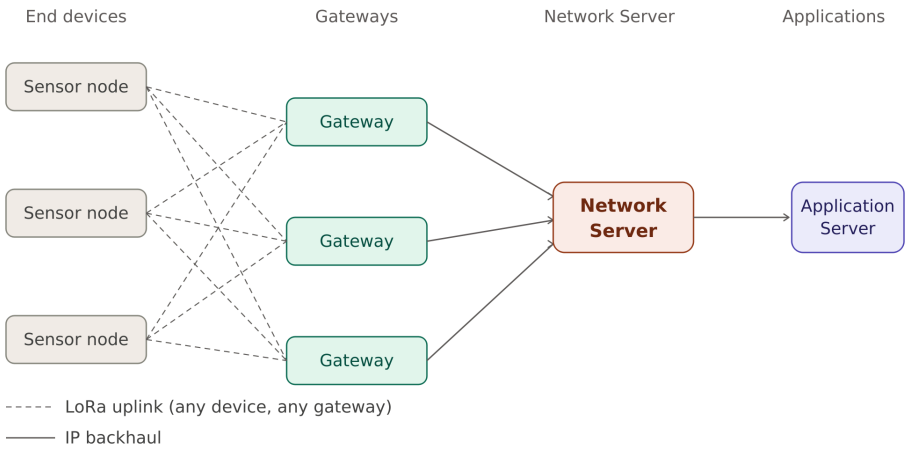


Figure 2.5. LoRaWAN star of stars topology.

identifier (`DevEUI`), and a device-generated nonce (`DevNonce`) used to prevent replay attacks [1]. Provided the request is authenticated by the Join Server, the network responds with a `join-accept` message carrying a network identifier (`NetID`), a session-specific device address (`DevAddr`), and downlink timing parameters [1]. From this exchange, both the device and the network independently derive a Network Session Key and an Application Session Key using AES-128, which respectively protect the integrity of MAC-layer signaling and the confidentiality of the application payload – providing the mutual authentication and end-to-end confidentiality referenced earlier in this chapter's comparison of short-range wireless technologies.

Above the join procedure, LoRaWAN defines three device classes that trade downlink latency for power consumption [30]. **Class A** devices, mandatory for all LoRaWAN end devices, transmit only when they have data to send, opening two short downlink receive windows immediately after each uplink and remaining asleep otherwise; this is the lowest-power class and the one adopted by the vast majority of battery-operated sensor nodes, including the one developed in this thesis. **Class B** devices additionally open receive windows at scheduled intervals, synchronized to a beacon broadcast by the gateway, allowing the network to reach the device with bounded latency at a moderate energy cost. **Class C** devices keep their receive window open continuously, except while transmitting, minimizing downlink latency at the price of a duty-cycle-like continuous power draw that is incompatible with battery operation.

At the MAC layer, LoRaWAN uses an ALOHA-based random access scheme: devices transmit whenever they have data, without a channel-sensing (listen-before-talk) mechanism, coordinating access only through the pseudo-random selection of channel and, indirectly, through Adaptive Data Rate (ADR), by which the network instructs each device to use the lowest spreading factor compatible with a reliable link, reducing airtime and, consequently, both energy consumption and collision probability [1]. This uncoordinated access is simple and requires no network-wide clock synchronization, but scales poorly:

reported measurements show the packet delivery ratio of standard LoRaWAN dropping below 50% once a single gateway serves more than roughly nine hundred devices [1], and simulation results comparing standard LoRaWAN against time-slotted alternatives show collision-driven data loss growing sharply with network density, directly reducing both delivery ratio and device battery lifetime as retransmissions accumulate [1].

2.3.3 Regulatory Constraints in the ISM Band

Because LoRaWAN operates in unlicensed spectrum, its use is governed not by a network operator's spectrum license but directly by regional radio regulations, and the protocol's channel access and duty-cycle enforcement mechanisms exist specifically to keep every device compliant with them. In Europe, the relevant band is 863–870 MHz, regulated under ETSI EN 300 220-2, which subdivides the band into sub-bands with independent duty-cycle limits: the sub-band 868.0–868.6 MHz (used for the three mandatory default LoRaWAN channels at 868.1, 868.3, and 868.5 MHz) is limited to a **1% duty cycle**, meaning a device may occupy that sub-band for no more than 36 seconds in any rolling one-hour period; the sub-band 869.4–869.65 MHz, used by default for the second receive window (RX2) at 869.525 MHz, instead permits a substantially more permissive **10% duty cycle**, which is why network servers preferentially schedule downlink traffic there [30]. As an alternative to a fixed duty cycle, EN 300 220 also allows a Listen-Before-Talk with Adaptive Frequency Agility (LBT+AFA) channel access scheme, though LoRaWAN gateways and network servers in practice almost universally rely on the duty-cycle model rather than on LBT. Regulatory maximum transmit power in the EU868 band is expressed as **Effective Isotropic Radiated Power (EIRP)**, defined as

$$\text{EIRP [dBm]} = P_{TX}\text{[dBm]} - L_{cable}\text{[dB]} + G_{ant}\text{[dBi]}, \tag{2.1}$$

where P_{TX} is the transmitter's conducted output power, L_{cable} accounts for feedline and connector losses, and G_{ant} is the antenna gain referenced to an ideal isotropic radiator. EIRP is capped at **+14 dBm** (25 mW) in the main sub-band.

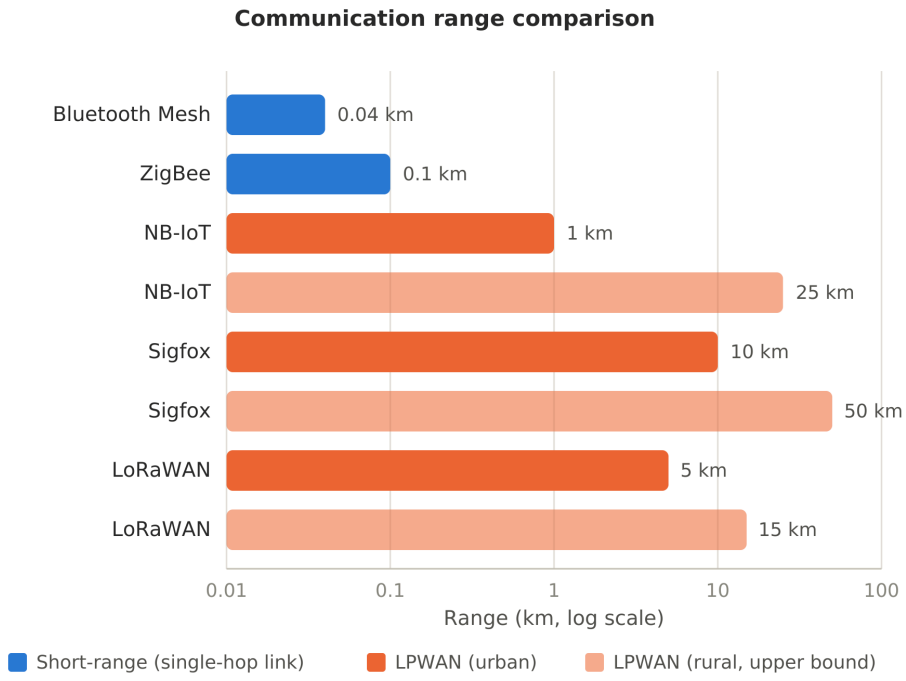


Figure 2.6. Communication range comparison between the main wireless architectures.

Chapter 3

Node Design and Low-Power Firmware

Compliance with the HSE, HACCP, and GxP protocols introduced in Chapter 1 requires continuous monitoring not only within the boundaries of a single production site, but along the entire supply chain – from raw material intake, through processing and storage, to distribution. Critical control points are rarely confined to a fixed plant: temperature-sensitive pharmaceuticals must remain within specification during transport, perishable food must be tracked from the cold room to the delivery vehicle, and environmental conditions must often be logged continuously across sites that share no common infrastructure. A monitoring system limited to stationary points inside a facility therefore covers only part of the problem that these regulations actually require solving.

This requirement exposes, in a particularly direct way, the limitations already discussed in Chapter 2. Wired architectures are structurally unable to address it: a cable can be run across a warehouse, but not onto a truck, a shipping container, or any other asset that is expected to move – the very mobility that transport monitoring requires is precisely what wired infrastructure cannot accommodate, regardless of installation cost. Short-range wireless technologies fare only marginally better. Even where deployment is physically possible, their limited coverage – typically bounded to tens or a few hundred meters, as shown in Section 2.2.1 – means that a node monitoring a vehicle in transit, or a remote point along a distribution route, has no way of relaying its readings to a server that may be located kilometers away, and their comparatively high power consumption makes continuous, unattended operation over the days or weeks a shipment may take impractical. A compliant monitoring architecture, spanning fixed facilities as well as mobile assets, therefore calls for a node capable of long-range, low-power communication – precisely the profile that motivated the adoption of LoRaWAN in Chapter 2 – combined with the flexibility to monitor whatever quantity a given critical control point happens to require, without becoming a different, purpose-built device each time.

This is the design philosophy on which this chapter builds. Continuing a line of research already pursued at the Laboratorio di Misure e Sensori of Politecnico di Torino, where a modular architecture separating a general-purpose Control Unit from interchangeable, quantity-specific Measurement Units (MUs) has been proposed for industrial

wireless sensor networks [4], the device described in this thesis extends that same principle: a single Control Unit (CU) coordinates communication, geolocation, and power management, while the actual sensing task is delegated to physically separate, hot-swappable Measurement Units. Unlike a fixed-sensor node, which must be entirely replaced or duplicated whenever a new quantity needs to be monitored, this architecture allows a single Control Unit to support up to four Measurement Units simultaneously, each carrying a different set of sensors and each independently detachable and replaceable without any modification to the Control Unit hardware or firmware. Compliance with a specific critical control point – whether it calls for temperature, humidity, or gas concentration monitoring – becomes a matter of attaching the appropriate Measurement Unit, rather than deploying an entirely different node.

The remainder of this chapter introduces the concept of interchangeable sensing in more detail, describes the hardware architecture of the Measurement Units and the Control Unit, and then focuses on the Control Unit firmware developed as part of this thesis, with particular attention to the low-power strategies adopted to meet the energy-autonomy requirements outlined in the Introduction.

3.1 Measurement Unit

At present, one model is available for the specific CU discussed in this thesis: Model 0x0001. It integrates four different sensors:

- LSM6DSMTR accelerometer
- MS5837 pressure sensor
- HPP845E031R4 relative humidity sensor
- NTCLP450E3104H NTC thermistor, 100 kΩ

It also integrates an STM32WB30CE microcontroller and an IS25LP032D (32 MB) flash memory for measurement data storage. More MU models, integrating gas concentration sensors for CO, NO_2, and Cl_2, are currently in the design phase and will be available in the future.

Each MU is identified by a 4-byte Extended ID (ExtID): 2 bytes identify the model, and the other 2 bytes uniquely identify the specific device.

3.2 Control Unit

The Control Unit is responsible for coordinating the entire system and acts as the gateway between the MUs and the Network Server. It consists of:

- **STM32WB55RGV6 microcontroller**, an ultra-low-power MCU;
- **Wio-E5 LoRa communication module**, used for data transmission. It integrates an STM32WLE5JC microcontroller, which embeds a powerful and ultra-low-power LPWAN-compliant radio solution. It communicates with the STM32WB55RGV6

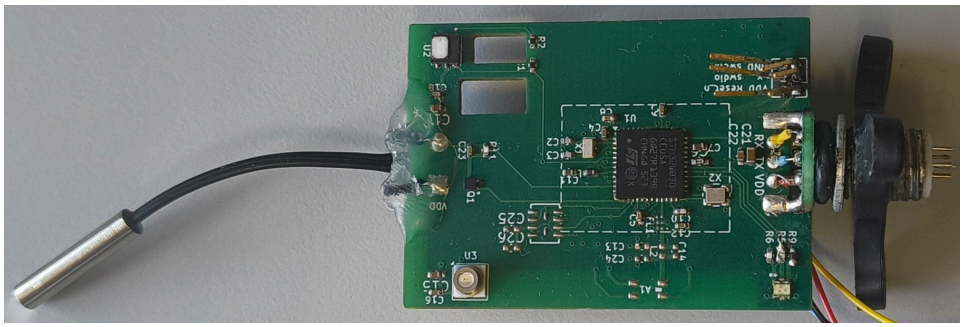


Figure 3.1. Photograph of a Measurement Unit – Model 0x0001.

microcontroller via LPUART (8N1 encoding), a low-power UART interface, allowing communication even during the MCU's low-power modes;

- **Teseo-LIV3R GNSS module**, used for geolocation. It communicates with the microcontroller via I2C.

together with a buzzer and an RGB LED for feedback, and various peripherals, such as a tactile button, a USB-C interface, four custom UART interfaces (MSUART), and others for specific purposes described below.

Communication between the CU and the connected MUs takes place over a custom UART interface, called MSUART, with 8N1 encoding. The four MSUART lines are connected to the microcontroller's UART pins through a common UART bus. Communication is always initiated by the CU, which notifies the target MU by specifying the corresponding ExtID in the command; in this way, only the selected MU replies, avoiding collisions. It is important to note that the ExtID would occupy too many bits of the LoRa payload if used to report measurement results directly. To minimize this limitation, the low-power strategy adopted replaces the long ExtID with a 1-byte Local ID (LID), yielding a shorter uplink size and, consequently, lower power consumption during data transmission. The correspondence between the ExtID and the LID of each connected MU is assigned by the CU during the discovery phase. This strategy is discussed in detail in the following section of this chapter. MU detection relies on a set of four voltage amplifiers, each connected to the interface between the CU and one MU. When an MU is plugged in, the corresponding amplifier detects a voltage gain and notifies the microcontroller through its dedicated pin.

Power for the entire device (CU and MUs) is supplied via a USB-C cable connected to the CU. Since the device is designed to be portable, it must also be able to run on battery power. The power supply management unit handles this dual capability, notifying the microcontroller which of the two sources is in use through dedicated pins.

3.3 Low-Power Firmware

The firmware running on the Control Unit is built around a single architectural principle: nothing in the system is allowed to block. Every activity that would traditionally be

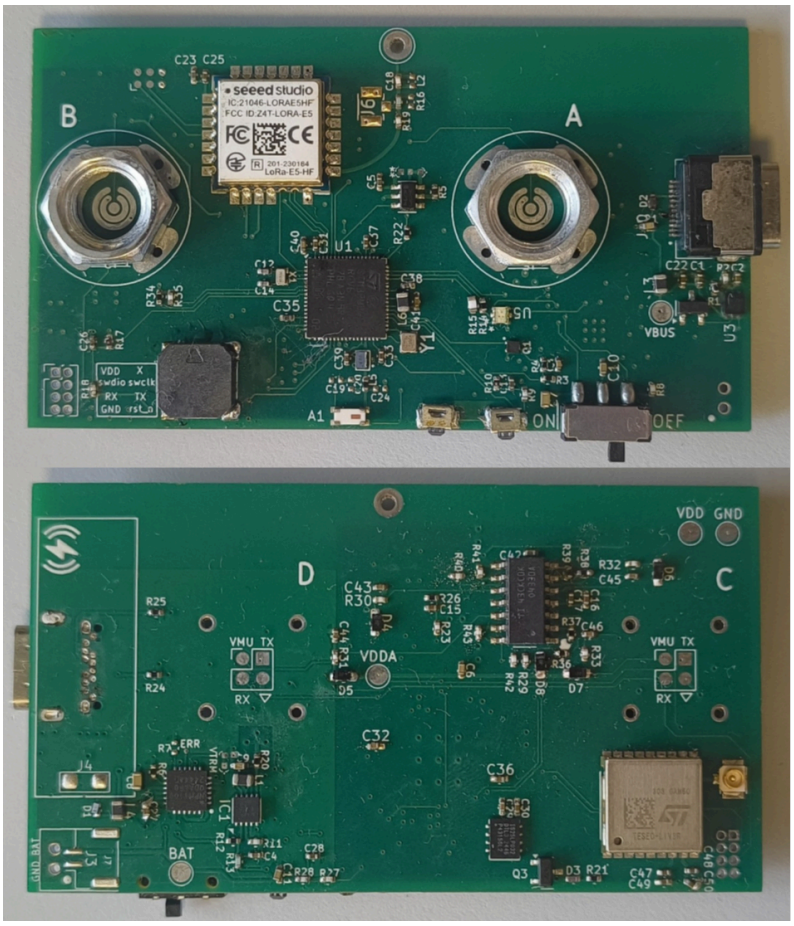


Figure 3.2. Photograph of a Control Unit prototype.

implemented as a blocking wait – polling a peripheral until it responds, looping until a flag is set, waiting for a transmission to complete – is instead modeled as a state machine that advances by exactly one step per scheduler tick and immediately returns control. This constraint is not a stylistic preference; it is what makes aggressive low-power operation possible at all. A microcontroller cannot enter a deep-sleep mode while it is spinning in a wait loop, so eliminating blocking code is the precondition for every power-saving strategy described later in this chapter, rather than one strategy among others.

At the center of this architecture is a cooperative, single-threaded scheduler operating on top of a minimal tick-driven main loop: the `main` function itself contains no application logic beyond a single call to the scheduler on every tick. At each tick, the scheduler performs a fixed sequence of operations – updating the elapsed time, translating hardware events into pending requests, selecting which task should run, advancing that task's state machine by one step, advancing any pending feedback pattern, and finally deciding whether and how to sleep before the next tick. Tasks are ordered by priority, with

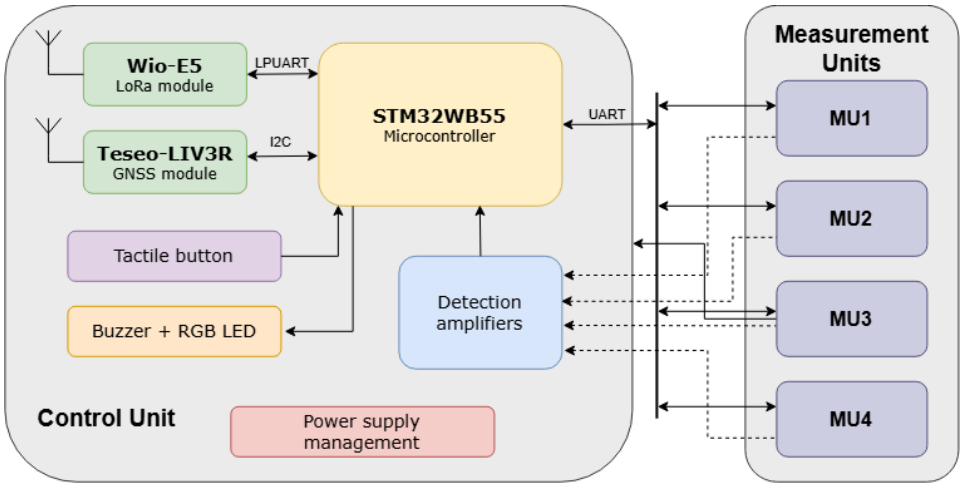


Figure 3.3. Block diagram of the system.

Measurement Unit handshaking taking precedence over notifying the network server, which in turn takes precedence over the periodic poll; only one primary activity is ever active at a time, alongside lightweight secondary activities such as LED or buzzer patterns that do not compete for the scheduler's main slot. Because a state machine's progress depends on state that must persist not only across ticks but across sleep cycles – the current step, the number of retry attempts, the index of the Measurement Unit being processed, the timestamp at which the current step was entered – this state is deliberately kept outside the call stack, so that it survives both the return from a function call and a wake-up from deep sleep.

This chapter describes the firmware built on top of this scheduling model, from the initial power-on sequence and LoRaWAN network join, through Measurement Unit discovery and periodic reporting, to the sleep policy that ties these activities to the device's energy budget. Particular attention is given to how each mechanism was specifically designed to minimize energy consumption.

3.3.1 Power-on Sequence

Right after device power-on, the first steps to be executed are the initialization of the RGB LED and buzzer, and the battery level measurement. The buzzer plays a melody, and the blue and red LEDs blink to notify the user that the device has been powered on. After these initial actions, the LoRa configuration and join sequence is executed. The theoretical background behind these steps is discussed in detail in Section 2.3.2.

Join LoRaWAN

After a preliminary hardware reset of the Wio-E5, the first communication attempt between the MCU and the LoRa module is performed. To verify that the module connection is working correctly, the MCU sends a dummy AT command. With a maximum of three

attempts, if the LoRa module replies correctly, as shown below, the sequence continues; otherwise, it exits the join procedure and the program continues.

```
[Tx]:  AT
[Rx]:  +AT: OK
```

The module configuration is the next step. First, the MCU reads the DevEUI and DevAddr parameters from the Wio-E5 using the following commands:

```
[Tx]:  AT+ ID= DevEui
[Rx]:  +ID: DevEui, xx:xx:xx:xx:xx:xx:xx:xx

[Tx]:  AT+ ID= DevAddr
[Rx]:  +ID: DevAddr, xx:xx:xx:xx
```

These two parameters are specific to each device, so they must be read at least once and then saved to flash memory. After the first power-on, the program can retrieve their values from memory, avoiding the two transmissions described above. Next, LoRaWAN configuration parameters such as Region, Class, ADR, and others must be set using the dedicated commands:

```
[Tx]:  AT+ ID= AppEui
[Rx]:  +ID: AppEui, xx:xx:xx:xx:xx:xx:xx:xx

[Tx]:  e.g.  AT+KEY=NWKSKEY, "2B7E151628AED2A6ABF7158809CF4F3C"
[Rx]:  +KEY: NWKSKEY 2B7E151628AED2A6ABF7158809CF4F3C

[Tx]:  e.g.  AT+KEY=APPSKEY, "2B7E151628AED2A6ABF7158809CF4F3C"
[Rx]:  +KEY: APPSKEY 2B7E151628AED2A6ABF7158809CF4F3C

[Tx]:  e.g.  AT+KEY=APPKEY, "2B7E151628AED2A6ABF7158809CF4F3C"
[Rx]:  +KEY: APPKEY 2B7E151628AED2A6ABF7158809CF4F3C

[Tx]:  AT+DELAY=RX1, 1000 // Unit:  ms
[Rx]:  +DELAY RX1, 1000

[Tx]:  AT+DELAY=RX2, 2000 // Unit:  ms
[Rx]:  +DELAY RX2, 2000

[Tx]:  AT+ DR= EU868 // Region: Europe (868 MHz)
[Rx]:  +DR: EU868

[Tx]:  AT+MODE= LWOTAA // Over-The-Air-Activation
[Rx]:  +MODE: LWOTAA
```

```
[Tx]:  AT+CLASS=A
[Rx]:  +CLASS: A

[Tx]:  AT+ADR=ON
[Rx]:  +ADR: ON
```

The last step consists of the actual Join request to join the known Network Server:

```
[Tx]:  AT+JOIN
```

If the join completes successfully, the following set of return messages is received by the MCU, and the Wio-E5 is finally put into extremely low-power mode:

```
[Rx]:  +JOIN: Starting
[Rx]:  +JOIN: NORMAL
[Rx]:  +JOIN: NetID 000024 DevAddr 48:00:00:01
[Rx]:  +JOIN: Done

[Tx]:  AT+LOWPOWER=AUTOON
[Rx]:  +LOWPOWER: SLEEP
```

If, instead, the join fails:

```
[Rx]:  +JOIN: Join failed
```

or if the join process is still ongoing:

```
[Rx]:  +JOIN: LoRaWAN modem is busy
```

In this mode, the modem enters deep sleep whenever it is idle – that is, when it has no commands to receive and no LoRaWAN service pending. With this mode enabled, every AT command sent to the modem must be preceded by at least four `0xFF` bytes, and every response from the modem is likewise prefixed with four `0xFF` bytes. To turn off low-power auto mode using the `AT+LOWPOWER=AUTOOFF` command, four 0xFF bytes must also be added at the start of the command. The described procedure is structured by

0xFF	0xFF	0xFF	0xFF	A	T	+	L	O	W	P	O	W	E	R	=	A	U	T	O	O	F	F	\r	\n

Table 3.1. Characters to transmit to disable the Low Power Auto Mode.

dividing commands into blocks and using the `lora_status` flag: for each command, if the correct return message is not received, the flag is set. At the end of the block, if the flag is still clear, the next block can be executed; otherwise, the join procedure is stopped and the program continues with the following phase. Indeed, the device must also be able to operate without radio availability: in that case, the measurements are simply stored in memory and extracted at a later time. The entire flow of this first phase is shown in Figure 3.4.

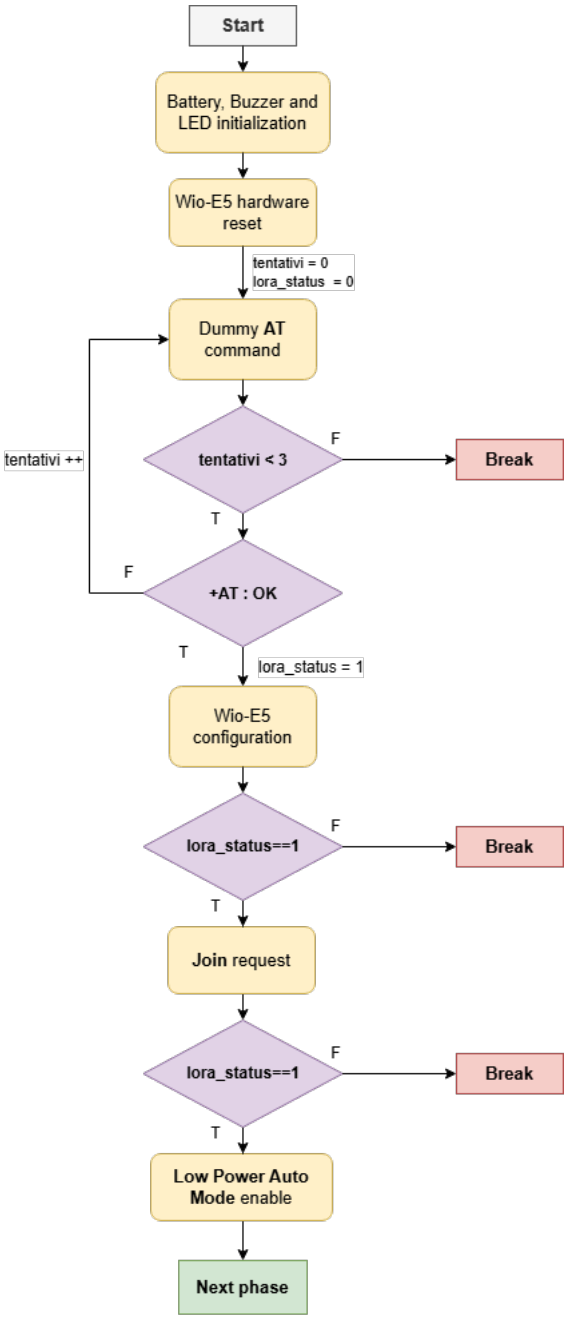


Figure 3.4. Power-on phase flow.

3.3.2 Measurement Unit Discovery

Measurement Unit discovery is triggered by pressing the tactile button on the circuit. The tactile button is connected to the MCU through an external-interrupt GPIO pin with hardware button debouncing. The button interrupt triggers the discovery phase,

which begins by enabling the analog voltage supply for all four MU connection interfaces. Once powered, the detection amplifiers report the number N of connected MUs. Three scenarios are possible:

- **N = 0**: no MUs are connected; the analog voltage supply is turned off and the program ends (no measurements are possible if no sensors are connected);
- **N = 1**: only one MU is connected to the UART bus, so no handshake is required; LID = 1 is automatically assigned to the MU. The MCU sends the connection command – described later in this section – and receives the MU's ExtID in response;
- **1 < N ≤ 4**: multiple MUs share the same UART bus to the MCU, so a handshake is required to avoid collisions.

UART Commands

Before describing the handshake procedure adopted in this thesis, it is useful to introduce the set of commands that enable communication between the microcontroller and the Measurement Units. These commands are extracted from MeasureStream Network documentation, version v0.5.

Identification. Executed right after the push-button interrupt, only if more than one MU is connected. The CU sends each connected device an identification request (Table 3.2), setting their Local ID to the default value and opening a 1 sec receiving window.

Byte	0	1
Data from CU	Default ID: 0x00	Opcode: 0x10

Table 3.2. Identification request – Many-MU case.

After a random delay (ΔT) between 10 and 110 ms, if the UART bus is free, each MU replies with its Extended ID (Table 3.3).

Byte	0 - 3	4
Data from MU	Extended ID	Check Byte: 0x10

Table 3.3. MU reply to the identification request.

If a correct message is received, the CU replies to that specific MU with its new Local ID (Table 3.4). Local IDs are assigned progressively, from 1 up to 4.

If, instead, the CU detects a collision, it sends a NACK (Table 3.5).

Byte	0	1	2 - 5	6
Data from CU	Default ID: 0x00	Opcode: 0x11	Extended ID	Local ID MU

Table 3.4. CU assignment.

Byte	0	1
Data from CU	Default ID: 0x00	Opcode: 0xFF

Table 3.5. NACK.

Connection. Executed right after the push-button interrupt, if only one MU is connected. The CU sends an identification request (Table 3.6) to the connected MU with the available Local ID. The MU replies with an Acknowledge message (Table 3.7), containing its Extended ID and the received Local ID.

Discovery. After MU identification and connection, the CU performs the discovery step by requesting the number of sensors mounted on each MU, using the following command (Table 3.8).

Commands are characterized by an FPort number, which acts as an opcode in LoRaWAN. Each FPort refers to a specific topic, such as identification, configuration, polling, or error handling; this value is used throughout the firmware to route incoming and outgoing messages to the correct handler, as detailed in the following sections.

Handshake Procedure

The adopted procedure follows a CSMA/CD-like structure:

1. The MCU sends an identification command to all devices, using the same default ID (0x00);

2. Each MU randomly selects a number between 10 and 110, corresponding to the number of milliseconds it must wait (ΔT);

3. Each MU listens on the line. After the elapsed ΔT, if the line is free, it sends its Extended ID and waits for a response containing its Local ID. If the CU detects a collision, it responds with a NACK. If the MU receives a NACK, receives no response, or finds the line busy, it retries after another random ΔT. This operation is repeated up to three times;

4. For each identified MU, the CU requests the number of sensors, in order of Local ID.

Byte	0	1	2
Data from CU	Default ID: 0x00	Opcode: 0x01	Local ID MU: 0x01

Table 3.6. Identification request – One-MU case.

Byte	0 - 3	4
Data from MU	Extended ID MU	Local ID MU

Table 3.7. Acknowledge.

If the handshake is successful, radio communication is available, and at least one MU is connected, the CU notifies the server with the newly acquired information, such as the number of connected MUs and their ExtIDs.

1. The Wio-E5 wakes up from low-power mode;
2. Payload building: a list of all identified Extended IDs, ordered by Local ID;
3. Uplink transmission on the application port (Fport: 0x11);
4. Non-blocking wait for transmission confirmation;
5. The Wio-E5 returns to low-power mode.

This strategy guarantees a plug-and-play experience for the user: the Measurement Units are easy to connect and configure, with everything managed by the Control Unit and the server. The flow of the discovery phase is shown in Figure 3.5.

3.3.3 Periodic Poll

Once the power-on phase is completed, the first periodic poll is scheduled. This is the longest phase, but also the least expensive in terms of average power consumption, since the CU spends most of its time in a low-power mode. Low-power strategies are discussed later, in Section 3.3.4. The steps composing this phase are listed below.

1. **Battery level measurement**. A first distinction is made:
   - `CRG_STAT = 0`: the battery is recharging, so the battery level is directly assigned the status value 0xFF.
   - `CRG_STAT = 1`: no connection to the mains, so the battery level must be measured. A 12-bit ADC converter translates the battery voltage into a discrete number, `raw`, which is converted into the percentage level `batt` using Equation 3.1.

$$batt = \frac{((raw - 2200) * 100 + \frac{1450}{2})}{1450} \tag{3.1}$$

Byte	0	1
Data from CU	Local ID MU	Opcode: 0x30

Table 3.8. Number of sensors request.

Byte	0
Data from MU	# Sensors

Table 3.9. MU number-of-sensors reply.

If the result is higher than 100% (`raw` > 3650), the device is connected to the mains only through the USB-C cable, with no battery connected; in this case, the battery level is directly assigned the status value 0xFE.

2. **Wio-E5 wake-up**. The module becomes able to transmit and receive;
3. P_{TX} **reading** from the LoRa module;
4. **Status packet assembly**. This contains the CU model, the battery percentage `batt`, and the transmit power P_{TX};
5. **Uplink transmission**;
6. **Microcontroller deep-sleep mode**. The reception of data on the LPUART wakes up the MCU;
7. **Downlink reception**. The message is assigned to the corresponding configuration handler based on the received FPort, which indicates the number of packets to be exchanged; accordingly, the CU opens an equal number of communication windows.

The poll period is configurable by the server with a minimum of 3 h. By default, it is set to 12 h. The flow of the polling phase is shown in Figure 3.6.

3.3.4 Sleep Policies

The low-power behavior of the CU during the low-power modes mentioned above – and, more generally, throughout its entire operation – relies on a set of sleep policies, described in this final part of the chapter. Since the device must operate for at least one year on battery power, keeping the microcontroller and the various modules in low-power modes for as long as possible is essential to meeting this requirement. A non-blocking firmware design strategy has been adopted for this purpose: since the MCU cannot wait for the next instruction or data-exchange window in normal mode, the classic `HAL_Delay()` function cannot be used. In general, waiting periods are spent in Sleep or Stop1 mode, with the MCU waking up at the desired moment thanks to a configurable low-power timer.

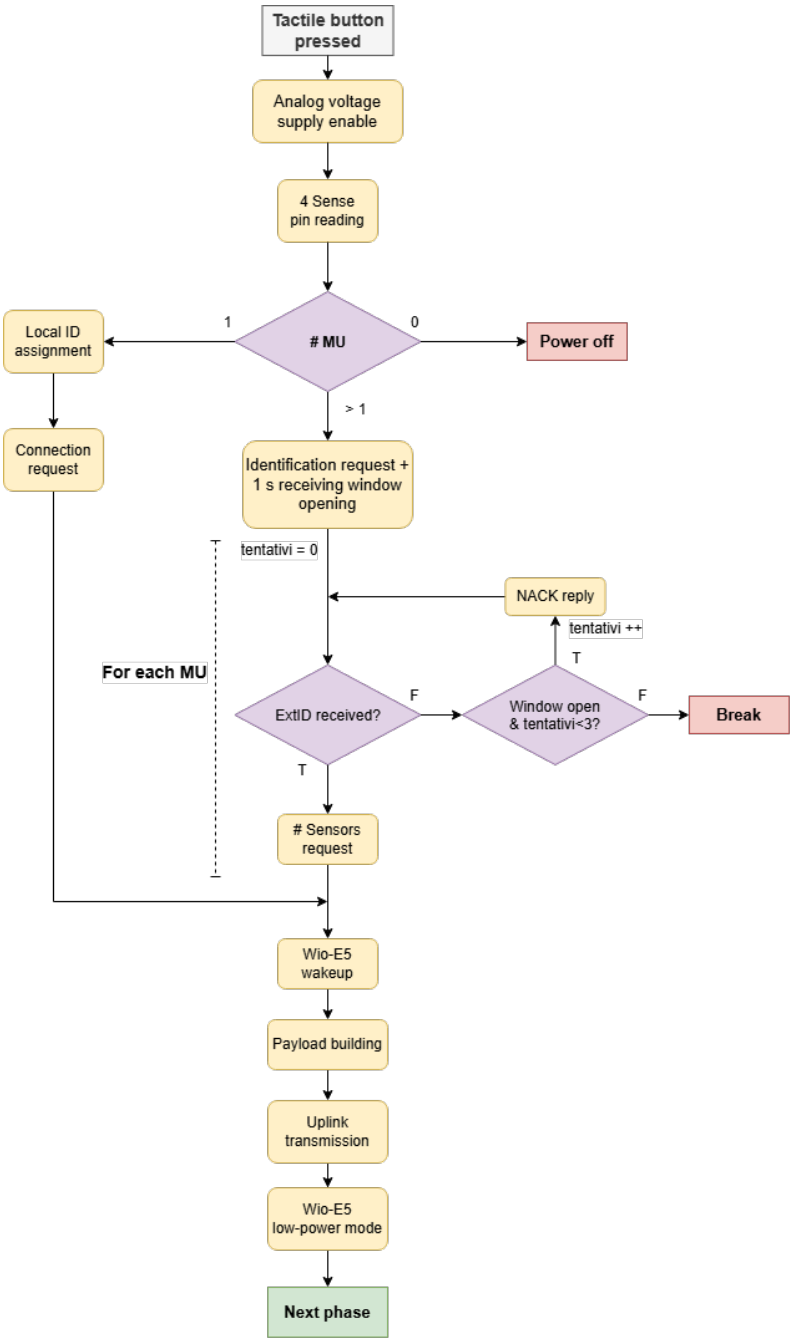


Figure 3.5. Discovery phase flow.

The enabled low-power peripherals are the following:

- **LPTIM1**: provides the time reference during sleep modes, substituting the system clock while the latter is disabled. It is configured as a one-shot timeout timer,

meaning that it starts counting only when requested and stops at the first trigger, without automatically resetting and restarting. If no external trigger occurs – such as data arriving on the LPUART or a button press – the timer generates an interrupt at timeout and wakes up the MCU; the time spent in sleep mode is stored and added to the virtual system time upon wake-up, ensuring a correct reference time regardless of the effects of the low-power modes;

- **LPTIM2**: provides the source for the PWM function used by the buzzer. Since it is a low-power peripheral, it allows the buzzer to keep playing notes during sleep modes;
- **LPUART1**: the communication interface between the microcontroller and the Wio-E5 LoRa module. It allows the Wio-E5 to send messages even while the MCU is in deep-sleep mode, triggering a wake-up upon reception.

For each peripheral, specific approaches are used to minimize power consumption during operation.

LED

The LED is the simplest peripheral to control, since it only requires waiting for the blinking action. During the waiting period between two consecutive pin toggles, the MCU enters deep sleep and wakes up at timeout, configured in this case to 500 m sec, giving a complete blink cycle of 1 sec.

Buzzer

The buzzer requires both low-power timers to function correctly. It generates notes using the PWM mode of LPTIM2, where the period of the waveform depends on the specific frequency of the note. The duration of each note is spent in deep-sleep mode, with the MCU waking up at LPTIM1 timeout only to switch to the next note. Thanks to the use of a low-power timer source for the PWM waveform, the buzzer can keep producing sound even while the MCU is in sleep mode, resulting in an overall reduction in power consumption.

Wio-E5 to Server

The main low-power actions concerning the LoRa module relate to the polling phase described in Section 3.3.3 and, more generally, to communication with the server. Payload transmission requires energy, and the longer the message, the higher the power consumption. The payload size is therefore always kept as short as possible: for this reason, every time a measurement is transmitted, the 4-byte ExtID of each MU is replaced with a shorter 1-byte LID. This strategy makes it possible to reduce power consumption during transmission while still preserving an identification number for the MU.

Another method used to minimize payload size consists in transmitting only aggregated data, such as the mean and variance of the measurements over the entire polling

period, instead of the full set of individual measurements. This approach drastically reduces the size of each transmission and, consequently, the power consumption associated with it.

As already mentioned, each uplink has a cost in terms of power consumption. It is therefore also necessary to limit the number of transmissions as much as possible, in order to save energy. This is the reason why the polling period is kept as long as possible – with a minimum of 3 h and a default value of 12 h.

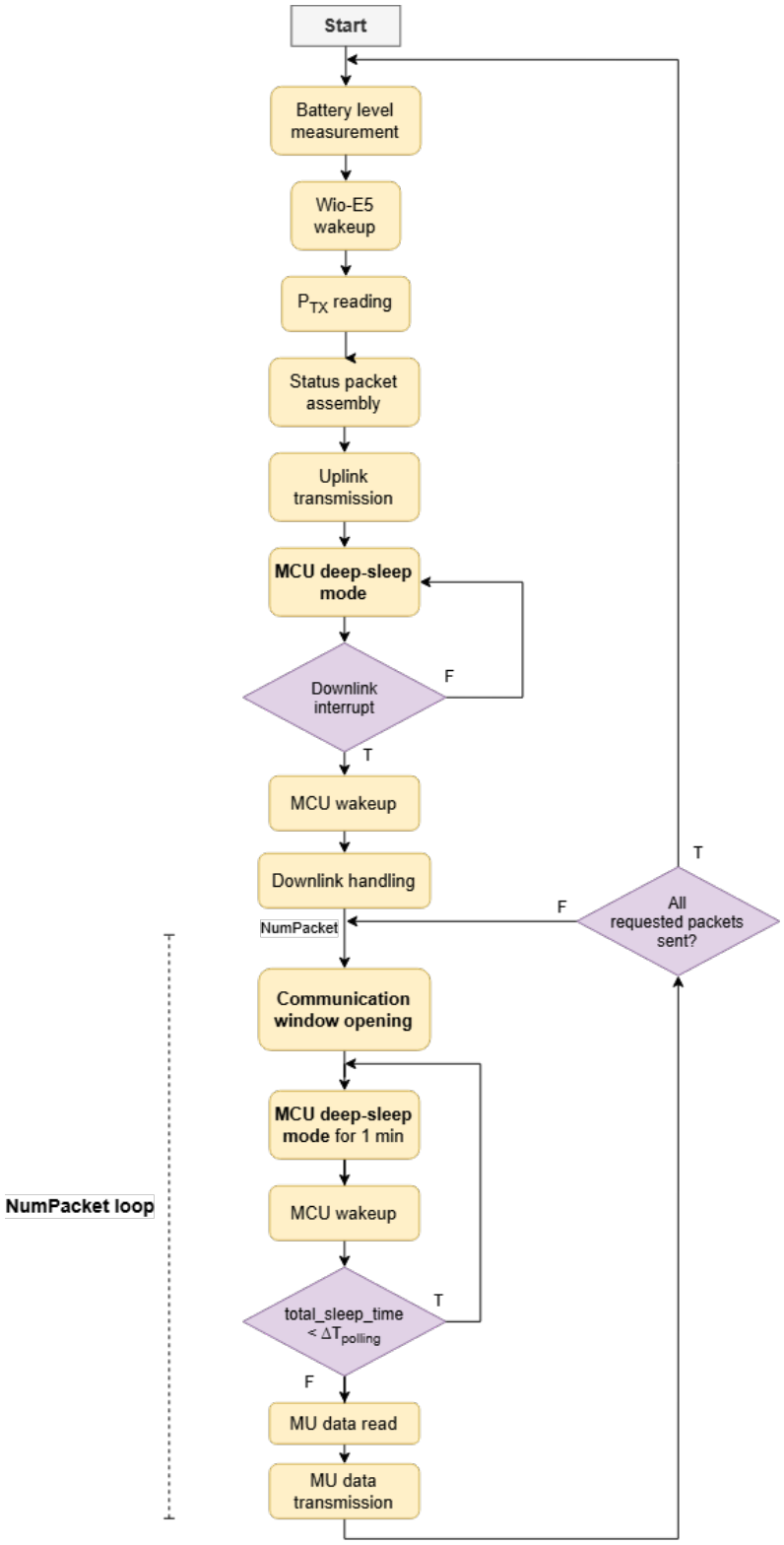


Figure 3.6. Polling phase flow.

Chapter 4

GNSS Module Evaluation

Geolocation is a core function of the device described in this thesis. Because critical control points are not confined to fixed facilities but extend to mobile assets – as discussed in the introduction to Chapter 3 – a node monitoring a shipment or a vehicle in transit must be able to report not only the measured quantity, but also where that measurement was taken. Without this information, a reading of an out-of-range temperature or an anomalous gas concentration cannot be tied to a specific point along a transport route, undermining exactly the kind of traceability that HSE, HACCP, and GxP compliance requires. The GNSS module integrated into the Control Unit is what makes this possible.

The module currently integrated in the device, the Teseo-LIV3R, was selected primarily for its compact form factor and its native integration within the STMicroelectronics ecosystem already used for the Control Unit's microcontroller. During testing, however, two significant limitations emerged. First, its power consumption, as reported in the manufacturer's datasheet, is considerably higher than that of competing modules in the same class, working directly against the energy-autonomy target that motivates the low-power firmware design described in Chapter 3. Second, and more critically, the module failed to acquire a valid position fix even after waiting periods of up to one hour. This test was conducted with the module mounted on a dedicated breakout adapter – a minimal printed circuit board routing the module's pins to a header, with no active circuitry beyond the module itself – connected to an Arduino Nano board, under the same environmental conditions and with the same antennas later used to test the candidate replacement modules described in this chapter. Since those candidate modules reliably achieved a fix under identical conditions, the failure can be attributed to the Teseo-LIV3R module itself, rather than to the test environment or setup, making it impossible to characterize its real-world Time To First Fix (TTFF) performance or to validate it as a reliable geolocation source for the device.

These limitations motivated a search for a viable replacement. This chapter presents a comparative evaluation of two candidate GNSS modules, each tested with two different antennas, for a total of four module-antenna combinations. For each combination, current consumption was measured through repeated sampling to characterize both its average value and its associated measurement uncertainty, while TTFF was measured under both cold-start and hot-start conditions to capture the module's acquisition performance in

the two scenarios most relevant to the device's duty-cycled operation. The chapter details the measurement methodology adopted, reports the resulting current and TTFF data for each combination, and combines them into a single power-consumption figure of merit, obtained through explicit uncertainty propagation, in order to identify the combination that offers the best replacement for the Teseo-LIV3R among those considered in this study.

4.1 The Currently Integrated Module: Teseo-LIV3R

The GNSS module currently integrated in the Control Unit is the STMicroelectronics Teseo-LIV3R, a single-die GNSS receiver supporting concurrent GPS and GLONASS reception, packaged in a compact 9.7 × 10.1 mm LCC-18 form factor. Its integration into the Control Unit was originally motivated by its compact footprint and by its native fit within the STMicroelectronics ecosystem already adopted for the Control Unit's microcontroller.

According to the manufacturer's datasheet, the module's power consumption in standby mode – with only the backup RTC domain active – is as low as 19.8 µW. In active tracking, however, the datasheet reports a power consumption of 93.4 mW under standard test conditions (GPS and GLONASS concurrently enabled, $T_c = 25\,^{\circ}\mathrm{C}$, $V_{CC} = 2.1$ V, $V_{CC_IO} = 3.3$ V) [26], considerably higher than that of the competing modules of the same class considered later in this chapter. Declared Time-To-First-Fix figures are 32 s (cold start), 36 s (warm start), and 1.5 s (hot start), with a declared cold-start acquisition sensitivity of −147 dBm.

Despite these nominal figures, the module could not be validated experimentally. The Teseo-LIV3R was tested on a dedicated breakout adapter – a minimal printed circuit board routing the module's pins to a header, with no active circuitry beyond the module itself – connected to an Arduino Nano board. Under these conditions, and using the same antennas later employed to test the candidate replacement modules described in Section 4.2, the module failed to acquire a valid position fix even after waiting periods of up to one hour. Since the candidate modules reliably achieved a fix under identical environmental conditions and with the same antennas, the failure can be attributed to the Teseo-LIV3R module itself, or to the specific breakout adapter used, rather than to the test environment or setup.

Notably, this behavior cannot be explained by the module's nominal TTFF or sensitivity specifications alone: as shown in Table 4.2, the Teseo-LIV3R's declared cold-start TTFF and sensitivity are broadly comparable to those of the candidate modules evaluated in this chapter. The discrepancy between datasheet specifications and observed behavior made it impossible to characterize the module's real-world TTFF or power consumption experimentally, and motivated the search for an alternative documented in the remainder of this chapter.

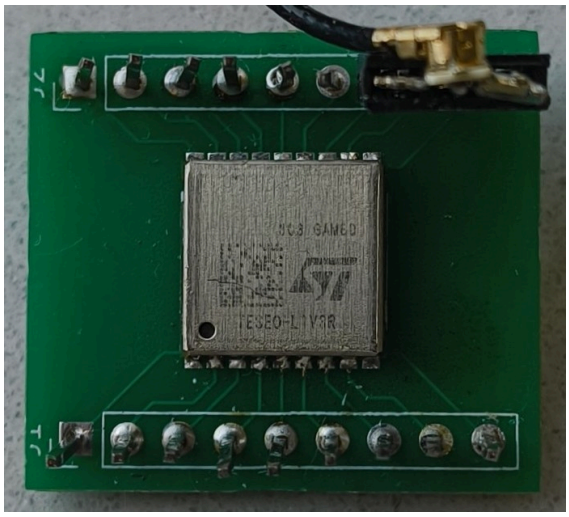


Figure 4.1. Teseo-LIV3R testing condition.

4.2 Candidate Replacement Modules and Antennas

Two candidate GNSS modules were selected for comparison against the Teseo-LIV3R, each evaluated with two different antennas, for a total of four module-antenna combinations.

4.2.1 u-blox NEO-6M

The NEO-6M is a GPS-only receiver chip from u-blox's NEO-6 family, widely available on low-cost breakout boards (commonly marketed as "GY-NEO6MV2"). According to the manufacturer's datasheet, the module supports multiple power modes that trade acquisition performance for reduced consumption: in Maximum Performance mode, the tracking current reaches approximately 39 mA (typical, $V_{CC} = 3.0$ V), while in Power Save Mode at a 1 Hz update rate, this drops to approximately 11 mA [31]. Declared TTFF figures are 27 s for cold and warm start and 1 s for hot start, with a declared cold-start sensitivity of -147 dBm.

4.2.2 ATGM336H

The ATGM336H is a multi-constellation GNSS module (supporting BeiDou, GPS, GLONASS, and Galileo) built around the AT6558 SoC. According to its datasheet, the module draws less than 25 mA in continuous operation at 3.3 V [34]. Declared TTFF figures are 32 s (typical, ≤ 35 s maximum) for cold start and ≤ 1 s for hot start and for signal reacquisition, with a declared cold-start sensitivity of -148 dBm.

4.2.3 Antennas and Combinations

Each of the two candidate modules was evaluated with two different GNSS antennas.

Antenna A is the Taoglas AP.25M.07.0080A, a 25 mm active patch antenna combining a ceramic patch element with an integrated single-stage low-noise amplifier (LNA) and a front-end SAW filter. According to its datasheet, the patch element alone provides a nominal gain of +2.0 dBic at zenith, while the complete assembly – patch, LNA, cable, and connector – reaches approximately 17 ± 3 dBic at 3.0 V [28]. The LNA draws 4 mA (typical) at 3.0 V, rising to 9 mA at the maximum rated voltage of 5.5 V. The antenna operates over 1575.42 ± 1.023 MHz, matching the GPS L1/Galileo E1 band, and terminates in an I-PEX MHF-I (U.FL) connector via an 80 mm coaxial cable.

Antenna B is a compact active antenna (model TX-AG011, 16×6×6.7 mm) manufactured by Dongguan Tengxiang Electronics, operating at 1575.42 MHz with a VSWR below 2.0 and a 50 Ω impedance, terminating in a U.FL female connector via an RG113 cable [6]. The manufacturer's product listing reports a gain of 22 dBi; given the antenna's small form factor, this figure is physically implausible for a device of this size, and is almost certainly a data-entry error or a value copied from an unrelated product template rather than a genuine measurement. No independent datasheet could be located for this antenna, and its actual gain was therefore not assumed a priori, but treated as an outcome to be inferred – together with the rest of its performance – from the TTFF and consumption measurements presented in this chapter.

This resulted in four module-antenna combinations, summarized in Table 4.1, each independently characterized in terms of current consumption and TTFF according to the methodology described in Section 4.3.

Combination	Module	Antenna
C1	NEO-6M	Antenna A (Taoglas AP.25M)
C2	NEO-6M	Antenna B (TX-AG011)
C3	ATGM336H	Antenna A (Taoglas AP.25M)
C4	ATGM336H	Antenna B (TX-AG011)

Table 4.1. The four module-antenna combinations evaluated in this chapter.

Parameter	Teseo-LIV3R	NEO-6M	ATGM336H
Tracking power/current	93.4 mW	∼11–39 mA	<25 mA
Standby current	19.8 µW	22 µA	10 µA
Cold-start TTFF (typ.)	32 s	27 s	32 s
Hot-start TTFF (typ.)	1.5 s	1 s	1 s
Cold-start sensitivity	−147 dBm	−147 dBm	−148 dBm

Table 4.2. Nominal performance comparison of the three modules, as reported in their respective datasheets. Current and voltage conditions vary slightly between datasheets; see text for details.

4.3 Methodology

This section describes the experimental procedure adopted to characterize each of the four module-antenna combinations introduced in Section 4.2, and the method used to combine the resulting measurements into a single power-consumption figure with an associated uncertainty.

4.3.1 Current Consumption Measurement Setup

For each combination, the module's supply current was measured with a Fluke 111 digital multimeter, connected in series between the module's V_{CC} pin and the 3.3 V output pin of the Arduino Nano board used to power and interface with the module under test. No configuration commands were sent to the module beyond power-up: the receiver's default output messages were read as produced, without selecting a specific operating mode (e.g., continuous tracking versus periodic acquisition), so the measured current reflects the module's out-of-the-box behavior rather than a mode explicitly configured for this test. All measurements were carried out outdoors, in order to maximize satellite visibility and avoid the additional signal attenuation – and the corresponding uncertainty on the module's actual operating state – that indoor testing would introduce.

For each of the four combinations, ten repeated current readings were taken under nominally identical conditions: the physical setup, antenna position, and module configuration were left unchanged between readings, so that the resulting sample reflects the combined effect of instrument reading noise and short-term fluctuations in the module's supply current, rather than variability introduced by changes in the test setup. It is worth noting that, because all ten readings for a given combination were taken in short succession without altering the physical setup, this sample does not capture slower sources of variability – such as changes in the number of visible satellites over time – that could affect current draw over longer observation windows; the resulting uncertainty should therefore be interpreted as a lower bound on the true variability of each combination's power consumption, rather than as a complete characterization of it.

4.3.2 TTFF Measurement Setup

For each combination, five acquisition events were recorded: one initial acquisition following power-up with no prior almanac or ephemeris data available to the receiver, and four successive acquisitions obtained by power-cycling the module at approximately 2-minute intervals. In each case, TTFF was measured as the elapsed time between module power-up and the reception of the first output message reporting a valid GPS fix status. Timing was not instrumented to high precision, as TTFF is not the primary figure of merit in this study: it is reported alongside the power-consumption results as supplementary information, useful primarily to discriminate between combinations that might otherwise show comparable power consumption.

Because no verified threshold could be established from the datasheets of the modules under test for the power-off interval that formally distinguishes a "hot start" from a "warm" or "cold" one, this thesis avoids that standard GNSS terminology and instead

reports the four successive acquisitions descriptively, as acquisitions following a 2-minute power-cycle interval.

4.3.3 Current Uncertainty Estimation

For each combination, the ten current readings $I_1, \ldots, I_{10}$ were used to compute the sample mean

$$\bar{I} = \frac{1}{n} \sum_{i=1}^{n} I_i, \qquad n = 10, \tag{4.1}$$

and the experimental standard deviation

$$s_I = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} \left(I_i - \bar{I} \right)^2}. \tag{4.2}$$

The standard uncertainty of the mean (Type A) is then

$$u_A(\bar{I}) = \frac{s_I}{\sqrt{n}}. \tag{4.3}$$

Current readings were taken with a Fluke 111 digital multimeter, whose datasheet specifies a DC current accuracy of $\pm(1.0\%$ of reading$+3$ counts), with a display resolution of 0.001 A (i.e., 1 count = 0.001 A). This specification defines the instrument's maximum error limit, $a(\bar{I})$, as

$$\delta(\bar{I}) = 0.010 \cdot \bar{I} + 3 \times 0.001 \text{ A} = 0.010 \cdot \bar{I} + 0.003 \text{ A}. \tag{4.4}$$

Assuming a rectangular distribution within this limit, as is standard practice for manufacturer-specified accuracy bounds, the corresponding Type B standard uncertainty is

$$u_B(\bar{I}) = \frac{\delta(\bar{I})}{\sqrt{3}}. \tag{4.5}$$

The Type A and Type B contributions were then combined in quadrature to obtain the combined standard uncertainty

$$u_c(\bar{I}) = \sqrt{u_A(\bar{I})^2 + u_B(\bar{I})^2}. \tag{4.6}$$

The expanded uncertainty is reported as $U(\bar{I}) = k \cdot u_c(\bar{I})$, with the coverage factor k chosen from the Student's t-distribution for $n - 1 = 9$ degrees of freedom at the desired confidence level (e.g., $t_{9,95\%} \approx 2.26$), rather than the asymptotic value $k = 2$, to account for the relatively small sample size. It should be noted that ten repetitions, while sufficient to compute a meaningful Type A estimate, remain a comparatively small sample: the resulting uncertainty is statistically valid but should not be interpreted as fully representative of all sources of variability affecting the module's current draw in the field.

4.3.4 Power Consumption Calculation

The power consumption of each combination was computed from the measured mean current and the voltage across the module. Since the multimeter's series resistance R_S is smaller than 10 mΩ, it is considered negligible in these calculations. Therefore, the voltage supplied to the modules is taken to be equal to the supply voltage provided by the Arduino Nano board, $V = 3.3$ V:

$$P = V \cdot \bar{I}. \tag{4.7}$$

The Arduino Nano's regulated output is subject to a manufacturer-specified tolerance, taken as ±1.5% of the nominal value. Since no repeated measurements of V were performed, this contributes a Type B uncertainty only. Assuming a rectangular distribution within the specified tolerance, as is standard practice for manufacturer-specified accuracy bounds, the maximum error limit on the voltage is

$$\delta(V) = 0.015 \cdot V = 0.015 \times 3.3 \text{ V} = 0.0495 \text{ V}, \tag{4.8}$$

and the corresponding standard uncertainty is

$$u_B(V) = \frac{\delta(V)}{\sqrt{3}} = 0.02858. \tag{4.9}$$

Since P is a function of two independent input quantities, V and $\bar{I}$, its combined standard uncertainty is obtained by applying the law of propagation of uncertainty:

$$u_c(P) = \sqrt{\left(\frac{\partial P}{\partial V}\right)^2 u_B(V)^2 + \left(\frac{\partial P}{\partial \bar{I}}\right)^2 u_c(\bar{I})^2}. \tag{4.10}$$

Since $P = V \cdot \bar{I}$, the partial derivatives are

$$\frac{\partial P}{\partial V} = \bar{I}, \qquad \frac{\partial P}{\partial \bar{I}} = V, \tag{4.11}$$

so that

$$u_c(P) = \sqrt{\bar{I}^2 \cdot u_B(V)^2 + V^2 \cdot u_c(\bar{I})^2}. \tag{4.12}$$

The expanded uncertainty is reported as $U(P) = k \cdot u_c(P)$, using the same coverage factor $k = 2.26$ (95% confidence, $t_{9,95\%}$, consistent with the degrees of freedom associated with the ten current readings) adopted in Section 4.3 for the current measurements. The resulting power consumption, $P \pm U(P)$, constitutes the primary figure of merit used in Section 4.4 to rank the four module-antenna combinations, with TTFF reported alongside as a secondary criterion for combinations whose power consumption values are not clearly distinguishable within their respective uncertainties.

Reading	C1 [mA]	C2 [mA]	C3 [mA]	C4 [mA]
1	46	34	51	40
2	47	34	52	40
3	47	34	52	40
4	47	33	52	40
5	47	34	52	40
6	47	34	52	40
7	47	34	52	39
8	47	33	52	40
9	47	33	52	40
10	47	33	52	40
Mean $\bar{I}$	46.90	33.60	51.90	39.90
Std. dev. s_I	0.32	0.52	0.32	0.32

Table 4.3. Current consumption readings (mA) for the four module-antenna combinations, ten repetitions each.

4.4 Results

4.4.1 Current Consumption

Table 4.3 reports the ten current readings collected for each of the four module-antenna combinations, together with the resulting sample mean and experimental standard deviation, computed as described in Section 4.3.

Applying the uncertainty model described in Section 4.3, the Type A uncertainty, the combined standard uncertainty, and the expanded uncertainty (at the 95% confidence level, $t_{9,95\%} \approx 2.26$) were computed for each combination. The results are summarized in Table 4.4.

Combination	$u_A(\bar{I})$ (mA)	$u_B(I)$ (mA)	$u_c(\bar{I})$ (mA)	$U(\bar{I})$ (mA)
C1 (NEO-6M, Ant. A)	0.10	2.00	2.01	4.53
C2 (NEO-6M, Ant. B)	0.16	1.93	1.93	4.37
C3 (ATGM336H, Ant. A)	0.10	2.03	2.03	4.60
C4 (ATGM336H, Ant. B)	0.10	1.96	1.96	4.44

Table 4.4. Uncertainty budget for the mean current of each combination.

Combination C2 (NEO-6M, Ant. B) shows the lowest mean current draw among those tested, at $\bar{I} = 33.60$ mA $\pm$ 4.37 mA, while combination C3 (ATGM336H, Ant. A) shows the highest, at $\bar{I} = 51.90$ mA $\pm$ 4.60 mA.

4.4.2 Time to First Fix

For each combination, five acquisition events were recorded following the procedure described in Section 4.3: one initial acquisition after power-up with no prior almanac or ephemeris data, and four successive acquisitions obtained by power-cycling the module at approximately 2-minute intervals. Table 4.5 reports the resulting TTFF values.

Acquisition	C1	C2	C3	C4
Initial (power-up)	17 min 30 s	28 s	2 min 48 s	1 min 15 s
Successive #1 (2 min)	16 s	3 min 30 s	6 s	7 s
Successive #2 (2 min)	15 s	2 s	3 s	9 s
Successive #3 (2 min)	37 s	8 s	7 s	7 s
Successive #4 (2 min)	24 s	13 s	11 s	7 s

Table 4.5. TTFF measurements for the four module-antenna combinations. As discussed in Section 4.3, timing was not instrumented to high precision, and TTFF is reported here as supplementary information rather than as the primary figure of merit.

As expected, successive acquisitions following a short power-cycle interval consistently show a markedly shorter TTFF than the initial, cold acquisition, across all four combinations, consistent with the retention of recent ephemeris and almanac data over the 2-minute interval between successive power cycles, with the exception of combination C2, whose first successive TTFF is much higher than both its initial value and its other successive values.

4.4.3 Power Consumption

The power consumption of each combination was computed from the mean current and its associated uncertainty, following the model described in Section 4.3, using the fixed supply voltage $V = 3.3$ V. The resulting values are summarized in Table 4.6.

Combination	Power (mW)	Expanded uncertainty $U(P)$ (mW)
C1 (NEO-6M, Ant. A)	154.77	15.26
C2 (NEO-6M, Ant. B)	110.88	14.58
C3 (ATGM336H, Ant. A)	171.27	15.54
C4 (ATGM336H, Ant. B)	131.67	14.88

Table 4.6. Power consumption of the four module-antenna combinations, with expanded uncertainty at the 95% confidence level.

Combination C2 (NEO-6M, Ant. B) exhibits the lowest power consumption among those tested, at $P = 110.88$ mW $\pm$ 14.58 mW. This value is comparable, within measurement uncertainty, to that of combination C4 (ATGM336H, Ant. B) ($P = 131.67$ mW $\pm$

14.88 mW), as the two uncertainty intervals overlap; in this case, the TTFF results reported in Table 4.5 are used as a secondary criterion to discriminate between the two combinations, favoring C4 for its shorter average successive acquisition time.

Chapter 5

Conclusion

This thesis addressed the problem of monitoring critical control points across distributed, and often mobile, industrial and logistics environments, where compliance with safety and quality protocols such as HSE, HACCP, and GxP requires continuous, real-time measurement of parameters that cannot always be tied to a single, fixed facility. Chapter 2 showed that wired industrial architectures, while robust and deterministic, are structurally unable to meet this requirement, and that short-range wireless technologies, although more flexible, remain limited by their coverage and, in most cases, by power consumption incompatible with long-term, unattended operation. LoRaWAN was identified as the most suitable long-range, low-power communication technology for this class of applications, combining multi-kilometer range, battery lifetimes on the order of years, and the possibility of a private, self-owned network infrastructure not available with cellular LPWAN alternatives such as LTE-M, NB-IoT, or Sigfox.

Building on this premise, Chapter 3 presented the design of a modular monitoring node, developed at the Laboratorio di Misure e Sensori of Politecnico di Torino, consisting of a single Control Unit (CU) and up to four interchangeable Measurement Units (MUs). This architecture allows the same node to be adapted to different critical control points – temperature, humidity, gas concentration, or others – simply by attaching the appropriate Measurement Unit, without redesigning or duplicating the device. The core contribution of this thesis lies in the Control Unit firmware, designed around a strictly non-blocking, state-machine-based architecture that allows the microcontroller to remain in deep-sleep modes for the vast majority of its operating time. Specific low-power strategies were implemented at multiple levels: a compact Local ID replaces the longer Extended ID during LoRa transmissions to reduce payload size and, consequently, transmission time; measurements are aggregated over each polling period rather than transmitted individually; and dedicated low-power peripherals (LPTIM1, LPTIM2, LPUART1) allow the LED, the buzzer, and the LoRa module to operate correctly even while the microcontroller is asleep. Together, these strategies were designed to support the target of at least one year of autonomous operation on battery power, a constraint of particular relevance given the device's intended use on mobile assets, such as vehicles, where recharging or replacing the battery cannot always be performed on demand.

Finally, Chapter 4 addressed the geolocation function of the device, essential for tying

each measurement to a specific point along a transport route. The GNSS module originally integrated into the Control Unit, the Teseo-LIV3R, proved unable to acquire a valid position fix under the test conditions adopted in this work, despite competitive nominal specifications, and was also found to draw more power than comparable alternatives. This motivated a comparative evaluation of two candidate replacement modules (u-blox NEO-6M and ATGM336H), each tested with two different antennas, based on measured current consumption, its associated uncertainty, and Time To First Fix. The combination of the NEO-6M module with the compact TX-AG011 antenna (C2) achieved the lowest power consumption among those tested, with a shorter successive-acquisition TTFF than most other combinations, although its markedly higher first-successive-acquisition time compared to its own initial and other successive readings (3 min 30 s) makes it a comparable candidate to the ATGM336H paired with the same antenna (C4), whose power consumption remained within the same uncertainty range.

Several directions remain open for future work. On the hardware side, the anomalous behavior of the Teseo-LIV3R module warrants further investigation, ideally through testing with additional breakout adapters or antennas, to determine with more confidence whether the fault lies in the module itself or in the specific test setup adopted in this thesis. The ATGM336H GNSS module could replace the Teseo-LIV3R if needed. On the firmware side, a direct, in-field measurement of the device's overall power consumption and effective battery lifetime – beyond the individual peripheral-level strategies discussed in Chapter 3 – would provide a more complete validation of the one-year autonomy target. Finally, extending the set of available Measurement Units, in particular by completing the integration of the gas concentration sensors (CO, NO_2, Cl_2) mentioned in Chapter 3, would broaden the range of critical control points the system can address, further increasing its relevance for HSE, HACCP, and GxP compliance across a wider range of industrial and logistics scenarios.

Bibliography

[1] Mukarram Almuhaya, Tawfik Al-Hadhrami, Omparakash Kaiwartya, and David J. Brown. A novel fair frame time slot LoRaWAN internet of things. In *2024 International Conference on Intelligent Computing and Next Generation Networks (ICNGN)*. Crown, 2024.

[2] Aucon Technologies. Comparing industrial communication protocols: Advantages and disadvantages. Online article. Accessed 2026.

[3] Kamaldeep Bansal. Bluetooth mesh device authentication & secure network commissioning. In *2024 4th International Conference on Innovative Practices in Technology and Management (ICIPTM)*. IEEE, 2024.

[4] Alessio Carullo, Silvia Corbellini, Franco Lombardo, Alberto Vallan, and Marco Sento. Measuring node for modular wireless sensor-networks in harsh industrial environments. In *2024 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*. IEEE, 2024.

[5] Vishnu V Das, Anish Sathyan, and Divya D S. Establishing LoRa based local agrisensor network through sensor plugin modules and LoRaWAN data concentrator for extensive agriculture automation. In *2022 IEEE 19th India Council International Conference (INDICON)*. IEEE, 2022.

[6] Dongguan Tengxiang Electronics. China Beidou GPS Module Locator BDS Satellite Positioning Navigator Compass Antenna Module, Dual Mode (model TX-AG011). Made-in-China.com product listing. Online product listing; no independent datasheet available, see thesis discussion.

[7] Edilizia Namirial. HSE: significato e applicazioni. Online article. Accessed 2026.

[8] European Commission. EU GMP guideline – annex 1: Manufacture of sterile medicinal products. GMP Compliance Adviser. Accessed 2026.

[9] European Union. Regulation (ec) no 852/2004 of the european parliament and of the council of 29 april 2004 on the hygiene of foodstuffs. EUR-Lex, 2004.

[10] Food and Agriculture Organization of the United Nations. Agrifood policy highlights – july 2026. FAO Policy Support and Governance, 2026.

[11] Samsul Huda, Hien Nguyen, Tuy Tan Nguyen, and Yasuyuki Nogami. LPWANs for IoT-enabled agriculture: Review of performance and security. In *TRON Forum*, 2025. ISBN 978-4-89362-416-1.

[12] Inductive Automation. What is SCADA? Online article. Accessed 2026.

[13] Industrial Monitor Direct. Fundamentals of industrial control systems: SCADA, DCS, and fieldbus. Knowledge base article. Accessed 2026.

[14] International Organization for Standardization. ISO 14001 explained. ISO official website. Accessed 2026.

[15] International Organization for Standardization. ISO 45001 explained: What it is. ISO official website. Accessed 2026.

[16] Mehzabien Iqbal, Abu Yousha Md Abdullah, and Farzana Shabnam. An application based comparative study of LPWAN technologies for IoT environment. In *2020 IEEE Region 10 Symposium (TENSYMP)*, pages 1857–1860. IEEE, 2020.

[17] Anna N. Kim, Fredrik Hekland, Stig Petersen, and Paula Doyle. When HART goes wireless: Understanding and implementing the WirelessHART standard. In *2008 IEEE International Conference on Emerging Technologies and Factory Automation*, pages 899–907. IEEE, 2008.

[18] MDPI Sensors. Industry 4.0 adoption barriers and enabling technologies for SMEs. *Sensors*, 21(2):385, 2021.

[19] MECT. Guida introduttiva ai PLC: cos'è un PLC e come funziona. Online article. Accessed 2026.

[20] Mikrodev. What is RTU and how does it work? Online article. Accessed 2026.

[21] Ministero dell'Agricoltura, della Sovranità Alimentare e delle Foreste. Codex alimentarius. Online resource. Accessed 2026.

[22] QS Group. Cosa sono le norme GxP. Online article. Accessed 2026.

[23] Shubham Saloni and Achyut Hegde. WiFi-aware as a connectivity solution for IoT: Pairing IoT with WiFi aware technology, enabling new proximity-based services. In *2016 International Conference on Internet of Things and Applications (IOTA)*, pages 137–142. IEEE, 2016.

[24] Amit Sharma and Girma S. Tewolde. Considerations in low power wireless sensor networks. *IEEE*, pages 626–631, 2015.

[25] Pankaj Kumar Singh and Om Jee Pandey. LPWAN technologies for IoT based SES. In *2025 IEEE 12th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)*. IEEE, 2025.

[26] STMicroelectronics. *Teseo-LIV3R: Tiny GNSS Module with Embedded Antenna, Datasheet*. STMicroelectronics, 2023. Datasheet, DS12063.

[27] Jing Sun and Xiaofen Zhang. Study of ZigBee wireless mesh networks. *2009 Ninth International Conference on Hybrid Intelligent Systems*, pages 264–267, 2009.

[28] Taoglas. *AP.25M.07.0080A: 25 mm Embedded GPS/GLONASS Active Antenna, Datasheet*. Taoglas Limited. Datasheet AP.25M.07.0080A.

[29] The Network Installers. Ethernet installation cost. Online article. Accessed 2026.

[30] The Things Network. LoRaWAN. The Things Network Documentation. Accessed 2026.

[31] u-blox. *NEO-6 u-blox 6 GPS Modules, Data Sheet*. u-blox AG. Accessed via SnapEDA, breakout board commonly marketed as "GY-NEO6MV2".

[32] Wi-Fi Alliance. Alternative topologies. Wi-Fi Alliance official website. Accessed 2026.

[33] Xiaodong Zheng, Shuangsi Xue, Hui Cao, Feng Wang, and Mingke Zhang. A cost-efficient smart IoT device controlling system based on Bluetooth mesh and cloud computing, 2020.

[34] Zhongkewei (ZK). *ATGM336H GPS/BDS Dual-Mode Module, Datasheet.* Zhongkewei Electronics.