

DOTTORATO DI RICERCA IN INGEGNERIA INDUSTRIALE E DELL'INFORMAZIONE

CURRICULUM INGEGNERIA ELETTRONICA

XXXVIII CICLO

TESI DI DOTTORATO IN

HARDWARE/FIRMWARE CIRCUITS AND SYSTEMS FOR THE MONITORING OF INDUSTRIAL SYSTEMS

SSD: IINF-01/A

Coordinatore del Corso
Prof. Katia Gallucci

Dottorando
Paolo Esposito

Tutor
Prof. Vincenzo Stornelli

*“The future belongs to those who believe in the beauty of their dreams”
Eleanor Roosevelt*

Abstract

The evolution of industry has fostered the development, refinement, and optimization of hardware-firmware systems. These systems are characterized by a wide range of implementation methods, owing to their inherently flexible architecture. This thesis investigates the application of Long Range (LoRa) technology within well-defined hardware-firmware frameworks. The proposed systems are deployed across three distinct industrial contexts. First, LoRa is employed in industrial and environmental monitoring, where extensive areas exposed to geohazards require continuous and reliable observation. Second, LoRa serves as a communication protocol among smart metering devices installed in regions with limited internet connectivity; its long-range capabilities enable data transmission to the nearest internet-accessible nodes. Third, LoRa is integrated into a Structural Health Monitoring (SHM) system to interconnect multiple nodes with a central master node—a solution particularly advantageous for monitoring large-scale structures. Beyond these applications, in this work several industrial use cases involving hardware-firmware systems are studied, developed and designed. Specifically, these include the development of 3D-printed sensors and circuits for cost-effective prototyping and manufacturing, the design of specialized sensor interfaces and microelectronic circuits, and the application of Design of Experiments (DoE) methodologies to model a high-power Ultrawide Band (UWB) circuit. Each case study is examined in detail with respect to its hardware architecture, control unit and framework, and firmware implementation. Hardware-firmware systems constitute the backbone of contemporary industrial and academic applications. Their comprehensive characterization and continuous enhancement, regardless of the industrial era, introduce significant innovations and streamline entire production processes.

Contents

List of Figures	8
List of Tables	12
1 Introduction	1
1.1 Industry 4.0 — State Of The Art	2
1.2 Industry 5.0 — The Future Industry	3
1.3 Test Methods In The Industry	4
1.4 Communication Standards In The Current Industry	6
1.4.1 Industry 4.0	6
1.4.2 Industry 5.0	7
1.5 Hardware and Firmware	8
1.5.1 Instruments for the industry	9
1.5.2 Microcontrollers: What Do We See	10
1.5.3 Hardware in Microcontrollers	12
1.5.4 The Microcontroller Unit	20
1.6 Purpose of the Thesis	26
2 LoRa-based Hardware-Firmware Systems	27
2.1 Geohazard Prevention	27
2.1.1 Introduction	28
2.1.2 Network Structure	30
2.1.3 Working Principle and Nodes Firmware	32
2.1.4 Implementations and Results	36
2.1.5 Comparison With Other Works	42
2.2 Reliable and Smart Gas-Metering	44
2.2.1 Literature Overview	44
2.2.2 Smart Metering and User Safety — Further Motivations	45
2.2.3 System Overview	46
2.2.4 Firmware: an Overview with Focus on the Flux Sensor	51
2.2.5 Flow Sensor Metrological Analysis and Data Transmission Test	55
2.2.6 Blockchain Structure	61
2.3 Structural Health Monitoring	65
2.3.1 Introduction	65
2.3.2 System Description	66
2.3.3 Setup and Results	75
3 Collateral Activities	81
3.1 3D-Printed Sensors and Circuits	82
3.1.1 Applications of Fused Deposition Modeling in Electronics	83

3.1.2	3D-printed Sensors	84
3.1.3	3D-printed Circuits	93
3.2	Circuits for Sensor Interfaces	99
3.2.1	Circuit Description	101
3.2.2	Mathematical Model	103
3.2.3	Circuit Manufacturing	105
3.2.4	Measurements	105
3.3	Circuits for Microelectronics	108
3.3.1	The circuit blocks	109
3.3.2	Results	111
3.4	DoE-based Mathematical Models	115
3.4.1	The Pulser	116
3.4.2	The Marx Generator	116
3.4.3	Simulations	117
4	Conclusions	122
	Bibliography	131

List of Figures

1.1	Main aspects of the Industry 4.0.	2
1.2	Graphical representation of the SIL model.	5
1.3	Graphical representation of the HIL model.	5
1.4	Graphical representation of microcontroller board parts.	11
1.5	Arduino MKR WiFi 1010. (https://www.mouser.it/ProductDetail/Arduino/ABX00023?qs=%252BEew9%252B0nqrAwxv2YQYWyPw%3D%3D) . . .	12
1.6	Representation of a basic General Purpose Input-Output (GPIO) interface.	13
1.7	Block representation of an Analog-to-Digital Converter (ADC).	14
1.8	A floating-gate transistor	15
1.9	Circuit structure of a Static RAM (SRAM) memory.	16
1.10	Representation of the Von Neumann architecture.	17
1.11	Representation of the Harvard architecture.	18
1.12	Representation of pipelining with four-phases instructions.	19
1.13	High-level representation of an Microcontroller Unit (MCU).	21
1.14	Deeper-level representation of an MCU.	21
1.15	Representation of firmware layering.	22
2.1	Representation of the Chirp Spread Spectrum. By Georg-Johann — Own work, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=11330709	28
2.2	High-level representation of the Wireless Sensor Network (WSN) structure.	30
2.3	User interface based on Node.js (a) and HTTP (b) for the proposed WSNs.	31
2.4	Variations on the communication strcuture of the reported WSNs	32
2.5	Internal structure of a front-end node for the proposed WSNs.	33
2.6	Internal structure of a gateway node for the proposed WSNs.	33
2.7	Firmware-related flow diagram for both WSNs front-end and gateway nodes.	34
2.8	Circuit configuration for smart nodes power consumption measurement. .	35
2.9	Power consumption over a transmission period for the LoRa smart nodes.	35
2.10	Implementation of the reported system in the Pantelleria island	36
2.11	Implementation of the custom WSN in the inhabited area of Bucchianico (CH).	37
2.12	Inclination results from the implementation in the Bucchianico (CH) locality.	37
2.13	Implementation of the reported system with crack meters for rockfall monitoring	38
2.14	Crack measurement with the proposed custom WSN.	38
2.15	Slave (a) and sink (b) nodes in the underwood monitoring implementation	39
2.16	Water level measurement in the underwood multi-level implementation. .	39

2.17	Water pressure (left) and inclination (right) measurements in the under-wood multi-level implementation.	40
2.18	Mounting map (a) and radio coverage (b) of the WSN implementation reported in Ischia (NA).	40
2.19	Results from the implementation in Ischia (NA): inclinations (a), temperature (b), wind direction and magnitude (c).	41
2.20	In-cave implementation of the reported system with deep-sleep capability	42
2.21	Visual representation of an Advanced Metering Infrastructures (AMI) structure.	45
2.22	Visual representation of a blockchain.	46
2.23	High-level representation of the gas-metering system communication structure.	47
2.24	Single gas-metering unit structure.	48
2.25	Gas meter measurement part circuit arrangement.	49
2.26	Gas meter interfacing part circuit arrangement.	50
2.27	Schematic representation of the gas metering unit wireless recharge part.	51
2.28	Flow diagram representing the measurement block firmware.	52
2.29	Flow diagram representing the interfacing block firmware.	54
2.30	Flow diagram representing the flux sensor firmware.	55
2.31	Setup diagram (a) and testbench (b) for the metrological evaluation . . .	56
2.32	Reference flow meters: in-line (a) and cased (b) flow meters	57
2.33	Pressure sensors (a) and reference pressure gauge (b)	58
2.34	Pressure sensors calibration diagram.	58
2.35	Pressure measurements at different positions in the test configuration, in three different flow conditions, and in two environmental conditions (a) and (b).	59
2.36	Nucleo-L452RE board from Mouser Electronics, Inc. https://www.mouser.it/ProductDetail/STMicroelectronics/NUCLEO-L452RE?qs=zwwtmr6GuIyXyX5dw%2FgKaw%3D%3D	60
2.37	Flow sensor measurement setup.	60
2.38	Flow sensor preliminary test results.	61
2.39	The proposed communication infrastructure	62
2.40	Distributor-consumer (a) and prosumer-consumer (b) tokens life cycle. .	63
2.41	Infrastructure used for the simulation.	63
2.42	High-level scheme of the system based on communication	67
2.43	Node scheme reporting the used hardware (a), and final assembled circuit (b)	70
2.44	Node algorithmic routine	71
2.45	Experimental tests operational flow diagram	72
2.46	Packet format according to each type of data	74
2.47	Structure under test	75
2.48	Nodes arrangement	75
2.49	Comparison of ADXL355 (a) and ADXL355 (b) under no stimulation . .	76
2.50	Time response comparison for the ADXL355 (a) and ADXL345 (b) accelerometers, on node 5	77
2.51	Time response comparison for the ADXL355 (a) and ADXL345 (b) accelerometers, on node 5, after structure alteration	77
2.52	X-axis spectrograms of the main (a) and secondary (b) accelerometers from node 5	78

2.53	Z-axis spectrograms of node 3 before (a) and after (b) structure alteration	79
2.54	Detail on the structure time response before (a) and (b) after the strcuture alteration	80
3.1	3D printing methods: Stereolithography (SLA) (a), Aerosol Inkjet Printing (AIP) (b), Fused Deposition Modeling (FDM) (c).	83
3.2	E3D ToolChanger 3D printer.	84
3.3	Working principle of the analyzed accelerometric sensor	85
3.4	Rendering of the accelerometric sensor in the slicing phase	85
3.5	Test setup for the proposed FDM accelerometric sensor.	86
3.6	Step responses from the three FDM accelerometric sensors	87
3.7	Frequency responses from the three FDM accelerometric sensors	87
3.8	Input-output characteristics from the three FDM accelerometric sensors .	88
3.9	Conductive patterns used in the analysed strain gauge sensors.	89
3.10	Strain gauges during (a) and after (b) the printing process.	89
3.11	Curved samples for the strain gauge characterization.	90
3.12	Resistance measurement for the wavy (a) and straight (b) sensors with 20 mm radius sample.	90
3.13	I/O characteristic of the straight FDM bend sensor.	91
3.14	Top (a) and bottom (b) sides of the strain gauge circuit interface.	91
3.15	Smart glove with bending sensors.	92
3.16	Flow diagram related to the smart glove firmware.	92
3.17	Movements of the thumb (a) and index (b) finger detected from the serial visualizer.	93
3.18	Protopasta-made samples for trace (a) and resistance (b) characterization.	93
3.19	Comparison between the theoretical and measured capacitance of predefined capacitor samples	94
3.20	Schematics of the 3D-printed active filters.	95
3.21	Printed active low-pass filter	95
3.22	Results from the 3D-printed active low-pass filters with Sallen-Key topology.	96
3.23	Results from the 3D-printed active high-pass filters with Sallen-Key topology.	96
3.24	Results from frequency sweeps on the low-pass filters used for the extraction of the cut-off frequency error DoE-based model	97
3.25	Distinction between single-ended (a) and differential (b) interface circuits.	99
3.26	Working principles of the differential capacitive sensors types.	100
3.27	Graphical high-level representation of the proposed sensor interface . . .	101
3.28	Schematic of the proposed interface circuit	101
3.29	Internal structure of the “Error Calculator” block	102
3.30	Representation of the modulation-demodulation procedure used in this work.	103
3.31	Interface H-bridge composed of sensor capacitors, Voltage-Controlled Capacitors (VCCs), and parasitic capacitors	104
3.32	Final Computer Numerical Control (CNC)-milled version of the presented circuit	105
3.33	Circuit transfer function: theoretical and measured trends	106
3.34	Response of the circuit interface to common-mode-related inputs	107
3.35	Equivalent circuit of the Second-Generation Voltage Conveyor (VCII) . .	108
3.36	Proposed internal circuit for VCII	110

3.37	Circuits of the proposed positive charge pump (CP+), above, and negative charge pump (CP-), below	111
3.38	Pumped supply voltages (a) and VCII output voltage dynamic (b). . . .	112
3.39	Circuit configuration for the α (a) and β (b) measurements.	112
3.40	Frequency domain result for the α (a) and β (b) parameters.	113
3.41	Frequency domain results for the X (a), Y (b), and Z (c) terminals. . . .	114
3.42	Time domain result for the transient test on the whole circuit	114
3.43	DoE screening table	115
3.44	Schematic circuit of the used Marx generator.	117
3.45	Schematic circuit of the used Marx generator with parasitic components. .	117
3.46	Pareto plots for the output voltage $V_{out,min}$ extracted from LTSPICE (a) and QSPICE (b).	119
3.47	Pareto plots for the fall time t_{fall} extracted from LTSPICE (a) and QSPICE (b).	120

List of Tables

2.1	LoRa systems comparison table.	43
2.2	Technical differences between ADXL355 and ADXL345 accelerometers. .	68
3.1	Summary performance table for the 3D-printed FDM accelerometric sensor.	88
3.2	Percentage error of the mathematical regressed model with respect to the strain gauge I/O average characteristic.	91
3.3	Values in the resistances counter-test.	94
3.4	Variation of the differential control voltage ΔV_{ctrl} with respect to input parameter x and circuit baseline capacitance C_{bl}	106
3.5	Comparison of the proposed interface with other capacitive sensor interfaces in literature.	107
3.6	After-screening input combinations per sample in the simulated Marx topology.	118
3.7	t-values and p-values from the $V_{out,min}$ parameter in the simulated Marx circuit.	120
3.8	t-values and p-values from the t_{fall} parameter in the simulated Marx circuit.	121

Chapter 1

Introduction

The industry, which is the economic sector focused on the manufacturing and production of goods, has evolved rapidly over the last century, enabling several paradigms and technologies that aim to improve the well-being of humanity. These technologies are mostly based on the use of Hardware (HW)-Firmware (FW) systems, which combine hardware and firmware to perform specific control tasks. HW-FW systems base the most recent industry in several points, such as:

- driving industrial automation since, with particular HW-FW combinations, they can execute real-time control and processing, essential to manufacturing workflows [1];
- enabling connectivity and data flow, achieved through field programmability and interfaces for communication standards;
- supporting predictive maintenance and analytics, since firmware can embed diagnostic capabilities and telemetry allowing the user (or Artificial Intelligence (AI)) to assess the future conditions of the systems [2];
- enabling human-machine collaboration, when hardware and firmware are implemented in cobots or adaptive systems;
- security, since the firmware design process has a specific step for the system security, and some solutions exploit novel technology [3];

These points assess the importance of hardware-firmware systems, which, in the modern industry background, are still being optimized in their form. For this reason, this thesis explores and reports some applications of full and partial hardware-firmware systems in an industrial background, in several specific topics, such as SHM, 3D-printing, or even DoE.

For a uniform understanding of the context, this chapter introduces the background of Industry 4.0 (Sec. 1.1) and 5.0 (Sec. 1.2), enabling the reader to assess the state of the art and directions for the future. Furthermore, the Hardware-Firmware paradigm is approached through the testing structures used in the industry, as discussed in Sec. 1.3, and the introduction of protocols used in both Industry 4.0 and 5.0, listed in Sec. 1.4. Once the HW-FW concept is introduced, it is deepened through its introduction in the microcontrollers domain, in Sec. 1.5.

1.1 Industry 4.0 — State Of The Art

Industry 4.0 is the current technological state that the industry is experiencing, and it is mainly characterized by the integration of digital technologies, such as the Internet of Things (IoT), cyber-physical systems, AI, big data analytics, and cloud computing, into the traditional manufacturing processes. Different from earlier industry revolutions, Industry 4.0 focuses on creating smart factories, where real-time communication involving hardware-firmware systems and decentralized decision-making is achieved.

These paradigms are mainly fulfilled through connectivity: smart networks, also known as industrial IoT, are multi-object sets where each part of the factory has an ID and can interact in an abstract space with the other parts of the industry. This interaction brings to interesting concepts, such as adaptive control, where a manufacturing process adjusts its parameters according to the response of each smart factory component. Additionally, data flow in the industrial IoT must be real-time to maximize time efficiency in the involved manufacturing processes.

Another important concept in the industry 4.0 is the predictive maintenance, where one or more parts of industrial machinery are monitored and, according to the wear, measured with specific methods in determined points, a hardware-firmware system is capable of predicting its lifetime and eventual maintenance needed. The importance of this method in the smart factories efficiency increase, imposed its popularity, with a wide variety of implementations [4].

On the other hand, software plays a crucial role in Industry 4.0, since advanced data analytics methods enable mass customization of products. In particular, data analytics plays a significant role in decision-making frameworks, since it combines statistical methods, computational techniques, and domain knowledge [5].

Other implications of software and firmware involve the use of augmented reality as a form of technological integration. Augmented reality overlays digital information onto the real-world environment in real time. The primary target of this technology is to enhance the physical world through the addition of interactive, computer-generated content, which can help the factory operators and workers to execute their jobs more easily and smoothly.

At its core, Industry 4.0 combines automation with intelligence, turning manufacturing processes into flexible, resilient, and competitive ecosystems.

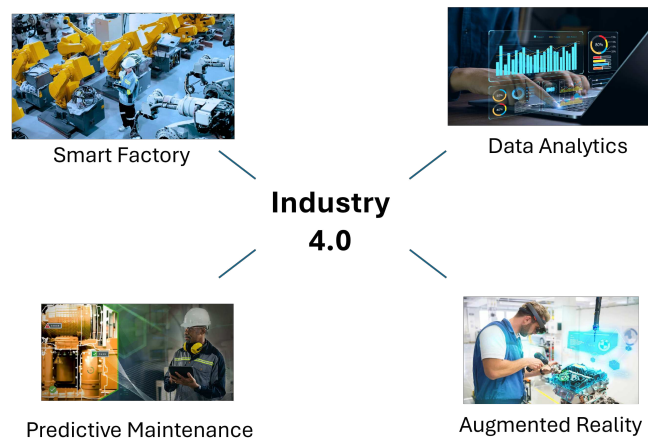


Figure 1.1. Main aspects of the Industry 4.0.

1.2 Industry 5.0 — The Future Industry

Industry 5.0 emerges as a natural evolution of the previous model, considering the beginning of the IoT-based integration, which then led to a relentless pursuit of efficiency and autonomy, sided by creativity, sustainability, and worker well-being. In this novel industry framework, the focus shifts from a technology-driven production to a human-centric sustainable model. These targets are assessed through tight collaboration, aiming for a symbiosis between human operators and advanced machines, with a specific focus on environmental goals due to the related current-decade crisis, as well as the creativity of users and technical operators. As a result, Industry 5.0 does not introduce novel tools aiming to replace instruments introduced by Industry 4.0; it rather employs them in new applications, integrating artificial intelligence.

Context understanding by machinery and smart devices is the first step in the human-machine symbiosis, and it is achieved through AI. In this regard, AI is the evolution of neural networks. Their working principle is applied to complex structures, such as the Large Language Models (LLM), Long Short-Term Memory (LSTM), using large word databases for their training. As a result, AI can be used in several ways, from research assistance to decision-making in several industry-related situations [6]. Additionally, human-machine symbiosis can be implemented through neurotechnology interfaces for direct machinery control using neural signals: in this regard, stimuli from the sensation of touch are still object of study in the neurosciences [7]; reversely, machine can partially control the human body through adaptive exoskeletons to support human operators during physically demanding tasks, nevertheless, the framework related to this matter is still under definition in the scientific community [8].

As mentioned, the environmental crisis has induced industry to target sustainability-related matters and circular economy, aiming to reverse the trend of pollution on the planet. In this regard, circular manufacturing platforms and tools, used to track, recover, and reuse materials across product lifecycles, are an important topic in the current and following industry framework [9]. On the same page, green materials engineering and the study of bio-inspired materials are another important subject for sustainability in industry; in this regard, important is the study of biomaterials for circuit substrates, especially for the healthcare [10]. Worth to mention is the use of specific carbon-absorbing concrete structures to additively manufacture buildings, as this technology could reverse the trend related to carbon emissions in the planet [11].

Human-centric design is an aspect that, in Industry 5.0, leads to more reliable and livable industry environments, as it employs AI to replicate human operators in simulation frameworks [12]. In particular, this factor is implemented through the creation of human digital twins to enhance ergonomic choices and simulate behavior affected by stress. Stress behavior, both in humans and machines, can significantly contribute to the implementation of safer designs from simulated smart factories; in this regard, workload control and labor flexibility are the primary paradigms pursued [13].

Finally, in Industry 5.0, wide importance is given to ethics. Ethics, both in terms of AI decision boundaries, safeguards for workers, and sustainability. The AI-related ethics is a problem coming directly from recent philosophy, with interest about topics ranging from machine awareness — although this can be turned into a measurable parameter [14] — to the technological singularity [15]; each topic

represents a foundation for the massive use of AI inside the industry. Additionally, AI-related ethics cover more practical topics such as the transparency and fairness of AI towards their users [16]; this topic, in the industry context, can be determinant since AI would hold entire important activities in the manufacturing processes, hence, a non-ethical AI would increase the manufacturing error considerably. Finally, ethics concerning sustainability represent a cornerstone in modern system design, as global warming, which causes the aforementioned environmental crisis, can lead to disastrous consequences for the planet, with the risk of irreversible climate change; hence, hardening the survival of humanity on the planet [17].

As a result, Industry 5.0 combines several elements of both industry and everyday life, with massive use of AI, promoting the psychology and creativity of the single human operator. This paradigm shift, from Industry 4.0, opens the way towards a more accessible concept of industry, where customization and the well-being of operators have priority over mass production of goods.

1.3 Test Methods In The Industry

The current industry is undergoing a smooth transition from the 4.0 to the 5.0 model, demonstrating the evolution in how the industry approaches efficiency, sustainability, and human-centric design. Both the industry models rely heavily on complex cyber-physical systems, advanced automation, and AI-driven decision-making, and these factors demand high levels of reliability. Reliability, by its purpose, is guaranteed through rigorous testing methodologies, such as the Hardware-In-the-Loop (HIL) and the Software-In-the-Loop (SIL), which enable safe, cost-effective, and realistic simulations of industrial scenarios. Furthermore, these methods allow hardware and firmware validation in controlled environments. In particular, hardware-firmware systems are coordinated by microcontrollers, which play a central role. Complex industry implementations require a multi-staged process to link the abstract components — software or firmware — to the physical parts — hardware. This process involves:

- Algorithmic tests: SIL models
- Instrumentation tests: HIL models

In the first case, SIL is based on simulating a specific firmware on a microprocessor, without any real-time requirements, for a specific application, as represented in Fig. 1.2. In particular, the firmware is compiled and executed in a hosted environment, e.g., Simulink; hence, sensor inputs, actuators, and environment models are simulated. Therefore, no hardware is involved in this model.

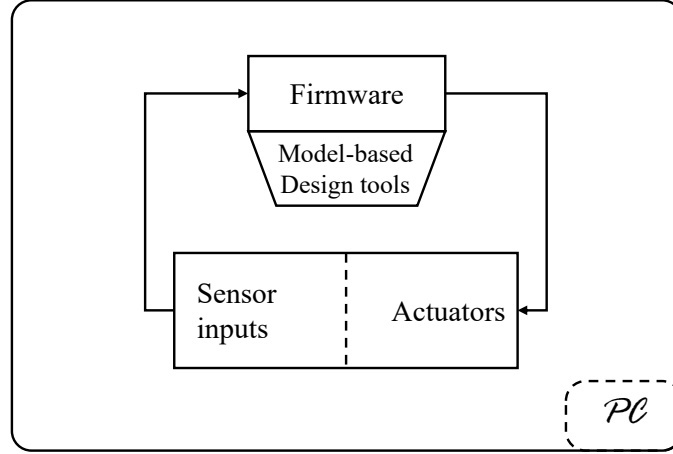


Figure 1.2. Graphical representation of the SIL model.

The SIL approach is generally used to test the logic behind a firmware, before its loading on the actual hardware. Consequently, the SIL aims to guarantee full observability of any variable, early debugging of logic and algorithms, and easy integration with model-based design tools.

On the other hand, the HIL, as represented in Fig. 1.3, tests the entire Device Under Test (DUT), composed of its hardware and firmware parts, using a real-time microprocessor. The real-time microprocessor reproduces, through real-time-based models and the use of Digital-to-Analog Converters (DACs) and ADCs, both the sensor inputs and the actuators.

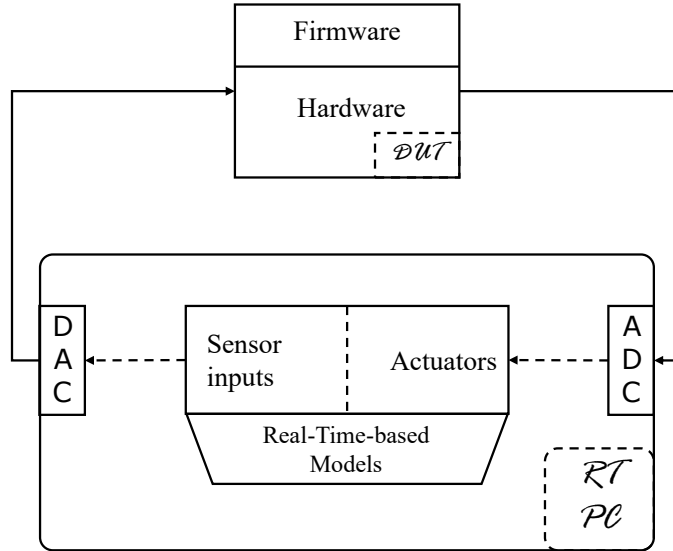


Figure 1.3. Graphical representation of the HIL model.

In this model, the DUT firmware differs slightly from the firmware tested in the SIL, as it is upgraded to consider both the hardware beneath and any real-time requirements. The HIL model primary target is to perform a validation test of the hardware-firmware system in situations approaching the reality, where real-time constraints, dangerous or edge scenarios, are simulated. In particular, due to the need for real-time operation, sensor inputs and actuators cannot be simulated in an ordinary microprocessor: they are simulated either on a real-time microprocessor

integrated with peripherals, or other devices such as Field-Programmable Gate Arrays (FPGAs). Consequently, HIL allows safe and repeatable tests without using real instrumentation; hence, by limiting its wear, it permits observing and debugging any timing issues in the system, e.g., interrupt conflicts, jitters, delays, and detecting any interface issues.

1.4 Communication Standards In The Current Industry

As mentioned above, industrial systems are based on the interrelationship between hardware, firmware, and software. Specifically, the best efficiency in industrial systems is accomplished through seamless and reliable cooperation between these parts. This interoperability is ensured through the use of industrial protocols. This section gives an overview of the industry protocols used in both the 4.0 and 5.0 industry frameworks.

1.4.1 Industry 4.0

Industry 4.0 is primarily characterized by the introduction of the Industrial Internet of Things (IIoT), where IoT protocols are re-adapted to the industry setting, introducing narrow time windows for each operation.

MQ Telemetry Transport (MQTT) [18] is one of the first protocols to be imported from IoT. This protocol was primarily designed for lightweight messaging, such as satellite links [19], and then adapted for industry applications including telemetry [20], edge-to-cloud communication [21], and SCADA integration [22].

Constrained Application Protocol (CoAP) [23], once used for peer-to-peer communication between low-power devices, then adapted in IIoT for machine-to-machine communication, resource discovery, and specific industrial services, e.g., REST [24].

Data Distribution Service (DDS) [25], initially used for IoT middleware and distributed systems, then extended to IIoT for real-time reliable pub-sub messaging.

Long Range Wide Area Network (LoRaWAN) [26], a wireless protocol famous for its range, reaching up to 90 km, used for IoT applications such as remote sensing and smart cities [27], and later used with the introduction of more rigorous requirements in the industry, for applications such as remote monitoring [28], predictive maintenance [29], and asset tracking for large industrial sites [30].

On the other hand, the stringent industry necessities, such as the real-time operability, the use of already-known communication standards, e.g., RS-232 [31] and RS-485 [32], and smooth migration among diverse systems, with different protocols, built the basics for several industry in-born IIoT standards. These protocols are foundational in enabling smart manufacturing, automation, and data security.

The oldest, but still used, industrial protocol is the Modbus [33], widely used for serial communication and evolved in three different directions depending on the used means, and encoding principle: Modbus RTU, using reduced terminal units, i.e., RS-232 and RS-485 standards, Modbus ASCII, using ASCII-encoded messages and Modbus TCP/IP, using the TCP/IP as protocol infrastructure, compliant for ethernet networks. Although its robustness, Modbus still lacks features such as security, to be implemented directly on the system front ends, or the migration to

other protocols, such as EtherCAT or Ethernet/IP, in specific regions of the industrial system.

Another old, but still important pair of protocols, is the couple PROFIBUS [34] & PROFINET [35]: standards for fieldbus communication and real-time ethernet. Their main characteristic is the high reliability combined with high-speed deterministic communication. While these features properly describe PROFIBUS and PROFINET, these standards are proprietary; hence, while this can imply a smooth migration between the two protocols themselves, it also leads to several difficulties in using non-proprietary protocols and standards in the industrial network.

For seamless motion control and high-speed automation, Ethernet for Control Automation Technology (EtherCAT) [36] was designed and introduced. It uses Ethernet frames in a “processing-on-the-fly” approach, where each slave device reads/writes data as the frame passes through. This industrial protocol is characterized by multi-device scalability, sub-microsecond latency, and high time determinism, excellent for robotics [37] and CNC applications [38]. Notwithstanding these strength points, EtherCAT is generally complex to configure and less focused on interoperability outside real-time automation. Nevertheless, this protocol proved to be an ideal choice for Industry 4.0 applications such as cobots [39], exoskeletons [40], and adaptive production lines.

Ethernet/IP, also known as Ethernet Industrial Protocol [41], is an industrial standard that extends the Common Industrial Protocol (CIP) to Ethernet. Widely adopted in North America, its producer-consumer model enables real-time data exchange and implements the easy integration of control, safety, and motion over a single Ethernet backbone. It requires a robust network design to ensure the right amount of time determinism under heavy traffic, while still providing a strong migration path for factories with existing CIP infrastructure.

For sensor/actuator-level communication, the IO-link standard was introduced and defined in IEC61131-9 [42]. It works as a point-to-point serial communication standard over 3-wire cables. In addition to its plug-and-play nature and its full compatibility with standard binary sensors and actuators, it transmits both process and diagnostic/configuration types of data. Unlike a fieldbus protocol, IO-Link typically connects to higher-level networks only through gateways, which limits its use.

Finally, Open Platform Communications Unified Architecture (OPC UA) [43], which pioneered the nowadays AI-based protocols and models, is a widely used protocol in automation and industry 4.0; differently from ordinary protocols, OPC UA has a semantic-based structure, allowing machines and applications to share contextualized information. Although being a resource-intensive IoT protocol, OPC UA is vendor-neutral, standardized, and secure. It is considered as Industry 4.0 major protocol, combined with custom specifications for digital twins, robotics and cloud integration.

1.4.2 Industry 5.0

As mentioned above, Industry 5.0 is built upon Industry 4.0 and emphasizes human-centric, sustainable, and resilient systems that integrate advanced communication technologies. In this regard, the most used Industry 5.0 standards and protocols are listed and deepened in this section.

As a wireless replacement for wired fieldbus systems, 5G technology offers ultra-low latency (down to 1 ms), high bandwidth, and massive device connectivity. Those features make this standard ideal for real-time industrial automation, remote monitoring, and autonomous robotics [44]. Additionally, 5G technology supports mobile robots, drones, and Automated Guided Vehicleless (AGVs) in smart factories. As an integration, Wi-Fi 6 (IEEE 802.11ax [45]) enhances 5G reliability, reduces interference between multiple devices, and improves the handling of dense device environments, thanks to its support for high-throughput, low-latency communication for industrial IoT devices operating in indoor environments.

Low-Power Wide-Area Network (LPWAN) is a concept focused on the use of long-range, low-power, low-data-rate communication technologies such as LoRaWAN, Narrow Band (NB)-IoT, and Sigfox [46]. In this sense, LPWAN is the key solution when the connection of battery-powered sensors [47], asset trackers [48], and remote monitoring devices in large areas [49] is the main problem. This set of protocols is primarily used in Industry 5.0 to help monitor environmental sustainability metrics, such as CO₂ levels, energy efficiency, waste tracking, etc., over both indoor [50] and outdoor [51] implementations.

Blockchain is a safety-based model introduced originally in Industry 4.0 to ensure trust and transparency in distributed industrial networks through the implementation of an immutable ledger for supply chain management, traceability, and certifications involved in processes. Industry 5.0 employs blockchain as a tool to support circular economy initiatives [52], enabling the proof of sustainable sourcing, recycling, and product lifecycle compliance by manufacturers. Protocols that implement blockchain in an industrial background are Ethereum [53], Hyperledger Fabric [54], and IOTA [55].

Zero-Touch Network Management [56] is another protocol and standards concept that is being introduced in Industry 5.0 to implement self-configuring, self-healing, and self-optimizing networks in complex industrial environments. This framework is based on the AI-driven automation for the humanless management of networks. In fact, the zero-touch management drastically reduces the need for manual setup in scalable IIoT deployments and boosts the human-machine collaboration paradigm implementation.

Finally, the introduction of deeper AI-based architectures and the rise of quantum computing caused the urge to increase the security behind industrial systems; for this reason, the Zero Trust Security models [57] have become a starting point in Industry 5.0 cybersecurity. These networks are based on the principle “never trust, always verify” where, differently from traditional perimeter-based security, every device, user, and service requires continuous verification, encryption, and identity checks as it is assumed as potentially compromised. In Industry 5.0, a cybersecurity framework with these features is crucial for protecting human-machine interfaces, wearable devices (such as exoskeletons) [58], and sensitive data, especially in collaborative workspaces [59].

1.5 Hardware and Firmware

This section deepens the hardware used in the industry, implementing the protocols discussed in Sec. 1.4, the test methods reported in Sec. 1.3, and the applications reported in Secs. 1.1 and 1.2. Initially, the various types of instruments used in

the industry are discussed, with particular attention to microcontrollers. Then, microcontrollers are deepened in both their hardware and firmware parts.

1.5.1 Instruments for the industry

As mentioned above, reliable automation in industry is a concept that can be implemented with a very wide set of instrumentation on various levels.

The most used and well-known instrumentation begins with the microprocessors, famous mostly because of their large usage domain. Nevertheless, industry relies on high-performance and real-time microprocessors as determinism in processes and tasks is required. Real-time processors, in industry, are often employed in industrial PCs which, differently from ordinary PCs, are specifically structured for real-time applications, input-output interfacing, and harsh-conditions operation. In particular, apart from branded implementations, e.g., Siemens, industrial computers, containing real-time processors, can be assembled and configured according to the industry requirements. High-performance processors, in the industry, are mostly used in workstations, intended for office environments where design is performed. As a matter of fact, workstations focus rather on performance than determinism and responsivity.

Furthermore, a subset of processors is designed for some specific tasks, like signal processing. For this purpose, Digital Signal Processors (DSP) are introduced and used. Mainly, DSPs are microprocessors optimized for the execution of Multiply-and-Accumulate (MAC) tasks, which are the basis of signal filtering. In particular, DSPs can implement Finite Impulse Response (FIR) filters: totally digital filters composed of a finite number of registers and a filtering window, which implements the corresponding function. DSPs represents the most electronics-based part of the industry using microprocessors.

Additionally, microprocessors are used in Programmable Logic Controllers (PLCs), which are an intuitively programmable instrument for the automation of large industrial production chains. Although it needs proper qualification to be used, a PLC can be programmed by using both graphical and written languages: in this sense, due to their ease of use, graphical languages are the most used by specialists. PLCs can be described as a little, simpler, implementation-friendly, industrial PC, with the possibility of expansion through special-purpose programmable modules. This instrument satisfies the real-time requirement more than the others as it is used to manage very large production plants and chains.

Processors are also used in Data Acquisition (DAQ) systems, acting as a fine linker between physical and digital worlds. DAQ systems are high-performance systems used to implement the methods described in sec 1.3. A DAQ system is mainly composed of

- sensors and transducers for measuring physical quantities and converting them into electrical signals;
- signal conditioning circuitry, where signals coming from sensors are amplified, filtered, isolated, and excitation for active sensors is provided;
- ADC conversion to convert physical signals into digital;
- Central Processing Unit (CPU) to process the digitalized signals;

- software, e.g., MATLAB or LabVIEW, for further signal processing and to implement specific functions;

As a consequence, DAQ systems are widely used in the testing subfield of industry.

Alternatively to microprocessors, FPGAs are the paradigm switch that allows easy and faster computing. In particular, the working principle of FPGAs is decentralized, based on the cooperation of several building blocks — known as cells — with a predefined structure. These blocks are interconnected through reconfigurable paths; hence, the main concern behind FPGA efficiency is the number of used cells and the longest paths selected. FPGAs have a central role in signal processing, as they can be explicitly programmed to perform the MAC function. Nonetheless, they can be re-programmed both in specific hardware-based languages, known as Hardware Description Languages (HDLs), and in high-level graphical languages, such as Simulink, which uses translators from high to low coding levels. Although the language choice is important due to optimisation-related reasons, language modularity allows easy programming and is inspired by the structure of digital systems. Furthermore, FPGAs can either have a central role in specific applications or can be used for prototyping reasons.

Finally, microcontrollers are another important alternative to microprocessors and FPGAs used in the industry. Microcontrollers are used when easy implementation, programming, and interfacing are needed, without necessarily needing real-time computation or high performance — although in some cases microcontrollers can satisfy these last requirements. In particular, microcontrollers can be used for sensor interfacing and to implement isolated operating modules in a system, such as slave nodes in multi-nodal systems. Microcontrollers, like industrial PCs, are composed of a central low-power processing unit connected to several peripherals for reading external signals and, eventually, generating signals for the external environment. For this thesis, the structure of a microcontroller, with specific details about programming, is discussed.

1.5.2 Microcontrollers: What Do We See

In the nowadays industry, the most well-known microcontroller manufacturers are

Arduino producing highly affordable and easy-to-use microcontrollers. In the years, Arduino manufactured several different variants of their boards, integrating a wide set of functionalities, from Wi-Fi to the integration of FPGAs. Arduino remains a good alternative for easy and affordable programming, with no demand for high performance;

ESP32 producing, like Arduino, highly affordable and easy-to-use microcontrollers. In particular, although ESP32 produces fewer variants than Arduino, ESP32 features higher power in terms of both dissipation and computation; hence, it is used for more specialized applications.

STMicroelectronics due to the computational power, internal complexity, and numerosity of variants, STM boards are designed for the industry. Although the high computational power, they use specific low-power architectures and high-performance processors, enabling their suitability for industrial applications.

Each manufacturer has its specific microcontroller structures, but all the microcontroller boards share a basic setup, reported in Fig. 1.4, to work properly.

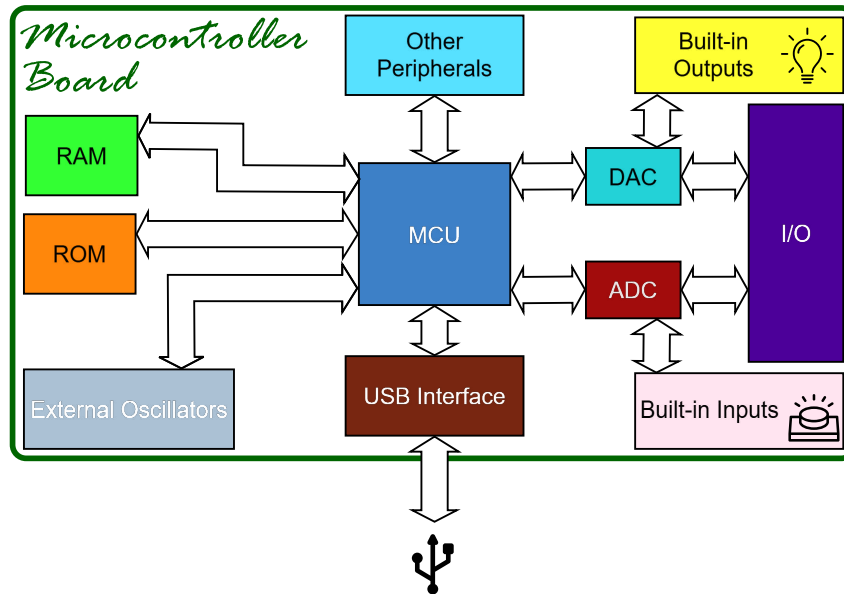


Figure 1.4. Graphical representation of microcontroller board parts.

As a matter of fact, the MCU is the center of the board since it directly manages the steps of every single instruction. In particular, the MCU fetches, decodes, and executes instructions according to the program stored in the non-volatile memory. Firstly, the MCU is connected to the input-output peripherals through ADCs and DACs, linking hardware analog and digital parts. Many microcontrollers have both built-in inputs and outputs to ease basic board tests, such as LED blinking, or the execution of code after pressing a button. Additionally, the MCU communicates with volatile and non-volatile memories: Random Access Memory (RAM) and flash, respectively. Mostly, oscillators providing signals at specific frequencies are used externally to the MCU. Since microcontrollers are reprogrammable, every microcontroller board comes with an ad-hoc USB interface for the MCU. Finally, some microcontrollers are built with a specific feature, e.g., Wi-Fi in Arduino MKR WiFi 1010 (fig. 1.5), or can be expanded with special-purpose modules, like the STMicroelectronics boards; in both cases, the MCU is connected to special-purpose peripherals depending on the context. In every MCU-peripheral connection, an ad-hoc protocol is used, depending on the nature of the peripheral. Each element of a microcontroller board will be discussed in the next subsection.

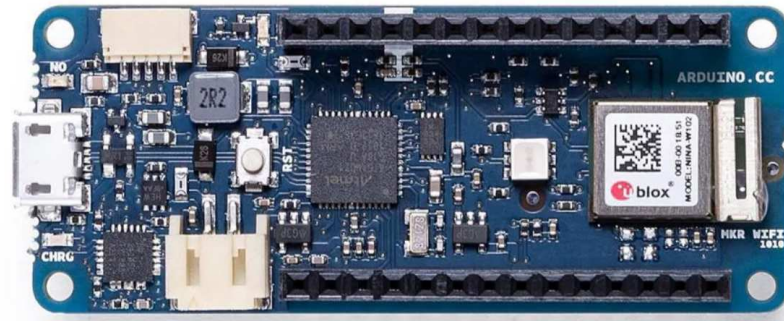


Figure 1.5. Arduino MKR WiFi 1010.

(<https://www.mouser.it/ProductDetail/Arduino/ABX00023?qs=%252BEew9%252B0nqrAwxv2YQYWyPw%3D%3D>)

1.5.3 Hardware in Microcontrollers

Microcontrollers are the most practical example of hardware-firmware linking, since they are composed of a central processing unit, known as MCU, and several peripherals. These peripherals fulfil communication-related functions, from the ADC/DAC conversion to the implementation of communication protocols and interfaces (Universal Asynchronous Receiver-Transmitter (UART), Inter-Integrated Circuit (I²C), GPIO, Serial Peripheral Interface (SPI) etc...), timing functions, through the use of crystal oscillators, or storage functions, through the use of external volatile and non-volatile memories. In particular, according to the availability of these peripherals and the timing of the MCU, there are several types of microcontrollers, from different manufacturers. Since microcontrollers are used in a wide range of applications, each with its specific requirements, it is essential to note that both the hardware and firmware must be tailored accordingly. In particular, each firmware can be related to a certain hardware, but the reverse is not true. According to this important clarification, the hardware of an MCU is specifically deepened in this section. Firstly, hardware from an MCU has a different performance than its CPU counterpart; this is due to the main targets behind the use of these two instruments. As a matter of fact, in non-real-time situations, the CPUs aims for the best performance per clock cycle, without considering time, while MCUs points to optimize the control per unit; this difference is one of the causes of decentered computational power in MCUs. As a consequence, unlike CPUs, a non-centered computational power causes fewer bottlenecks: CPU suffers from core-memory and Arithmetic Logic Unit (ALU) bottlenecks due to the centralization of computational power. The main parts of a microcontroller board, as mentioned in the previous section, are

Oscillators they provide an external frequency with respect to the MCU oscillators

and contribute to the generation of the clock signal. Oscillators express how fast operations in an MCU are performed and how fast peripherals and modules are driven. Mainly, there are three types of oscillators:

- RC Oscillators, based on the related circuit: they are low cost and low area, with quick start, but limited frequency accuracy;
- Crystals and Ceramics, based on piezoelectric components which, once powered, produce an almost-constant frequency alternating signal: the accuracy is high, they occupy little area and have a quick start, but they are expensive;
- Phase-Locked Loopss (PLLs), circuits that, given an external oscillating signal, multiply its frequency relying on mixing techniques: they are a cheap alternative to crystal-based oscillator, but their accuracy depends exclusively on the input oscillator; additionally, since it is a feedback circuit, they do not have a quick start when turned on and occupy more area than the previous oscillator types.

Oscillators are generally implemented outside the MCU, but in some specific situations, where strict frequency requirements are needed, they are implemented internally to the MCU.

I/O inputs and outputs, they enable the MCU interfacing and interaction with external sensors and circuits. Microcontrollers integrate a wide range of I/O peripherals directly on-chip, reducing the need for external components. Peripherals such as GPIO, protocol-related peripherals, or specific interfaces are employed for this purpose. GPIO is a digital peripheral based on the use of transistors, for the pull-up, pull-down, open drain configurations; tri-state buffers for the enabling of different input/output modes; schmitt triggers for the reading of values; and registers, for their storing. GPIO does not have an ad-hoc structure for communication with the MCU. A proper GPIO interface, whose schematic is reported in fig. 1.6, is composed of several control signals depending on the implementation.

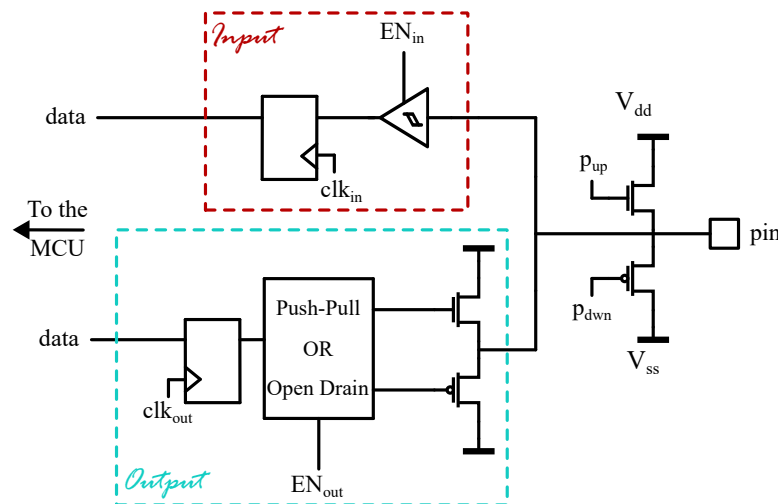


Figure 1.6. Representation of a basic GPIO interface.

Communication-based interfaces are a digital type of I/O whose structure is inspired by existing communication protocols, e.g., UART, SPI, I²C; hence, there is a fixed number of wires (signals), each one of them with a purpose, and implementation-independent. Specific interfaces are mainly drivers for predefined input types, such as touch, LCD, camera, or audio inputs. The primary aspects of I/O peripherals are their multiplexing and configurability, which define the capability, for a single pin, to be customized in terms of mode, function, speed, interrupt events support, and the ability to disable any pins for power management reasons.

ADC is a mixed-signal peripheral used to convert analog signals into digital. As depicted in fig. 1.7, it works by sampling and holding an analog value provided in a certain time instant, which is then quantized to a fixed value and encoded in binary format.

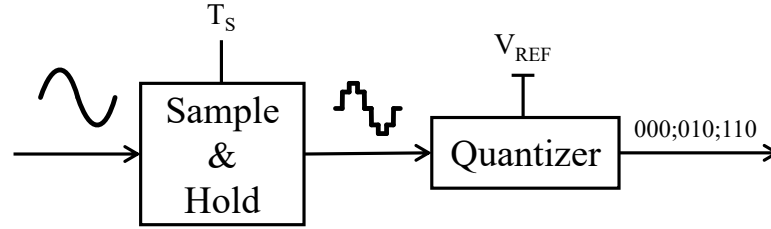


Figure 1.7. Block representation of an ADC.

The sample & hold operation is performed with a specific sampling time T_s , while quantization is performed with a specific resolution. Both these parameters constitute the main specifications for ADCs, since sampling time is related to the fastest digitizable analog signal, while resolution is related to the number of bits readable from the digital side, hence, to the precision of the ADC. Additionally, conversion speed is an important parameter to consider in ADCs, as it can be a determining factor among different ADC typologies. As an example, the Successive Approximation Registers (SAR) converter has a balanced trade-off between speed and accuracy, but it has an average resolution, $8 \div 16$ bits. On the other hand, the sigma-delta ($\Sigma\Delta$) converter can reach very high resolutions — around 20 bits — with high signal-to-noise ratios, but long settling times and complex structure, as it is feedback-based. Another worth mentioning ADC typology is the Flash ADC, known because of its high speed, compensated with a big area, high power consumption, and high costs. In most implementations, ADCs are placed outside the MCU and combined it with other MCU-external peripherals, such as the GPIO.

DAC reversely to the ADC, the DAC brings signals from the digital to the analog domain. It is generally used with output peripheral interfaces, so that signals incoming from the MCU can interact with analog external circuits or devices. Similarly to the ADC, one of the DAC main parameters is the resolution which is related to the bits number N on the digital side, converted into the number of steps dividing the output dynamic range, as given by the (1.1).

$$\text{Step Size} = \frac{V_{dd} - V_{ss}}{2^N} \quad (1.1)$$

Where V_{dd} and V_{ss} are the positive and negative supply voltages, respectively. Additionally, the actual output range and settling time are important parameters in the choice of a DAC. In this regard, there are different types of DACs:

- the resistor ladder (R-2R), implemented as a granular voltage divider where each branch corresponds to a specific analog voltage. This type of DAC is the average choice due to its simple structure, almost absent settling time, and limited noise; this DAC is generally employed in mid-resolution applications;
- the sigma-delta DAC, whose working principle and advantages/disadvantages are the same as the corresponding ADC-based topology;
- the current steering converter, which converts the digital input into a current that, when fed to controlled resistors, gives an output analogue voltage: this topology is used in high-performance designs.

Like ADCs, DACs are implemented outside the MCU to combine them with external peripherals.

flash memory non-volatile memory, it stores the program, which is the series of instructions for the MCU. Due to the long-term information retention, flash memories are implemented with floating gate transistors, represented in fig. 1.8. Floating gate transistors are a specific type of Metal-Oxide Semiconductor Field Effect Transistor (MOSFET), where a special non-accessible gate, hosting charge and giving the memory effect, is grown in the oxide underneath the external gate [60].

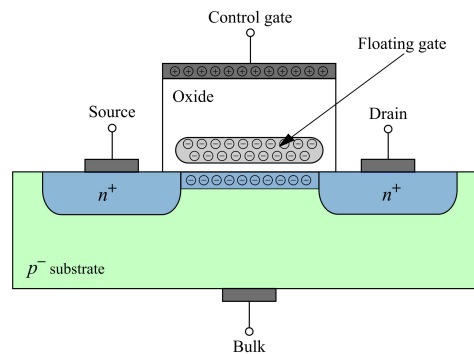


Figure 1.8. A floating-gate transistor [60].

Flash memories have memory capacities approaching the tenths of megabytes for advanced microcontrollers, with a variable granularity among the writing and erasing operations, the first performed in words or pages, the second in blocks or sectors, in order to have fine time management. In particular, read operations are the fastest, while write and erase operations are slower, and in some specific cases they need to be backed by the Direct Memory Access (DMA) peripheral. Some flash memories have a specific area for bootloader code hosting. This availability is crucial since the bootloader allows updates of the structural MCU firmware. Flash memories are generally integrated into the MCU to minimize time and resources during operations. Nevertheless, in some cases

flash memories can be implemented externally to the MCU chip and operate in parallel.

RAM volatile memory, it stores all the types of variables in the code, as well as addresses and function calls. RAM memory can be static (SRAM) or dynamic (DRAM): in the former case, it is implemented with flip-flop-based structures and requires refreshing only in presence of operations, while the latter uses capacitors to store data, hence needs to be refreshed at regular intervals. Due to the enhanced memory efficiency, SRAM is mostly used in microcontrollers. The circuit of an SRAM memory is represented in fig. 1.9.

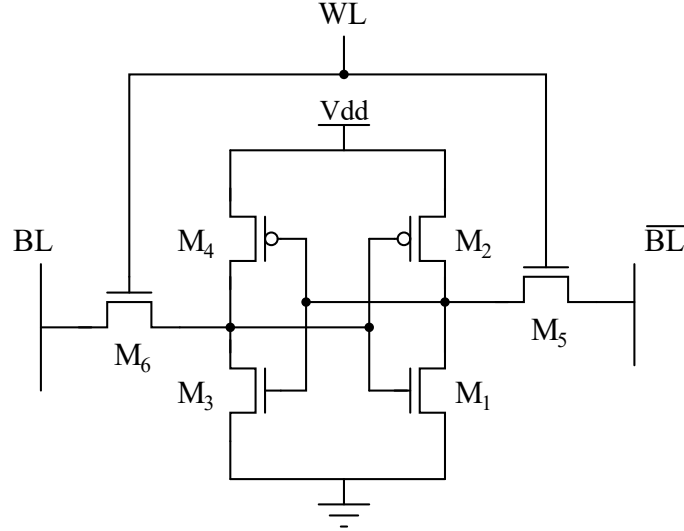


Figure 1.9. Circuit structure of a SRAM memory.

According to fig. 1.9, the flip-flop structure is delimited by the M_1 - M_4 transistors, whose programmability is directly controlled by the word line (WL) and the bit line (BL) bits. Due to their purpose, the capacity of RAMs are lower than flash memories, and amounts to the hundreds of kilobytes. Consequently, latency and data intervals are considerably lower. In particular, the data rate, measured in transmissions per second (T/s), amounts to

- $800 \div 2133 \text{ MT/s}$ for the older DDR3 RAM family;
- $1600 \div 3200 \text{ MT/s}$ for the DDR4 RAM family;
- $4000 \div 8000 \text{ MT/s}$ for the most recent DDR5 RAM family;

Generally, RAMs are integrated in the MCU to provide high speed, simpler design, lower costs, and more reliability.

The relationship between MCU and memory is crucial, since it can be the cause of any possible bottlenecks in the microcontroller structure. In particular, two main architectures are used in MCUs: the Von Neumann and the Harvard.

In the Von Neumann architecture, as reported in fig. 1.10, data and program instructions are stored in the same flash memory. Consequently, data and programs communicate with the MCU through the same system bus.

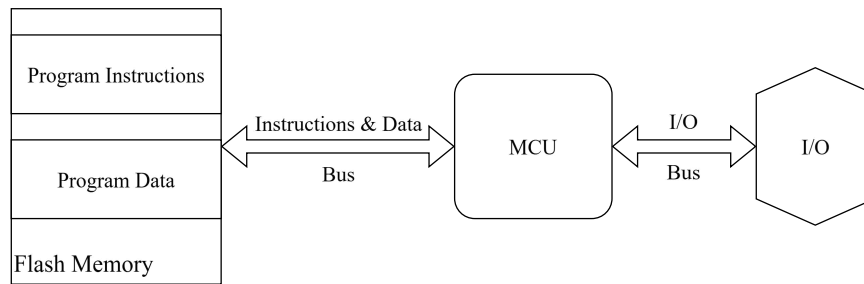


Figure 1.10. Representation of the Von Neumann architecture.

The common system bus for the MCU-memory communication brings both advantages and disadvantages. In particular, the main disadvantage that distinguishes Von Neumann architectures is the notorious “memory bottleneck”: since data and program memories are shared, hence, the flash memory has a relatively big size, the access of a specific register can take a non-negligible amount of time, lagging machine operations. This situation may be worsened when the MCU gets stuck in an access attempt, stalling the entire running program. On the other hand, the single bus feature enabled the use of several specific solutions, such as

- pipelining, where each processing action (fetch, decode, execute, store) is parallelized in the MCU to decrease the processing time (discussed below);
- caching, where frequently used data is stored in a smaller memory inside the MCU, called cache, in order to be easily and fastly accessed when needed;
- fetching, where the system predicts the needed data in the next instructions, hence it is fetched from the flash memory to the cache anticipatively;
- self-modifying code, where code alters its own instructions during its execution to reduce the instruction path length, thus, improving storing efficiency.

These solutions were mainly designed to mitigate bottlenecks through efficient time and space management in the flash memory. Von Neumann architectures, simple and costs-efficient, are mainly used in microprocessor-based systems, like PCs, smartphones, servers for cloud computing, and gaming console, where high-performance MCUs are used, in order to compensate for the Von Neumann bottleneck. In particular, the Von Neumann architecture, generally, invests rather on the MCU performance rather than structural complexity.

Differently, Harvard architecture, represented in fig. 1.11, relies on the use of separate memories for program instructions and data. Therefore, the central MCU communicates with the program and data memories through separate buses, thus, making it possible to fetch an instruction and read or write data at the same time.

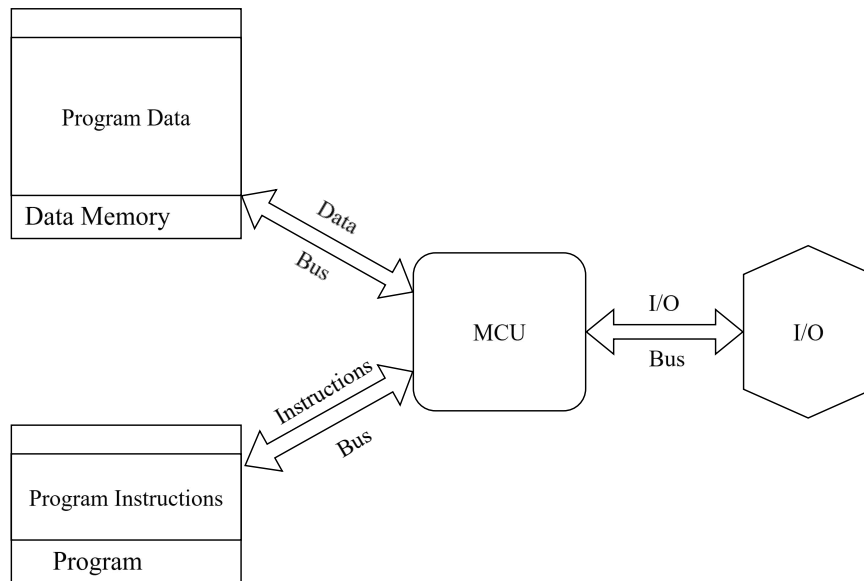


Figure 1.11. Representation of the Harvard architecture.

Consequently, the whole system performance is enhanced with the Harvard architecture due to the hardware-enabled pipelining; this allows the implementation of program and data memories with specific technologies according to the implementation requirements. Additionally, this structure increases the operations predictabilities, as data is stored in a specific memory. In fact, Harvard architectures are suitable for real-time applications. Although this structure overcomes the Von Neumann memory bottleneck, Harvard systems are complex and expensive. Additionally, in strict Harvard implementations, instructions cannot be treated as data, hence, self-modifying code is restricted. To address these limitations, the following modification have been integrated over the years:

- unification of addressing for data and program, division is left to the hardware;
- implementation of caching and prefetch on the MCU unit;
- implementation of self-modifying code with specific bridging techniques to read data from program memory.

Due to their characteristics, Harvard architectures are mainly employed in performance-critical applications, such as DSPs, embedded systems, network processors, or automotive systems. Consequently, the Harvard architecture invests in design complexity rather than MCU performance; hence, they can be implemented with less performant MCUs than the Von Neumann architectures.

As mentioned, for efficient MCU resources usage, pipelining is a widely used technique in both the Von Neuman and Harvard architectures. Pipelining consists of dividing each instruction in defined, repetitive phases, to increase the predictability of each MCU operation. Instruction phases are executed in fixed time slots such that, through parallelization, each one of them is executed in a single time slot. Fig. 1.12 reports an example of this method, where the single instruction is divided in the following phases:

fetch the instruction code is loaded from the instruction memory;

decode the instruction code is decoded, hence, the MCU decodes the operation related to the instruction;

execute the instruction is executed;

store the execution result is stored in the data memory – with no consideration of program branches and jumps as instructions, for the sake of simplicity.

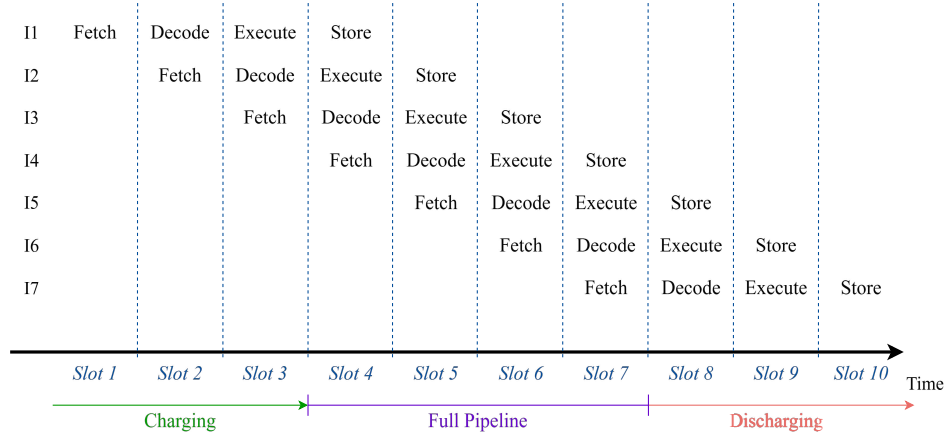


Figure 1.12. Representation of pipelining with four-phases instructions.

Primarily, this method increases the instruction throughput in a single MCU, reaching full efficiency when one instruction per clock cycle is executed. On the other hand, the pipelining technique enables the clock frequency increment with short instruction phases. In particular, instruction phases are parallelized according to their suitability on different MCU parts. In fact, the MCU architecture defines the number of instruction phases chosen for the pipeline; consequently, the number of instruction phases affects MCU complexity, execution time, and power consumption.

Pipeline can both improve resource usage and bring some designing hazards in an MCU. In particular, these hazards are caused when two or more instructions request the same resource or variable, namely, when instructions inter-dependence occurs. This issue is important to consider with conditional branches and jump instructions: in both cases, the next instruction to fetch is uncertain. Moreover, pipelining can cause structural hazards, where hardware resources are insufficient for multiple simultaneous operations. The introduction of pipeline stalls, i.e., specific wait cycles, can mitigate the resource conflicts in a pipeline-friendly MCU. Alternatively, the forwarding of intermediate results to dependant stages, or the outcome guessing from conditional branches, are feasible solutions for the management of pipeline-related hazards. As a result, low-end MCUs feature pipelines with two or three stages; mid-range MCUs integrate branch prediction and prefetching mechanisms for pipelines; while in high-performance MCUs more complex and specific pipeline architectures are implemented.

Finally, a determinant part of MCUs is the bus matrix. The bus matrix not only interconnects every part of a microcontroller board but also manages data transfers with specific methods depending on the vendor. In particular, the bus matrix directs requests from masters to slaves and, in multi-master systems, such as boards with MCU and DMA, it directly manages and contains delays. As a result, a simple bus

matrix can handle one request per slave, while a multiple bus matrix can manage more requests with prioritization schemes. Some examples of bus matrices are

CoreConnect the oldest bus matrix, used for high-performance systems on chips, guaranteeing modularity, with different bus types related to the microcontroller area: a high-performance bus for the MCU-memories communication, a peripheral-dedicated bus, and a bus dedicated to the traffic configuration;

AMBA Advanced Microcontroller Bus Architecture, which, like the CoreConnect, features more bus types for different tasks, e.g., the AHB bus for high-performance communication between MCU and memories, and the APB bus for the peripherals;

AvalonBus bus matrix for FPGAs, it supports multi-master structures, but it focuses more on the single FPGA implementation rather than referring to standard structures;

WishBone open source bus, featuring flexibility, simplicity, interoperability, and allowing for the integration of IP blocks with no licensing constraints.

Additionally, buses can be synchronous and asynchronous: synchronous buses contain a clock path among their signals and feature short master-slave distances, with low flexibility on the diversity of the connected devices, as a clock port is needed. On the other hand, asynchronous buses do not comprise a clock signal, as specific moments in the communications are acknowledged through specific level transitions. For this reason, asynchronous buses offer a wider range of diversity and are primarily used in input-output devices.

1.5.4 The Microcontroller Unit

This section delves deeper into the internal structure of the microcontroller unit, examining both its hardware and firmware.

Hardware

MCUs are digital systems; hence, their structure, represented in fig. 1.13, is composed of a datapath and a controlpath. The data path is defined as the part that samples input data, manipulates internal data, and produces output data with respect to the MCU. The control path is the part that controls, through digital signals, each single operation performed in the data path.

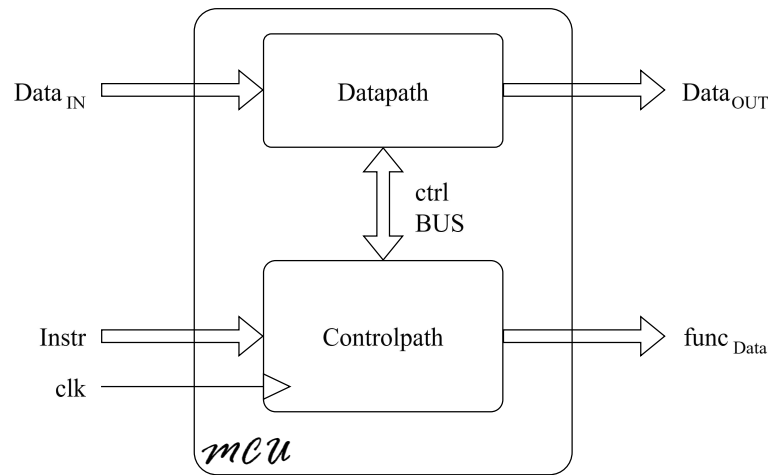


Figure 1.13. High-level representation of an MCU.

Consequently, fig. 1.14 shows a more detailed description of the MCU.

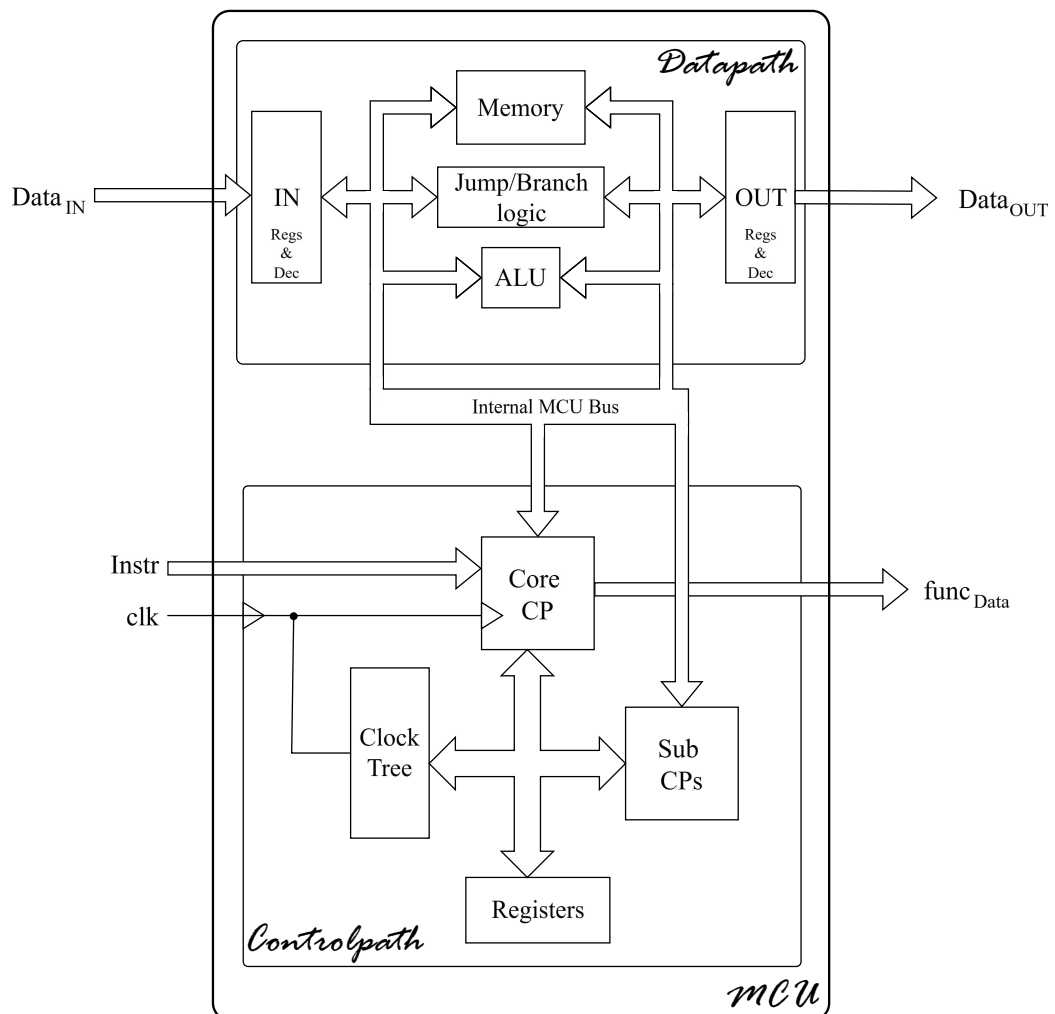


Figure 1.14. Deeper-level representation of an MCU.

In particular, the datapath comprises

- a local memory, accessible by the MCU to store local variables or executing caching and/or prefetching;

- the set of input registers and decoders, when a part of the inputs is encoded with a specific protocol;
- the set of output registers and encoders, with the same purpose as the inputs;
- an internal ALU, for arithmetic calculations;
- a jump/branch logic part to implement prefetching or branching predictions;

These parts are entirely controlled by the control path, which implements a finite state machine, the clock division tree, local registers for instruction decoding, and other local variables. Additionally, the control path employs slave control paths to manage complex data path parts, such as the ALU.

The control path can either be a Mealy or Moore machine. In particular, outputs in Mealy machines directly depend on both present and past inputs. On the other hand, in the Moore machine, outputs depend exclusively on previously processed inputs. Consequently, a Moore machines exhibit better hardware controllability due to the use of deterministic logic based on processed inputs. MCUs are the central element in microcontrollers due to their internal complexity.

Firmware

While choosing hardware can be straightforward, the choice of firmware is not as predictable, as each code line in the firmware has a specific execution time. As a result, firmware engineering is a field where the firmware structure is studied according to the underlying hardware. To smoothly approach the paradox of flexible programming on specific hardware, it is common to stratify the firmware framework. Generally, as reported in Fig. 1.15, layers have the following order, from the bottom to the top:

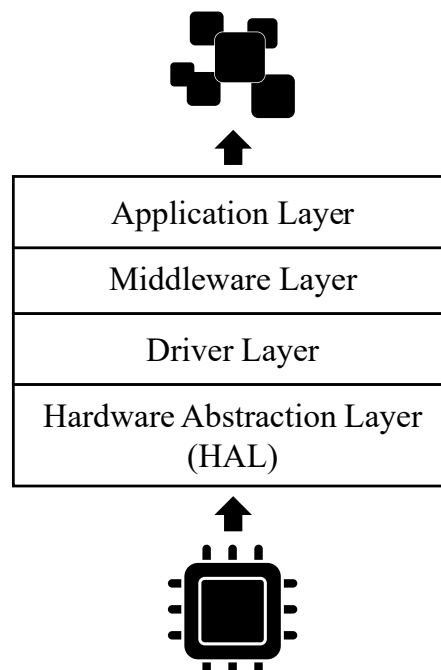


Figure 1.15. Representation of firmware layering.

1. Hardware abstraction layer: it is the layer where bare metal (microcontroller pins) is linked to low-level variables
2. Driver layer: in this layer, the pins defined in the layer below are coordinated to construct their use with specific microcontroller modules, such as I²C, GPIO, external oscillators, and eventual displays
3. Middleware layer: this layer uses the drivers defined in the previous layer to perform communication operations in an extra-microcontroller domain, e.g., implementation of Universal Serial Bus (USB) standard, memory data format, like Generic FAT Filesystem Module (FATFS)
4. Application layer: the highest-level layer, user-available, where it is possible to fulfil the required tasks with user-friendly functions and variables

In most cases, the user-microcontroller interaction occurs at the application layer, as the lower-level layers, such as peripheral driver codes, are generally already programmed by the constructor. Nevertheless, some microcontroller manufacturers, e.g., ST Microelectronics, provide a structure similar to Fig. 1.15, which is explicitly clear and modifiable. One of the main concerns in firmware engineering is code portability among various layers, as well as the choice of firmware over predefined hardware, and vice versa. In this choice, the determining factor is the approach chosen by the designer; namely

Top-Down Functional requirements are defined first, hence firmware is built on them. Hardware requirements, such as memory size and I/O needs, are estimated, and an MCU is chosen. This approach is generally used in academic contexts and general-purpose applications

Bottom-Up Hardware and cost constraints are defined first. Hence, firmware is developed separately from project requirements and then adapted to the MCU according to its hardware capabilities. This approach is used when costs need to be kept under a threshold, hence, in small industries or start-ups, or when MCUs are self-manufactured.

Co-Design Hardware and firmware are developed in parallel. As a result, there are specific teams, such as hardware, software, and system teams, working on determined parts of the final system. The determining factor in this approach is clear communication among teams at each design step. This design strategy is merely industrial and is applied in big companies that can afford highly qualified teams.

Writing firmware means defining the action set of the entire hardware-firmware device. For this reason, when designing firmware, it is highly recommended to consider the following technical paradigms.

One of the most used practices is state-machine-based design: the entire hardware-firmware device is a discrete-time system; hence, its operation modes can be related to the states of a state machine. Consequently, at higher levels, firmware is seen as a sequence of actions that lead the device from one state to another. Therefore, when lower layers are already defined, higher layers can be designed by following a state-machine approach. Generally, states represent fixed working modes, where all

variables have a predetermined value. Then, the execution of code constructs, such as “if” and “while”, as well as the execution of functions and external events, like pressing a button, represent transitions between states.

The management of interrupt and polling structures is another important aspect for firmware high-level layers, especially when considering interaction between the MCU and peripherals. Namely, in a polling structure, the MCU assesses the state of peripherals by checking each one individually. On the contrary, in interrupt structures, MCU routine is interrupted by an interrupt signal generated by any peripheral when a variation occurs. The interrupt triggers the Interrupt Service Routine (ISR), a special code pattern that is executed immediately after the interrupt is handled. While the polling-based firmware design offers a simple and predictable structure, it is avoided due to the resource waste it causes in microcontrollers. Resource waste is motivated by the occurrence of peripheral monitoring when there are no events at all; consequently, with a high number of peripherals, the polling method leads to time inefficiency. Despite their design complexity and stringent requirements, interrupt-based frameworks are mostly chosen in microcontroller firmware design due to the efficient MCU usage and immediate response to time-critical events.

Additionally, memory management is an important aspect to consider for efficient MCU resource usage. Memory management significantly affects the highest firmware layers, as it acts on variables and elements “placed” in lower-level layers. Mainly, microcontrollers offer a limited and fixed amount of available memory; consequently, firmware structure should involve strict variable allocation. Specifically, static allocation should be preferred to dynamic allocation, as the former is deterministic and immune to memory fragmentation, while the latter significantly slows microcontroller operation.

Middle layers, such as middleware and driver layers, are affected by real-time constraints. Specifically, these constraints shape the firmware structures in these layers, as they use variables and functions from lower levels. In this regard, context-switching features, improper management of multi-master structures, clock stability from oscillators, and access time to memory induce non-determinism in code execution. Some commonly used strategies to minimize the effects of non-determinism are:

- defining a bounded execution time for functions;
- using elementary constructs for functions at lower levels;
- using watchdogs as a counteraction to system stalls;
- prioritizing ISRs so that critical peripherals can affect the system more quickly.

At the lowest firmware layers, the program must be structured safely and robustly to ease its debugging. Safety precautions in firmware programming can be translated into preventing memory corruption and invalid peripheral configurations, both from external perpetrators and the code itself. In this regard, it is good practice to use bounds checking on arrays and communication buffers to avoid overflows. This is performed using fixed-width integer types to prevent architecture-dependent errors and to initialize peripherals and variables to safe states at startup. Robustness in coding is assessed through functions and practices that keep the correct functioning of the system in abnormal conditions. In this sense, robust code is implemented through redundant structures and cross-checks, task prioritization, and the implementation

of error codes with the related solving routine. Finally, debugging is a substantial code feature, as the more a code can be debugged, the more time is saved in spotting errors and understanding any functioning anomalies. For this scope, there are several debugging interfaces, with specific commands, such as the Joint Test Action Group (JTAG), which can considerably ease the debug operation. The integration of these standards into the firmware for its debugging may be a good practice, as microcontrollers lack proper debugger frameworks and human-machine interfaces that simplify code debugging. Additionally, it is good practice to use a log to report events, informing the user about the system status, and using conditional debug code (`#ifdef DEBUG` structures) to ease firmware design and understanding.

As a result, writing a firmware from scratch can be summarized in the following steps:

1. requirements and constraints definition: defining target applications and quantifying deadlines;
2. hardware understanding: identifying the MCU peripherals and learning about internal registers management, in particular, the purpose of any specific addresses;
3. development environment setup: choosing the proper coding interface to code, setting up the compiler, linker, and debugger;
4. structuring the Firmware in layers according to the structure reported in fig. 1.15;
5. memory usage planning: definition of global and static variables, space planning for stacks, evaluation of dynamic allocation use, and placing large buffers in well-defined memory regions;
6. core functionalities implementation, such as basic drivers, core logic — in this step, the use of state machines can be beneficial;
7. real-time and determinism handling: evaluation of any polling structures if needed, design of ISRs, fine use of timers, and design of task priorities;
8. microcontroller initialization: in particular, system clock configuration, peripherals enabling, setup of stack pointer and interrupt vector table, watchdogs initialization in safe mode;
9. safety and robustness assessment: achieved through the integration of specific block codes for input validation, protection of shared resources with bit-wise operations, watchdogs development;
10. test and debug units addition: integration of standard debugging frameworks, addition of conditional logging, definition and test of edge cases, profiling and management of worst-case scenarios;

1.6 Purpose of the Thesis

This thesis aims to demonstrate the implementation possibilities around hardware-firmware systems in industry-related contexts. In particular, academic experimentation is rarely employed in these contexts; therefore, this work demonstrates how the academic approach could be ported to industrial applications. This thesis primarily consists of applications where known elements are combined and employed to introduce novelty in industrial applications. Most importantly, the application context can vary; hence, the academic approach does change accordingly. Moreover, it is worth mentioning the use of academic instruments to achieve close-to-industry implementations, which are realized entirely within the scientific environment, involving 3D-printing, sensor interfacing, and UWB circuit simulation and modeling.

Chapter 2

LoRa-based Hardware-Firmware Systems

This chapter introduces the hardware-firmware systems developed and implemented during the PhD studies. In particular, three hardware-firmware systems were deepened: a WSNs for landslide and rockfall monitoring (Sec. 2.1), a smart gas meter (Sec. 2.2), and a spectrum-based SHM monitoring system (Sec. 2.3). Before the description of the systems in object, a literature overview is reported; afterwards, the structure of each system is approached with a top-down level, towards the actual components used in the systems — namely, the hardware — and their working routines — the firmware. The structures introduced in this chapter are based on LoRa technology. Each structure uses LoRa for a specific function, as its benefits can significantly contribute in the above reported fields.

2.1 Geohazard Prevention

WSNs have become a reliable method for preventing environmentally hazardous events, such as rockfall and landslides, in recent years. In this sense, part of the PhD work involved implementing WSN based on the LoRa technology. This choice is mainly motivated by the range achievable with LoRa and, most importantly, the low power capability that the base protocols give. In particular, LoRa exchange is managed by the Chirp Spread Spectrum (CSS) modulation where, as reported in fig. 2.1, chirp signals, i.e., continuous alternate signals with linear continuous constant frequency variation, are sent in the air.

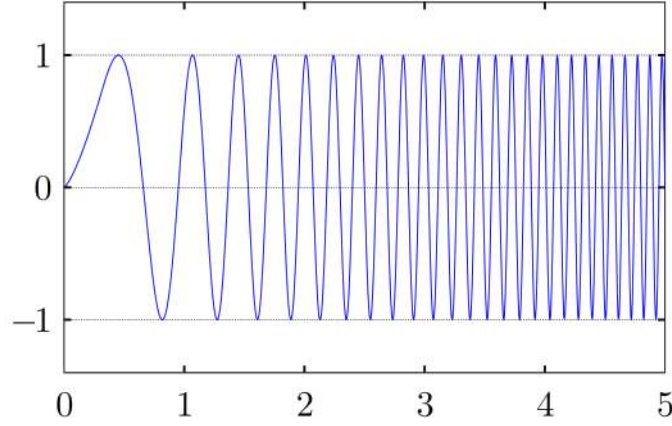


Figure 2.1. Representation of the Chirp Spread Spectrum. By Georg-Johann — Own work, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=11330709>.

The frequency variation of the chirp determines the binary encoding:

- when the communication is binary, a decreasing and an increasing frequency are related to zeros and ones, respectively;
- when the communication involves more than two symbols, the initial frequency of the chirp represents a specific symbol, i.e., in the ASCII table.

. This transmission method guarantees robustness to the communication channel, as it is resistant to both the Doppler effect and the multi-path fading when front-end apparatuses are operating at low power. Furthermore, its low-power and long-range characteristics enable a smooth IoT integration of the systems that use this technology. With this premise, the primary target behind this set of works is to use the LoRa technology for WSN implementations specialized in environmental monitoring. The status and implementation methods of the system evolved through each implementation. In particular, sec. 2.1.1 introduces the problem of rockfall and landslides; sec. 2.1.2 shows a high-level version of the communication infrastructure characterising the discussed implementations; sec. 2.1.3 deepens the working principle of the systems, with details on the structure of nodes; and sec. 2.1.4 reports briefly each implementation of these systems, with results from some of them.

2.1.1 Introduction

Landslides and rockfalls are events involving downhill movements of debris and rocks. In particular, rockfalls include the free fall of rocks. Both events can cause catastrophic consequences; in particular, their occurrence frequency has increased in the last decade due to the hydrogeological instability that characterizes recent decades; in this regard, according to Wang *et al.*, results from statistical long-term monitoring predict a doubling of landslide occurrences over the next three decades [61]. Furthermore, landslides result in approximately 4600 fatalities annually, with a frequency of 370 occurrences per year [62].

In this regard, documentation of small-volume rockfall episodes is limited, particularly in cases where land movements are not the primary cause of these events. Based on non-seismic events in the Spanish country, Coromias *et al.* examined the effects

of deadly rockfalls and concluded that, on average, 60% of fatalities result in up to five deaths for each event [63]. These numbers support the necessity of containment systems and monitoring networks for predicting and managing landslides and rockfall risks.

Containment systems can be broadly classified into two groups: electronic-based systems and mechanical-based systems. The former includes all systems based on the physical containment of rocks and debris, such as rock drapery systems [64], protection embankments [65, 66], and rockfall and debris barriers [67–69]. According to Fernandez *et al.* [70, 71], the electronic-based solutions are networks that notify the user of the rockfall hazard before its occurrence, sometimes through the use of open geographical data infrastructures. Originally based on wired standards [72], electronic-based monitoring systems were implemented using a variety of technologies, including optical Very High-Resolution (VHR) data combined with analytical models [73], Terrestrial Laser Scanning (TLS) [74], Infrared Thermography (IRT) [75], Synthetic Aperture Radar (SAR) [76], or even the assistance of external resources, such as satellites, through the Global Navigation Satellite System (GNSS) [77] or Unmanned Aerial Vehicles (UAVs) [78].

Recently, laser scanning [79] and optical fiber-based systems [80] have been developed as alternative methods for tracking and managing land movements. Notwithstanding the efficiency and high specialization of the solutions reported above, WSNs were among the first to be used for rockfall and landslide monitoring.

Although the solutions reported above enable highly specialized and reliable systems based on known protocols and standards, they frequently lack flexibility, total customizability, and, especially in some cases, such as TLS, optical fibers, or UAVs, affordability in costs and implementation. Specifically, one of the first documented WSNs in the literature employs 2-axis strain gauges to evaluate the terrain condition and, consequently, the risk of landslides [81]. Subsequently, this technique was refined by employing more advanced sensors and infrastructures, such as tilt sensors and Micro Electro-Mechanical Systems (MEMS) sensors, or even a combination of them [82]; furthermore, WSNs were created by using the IoT standards available at the time, such as ZigBee [83, 84]. Furthermore, as WSNs can be outfitted with energy-efficient communication protocols [85] and renewable energy recharging mechanisms [86], energy efficiency is a crucial factor as it boosts the entire system’s autonomy. Giorgetti *et al.*, in this regard, developed an energy-efficient protocol featuring event-driven functionality and based on fault recovery.

In recent years, LoRa technology [87] has emerged as a famous industry standard for long-range wireless applications. Recently, its application as a WSN implementation standard for rockfall monitoring has grown [88–91]. Since their system monitors an uninhabited hillside, Romdhane *et al.* are specifically among the first to investigate the viability of LoRa technology in this context [refPanni28], as they assess the node position through the measurement of Received Signal Strength Integrity (RSSI) coming from each node. Furthermore, recently, low-cost Arduino-based WSNs for the landslide monitoring have been implemented, in particular, the system proposed by Sigiuro *et al.* monitors a hillside road in a hazardous area [92]; the multi-sensor system implemented by Pancane *et al.* monitors landslide events in the installation spot of high-voltage towers [93].

2.1.2 Network Structure

The general structure of the proposed WSN, represented in Fig.2.2, relies on several protocols, in accordance with the Open Systems Interconnection (OSI) model. This framework depends strictly on the site of implementation; however, a general structure is discussed below, with a clarification of specific changes made in past implementations

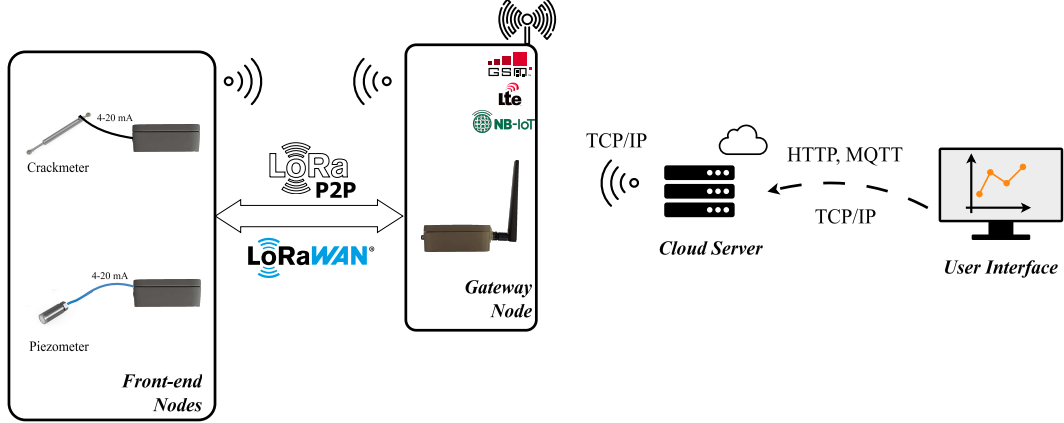


Figure 2.2. High-level representation of the WSN structure.

Monitoring parameters are measured by front-end nodes installed at specific critical spots. These front-end nodes, equipped with internal sensors, are either installed alone or integrated with an external sensor, e.g., a piezometer or a crack meter, using the 4–20 mA standard. Each front-end node transmits its measurement through the LoRaWAN or the LoRaP2P protocol to a gateway node. The choice between LoRaWAN and LoRaP2P protocols depends on the installation requirements: LoRaWAN is chosen when a standard protocol, operating in the data link layer, is needed; LoRaP2P is chosen when it is necessary to limit the packet size, operating directly on the physical layer. In specific situations, such as when a commercial component employing LoRaWAN is used, it is possible to use both LoRaWAN and LoRaP2P protocols directly in the gateway node.

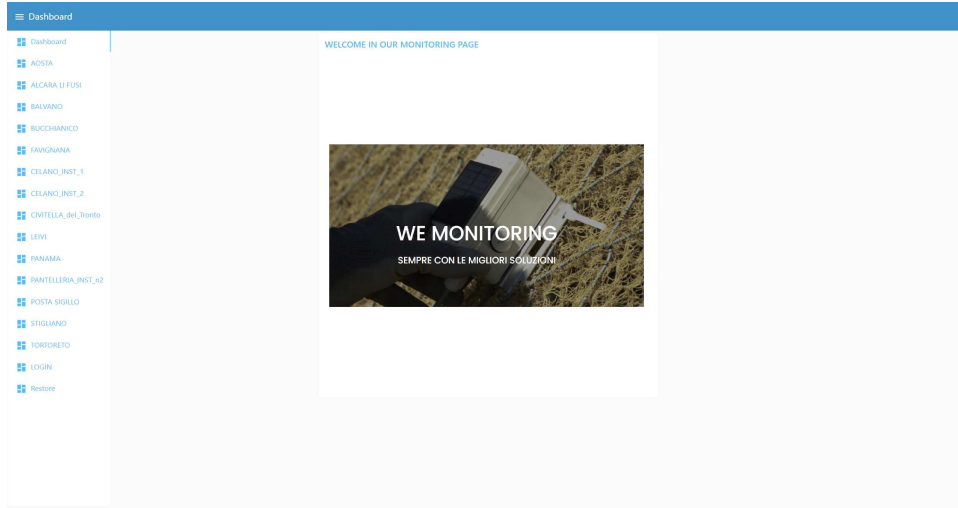
Data are sent to the gateway node, whose purpose is to shift the communication level from the physical/data link layer to the network/transport layer. As a matter of fact, the gateway node encapsulates the received LoRaWAN or LoRaP2P packets in an Internet Protocol (IP) packet so that, through Global System for Mobile Communications (GSM), Long-Term Evolution (LTE), or NB-IoT, it can be sent to the internet. The choice of GSM, LTE, or NB-IoT depends on the zone coverage.

Subsequently, network-layer data is sent to a server, which is then accessed by the User Interface (UI), depicted in Fig. 2.3. UI can either be based on Node.js [94] or on the Hypertext Transfer Protocol (HTTP). The UI based on Node.js accesses the server using the MQTT protocol, when LoRaWAN is used on the lower levels, or the Transmission Control Protocol (TCP) when LoRaP2P is used on the physical layer. On the other hand, with an HTTP-based UI, the server is accessed through the corresponding HTTP protocol.

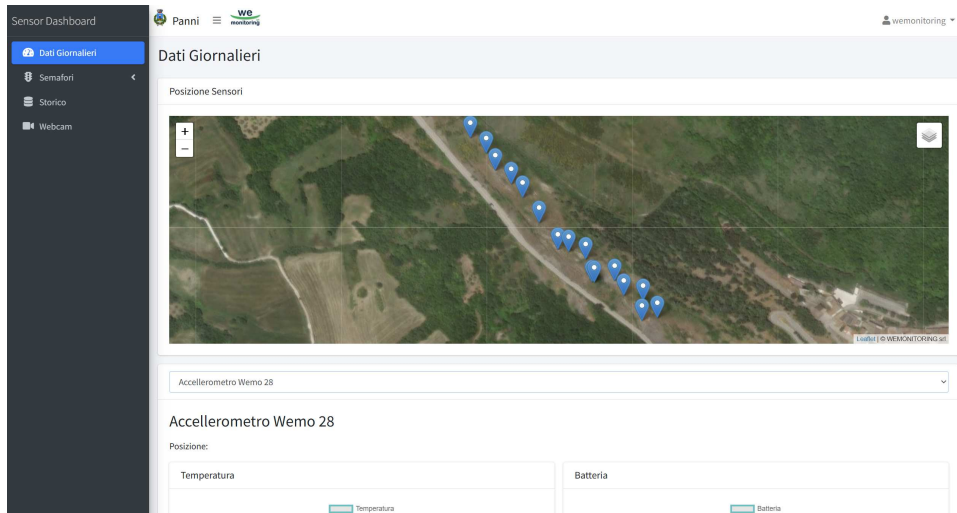
The first WSNs featured an interface based on the Node.js platform, reported in Fig. 2.3a. As shown in the image, this user interface gives easy interaction with the system due to the presence of configurable buttons and a side menu that allows

switching between various implementations. Each implementation has a different disposition of visual tools depending on the instrumentation used.

In more recent applications, the user interface has been modified and customized. In particular, it has been designed for a specific internet domain, with important changes to its appearance, as reported in Fig. 2.3b. In this novel UI, daily and historical measurements can be accessed, with the possibility to download a related number table in several useful formats, including CSV, TXT, and XLSX, to reproduce graphs and perform further data analysis.



(a)



(b)

Figure 2.3. User interface based on Node.js (a) and HTTP (b) for the proposed WSNs.

As mentioned, the network structure described can be changed according to the installation site. In fact, environmental monitoring may be needed in underwood scenarios; hence, dense foliage would suppress the LoRa signals sent from the system front-end part. In these conditions, the system can be upgraded with a local sink node, gathering information from all or part of the sink nodes, and a signal refreshing node, installed between the front-end zone and the gateway node, as reported in Fig. 2.4.

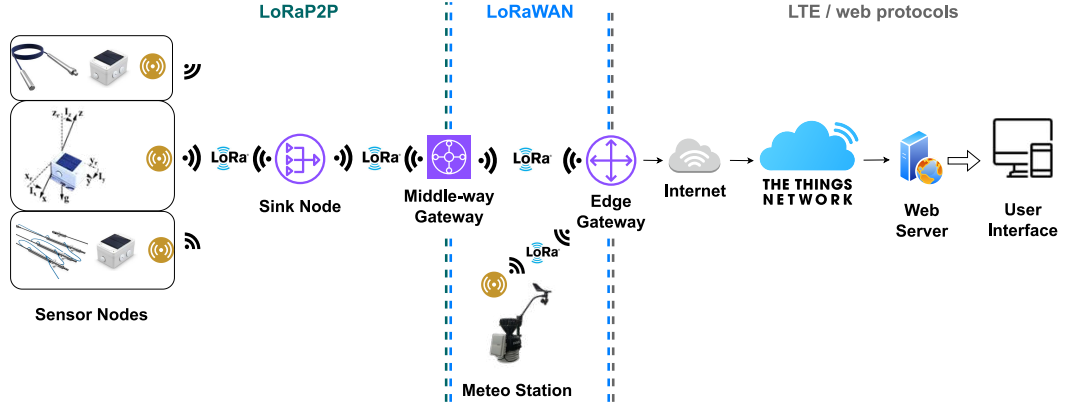


Figure 2.4. Variations on the communication structure of the reported WSNs [95].

2.1.3 Working Principle and Nodes Firmware

Internally, both front-end and gateway nodes are based on a fixed structure used in almost all the implementations. The primary role of nodes is to provide environmental measurements; hence, the exposure to environmental conditions introduces the need for a robust external structure. In this regard, all the nodes are equipped with an IP65 package. Furthermore, front-end nodes and the gateway are organized in a star topology, a choice made to maximize the system modularity.

Both types of nodes are based on low-power microcontrollers. Firstly, the STM32L microcontroller was used as MCU; afterwards, for the integration of the NB-IoT feature, the Nordic nRF52 was chosen. The Semtech SX127x series chips, along with a corresponding antenna, enable LoRa communication in the nodes. Additionally, nodes are designed to be reprogrammable; hence, they use the UART-to-USB CP2102 converter. Finally, the entire system is solar-powered, enabling autonomy in hardly reachable areas and use based on green energy. For this reason, the presented WSN front-end and gateway nodes feature a power-management section, which charges a LiPo battery through the BQ21040 battery charger and regulates its voltage to a 3.3 V constant value using an RP104 voltage regulator.

In fact, the different role of front-end and gateway nodes introduces some fine differences into their structure. Specifically, as shown in Fig. 2.5, front-end nodes are specialized in environmental measurement; hence, they have both internal and external sensors, namely, the LIS3DH accelerometer and the BME680 environment sensor as internal, and crack meters and piezometers as external sensors, if needed. In particular, external sensors are connected to a node internal voltage divider, whose output voltage is read by the MCU ADC.

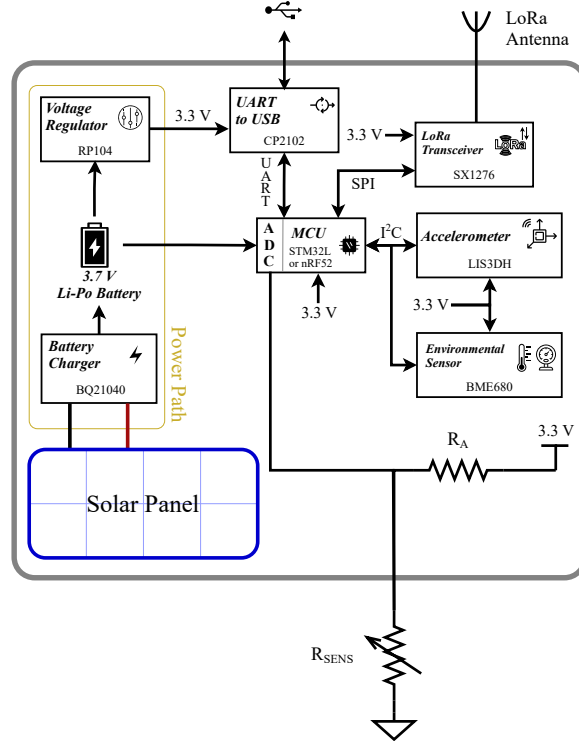


Figure 2.5. Internal structure of a front-end node for the proposed WSNs.

On the other hand, gateway nodes can be either commercial structures, such as the Milesight UG65 usable for LTE connection, or custom structures, used in the latest works, whose structure is reported in Fig. 2.6. Custom gateway nodes feature a SIM slot and a data-link modem, to perform the operations discussed in Sec. 2.1.2. In particular, the data-link modem can be either:

- an MC20CE modem for the GSM connection;
- a BG77 modem for either the LTE or the NB-IoT connection;

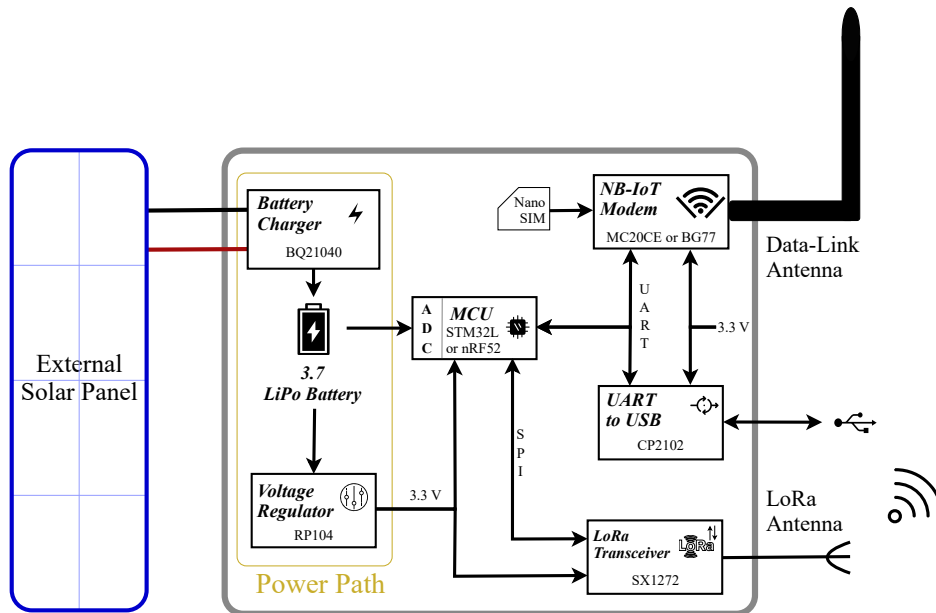


Figure 2.6. Internal structure of a gateway node for the proposed WSNs.

Both front-end and gateway nodes work in a low-power mode. In particular, as shown in Fig. 2.7, after initializing the sensors (front-end nodes), the modem, and the data-link session starting (gateway nodes), both nodes enter a listening mode.

The front-end nodes listen to events, which can be classified as either a sudden variation in any sensor of the node or the expiring of a timer. The event causes an interrupt, whose routine consists of:

1. reading of internal accelerometer and environmental sensor;
2. inclination calculation through the following formula:

$$INC_i = \frac{180 \cdot \left(\cos^{-1} \left(\frac{a_i}{\sqrt{a_x^2 + a_y^2 + a_z^2}} \right) \right)}{\pi} \quad (2.1)$$

3. reading of the MCU ADC. For external sensors, the ADC is read to assess both the sensors and battery values; otherwise, the reading only interests the latter value.
4. LoRaWAN or LoRaP2P packet preparation;
5. packet transmission;
6. in case of transmission success, the node returns to idle mode; otherwise, transmission is re-attempted a specified number of times.

On the receiving side, after their setup, gateway nodes are in a low-power listening mode, where the reception of a packet causes an interrupt. The gateway nodes interrupt service routine consists of packet encapsulation for the data-link OSI layer, and its sending, so that the listening mode is restored.

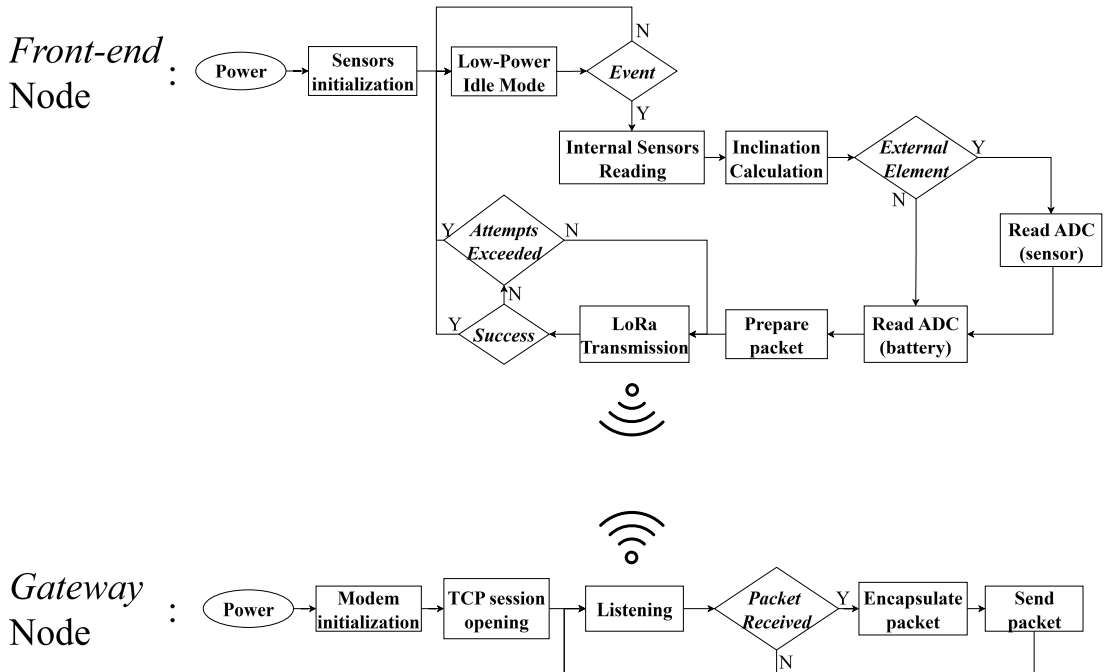


Figure 2.7. Firmware-related flow diagram for both WSNs front-end and gateway nodes.

The system is classified as autonomous, with supply independence being its primary characteristic. To further prove these features, the power consumption of a single node is reported below. To perform this measurement, an INA229 current monitor has been used. In particular, as shown in Fig. 2.8, the Instrumentation Amplifier (INA) has been connected in series with the smart node LiPo battery and to a microcontroller through the $40\text{ m}\Omega$ shunt resistor R_{sh} . The microcontroller has been programmed such that every current value detected by the INA is read serially, through an USB connection.

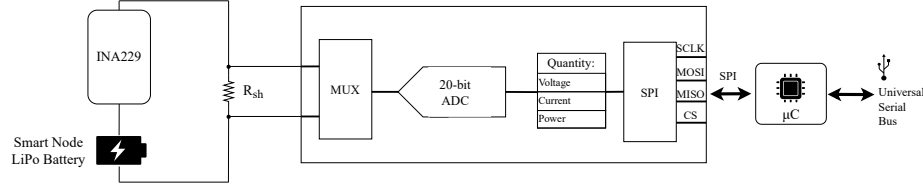


Figure 2.8. Circuit configuration for smart nodes power consumption measurement.

With this setup, the power consumption of the smart node has been assessed over a transmission period and is represented in Fig. 2.9.

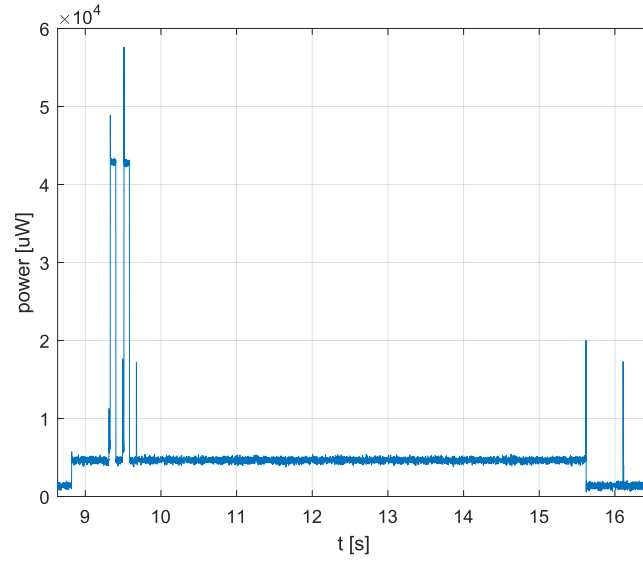


Figure 2.9. Power consumption over a transmission period for the LoRa smart nodes.

With the firmware documented above, the packet transmission process has a duration of 8 seconds, with a node power consumption of 1.4 mW in resting conditions. When the transmission starts, the power rises to 4.5 mW, with peaks reaching 43 mW. According to the slave node firmware, the transmission procedure begins when either a timeout is reached or an abrupt acceleration change is detected. The timeout value depends on the application; however, it is set to six hours by default. Therefore, in the absence of abrupt changes and over a day, the node has a power consumption of 34.68 mWh. The average node current during the packet sending process is around 1.36 mA, but the average consumption under resting conditions is 424 μA .

2.1.4 Implementations and Results

This subsection documents the different implementations of the proposed WSN architecture. In particular, the changes introduced between implementations will be documented, as well as results from any of them.

The proposed WSN structure was firstly implemented in the Pantelleria island, in Italy, in a zone subject to high rockfall and landslide risks [96]. In particular, as reported in Fig. 2.10, front-end nodes were installed on the steep part of a hill, monitoring its inclination drift using sensors inside the nodes. Additionally, the LoRaWAN protocol was used for communication between the front-end nodes and the gateway.



Figure 2.10. Implementation of the reported system in the Pantelleria island [96].

The system was then installed in an inhabited area whose buildings were erected in a high landslide risk zone due to the terrain morphology [97]. No external sensors were used in this application, as the primary parameter to monitor is the inclination of the terrain, assessed through twelve front-end nodes installed on the walls of the buildings, as shown in Fig. 2.11. In particular, the firmware internal to nodes is structured so that, in the case of exceeding a determined threshold, an evacuation alarm is activated.



Figure 2.11. Implementation of the custom WSN in the inhabited area of Bucchianico (CH). [97].

As a result, after a 2-week period, it has been possible to assess the landslide hazard. Specifically, as reported in Fig. 2.12, the inclination of nodes with ID 1, 3, and 4, slowly drifts towards lower values. This trend confirms the hazard risk and makes it possible to assess its temporal action.

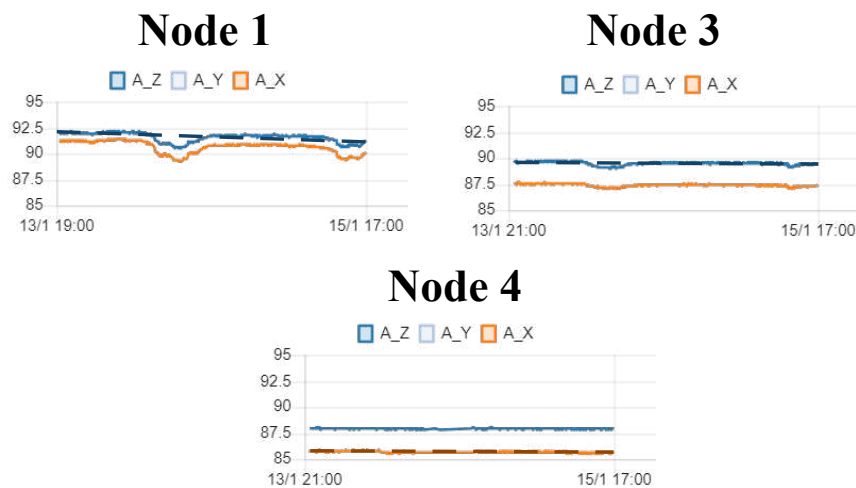


Figure 2.12. Inclination results from the implementation in the Bucchianico (CH) locality. [97].

The integration of external sensors became necessary with the implementation

of crack meters for monitoring a rock complex that experienced past modifications related to human activity [98] (Fig. 2.14). In particular, the reported rock complex hosts a castle at its peak, and overlooks a public street; for this reason, a threshold check is applied, and its overreaching triggers a stoplight.



Figure 2.13. Implementation of the reported system with crack meters for rockfall monitoring [98].

In particular, Fig. 2.14 reports crack measurements from the discussed implementation. Fig. 2.14 reports the most critical cases where it is clear, after a one-year monitoring, the drift that cracks experience.

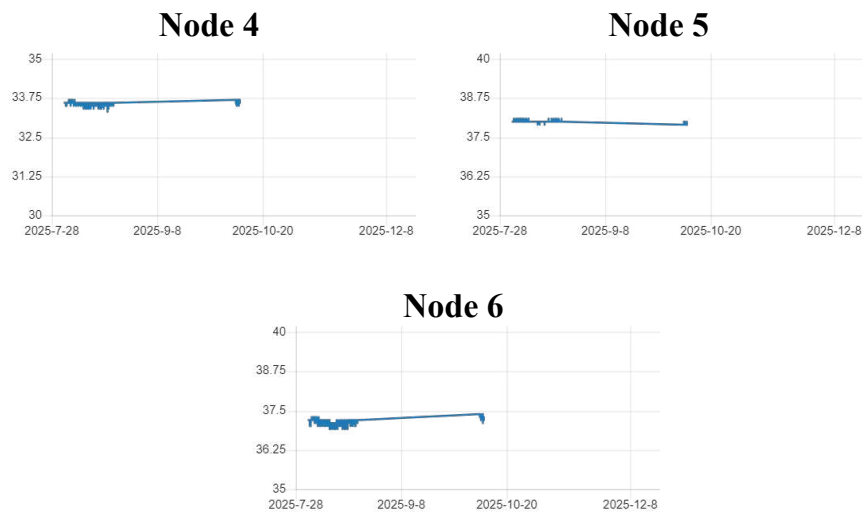


Figure 2.14. Crack measurement with the proposed custom WSN. [98].

The high flood and landslide risk in the Posta, Lazio, Italy, location motivated

the implementation of a more complex version of the proposed WSN. In particular, this implementation, underwood, and multi-sensor and multi-level monitoring were implemented [95]. The multi-sensor implementation comprises inclinometers from both internal front-end node sensors and external pole inclinometers, as well as piezometers to measure both water level and water pressure (Fig. 2.15a). Furthermore, the multi-level side is mentioned as the system implements both LoRaP2P for communication between front-end and gateway nodes, and LoRaWAN for the use of a commercial meteorological station. In fact, due to the vastness of the area and the dense foliage, this network also employed signal refreshing nodes (Fig. 2.15b).

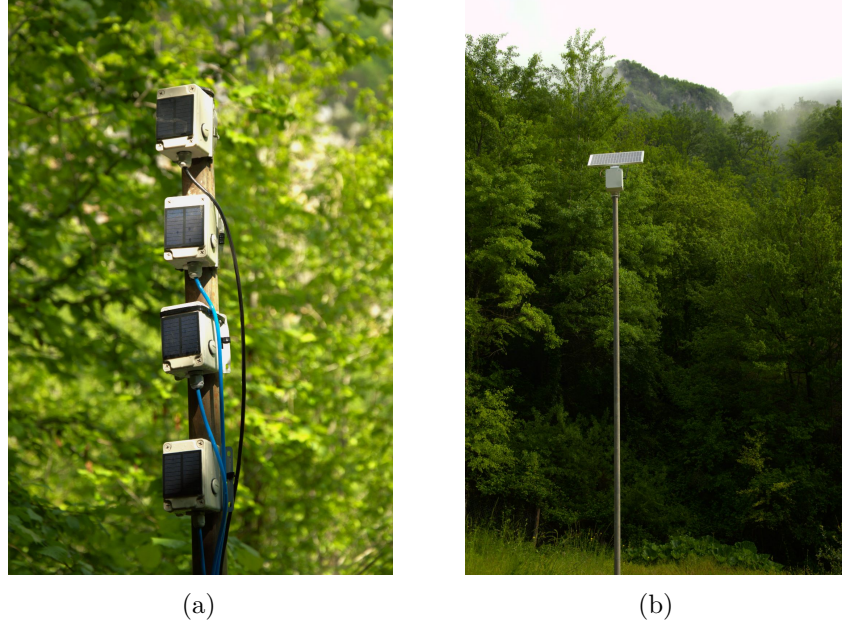


Figure 2.15. Slave (a) and sink (b) nodes in the underwood monitoring implementation [95].

As mentioned, this application has a multi-sensor structure, i.e., the hazard is assessed by measuring multiple factors such as the water level, water pressure, and inclination of specific areas. Fig. 2.16 reports the results from water level monitoring. As a matter of fact, there is a drift in the water level, ranging from 0.2 cm to 1 cm for node WL1 and from 0.5 cm to 0.2 cm for node WL2. This situation confirms the landslide hazard in the monitored place.

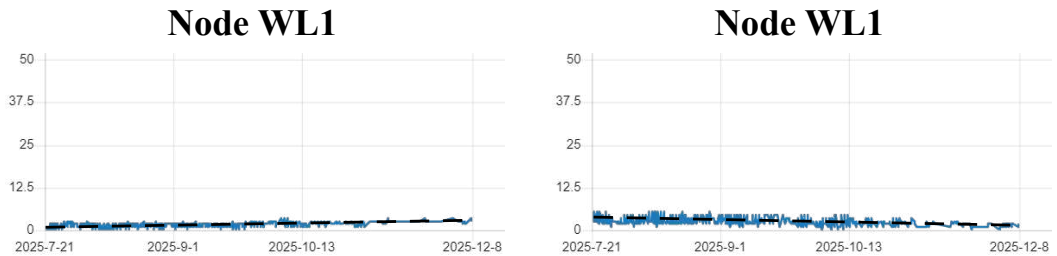


Figure 2.16. Water level measurement in the underwood multi-level implementation. [98].

On the other hand, as reported in Fig. 2.17, water pressure keeps an almost-constant average value, while inclination experiences slight temperature-related drifts.

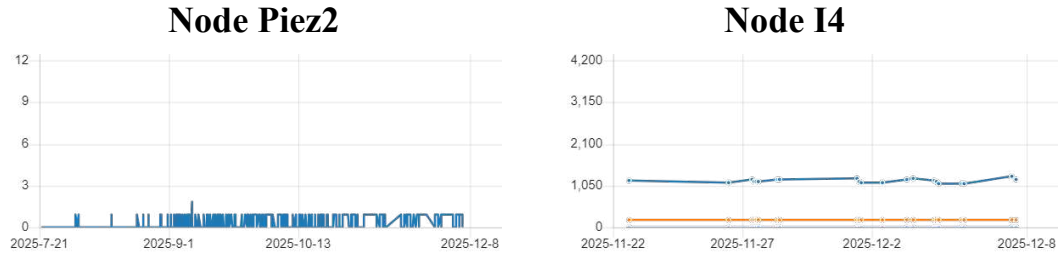
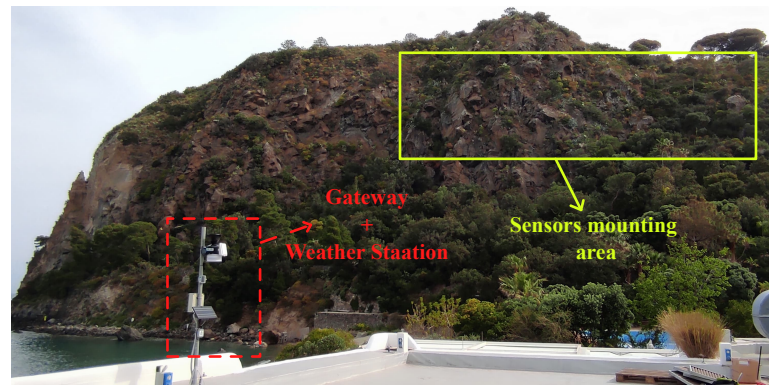
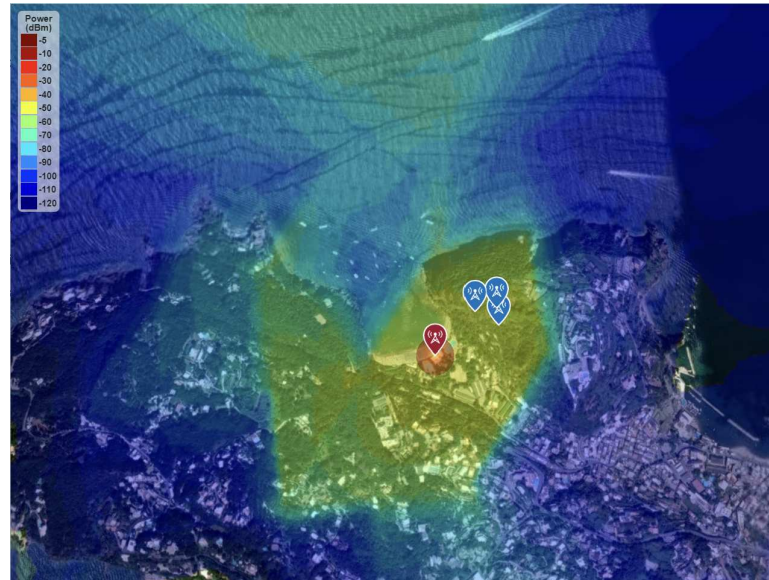


Figure 2.17. Water pressure (left) and inclination (right) measurements in the underworld multi-level implementation. [98].

In the next implementations, the NB-IoT was integrated as a communication standard into the system. In particular, this updated version of the system was used both in a touristic area with high seismic risk [99], and in an inhabited complex under landslide risk [100]. The system documented in [99] was installed in the locality of Ischia (NA), interested, in the past, by active seismic-volcanic activity. As a consequence, inclinometers were chosen as the main sensors for the nodes. Fig. 2.19 reports the system installation and the zone radio coverage.



(a)



(b)

Figure 2.18. Mounting map (a) and radio coverage (b) of the WSN implementation reported in Ischia (NA). [99]

Results (Fig. 2.19a) show slight variations in inclinations over a 10-day period due to, mostly, temperature drifts, reported in Fig. 2.19b. Additionally, the strong winds detected in the area (Fig. 2.19c) contributed to these drifts.

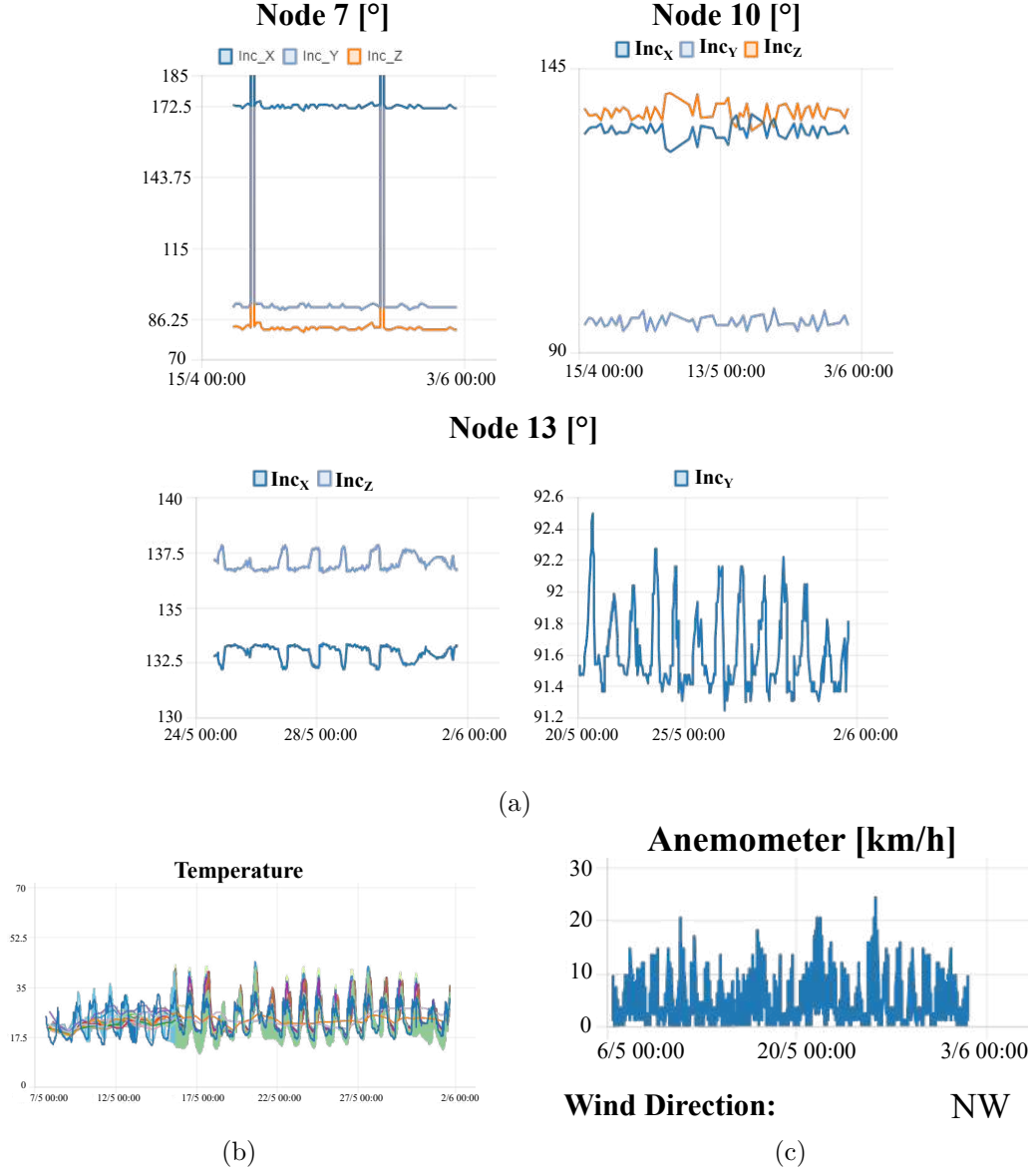


Figure 2.19. Results from the implementation in Ischia (NA): inclinations (a), temperature (b), wind direction and magnitude (c). [99]

Furthermore, the proposed WSN structure was used for environmental monitoring in caves both externally [101] and internally [102]. In particular, the system installed internally to the cave, reported in Fig. 2.20, features an asynchronous behavior where long sleep-awake cycles are implemented to assess longer power autonomy, as solar energy is lacking inside the cave.



Figure 2.20. In-cave implementation of the reported system with deep-sleep capability [102].

2.1.5 Comparison With Other Works

The system presented in this study employs LoRa technology for geosite monitoring, prioritizing long-range wireless communication under stringent low-power constraints. A comparative overview between the proposed solution and related works is provided in Tab. 2.1.

Although LoRa modulation can be operated across several frequency bands—including 915 MHz [89] and 433/470 MHz [90] — the use of the 868 MHz band is consistent with common European LoRa deployments. This choice also ensures compatibility with earlier WSN infrastructures, such as the SLEWS system [70, 82].

A further distinguishing feature of LoRa and LoRaWAN networks is their adoption of a star topology. This design diverges from the multi-hop configurations that characterized earlier approaches, such as SLEWS [70], as well as from the tree-based topologies utilized in certain IEEE 802.15.4 WSN implementations [85]. The LoRa star architecture inherently reduces the need for individual nodes to support long-range transmissions, thereby enabling lower transmission power across the network.

Regarding communication protocols, LoRaP2P provides a simplified or proprietary configuration designed to support direct, low-latency data exchange. This contrasts with the more elaborate LoRaWAN protocol stack [89], which integrates advanced security features for star-topology network management and application-layer services built on established software frameworks. In comparison, systems not based on LoRa rely on dedicated interoperability standards—such as the Sensor Observation Service (SOS) and XMPP/Jabber [70] — or on hybrid MAC and network-layer solutions combining CSMA/CA with TDMA. These approaches aim to guarantee energy-efficient operation and strong resilience to faults [85].

Furthermore, the use of solar energy as a power source remains a well-established strategy to ensure long-term autonomy in remote or environmentally challenging locations. The measured 420 μA sleep-mode current of the system is notably low when compared with other reported designs, including the approximately 1 mA consumption documented by Wang *et al.* [91]. Although some optimized WSN platforms report even lower quiescent currents—such as the 30 μA by Giorgetti *et*

al. [85] and the 100 μA from SLEWS nodes [70, 82] — achieving an average standby consumption of 420 μA still reflects an efficient system configuration. These observations indicate that incorporating a deep-sleep mechanism could further improve energy performance, which is essential for sustaining long-term network operation in challenging mountainous environments.

Work	Network Technology	Frequency Band	Network Topology	Communication Protocol	Type of Supply	Current Consumption (Sleep Mode)
[70]	Custom	868 MHz	Multi-Hop	Custom	Solar Energy	—
[72]	MicaZ	2.4 GHz	Multi-Hop	—	Battery	—
[81]	Mica2	—	—	Beacon Vector Routing (BVR)	—	—
[82]	Custom	868 MHz	Multi-Hop	—	Solar Energy	100 μA
[83]	ZigBee	—	—	—	—	—
[84]	ZigBee	—	—	ZigBee	Solar Energy	640 mA
[85]	IEEE 802.15.4 standard	2.4 GHz	Tree Logical Topology	Custom	Lead Batteries	30 μA
[88]	LoRa	868 MHz	Star Topology	—	—	200 μA
[89]	LoRa	915 MHz	Star Topology	LoRaWAN	Solar Energy	55.5 mA
[90]	LoRa	433–470 MHz	Three-layer IoT Structure	—	Battery	—
[91]	LoRa	—	Star Topology	—	Solar Energy	1 mA
Presented Implementations	LoRa	868 MHz	Star Topology	LoRaP2P	Solar Energy	424 μA

Table 2.1. LoRa systems comparison table.

2.2 Reliable and Smart Gas-Metering

The use of LoRa technology is not limited to environmental monitoring. In fact, versatility is one of the primary advantages of LoRa technology, enabling its use in several contexts. Among these contexts, smart metering gains importance, especially when long transmission distances are considered, and timing requirements are relaxed.

This section is dedicated to the design and implementation of a smart gas meter for civil applications. Issues, including incorrect billing, fraud, and ineffective energy management, are caused by the frequent problems with data accuracy, security, and transparency that characterize the traditional gas metering systems. These issues specifically occur when the gas meter is replaced with a new one or in the case of a change in the energy provider. To address these issues, a novel gas meter that combines LoRa technology and blockchain technology is introduced in this section.

The gas meter architecture is based on the principle of modularity, obtaining a unique structure. Modularity is achieved through the use of two decoupled blocks: an interface block and a measurement block.

Considering safety and the Internet of Things (IoT), the suggested method securely, irrevocably, and verifiably records energy usage and transactions.

According to a review of the literature, this novel strategy has the potential to completely transform the gas metering sector, offering a dependable and secure energy management solution.

The chapter is structured as follows: Sec. 2.2.1 better introduces the topic in subject, in addition to some scientific literature; Sec. 2.2.2 reports further motivations behind the implementation of the proposed smart gas metering system; Sec. 2.2.3 reports an overview of the system, with thorough explanation of each part; Sec. 2.2.4 reports in detail the firmware written for both the system modules and the flux sensor used in the system, with some results.

2.2.1 Literature Overview

Smart metering is one of the fundamental developments in the control of gas, water, and electricity. In contrast to conventional metering techniques, smart meters are designed to provide automatic real-time data gathering and transmission. In particular, gas metering has evolved as a result of the introduction of smart technologies that address specific issues [103], such as fraud prevention, precise flow monitoring, and gas leak detection.

Due to their reliance on mechanical components, gas meters are often prone to wear and tear. Additionally, their offline operating concept may eventually lead to mistakes and erroneous billing. On the other hand, smart gas meters utilize digital sensors and communication technology to ensure accurate readings and constant observation.

Existing choices for natural gas metering enable flow rate measurement without the use of moving parts. These options are mostly based on the calorimetric principle, based in turn on the detection of the temperature effect caused by the passage of a flow, thus the flow rate, close to the sensing element [104]. Modern technologies offer the potential to integrate sensing and computation into a chip. In any case, the installation of commercial measuring systems is complex; furthermore, to achieve a satisfactory uncertainty level required for legal measurement, the impact of geometry

integration, fluid-dynamic effects occurring in the meter, and their effect on the meter accuracy should be carefully examined.

Taking this course of action significantly enhances the management of gas-economic transactions. In particular, Cascetta *et al.* describe several recent technologies that solve billing fraud problems with specific principles, such as the transmission of flow rate detected from the movement of the gas meter index wheels through encoders [105]. As introduced in Sec. 1.4.1, Long-distance wireless internet connectivity for electronic devices is made possible by LoRa technology: a low-power, wide-area networking protocol. The low power consumption and the ability to handle a set of devices in a single network are guaranteed by the CSS modulation. As a consequence, LoRa has become a well-suited technology for the IoT and its applications in the industry involving smart metering.

The usage of various gas sensing technologies [106], AMI, as shown in Fig. 2.21, and specialized devices [107, 108] resulted from the integration of smart metering and IoT. Using an IoT-compliant framework, each implementation aims to transform gas meters into intelligent components within their local smart grids. These structures are low-power but lack safety due to the absence of specific security structures for gas meter data storage; hence, user data protection is not guaranteed.

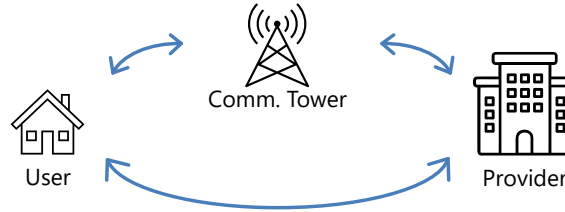


Figure 2.21. Visual representation of an AMI structure.

In this chapter, an enhanced gas meter for effective power consumption management is suggested. The device in subject guarantees safety in two areas: data security and physical protection. While the latter results from the use of blockchains for data storage, the former is ensured by a distinct, unique system structure.

2.2.2 Smart Metering and User Safety — Further Motivations

Due to their size and weight, traditional gas metering devices (generally based on diaphragm technology, revolving pistons, or turbines) are normally not appropriate for supporting modern metering. As a consequence, their support for modern transducers and technologies has become an important challenge, especially when accuracy, dependability, and risk management of safety are considered. Furthermore, monitoring large gas amounts can ensure continuity of use, prevent safety risks, and guarantee data availability to the data manager, in addition to fulfilling the targets described above. Furthermore, typical smart meters show high noise levels and significant pressure losses due to the movement of internal mechanical parts and interference in the fluid-measurement unit.

New static technologies for gas sensors can share data with the network and allow the integration of additional sensors. The proper smart metering act is expected to be implemented by these specific sensors.

Additionally, the use of environmental parameters enables the automatic correction of the main parameters when affected by temperature and pressure conditions. These elements are crucial for enhancing the metering process in terms of measurement accuracy. In fact, several tests should be performed to study the new smart metering devices and define the most suitable measuring layout, considering fluid dynamic effects on the measurement accuracy. Additionally, there are several advantages for both distribution firms and consumers, as well as from a sustainability perspective, that should motivate the adoption of smart metering in natural gas distribution, including:

- the operative costs reduction caused by remote reading;
- identification of unusual consumption and device failures;
- avoidance of unexpected breakouts and a better safety for the gas supply network (by eventually deactivating the gas supply remotely);
- the usage of the effective consumption, instead of the estimated, to define bills;
- a more lucid use of natural gas by users.

Each of these chances would enhance the measurement accuracy and dependability of the information gathered.

Simultaneously, IoT plays a pivotal role in wireless communication over the last decade; thus, defining a dependable and secure system to protect users is a priority. Blockchain technology, in this regard, is a safe digital tool for recording and validating transactions; it stores information as a sequence of linked blocks, each one related to its temporally preceding neighbor. For this reason, data cannot be altered or tampered with once it has been recorded, thanks to this structure. Blockchain is suitable for applications that require a high degree of security and accuracy, as it guarantees a transparent and reliable method for storing information. Fig. 2.22

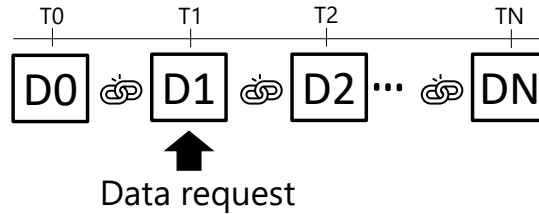


Figure 2.22. Visual representation of a blockchain.

2.2.3 System Overview

In this subsection, the system structure is reported with a top-down approach. In particular, the structure is deepened from a communication perspective, and subsequently, the structure of a single instance of the gas-metering system is deepened.

Communication-level structure

As represented in Fig. 2.23, the gas-metering system is built in a way such that more gas meters in a neighborhood could communicate with a central node, acting as a gateway, through the LoRa protocol.

In particular, for this purpose, LoRaP2P was chosen as a protocol due to its high customizability, as it is placed at the lowest level of the OSI model. As a consequence, it is possible to customize the packet format freely towards lighter structures, resulting in lower power consumption.

The gateway has the purpose of gathering data from the gas metering units in the neighborhood where it is installed, performing a data level-shift: from the machine to the data-link levels, accordingly to the OSI model. Through an internal NB-IoT modem, data is sent to a proprietary server, where it is manipulated to obtain the blockchain feature and represent it on the custom user interface.

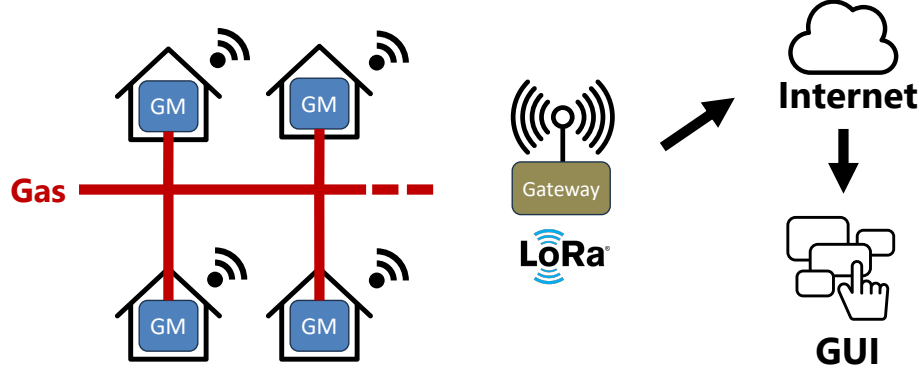


Figure 2.23. High-level representation of the gas-metering system communication structure.

In the internet domain, the blockchain-based architecture was chosen to ensure data transparency and integrity. In particular, data movements are stored in a fixed chain whose blocks integrate cryptographic linking. With this solution, energy usage and transactions are stored in a tamper-proof record; hence, trust and accountability of the gas metering device are maximized.

Data integrity is verified through distributed and proper user-level applications, which compare retrieved data with past values stored in a distributed network. If the comparison has a positive outcome, the distributed network is updated with the incoming novel data; otherwise, the app sends a warning to the user.

Gas-Metering Unit

As shown in Fig. 2.24, the single gas metering unit is composed of two main parts: a measurement block and an interfacing part. The measurement block specializes in direct gas flow measurement, while the interfacing part forwards the measurement block data, through LoRa, to the nearest neighborhood gateway. The blocks are connected magnetically and communicate through Wireless Power Transfer (WPT), used both as a power supply and a data means for the communication block. In fact, the power supply is directed from the interfacing module to the measurement module, ensuring safety in the case of tampering. This structure ensures modularity, allowing the interfacing part to be modified according to the zone coverage and the gas provider.

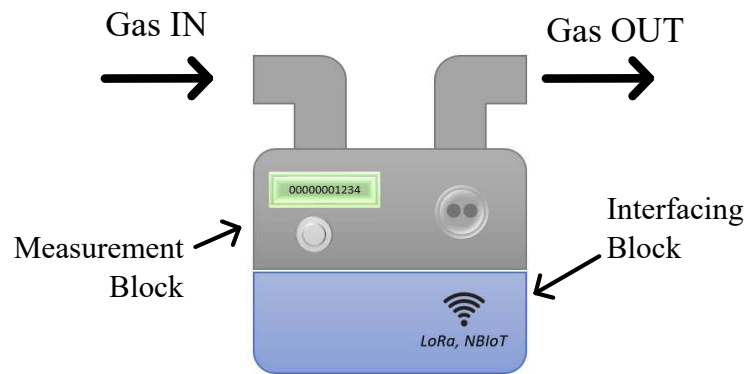


Figure 2.24. Single gas-metering unit structure.

As a matter of fact, the measurement block, integral with the gas tube, is internally composed of:

- a flow sensor, measuring the gas flow, in cubic meters per hour (m^3/h) and performing a preliminary temperature measurement;
- an environmental sensor, measuring temperature, pressure, and humidity, to provide data for environmental compensations of the gas flow measurement;
- a gas sensor and a cut-off valve, used for safety reasons, in case of gas leakage internally to the measurement module;
- drivers for an LCD;
- an ST-based MCU for the management of the peripherals, and its serial-to-USB converter for its re-programming feature;
- an internal power management path to switch the system power supply between an internal lithium battery, and the supply provided from the MCU when, through USB connection, it is connected to an external power source;
- a WPT driver for the communication and power supplying of the interfacing block.
- an Liquid Crystal Display (LCD) display for the data visualization;

This structure is shown in Fig. 2.25.

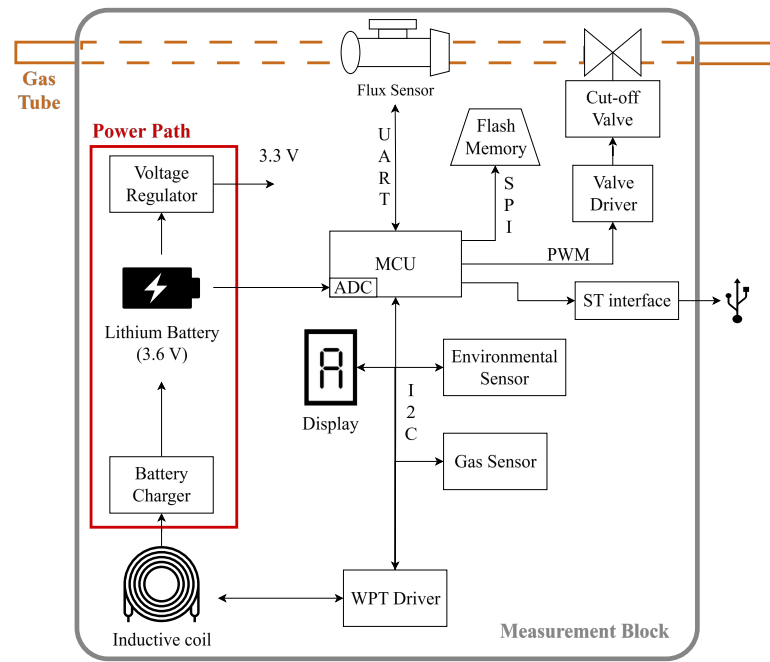


Figure 2.25. Gas meter measurement part circuit arrangement.

On the other side (Fig. 2.26), the interfacing side hosts:

- the ST-based MCU and related serial interface for peripherals management;
- the WPT driver, connected to an inductive coil, for wireless communication between the gas meter modules;
- a power path powered by an external battery, whose voltage is, at the same time, regulated at 3.3 V and boosted at 5 V for the various peripherals;
- a tri-axial accelerometer for tampering detection;
- a flash memory to store the most recent values from the measurement unit;
- an electronic circuit interface for the LoRa antenna;
- an NB-IoT modem, supporting a NanoSIM, for the related communication;
- an Infrared (IR) transceiver to command the system from the related means.

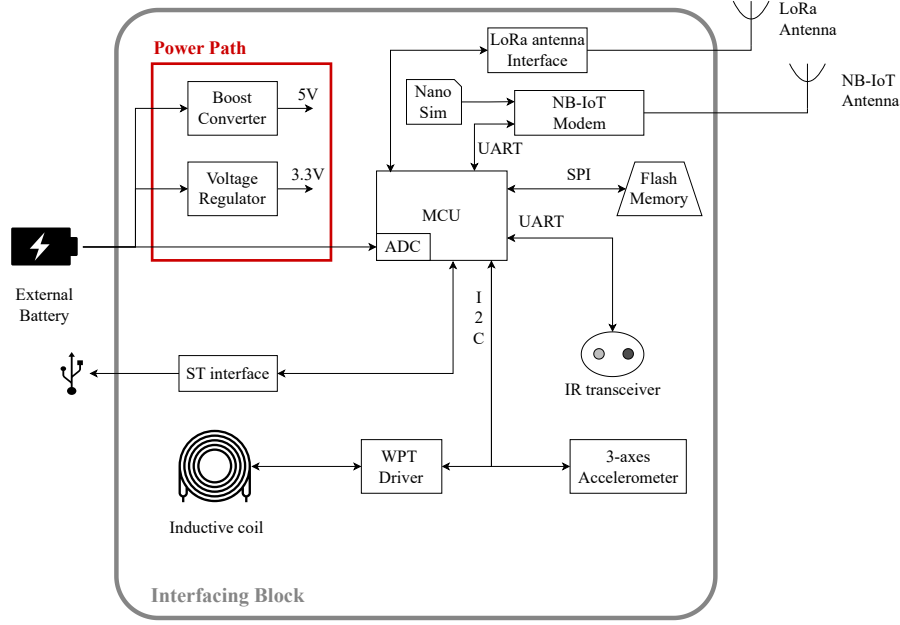


Figure 2.26. Gas meter interfacing part circuit arrangement.

The combined utilization of the LoRa and NB-IoT communication protocols in the design of the smart gas meter is principally dictated by the need for operational flexibility corresponding to the varied installation environments of the device. This dual-technology approach ensures robust data transmission capabilities regardless of the meter's geographical or network density setting.

Specifically, the justification for employing both standards is delineated by distinct deployment scenarios.

Firstly, NB-IoT functionality and compliance are important, as the inclusion of NB-IoT facilitates the direct transmission of data to the internet. This protocol is chosen because it enables the system to comply with Industry 5.0 standards and ensures seamless integration within the existing LTE ecosystem, which is a crucial factor in maintaining device internet connectivity. NB-IoT is the designated communication mode when the smart gas meter operates in isolation within a location, defined as having no other smart gas meters installed within a 50-meter range.

Secondly, the LoRa protocol is deployed as a supplementary or primary method under specific conditions; LoRa is necessary when NB-IoT coverage is limited; in this instance, data is transmitted via LoRa to the nearest covered point.

Lastly, LoRa technology is also strategically implemented in environments where multiple gas meters are installed in proximity, such as within a neighborhood (e.g., within a 50-meter range). In this configuration, data from each device is forwarded using the LoRa protocol (specifically LoRaP2P, chosen for its low-power feasibility and packet customizability) to a local gateway, which then manages the transmission of aggregated data to the internet.

In summary, this integrated LoRa and NB-IoT structure provides adaptability, leveraging NB-IoT for direct internet access and compliance, and utilizing LoRa to extend range, mitigate coverage limitations, and efficiently manage concentrated clusters of devices via local gateways.

Additionally, as shown in Fig. 2.27, the wireless recharge system is implemented through a power management chip and a driver circuit based on the Qi standard [109].

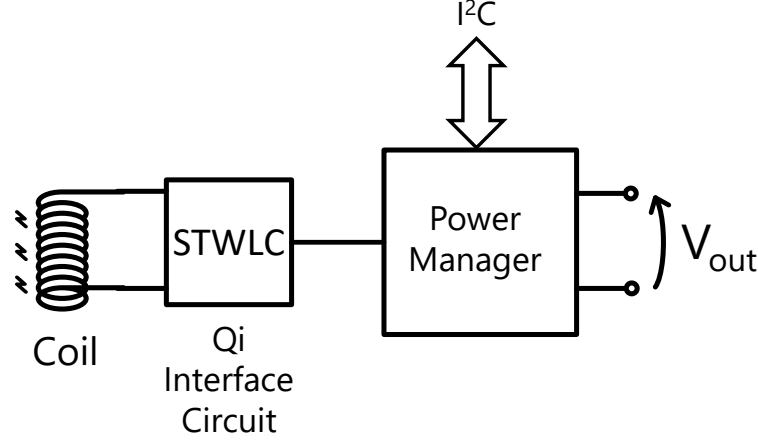


Figure 2.27. Schematic representation of the gas metering unit wireless recharge part.

The power management chip is a battery charger integrated circuit designed for use in chips that require safe and reliable battery charging solutions. This chip is compatible with USB power, as its operative range is 4.35–10.2 V, with provision of a 4.2 V regulated voltage. Additionally, the maximum charge current for the power management chip can be programmed through the I²C standard, with a maximum value of 1.5 A. Most importantly, current-related power consumption reaches approximately 50 μ A in standby mode, with an internal thermal regulation system for overheating prevention, occurring around 120°C.

Centrally, the gas metering interfacing block is composed of an ST-based MCU with integrated LoRa modulation capability, which manages the external peripherals, including an internal accelerometer to detect eventual unwanted removals of the detachable part of the object. Additionally, similarly to the measurement block, a power management path for its power supply is provided.

This block subdivision of the system was chosen for safety reasons, as this modularity mechanically decouples the modules in case of explosions caused by sparks occurring in the measurement block. As a result, the choice of elements in the measurement block preserves the integrity of the gas tube. Additionally, the physical splitting between measurement and interfacing parts brings modularity to the system and eases its maintenance.

2.2.4 Firmware: an Overview with Focus on the Flux Sensor

Firmware is the main non-physical part of the introduced system; in particular, its purpose is to use and manage the peripherals on both the measurement and interfacing block, accordingly to the MCU hardware capabilities. In particular, the system blocks work synchronously, with interrupt-based frames.

In this subsection, the system firmware will be explained in detail, following a top-down approach, with a final focus on the flux sensor firmware, the system main component. Firmware flow diagrams related to both the measurement and the interfacing parts are deepened.

In fact, safety is the main target to assess during the measurement part of the working routine. In particular, as reported in Fig. 2.28, after the startup procedure, where internal sensors are initialized, the MCU enters a low-power idle mode. During the idle mode, the MCU waits for an event to happen. In particular, the event can

either be a timeout for the periodic reading of the internal sensors or an interrupt triggered by either an input on the touch sensor or a possible gas leak. In the eventuality of a timeout, the MCU performs the ordinary periodic reading of battery level through the battery fuel gauge, and the reading of the internal sensors described in Sec.2.2.3. On the other hand, if the event involves an interrupt, its nature is decoded. In particular, for an input command, its decoding and execution are performed. In the case of a non-command, there is a high risk of a gas leak; hence, the gas sensor is read. In the event of a gas leak, an emergency routine is executed; otherwise, an error code is sent. The emergency routine consists of

- the cut-off valve opening;
- the alarm mode enabling;
- the LCD monitor update;
- a final check of the presence of any gas leaks by performing a repeated reading of the gas sensor until the gas leak is detected;
- in the absence of a gas leak, the cut-off valve is closed and alarm mode is disabled.

Subsequently, the LCD display is updated, the packet to send through WPT is prepared, and then it is transmitted.

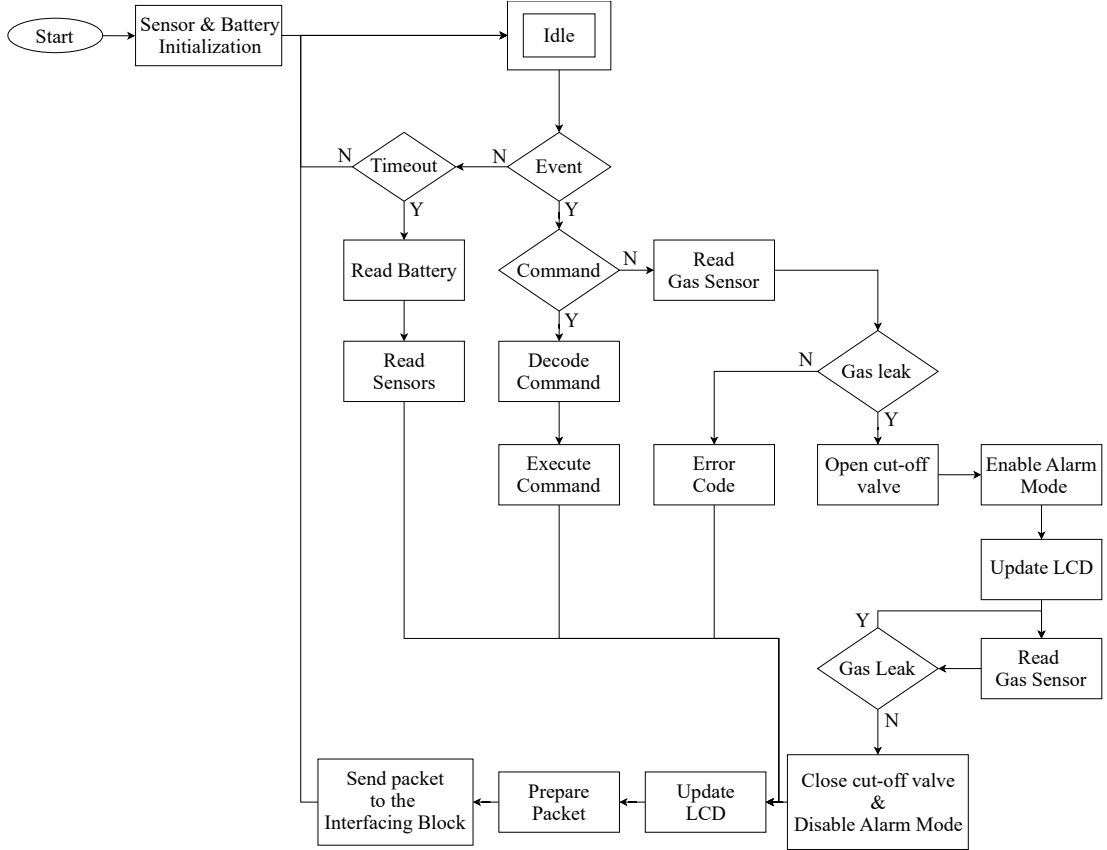


Figure 2.28. Flow diagram representing the measurement block firmmware.

On the interfacing block side, the primary objective is to correctly receive, decode, and forward data through the LoRaP2P data format. Furthermore, since

the interfacing block is a removable part, it is responsible for detecting gas meter tampering, which is performed through periodic checks on the internal accelerometer. In fact, the interfacing block, after the startup routine where sensors and battery are both initialized, enters a listening mode which, similarly to the idle mode, waits for any events in a low-power setting. In case of an event, the battery level is read, and the event is decoded; an event, in this situation, is defined as

1. an incoming packet;
2. a tampering event;
3. a command from the IR optical peripheral.

In the first case, the packet is decoded and re-encoded in the LoRaP2P format. In the case of a non-packet event, the accelerometric sensor value is checked. Exceeding a threshold in the differences of acceleration values causes the detection of system tampering. In case of tampering, the power supply to the measurement block is interrupted, and an alarm packet is prepared and encoded in the LoRaP2P format. In case of non-tampering, it means that an IR command has been sent from the corresponding peripheral; hence, the command is decoded, executed, and the system returns to the listening mode. In both cases of measurement packet or tampering event, the prepared LoRaP2P packet is transmitted to the closest gateway node. In the event of success, the system returns to the low-power listening mode; otherwise, transmission is reattempted a finite number of times before returning to the listening mode. Fig. 2.29 reports the corresponding flow diagram that describes this routine.

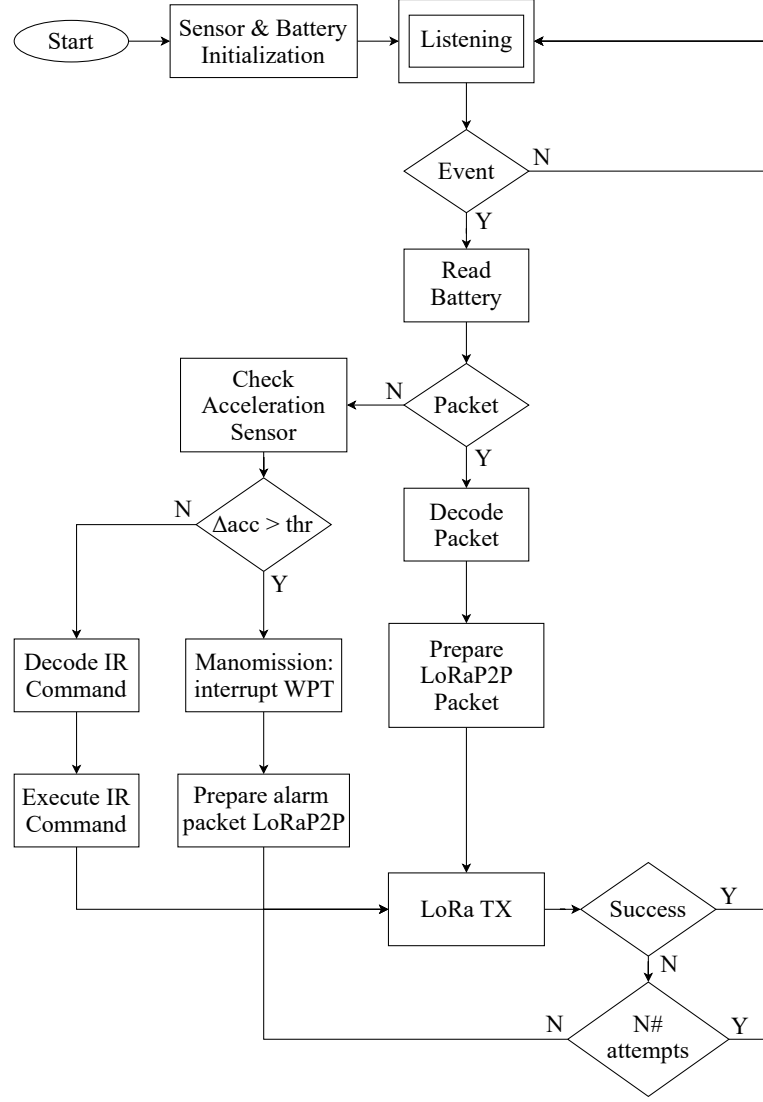


Figure 2.29. Flow diagram representing the interfacing block firmware.

Finally, the flux sensor firmware is discussed below. Since this system part provides the exact flow rate, it is the most important physical part of the gas metering system, having the most influential role. As reported in Fig.2.30, at the startup, the sensor initializes its internal variables, cleans the TX/RX serial buffer, is enabled, and its flow and data rate are set. In particular, since the sensor can provide different types of data, such as the current and cumulative flow, both with and without pressure and temperature compensation, in this initial part of the session, the data type, defined with this difference, is set. Once the startup process is finished, the sensor enters an interrupt-based UART receive mode. Once a packet is received, its Cyclic Redundancy Check (CRC) is first checked to ensure the integrity of the packet. The gas measurement, with the requested data type, is performed and returned to the MCU — the microcontroller can send these values to both the interfacing block through the WPT coil and the LCD screen with an external firmware. Once the flow measurement is performed, the node returns to the receive mode and remains in this status until the system is reset or turned off.

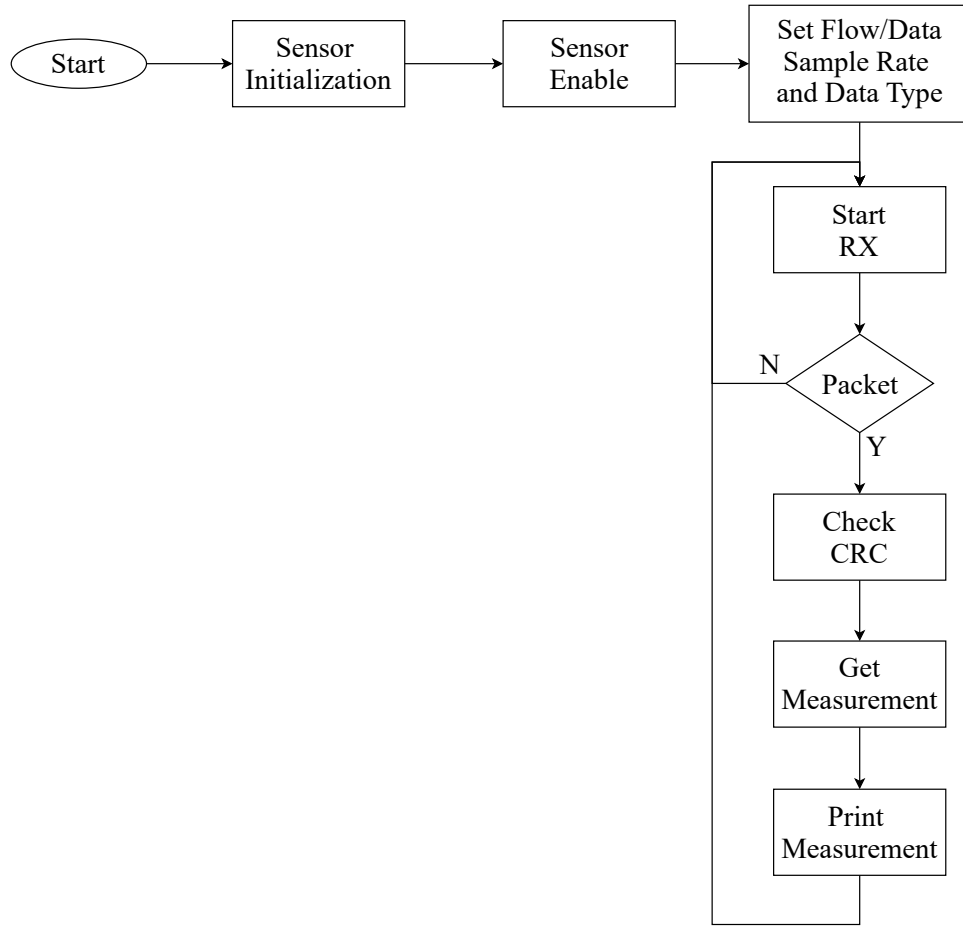
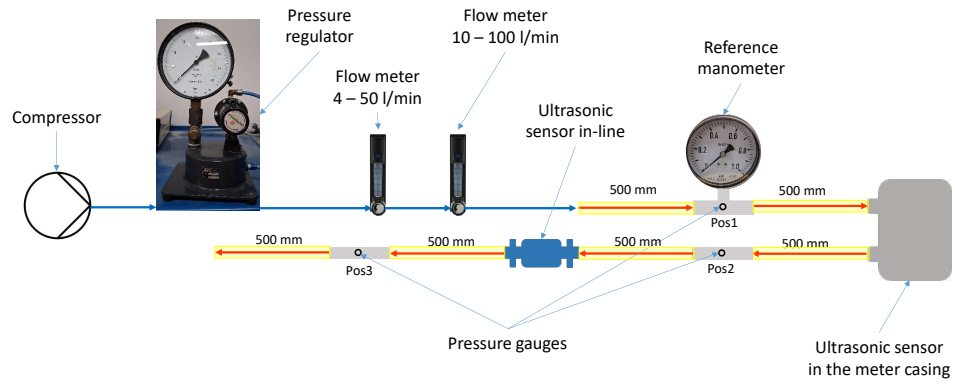


Figure 2.30. Flow diagram representing the flux sensor firmmware.

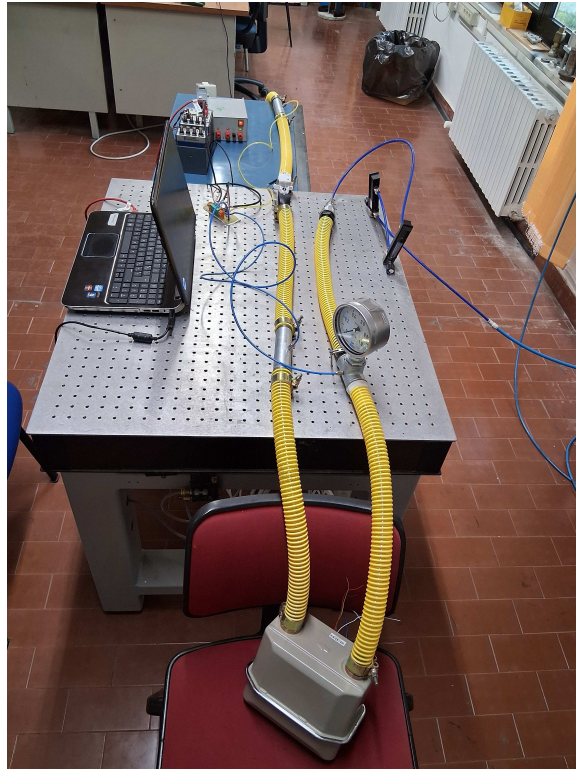
2.2.5 Flow Sensor Metrological Analysis and Data Transmission Test

The validation methodology was designed to assess the accuracy of measured data by studying the fluid dynamic effects of the geometrical layout on the ultrasonic flow sensor, and by quantifying the impact of environmental temperature and gas pressure across varying flow rate values.

The core of the validation consisted of an experimental reference circuit, reported in Fig.2.31a. This setup utilized a compressor and a pressure regulator to maintain a stable airflow, and two reference flowmeters (with an uncertainty of 2.5%) were connected in series for comparison. The testbench used in the laboratory is shown in Fig.2.31b



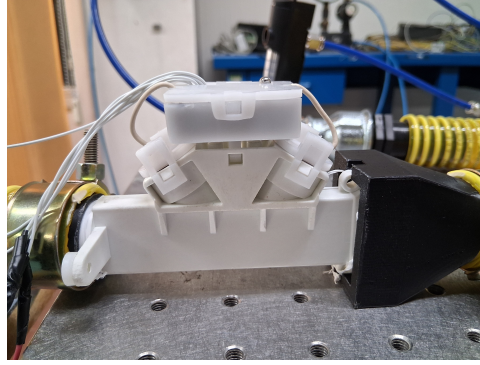
(a)



(b)

Figure 2.31. Setup diagram (a) and testbench (b) for the metrological evaluation [110].

Two ultrasonic flow sensors (Panasensing Spiro-L-Horiz, class 1.5) were tested. As shown in Fig.2.32, the first sensor was installed in-line with connectors, while the other was inserted into a classic meter casing.



(a)



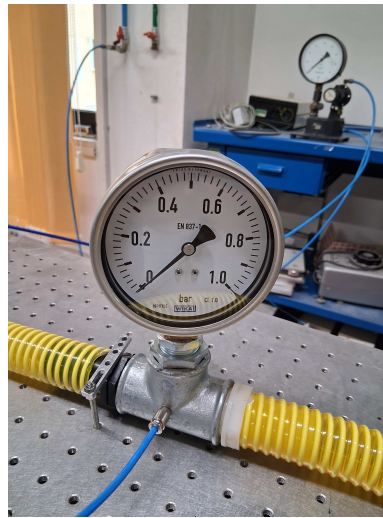
(b)

Figure 2.32. Reference flow meters: in-line (a) and cased (b) flow meters [110].

To monitor gas pressure, three MEMS Gauge Pressure Sensors (Omron 2SMPP, Fig.2.33a) were strategically positioned upstream and downstream of the two ultrasonic flowmeters at points labeled “Pos1”, “Pos2”, and “Pos3” (Fig.2.31a). The outcome from the MEMS sensors is compared with a reference manometer, reported in Fig.2.33b. The setup ensured compliance with standards by maintaining straight pipes leading to the ultrasonic meter inlets at greater than ten times the pipe diameter (30 mm).



(a)



(b)

Figure 2.33. Pressure sensors (a) and reference pressure gauge (b) [110].

Sensor calibration was performed against a reference Bourdon Tube Pressure Gauge (Wika). The resulting calibration, shown in Fig.2.34, yielded an uncertainty of 0.005 bar.

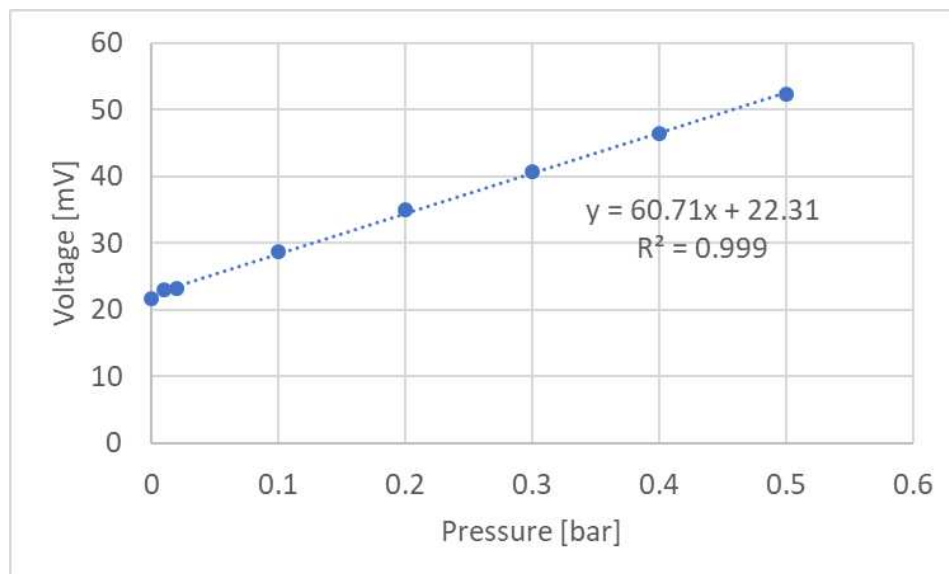


Figure 2.34. Pressure sensors calibration diagram. [110].

Tests were conducted across three flow conditions: $3.3 \text{ m}^3/\text{h}$, $4.8 \text{ m}^3/\text{h}$, and $6.0 \text{ m}^3/\text{h}$, covering the full operating range for domestic gas pipes.

Preliminary tests focused primarily on pressure behavior (fluid dynamics) within the gas loop. These tests were executed under two environmental conditions: Condition (a) (884 mbar, 27.0°C) and Condition (b) (889 mbar, 25.5°C).

The results, shown in Fig.2.35 demonstrated that at lower flow rates, the ultrasonic meter did not introduce significant pressure drops. However, at the highest tested flow rate ($6.0 \text{ m}^3/\text{h}$), the pressure drop associated with the sensor in the casing exceeded the 2.0 mbar standard limit. This deviation was attributed to both the fluid dynamic path and an imperfect system seal, indicating a need for refined sealing solutions in the final design.

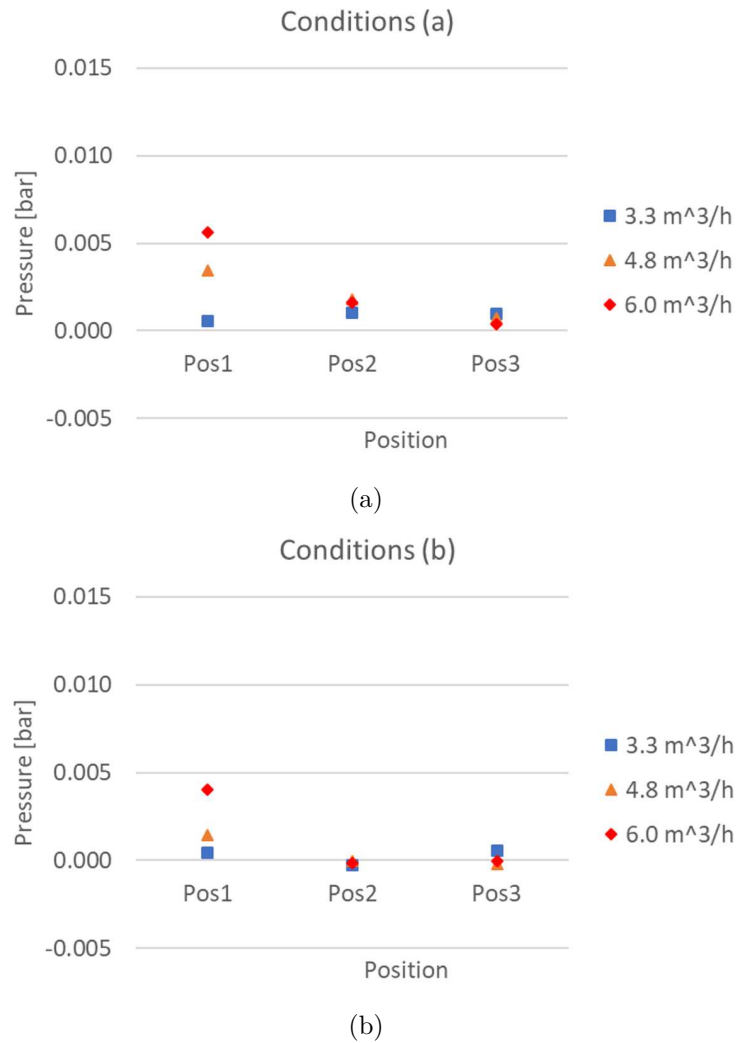


Figure 2.35. Pressure measurements at different positions in the test configuration, in three different flow conditions, and in two environmental conditions (a) and (b). [110].

Regarding flow rate validation, five repeated measurements were performed for each flow condition. The results showed that:

- the standard deviation of repeated measurements was approximately 2%;

- the maximum relative error of the mean flow rate, when compared to the reference flowmeters, ranged from 4% to 5%.

These initial findings confirmed that the proposed geometrical layout simplifies the data normalization process using a minimal set of sensors.

Additionally, for experimental reasons, the Spiro-h-horiz flow sensor firmware described in Sec . 2.2.4 has been adapted and implemented on a Nucleo-L452RE board, as reported in Fig. 2.36.

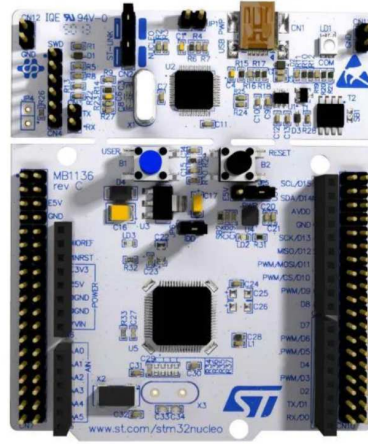


Figure 2.36. Nucleo-L452RE board from Mouser Electronics, Inc.

<https://www.mouser.it/ProductDetail/STMicroelectronics/NUCLEO-L452RE?qs=zwwtnr6GuIyXyX5dw%2FgKaw%3D%3D>.

In particular, firmware enhancements involved the predisposal for serial communication between MCU and an external operator, which reads sensor measurements through USB. For this purpose, an FTDI FT232RL UART to USB converter was used. In fact, the MCU firstly retrieves data from the flow sensor; secondly, data is analysed and enclosed in a package with an established format; finally, the packet is sent, through the FTDI chip, to the USB port of the user's machine, as represented in Fig. 2.37.

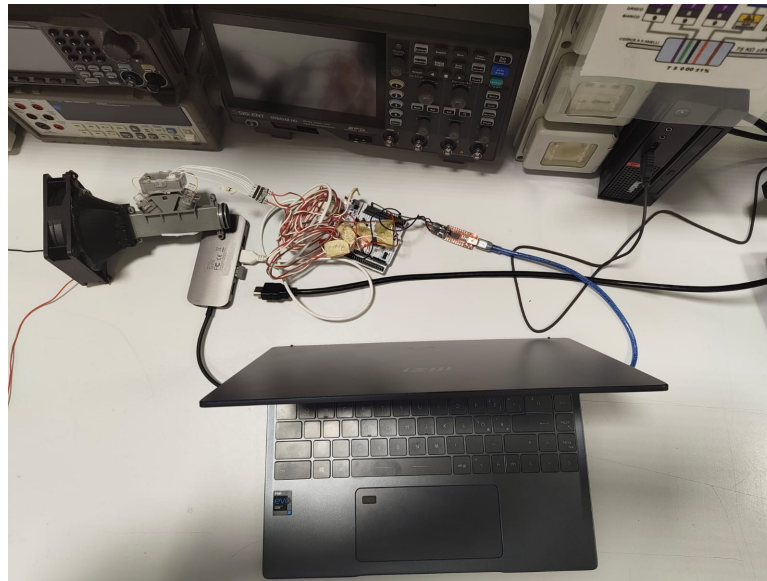


Figure 2.37. Flow sensor measurement setup.

Data is read through the SerialPlot software [111]. With this solution, it is possible to plot sensor data over time, with the chance to export data to a separate file for its fine graphical representation. The sensor is enabled through an ad-hoc fan mounted on its input side. The fan speed is linearly varied through a DC voltage generator. Data from the sensor is split into three main channels:

- temperature, measured in Celsius degrees ($^{\circ}\text{C}$);
- instantaneous flow, which can be both positive and negative, measured in cubic meters per hour (m^3/h);
- cumulative flow, which, by definition, is the integral of the instantaneous flow, incrementing each time a positive instantaneous flow sample is detected; this parameter is measured in cubic meters (m^3).

Results from a test measurement are reported in Fig. 2.38.

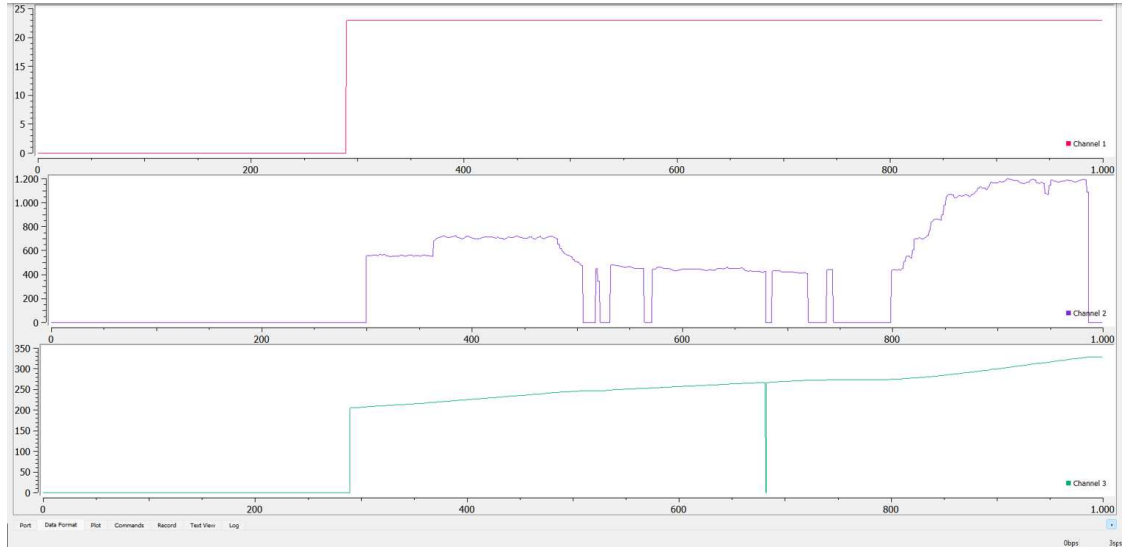


Figure 2.38. Flow sensor preliminary test results.

The fan speed was set to several values to check the sensor response and the slope variation in the cumulative flow. As a result, the sensor detects changes in input flow successfully, with a sampling rate of 0.5 samples per second.

2.2.6 Blockchain Structure

The data security strategy leverages blockchain technologies and predefined rules to manage “energy tokens”—digital representations of energy transactions. This system guarantees traceability, transparency, and safety, aiming to foster new market models that support direct peer-to-peer interactions between energy actors, moving away from centralized control.

A three-level model, shown in Fig. 2.39, was implemented to optimize resource usage for devices with limited computational resources:

1. edge/gas meter nodes: in-situ devices performing local measurements, configurable as consumers, producers, or prosumers;

2. aggregating nodes: responsible for data acquisition, processing, and transmission from the edge nodes;
3. authority nodes: responsible for system validation and supervision.

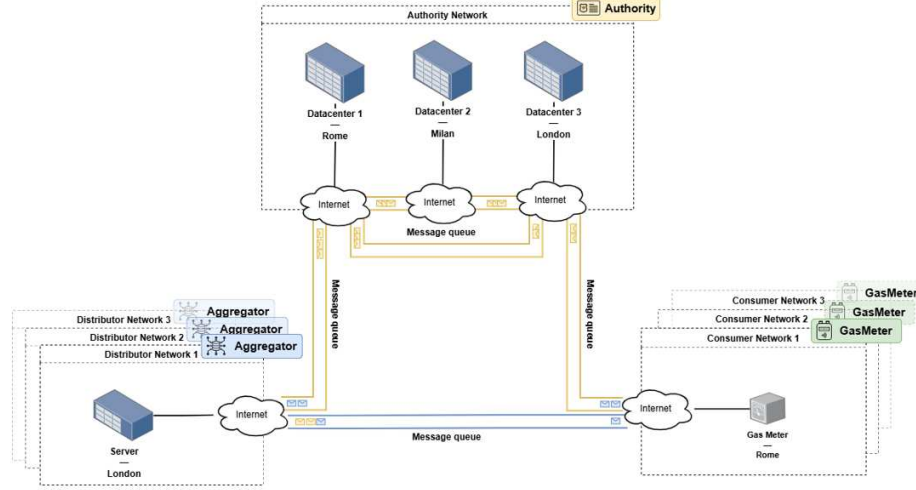


Figure 2.39. The proposed communication infrastructure [110].

The experimental system utilizes a private permissioned blockchain, specifically Hyperledger Besu, and employs the Proof of Authority (PoA) consensus algorithm. PoA mandates a minimum of three authority nodes to prevent a single point of failure. A consensus threshold of slightly over $2/3$ (approximately 67%) of authorities is required to approve transactions.

Key operational parameters include a block time of approximately 5 seconds in the test configuration, and a theoretical capacity estimated at 200–250 energy transactions per block.

The distributed validation process involves the following steps:

1. edge node sends the transaction to an aggregator;
2. aggregator sends the batch to three or more authority nodes;
3. authorities validate the batch independently;
4. if 67%+ of authorities approve, the batch is created and shared;
5. all nodes update their local site.

Tokens exist exclusively in the distributed ledger, and their ownership and status are managed via signed transactions using cryptography. Tokens can hold three statuses: created (available), transferred (available), or consumed (final status, used to assess consumption).

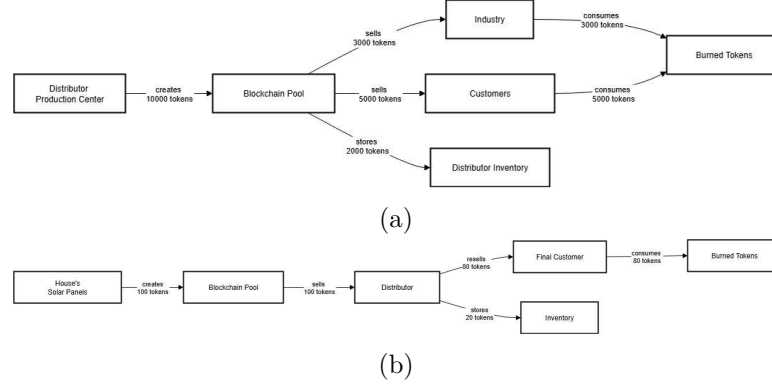


Figure 2.40. Distributor-consumer (a) and prosumer-consumer (b) tokens life cycle. [110].

The infrastructure was simulated using the software “Graphical Network Simulator 3” with Docker containerization to verify compliance with the predicted computational load. The simulation topology, shown in Fig. 2.41, included dedicated switches for the authority, aggregator, and consumer edge layers, connected via a central ISPSwitch to simulate a realistic geographical distribution.

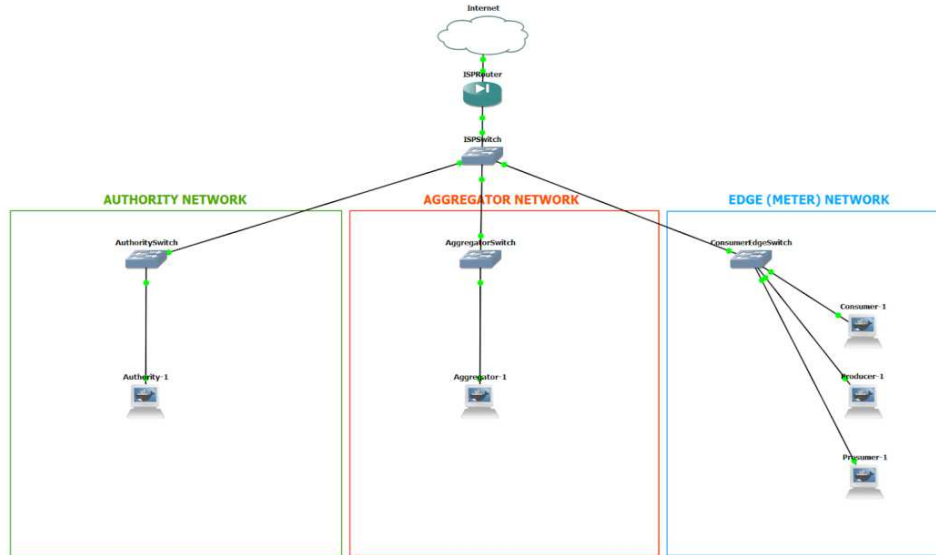


Figure 2.41. Infrastructure used for the simulation. [110].

The aggregator was configured with a batch interval of 30 seconds. The observed throughput in nominal test scenarios was approximately 100 transactions/minute, with a maximum theoretical capacity estimated at $\simeq 240$ transactions/minute. The batch architecture was specifically designed to amortize the initial fixed gas cost, reducing the average cost per transaction on the blockchain.

End-to-end latencies were measured during validation tests:

- edge-aggregator latency (via MQTT): 50–150 ms.
- smart contract execution: 800–1200 ms.
- block confirmation (PoA time): $\simeq 5$ s.
- total end-to-end latency: ranged from 6 to 37 seconds (average 20 s).

The variability in total latency is a significant function of the transaction's arrival time relative to the 30-second batching cycle. Preliminary checks confirmed the suitability of bidirectional communication between all node types. The simulation demonstrated that the infrastructure could manage a network of up to a few thousand end-users with an allowable computational load.

2.3 Structural Health Monitoring

As introduced in chapter 1, LoRa is widely used in industry. The previous sections showed how the characteristics of LoRa can enhance critical operations such as environmental monitoring and smart metering. Still, SHM is an important field where LoRa can make a significant difference, especially when large structures are considered.

In this section, a work based on the SHM through spectrum analysis is reported. SHM is defined as a sensor-based process where electronic systems are merged with data analysis to observe and determine the structural integrity of a building or infrastructure, ensuring safety and reducing risk factors.

Monitoring is implemented through periodically sampled measurements for material and geometric variations. The factors that identify the building status and the system technologies have changed over time; among the various techniques, in the last decade, modal harmonics analysis received special attention. In these implementations, modal harmonic analysis is performed on the results from accelerometer-based systems integrated with the building itself.

The activity presented in this section involves the use of a multi-nodal redundant system for industry purposes, where the integrity of a structure is analyzed through the examination of its modal frequencies. The primary characteristics of this system are its internal redundancy in multiple areas, which ensures a durable structure, and its autonomy due to battery compatibility. A modular construction that is subject to disturbance has been used to test the system. Results in the time and frequency domains have been acquired both before and after a structural modification.

The section is structured as follows: in Sec. 2.3.1, an introduction involving scientific literature and state of the art is presented, in Sec. 2.3.2, the system is described, while in Sec. 2.3.3, measurement results are reported and technically commented.

2.3.1 Introduction

The topic of SHM has gained increasing importance in recent years, particularly with the evolution of electronics and network frameworks for their implementation. This importance is especially due to the daily threats to human life posed by climate change that are impacting world nations. For this reason, the development of intelligent structures is becoming a highly categorized necessity; hence, the topic itself is generating an ever-increasing interest in the scientific area. In this regard, numerous studies conducted in recent years have attempted to emphasize the well-being of structures by taking into account their surroundings, hence, conducting the “all-round” monitoring [112–117].

The high importance of the SHM, in light of the recent research, is the proper notification of infrastructure users to enable the building evacuation and adequate maintenance before the occurrence of serious damage or even collapse [118, 119]. Additionally, the implementation of proper SHM has a good economic impact due to the elimination of the need for expensive traditional diagnostic techniques such as ultrasonography [120]. Besides, the achievement of an optimal, objective, and general SHM is impossible, since each structure has unique features, e.g., size, usage patterns, age of construction, type of stresses that it experiences, usage nature (building or bridge) [121]. Another challenge is managing two distinct circumstances: on the

one hand, the low safety resulting from natural obsolescence, and on the other, the consequence of catastrophic, destructive events like floods and landslides [122]. Therefore, a proper SHM system is based on a system comprising several general-purpose nodes, housing specific hardware such as sensors and data processing systems, and a health assessment system to discover a cohesive solution to the issue [123]. According to the literature [124–127], the excellence of these systems is given mainly by factors such as:

- flexibility
- reliability
- time in information transfer
- degree of detection accuracy
- costs
- non-destructivity of the approach
- management easiness
- capability of storing and reading large amounts of data smoothly

With this premise, the SHM theme has been occasionally reinterpreted in various ways, primarily considering the specific monitoring requirements. As a matter of fact, monitoring can be dynamic if done continuously over time, or static if limited to structurally crucial points [128–132]. From a different perspective, monitoring can be wireless [133, 134], a more suitable solution for enormous infrastructures and generally for places where wiring is not feasible. Alternatively, wired monitoring [135–137] is more reliable and appropriate for small buildings.

Additionally, the system complexity and the subject matter of study can vary extensively, encompassing modal frequencies [138, 139], hybrid systems [140], neural networks [141], MEMS sensors [142–146], and damage index [125, 130, 147, 148]. Generally, as mentioned above, it is impossible to determine the optimal strategy. The system discussed in this chapter is a master-multi slave, based on accelerometer sensors, and is employed for modal harmonic analysis. The primary objective of this system is power autonomy and internal redundancy, which are achieved through the use of double accelerometric units, dual-protocol communication, and the implementation of CRC for packet integrity.

2.3.2 System Description

In this section, the presented SHM system is described on different layers of abstraction. The primary characteristic of this system is redundancy, which is achieved in various aspects of the system, ranging from communication between on-field nodes and gathering nodes to measurement, where two different accelerometers have been used. Firstly, the communication part of the system is covered, highlighting its master-slave nature. Secondly, the hardware of each node is deepened to show the base chosen for the firmware. Lastly, the operating principle of the system, both for the nodes and the master, is discussed and depicted through flow diagrams. Specifically, this latter part regards the firmware of the entire system.

Communication

As mentioned above, the system is multi-nodal; therefore, it consists of several slave nodes that retrieve data and communicate with a master datalogger. As depicted in Fig. 2.42, communication is managed with redundancy: each slave can communicate with the master by using both a wired and a wireless protocol. The datalogger post-processes input node data. Among the various types of data, the Fast Fourier Transform (FFT) calculation for each node displacement constitutes a usual example.

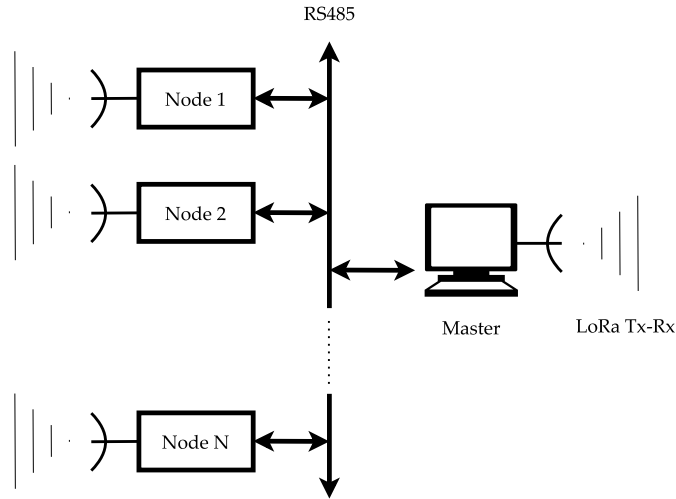


Figure 2.42. High-level scheme of the system based on communication [149].

The redundancy in communication protocols increases the system robustness and its usability in SHM; particularly, the wired standard is RS-485, while the wireless standard is LoRa. The choice of LoRa is related to the low-power feasibility and range that it can reach. In particular, it is a widely used standard, operating in standard frequency ranges. Additionally, due to the full communication control, customization of packets, and peer-to-peer operability, the LoRaP2P, on the physical layer, is selected for this purpose. On the other hand, for test reasons, or when data integrity and high data transfer frequency are needed, the wired protocol, the RS-485, is chosen over the LoRa. With the RS-485 standard, slave nodes can be placed with a distance up to several hundred meters from each other. As a result, redundancy is reached as a trade-off between data integrity and installation requirements; hence, the user can set the node communication language; in particular

- when the node distance is under one kilometer, both communication standards can be used promptly, but RS-485 is the main protocol, while LoRa is a backup;
- when the node distance exceeds one kilometer, LoRa is chosen as the only communication protocol since it can reach these distances.

Data integrity is additionally assessed by the use of the CRC between master and slave nodes. The packet error detection CRC technique, used in digital data transmission, is based on the addition of a mathematical expression, known as a checksum, to data packets. Throughout the entire working interval of the system, to improve reliability, the datalogger generates a checksum and sends it to the slave units as part of a data

packet. As a result, the master-generated checksum must be included by the slave nodes in their packets. Specifically, in the proposed custom architecture, there are a CRCI and a CRCh in each packet; these two values refer, respectively, to the low and high bytes.

Hardware Architecture

The single slave node in the system is composed of:

- An MCU for peripheral management;
- A Global Positioning System (GPS) to assess node positions;
- Two accelerometers for robust and redundant acceleration measurement;
- a LoRa unit for wireless communication implementation;
- an RS-485 interfacing unit for wired communication implementation;
- A 1 Gb NAND flash unit for data storing;
- A power path to provide a specific and constant supply to each device operating in the node.

The ESP32-WROOM-32UE from Espressif Systems [150] was chosen as MCU due to its predisposition to IoT applications. As mentioned, nodes have a GPS device, the FGPMMPA6H [151], which uses the National Marine Electronics Association (NMEA) 0183 standard to communicate with the MCU. This device was chosen due to its -165 dBm signal sensitivity, instant Time To First Fix (TTFF), and low power consumption, since in our system, negligible power consumption must be guaranteed, whereas nodes have hidden positions.

The use of two accelerometers provides tri-axial node displacement. In particular, there is a main accelerometer, the ADXL355 [152], and a backup accelerometer, the ADXL345 [153]. This choice ensures redundancy in measurements without considerably losing resolution from one accelerometer to the other. As reported in Table 2.2, the ADXL355 has less noise sensitivity, higher accuracy, and higher resolution with respect to the ADXL345, which, on the other hand, presents a lower current consumption and uses an easier communication protocol.

Parameter	ADXL355	ADXL345
Noise Density [$\frac{\mu g}{\sqrt{Hz}}$]	25	150
Accuracy [mg]	2	20
Resolution [bit]	20	10
Current Consumption [μA]	200	23
Communication Protocol	SPI	I2C

Table 2.2. Technical differences between ADXL355 and ADXL345 accelerometers.

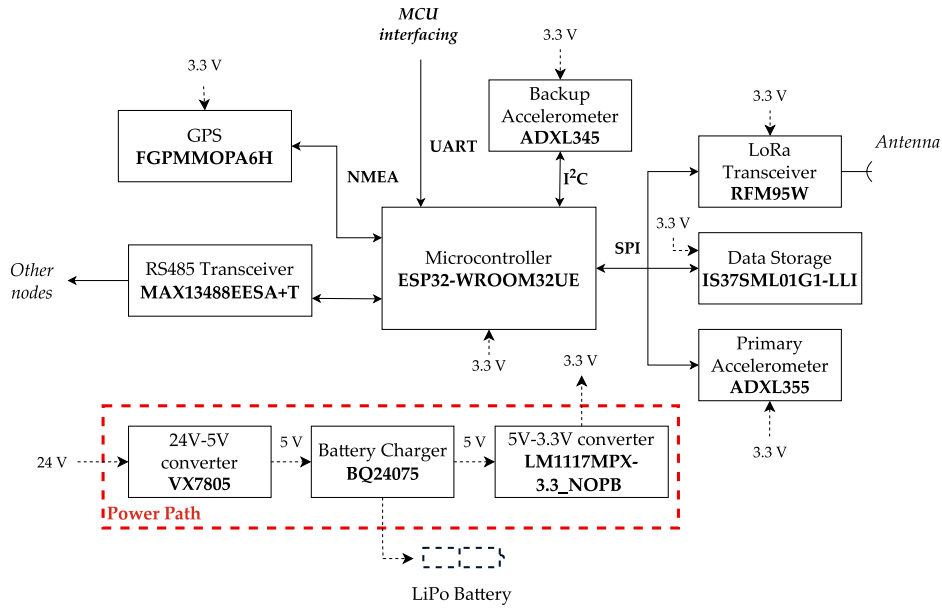
After each measurement, data from the main accelerometer is saved in the IS37SML01G1–174 LLI [154] 1 Gb NAND flash memory. This choice is due to the high sample rate and the single-node backup provision. This system part

communicates with the MCU through SPI, hence, it shares the communication channel with the ADXL355 accelerometer.

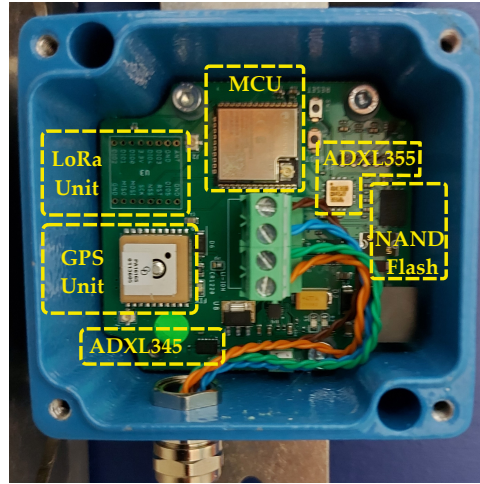
Additionally, since data interchange is performed on both RS-485 and LoRa, the node contains data transceivers for both technologies. In particular, the MAX13488EESA+T [155] RS-485 transceiver is characterized by AutoDirection control, a proprietary technology that automatically manages data direction through hardware without relying on any additional control signals. Furthermore, the transmission can be sped up to 16 Mbps, with an error-free rate of 500 kbps. On the other hand, the RFM95W [156] LoRa transceiver is characterized by a -148 dBm sensitivity, a 10.3 mA current consumption in reception mode, which is relatively low, and a bit-rate up to 300 kbps.

Finally, power supply to each hardware block in the node is provided by a power path. The first unit, the VX7805 [157], converts the node voltage supply from 24 V, used for the RS-485 interfacing, to 5 V. Afterwards, the BQ24075 [158] battery charger unit is integrated in the node for the eventual integration of a LiPo battery in future applications. The 5 V power supply is then converted to 3.3 V through the use of an LM1117MPX [159], providing the supply voltage for all modules in the single node.

The node components described above have been placed and designed on a board, which, once manufactured, was installed in a metal enclosure using screws. To minimize the noise caused by Printed Circuit Boards (PCB) oscillations, both accelerometers were strategically placed in proximity of the PCB anchor points. The single node, both as a schematic and as a circuit, is shown in Fig. 2.43a and Fig. 2.43b, respectively.



(a)



(b)

Figure 2.43. Node scheme reporting the used hardware (a), and final assembled circuit (b) [149].

Operating Principle

This section dives deeper into the sequential operation performed by each slave node. In particular, the reported actions concern node operations from the activation to the stable working state. The first action performed by the node is the setup of the connected peripherals according to two parameters sent by the master block: test time and accelerometer sampling time. These actions prepare the node for an idle phase, where the node waits for a command from the master unit, such as data sampling, mission execution, or internal memory control. The command in object is referred to the RS-485 protocol command set since LoRa, on its physical layer, does not feature specific commands, and RS-485 is used in the testing and prototyping phases. The mission mentioned above is defined as the sequence of measurements, i.e., data sampling, performed within a pre-defined time window, and data sent to the master unit. As a result, if the master requests mission initialization to the node,

its internal accelerometers are set up with the parameters sent by the master. Once the accelerometers are configured, the node checks for the presence of GPS signals. If a GPS fix is detected, the node downloads its GPS data, i.e., geographical and time data from the satellite. If, by contrast, there is no detection of the GPS fix, the node will execute the command sent by the master node likewise, without taking into account the node positions. This process is graphically represented in Fig. 2.44.

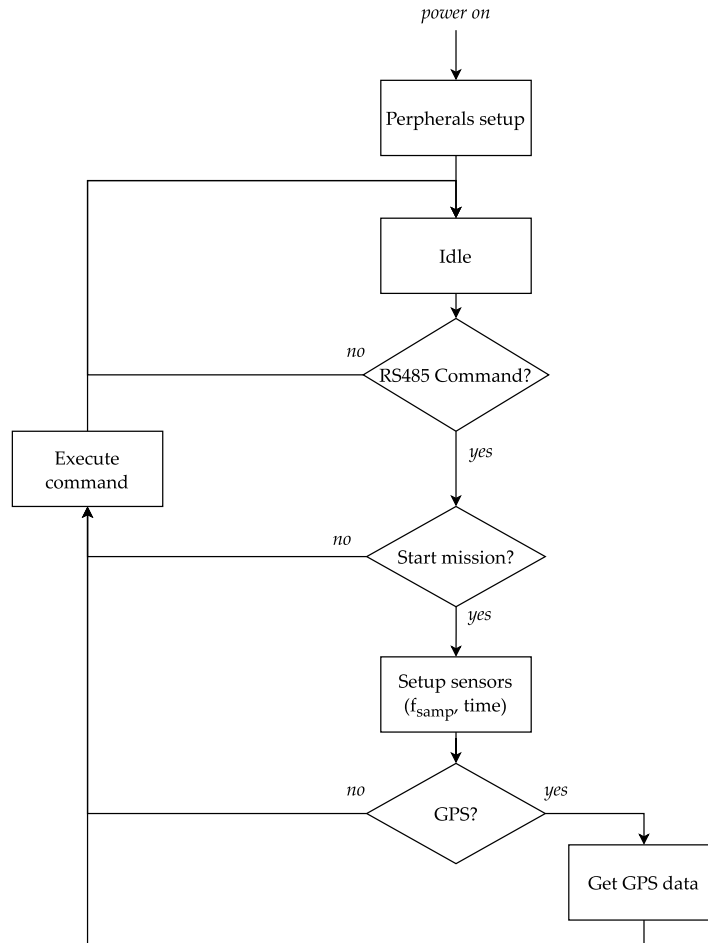


Figure 2.44. Node algorithmic routine [149].

Functional and Data Description

The main system functions and the exchanged data structure between master and slave nodes are described in this section.

Regarding system functionalities, a single slave node can interact with the system depending on its status. In particular, each node is distinguished by the master through an individual ID; the master uses the “rollcall” function to test system operability, vary the node IDs, and bus baud rate. Specifically, in the rollcall function, the master broadcasts a ping packet to all nodes and, depending on the number of response messages, the master can evaluate the system integrity and detect faulty nodes. This command is typically sent either after the system first starts or in the time interval between two missions. An ordinary mission session performed by the system, where the structure under test gets altered, as described in Fig. 2.45, has the following steps:

1. as the test starts, the measurement type, testing time, and sample rate parameters are set by the user;
2. the MCU stores the input parameters provided in the previous step;
3. nodes are started through the rollcall function, used as an awakening function;
4. once the master learns about the system integrity, it sends a command to the nodes, related to the measurement type defined by the user, so that they start their measurements;
5. as a result, each node acquires a data sample and sends it to the master;
6. the master makes a comparison between the passed time, as far as it is less than the testing time provided as input, the previous two steps are repeated;
7. otherwise, if the time passed is higher than the testing time, the master does not send an additional “measure” command to the nodes, thus, data is downloaded from the master;
8. after this first test part, the structure should be altered to repeat the test and gather additional data that can be compared with the pre-alteration numbers;
9. once the test, before and after alteration, has been performed, the whole process is finished.

These steps constitute the procedure that will be followed during the tests, leading to the presented results. After measurements, the stored data is analyzed through a spectrogram, which allows for observing the data frequency spectrum as a function of the mission time. This last step represents a post-processing algorithm performed on a separate calculator, since stored data can be downloaded from the master unit.

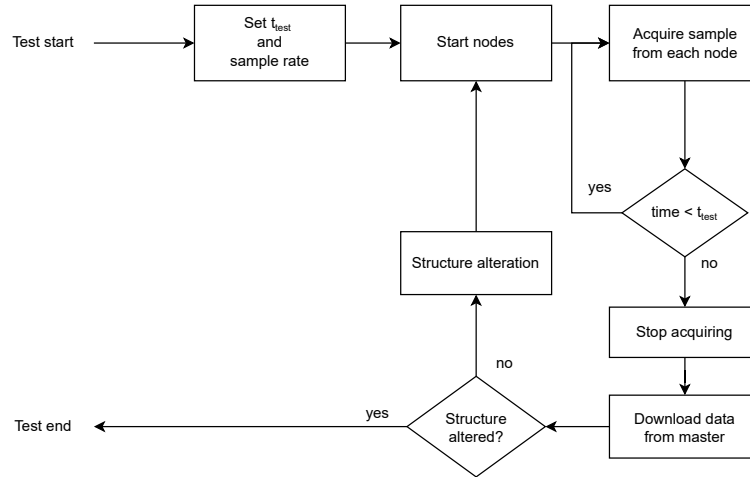


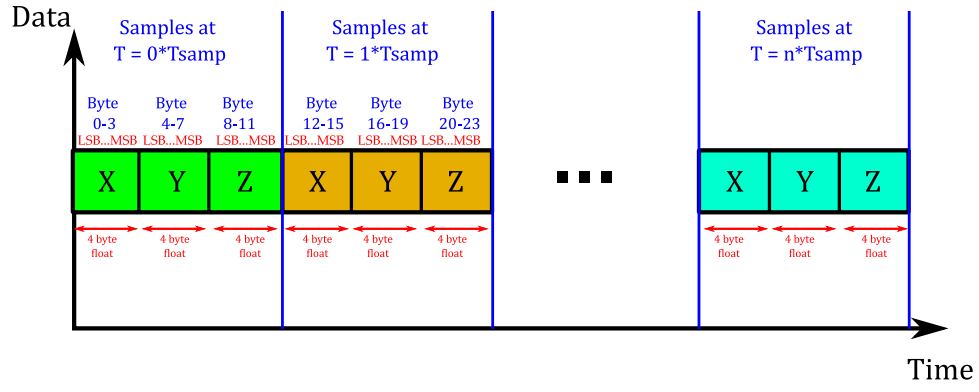
Figure 2.45. Experimental tests operational flow diagram [149].

As mentioned previously, the user can choose among several measurement types. According to this choice, the master will receive a different kind of data from the nodes. The main data types considered for this system are now discussed and deepened. Firstly, data can consist of the nodes inclinations during the entire mission time window, expressed through time-varying data from their internal accelerometers,

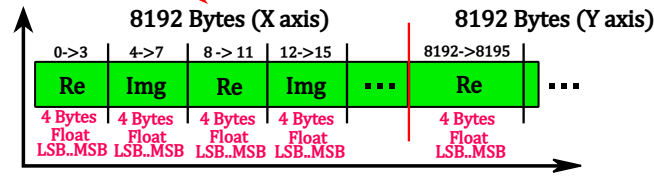
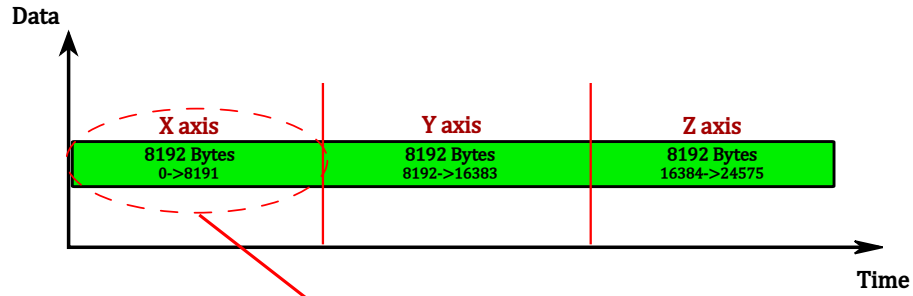
as reported in Fig. 2.46a. The packet related to this data type is composed of 12 bytes in float format, containing three 4-byte groups, one for each of the directional axes.

Secondly, the cumulative FFT can be extracted from accelerometric data. In particular, the calculation of the FFT is performed over a time-based average of inclinometric data gathered in the mission. The related packet, reported in Fig. 2.46b, occupies 24.5 kB, equally composed of 8.2 kB FFT data for each axis. Specifically, each 8.2 kB data chunk is composed of a repetitive sequence, occupying 8 bytes. Each sequence is equally divided into the FFT real and imaginary parts.

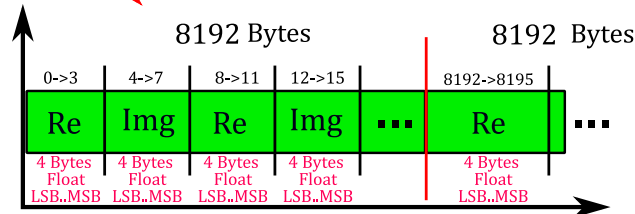
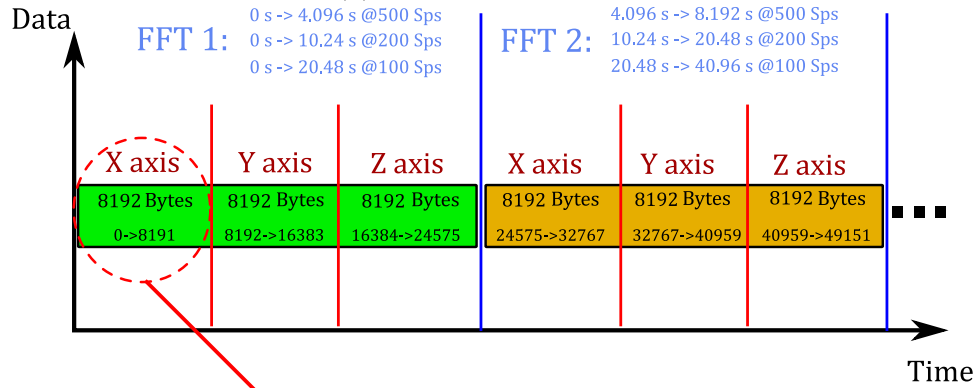
Finally, the spectrogram is the third data type retrievable from the slave nodes and is shown in Fig. 2.46c. The spectrogram is represented as an axis-wise FFT reported throughout the mission time, hence, a three-dimensional graph with color gradients. This data representation allows the analysis of modal harmonics over time. For this data type, the bit organization is similar to the cumulative FFT packet, with the addition of the sample number per second. The programmable sampling frequency, chosen by the user, between 100 and 500 Samples Per second (SPs), is considered in this type of measurement, as it directly depends on the testing time. The sampling frequency is a crucial parameter, as its impact on measurement resolution and accuracy is important for structural damage analysis.



(a) Inclinometer Data.



(b) Cumulative FFT data.



(c) Spectrogram data.

Figure 2.46. Packet format according to each type of data [149].

2.3.3 Setup and Results

The analyzed structure, reported in Fig. 2.47, is a solid wood modular box. A system composed of three hinges and three hooks holds this modular box together, enabling it to be opened and closed. In particular, the structure was tested by considering a first scenario with a closed box, representing the entire status of the structure, and a second scenario where the box is opened, simulating the structural alteration. The altered scenario is reported in Fig. 2.48.



Figure 2.47. Structure under test [149].

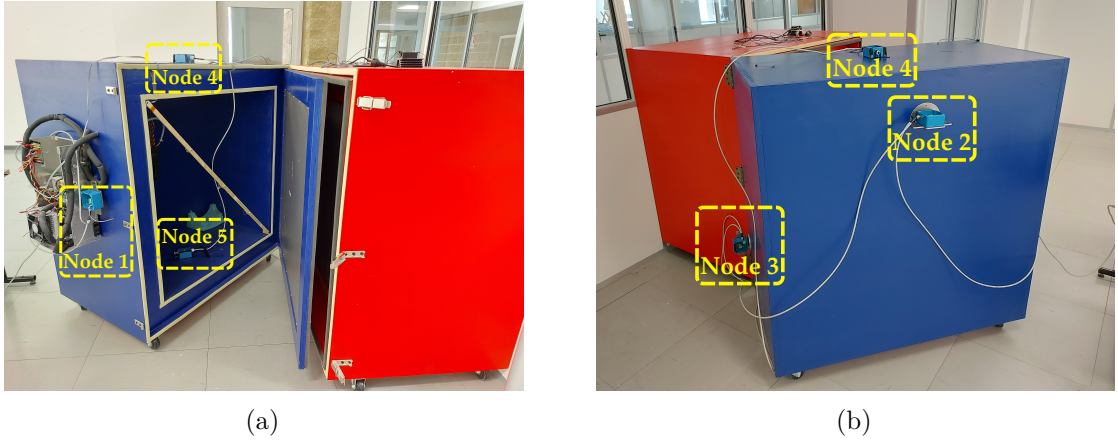


Figure 2.48. Nodes arrangement [149].

As is clear from Fig. 2.48, five nodes are used; in particular, four nodes are placed on the external side of the box, while the last one is placed inside. With this choice, the primary target is to avoid any overlapping of nodal points, ensuring the maximum clarity and completeness possible in the measurements. As mentioned, in this preliminary testing phase, RS-485 was selected as communication protocol. Additionally, this setup was chosen to validate the sensor networking performance in detecting structural modifications. To accomplish this goal, data acquired from each node is compared before and after the structural alteration.

Consequently, the experimental part of this work reports three different tests:

1. Firstly, no stress was applied to the system. The purpose of this test is to demonstrate the differences between main and secondary accelerometers.
2. In the second test, the structure was subject to repetitive stress by a mechanical piston. The purpose is to acknowledge the system response when the structure is intact.
3. In the third test, the structure was altered and, again, subject to the same repetitive stress applied in the second test. Therefore, the purpose is to gather data related to the modified structure.

In each test, the sampling frequency was set to 500 Hz to maximize data reliability. At the same time, a scale of ± 2 g was chosen for both accelerometric sensors, as in real situations, the system is subject to environmental stimuli, thus, low-amplitude accelerations. After these three tests, signals and spectrograms from both before- and after-alteration responses are compared.

For the first experiment, pre-test expectations were high for the main sensor compared to the backup one, due to their features. Due to the considerable resolution difference between the two accelerometers, it is expected that the detection of certain signals will be by the main accelerometer only. For this test, testing time was set to 30 seconds. Results are reported in Fig. 2.49.

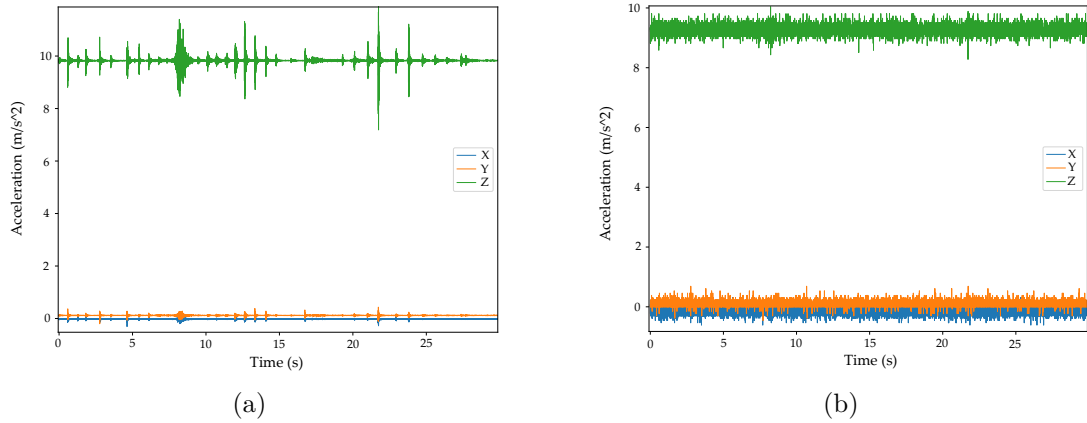


Figure 2.49. Comparison of ADXL355 (a) and ADXL355 (b) under no stimulation [149].

As a result, minimal indirect structural perturbations, e.g., step-induced accelerations by pedestrians passing near the box, were clearly detected by the main accelerometer, and hardly detected by the backup accelerometer, since these pulses are buried in environmental noise.

In the second test, as mentioned, the structure was subject to periodic perturbation through directly induced stimuli. A broader time window was needed to put the system in proper operating conditions, a process that involves, e.g., setting up the mechanical piston. For this reason, the testing time was set to 60 seconds. Node 5 is the node with the most valid data; hence, its results are reported in Fig. 2.50.

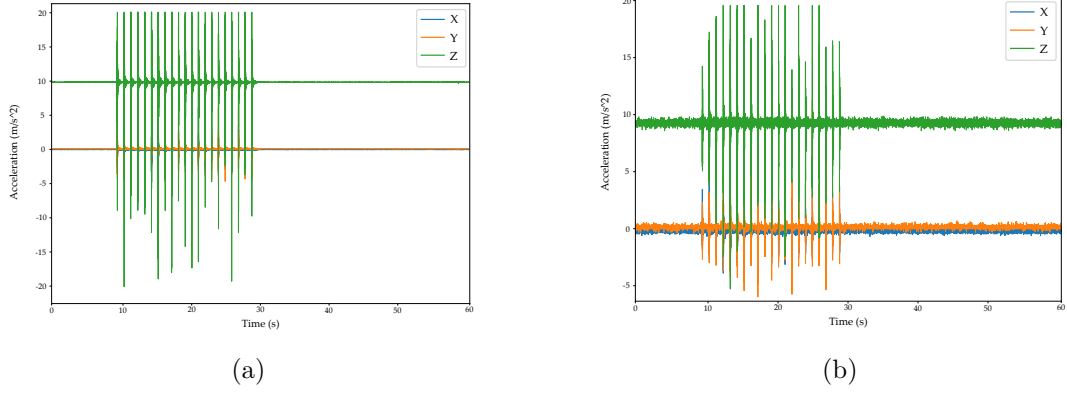


Figure 2.50. Time response comparison for the ADXL355 (a) and ADXL345 (b) accelerometers, on node 5 [149].

As per results from this second test, peaks derived from the periodic stimulus are detected successfully by the accelerometric sensors. In particular, as demonstrated in the first test, the peaks detected from the secondary sensor are immersed in a relatively high environmental noise. Nevertheless, this analysis demonstrates that the backup accelerometer can be successfully employed as a backup solution in the case of a fault by the main sensor.

In the third test, the structure was kept intact for the first few seconds, then it was altered and subjected to periodic stimuli. For the same reasons explained above, in the second test, 60 seconds was chosen as the testing time parameter. Results from this test are reported in Fig. 2.51. The main expectation from this test is an increase in settling time and a higher oscillation frequency in the stimulus response. This expectation is due to the lower inertia that characterizes the altered structure.

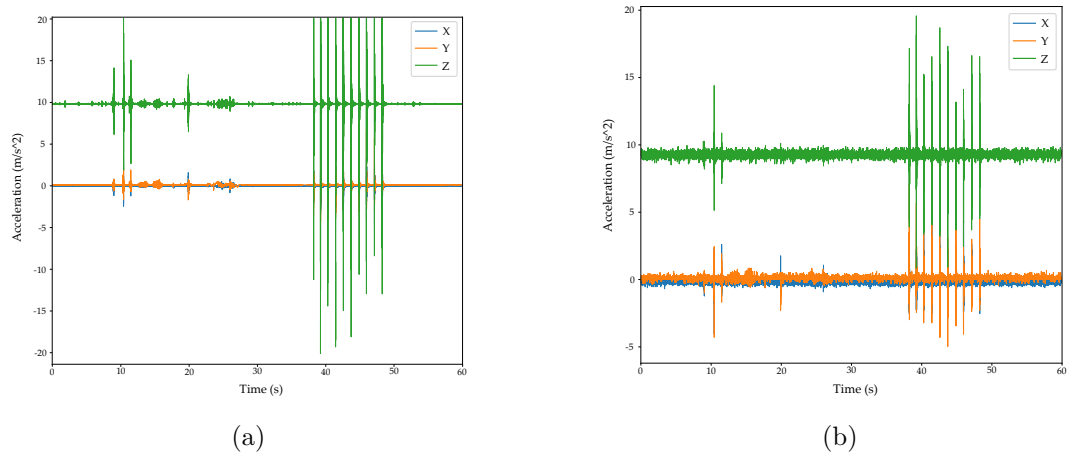


Figure 2.51. Time response comparison for the ADXL355 (a) and ADXL345 (b) accelerometers, on node 5, after structure alteration [149].

Fig. 2.51 shows how, again, the backup sensor can successfully detect the applied stimuli, hence, it can be used as a proper backup in case of failure of the main sensor. By analyzing the spectrograms related to this last test, direct information about the alteration event can be read, as demonstrated in Fig. 2.52. In particular, the selected frequency range for these measurements is 0–250 Hz to distinguish modal

harmonics as clearly as possible; additionally, Fig. 2.52 takes node 5 as reference, and shows the spectrogram related to its most stressed axis, the X-axis.

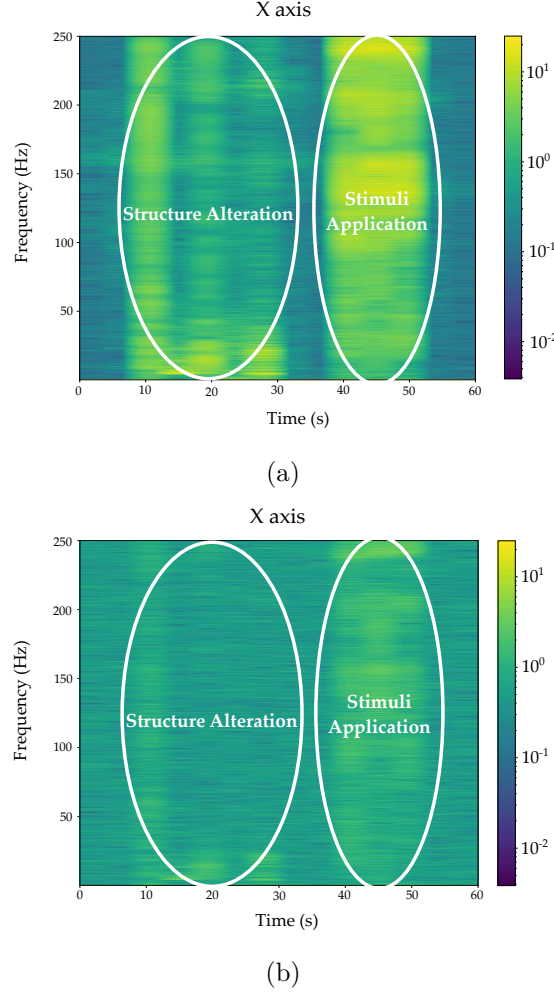


Figure 2.52. X-axis spectrograms of the main (a) and secondary (b) accelerometers from node 5 [149].

The states of the structure before and after the alteration are clearly distinguishable in the sequence: rest, structure alteration, rest, impulsive stimulus, rest. Furthermore, spectrograms offer a unique means of distinguishing the resolution difference between the two on-board accelerometers, as frequency components can be identified more clearly.

As mentioned previously, after the tests, a comparison is presented between the data before and after the structure alteration. Firstly, this comparison is achieved on the spectrograms, as reported in Fig. 2.53. For this test, node 3 was analyzed, since it contained the clearest data. In particular, Fig. 2.53a reports the spectrogram of the structure before its alteration, hence, the first seconds of the third test, while Fig. 2.53b reports the spectrogram sensed by node 3 after structure alteration, hence, in the final seconds of the third test.

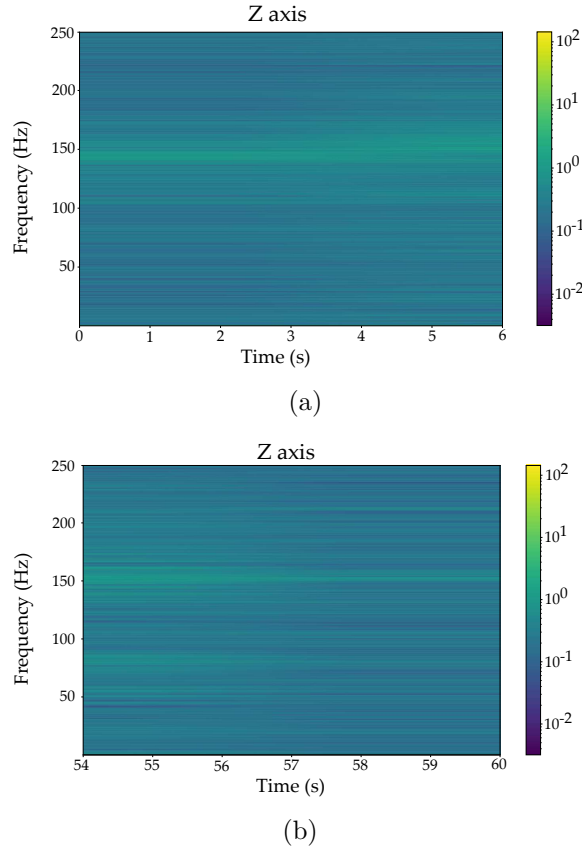


Figure 2.53. Z-axis spectrograms of node 3 before (a) and after (b) structure alteration [149].

Interestingly, differences in both the spectrogram parts can be acknowledged, confirming that structure alteration causes instant modal frequencies change, independently of the applied stimuli. The first difference that confirms the hypotheses drawn at the beginning of the section is the resonant frequency shift, in the altered structure spectrogram, of around 5 Hz. Secondly, the modal frequencies shift is noticeable as well. After the structural alteration, some frequencies, very close to the resonant frequency, shift to higher frequencies, around 225 Hz, while the low-frequency harmonics experience a drift to 60 and 90 Hz, respectively.

Finally, Fig. 2.54 reports a comparison exclusively in the time domain, before and after the structure alteration, for node 5. As a matter of fact, this comparison is achieved by considering data from both the second and third tests.

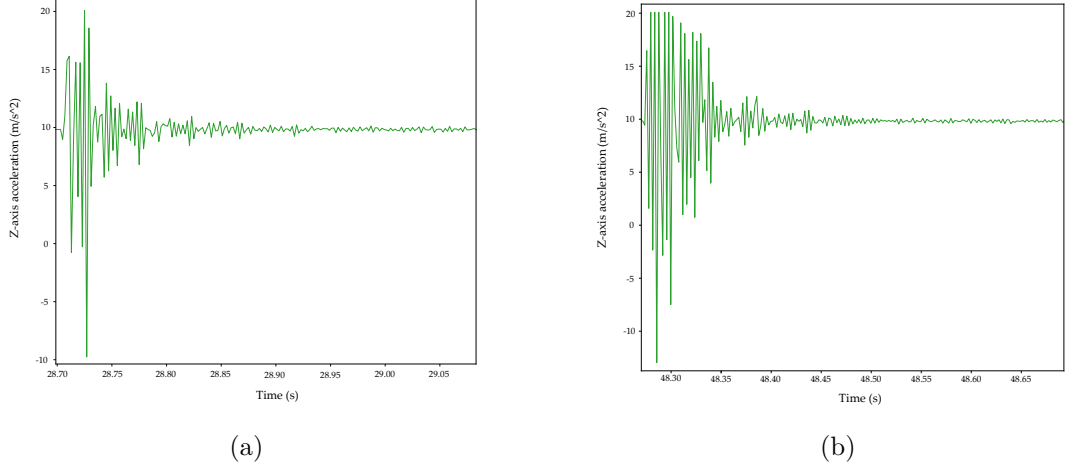


Figure 2.54. Detail on the structure time response before (a) and (b) after the strcuture alteration [149].

A first important difference is the oscillation frequencies of the damping oscillations recorded by node 5. These oscillation frequencies represent the resonant frequency of the structure and, coherently with the spectrograms, the resonant frequency of the non-altered structure equals 145 Hz, while in the altered structure the number equals 152 Hz. Additionally, settling time is different among the two structural statuses; in the non-altered situation, settling time equals 100 milliseconds, while in the altered situation, it becomes three times larger, reaching 300 milliseconds. These results are mainly due to the lower inertia of the altered structure with respect to its whole version, since altering means detaching a box, thus, lighting the box under test. Therefore, lower inertia causes an increase in settling time and a shift of its resonant frequencies to higher values. In this case, since the structure alteration was highly controlled and it did not deform the box under test considerably, the amount of resonant frequency shift is limited.

Chapter 3

Collateral Activities

Although the main body of this thesis is dedicated to the development of hardware-firmware solutions for industrial applications — specifically focusing on long-range wireless communication through LoRa technology — several parallel research activities were pursued during the PhD program. These works, while not directly related to LoRa-based systems, contribute to a broader understanding of electronic design methodologies, sensor interfaces, and advanced circuit architectures. Each activity explores a specific technical challenge, often involving different frequency regimes, fabrication approaches, or modeling strategies; collectively, they reflect the multidisciplinary nature of modern hardware design for industrial environments.

The remainder of this chapter presents a concise overview of these activities, which include:

- additively manufactured circuits and sensors, where 3D printing is investigated as an alternative fabrication approach for rapid prototyping and unconventional geometries;
- a fully differential interface for differential capacitive sensors, developed to improve noise immunity and measurement accuracy in harsh environments;
- the design and simulation of a Second-Generation Voltage Conveyor (VCII), exploring the analog signal-processing architectures that can benefit sensor front-ends and mixed-signal systems;
- a high-voltage ultra-wideband (UWB) circuit model, derived through a Design of Experiments (DoE) approach, aimed at understanding parameter sensitivities and optimizing high-speed/high-voltage behavior.

Although heterogeneous, these activities share a common denominator: they provide additional perspectives and technological insights that complement the main topics on embedded industrial systems described in this thesis. They demonstrate how industrial electronic design often requires integrating knowledge from diverse domains—ranging from advanced materials and additive manufacturing to analog architectures and statistical modeling.

3.1 3D-Printed Sensors and Circuits

Additive Manufacturing (AM) is a widely used technique for producing tools, cases, and mechanical components through material addition rather than removal, unlike subtractive or mixed manufacturing. This approach minimizes waste and is being explored for applications ranging from micron-scale objects [160] to houses [161] and even food [162].

In electronics, AM is emerging as a promising alternative to traditional methods, such as CNC milling and industrial processes, for sensor fabrication, PCB production, and chip manufacturing. Thanks to its reduced time-to-market, 3D-printed sensor applications are rapidly expanding. Mainly, AM is implemented with 3D-printing methods such as SLA, AIP, and FDM, and combinations of these three.

In particular, acfSLA is an AM technique which works by selectively curing a liquid photopolymer resin using a light source, typically a Ultraviolet (UV) laser or a digital light projector, in a layer-by-layer approach, as depicted in Fig. 3.1a. SLA can reach around 25 microns of resolution; hence, it is used for precision printing. In electronics, SLA-printed parts are often made conductive through electroplating, a step that significantly increases fabrication time. Despite this, SLA is widely used in high-frequency applications, enabling the production of waveguides [163] and resonators [164] in the K and Ka bands with promising performance.

Aerosol Inkjet Printing (AIP) is an advanced AM technique that atomizes conductive ink into nano-scale droplets, which are collimated and deposited onto a separately printed substrate, as shown in Fig. 3.1b. Although typically used on planar surfaces, AIP has also been demonstrated on non-planar and flexible substrates [165, 166]. It offers some of the highest resolution in AM, though surface inhomogeneities may affect trace quality [167]. AIP supports a wide range of materials and operates at low temperatures, but requires careful ink rheology control to prevent clogging or overspray, and the deposited traces must undergo thermal or UV curing [168]. Despite these limitations, it remains a promising option for high-resolution, flexible, and miniaturized electronics. It has been applied to circuits [169], sensors [170], self-heating structures [171], and interconnects on rigid [172], porous [173, 174], and curved substrates [175–177].

Fused Deposition Modeling (FDM) is an AM technique standardized by ISO-ASTM 52900 [178], based on the layer-by-layer extrusion of thermoplastic material, as shown in Fig. 3.1c. Although it offers the lowest resolution among the methods discussed, it is fast, cost-effective, and highly flexible in design—particularly for electronic circuits and components. In electronics, Polylactic Acid (PLA) is commonly used as a dielectric substrate, while copper-based Electrifi and carbon-based Protopasta filaments serve as conductive materials. Owing to its growing relevance in this field, FDM applications in electronics are further detailed in Sec. 3.1.1.

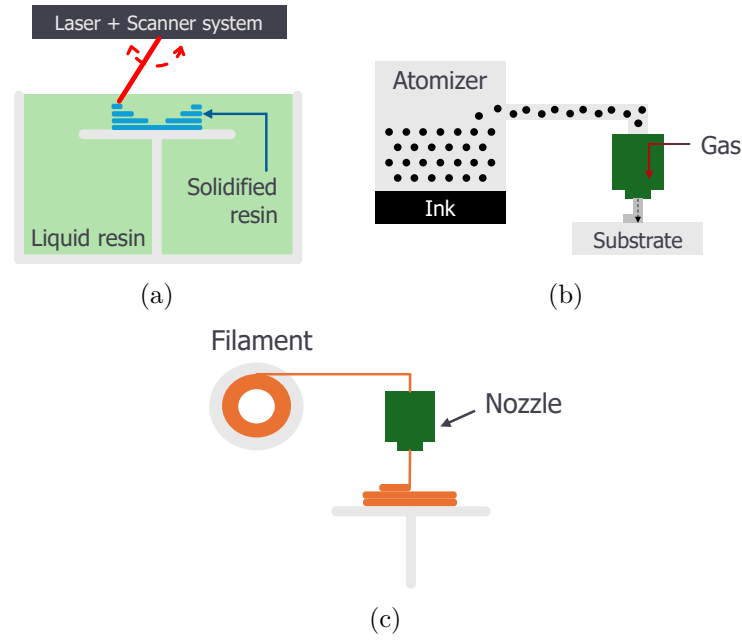


Figure 3.1. 3D printing methods: SLA (a), AIP (b), FDM (c).

3.1.1 Applications of Fused Deposition Modeling in Electronics

This section gives a brief discussion of this method feasibility for its use in electronics applications, high-frequency implementations, and sensors manufacturing. Finally, the equipment used in the FDM applications reported in this thesis, is shown.

Feasibility of FDM Over Electronics

The feasibility of FDM for electronic circuits and devices has been demonstrated in several works [179, 180]. Cho *et al.* electroplated an FDM-printed X-band horn antenna, achieving enhanced gain thanks to an integrated metalens [179]. Church *et al.* analyzed FDM under the Direct Digital Manufacturing (DDM) paradigm [180], highlighting that its extrusion-based process limits conductivity to the printing direction. Using microdispensing, they quantified variations in resistance due to trace orientation and strain.

At high frequencies, FDM has been evaluated as an alternative manufacturing method [181–183]. Most implementations use copper tape for the conductive parts due to its superior performance. Alias *et al.* produced a horn-shaped UWB antenna with PLA, showing stable performance across different conductive tapes [181]. Bozzi *et al.* fabricated Substrate Integrated Waveguides (SIWs) with Acrylonitrile Butadiene Styrene (ABS) substrates and copper tape, obtaining GHz-range filters with promising insertion and return losses [182]. Piekarczyk *et al.* instead 3D-printed the ground plane using Electrifi filament, demonstrating its potential—though resistive—use in microstrip devices [183].

FDM has also been widely used for sensor fabrication thanks to its flexibility and rapid prototyping capabilities. Capacitive buttons [184], batteries [185], and optical sensors [186] have been demonstrated with various material combinations. Flexible filaments enable strain-based devices, such as tri-axial accelerometers printed with carbon-loaded materials [187], bend-sensitive artificial hands [184], and more complex

sensorized gloves with embedded Human-Machine Interface (HMI) systems [188]. Additional strain sensors have been manufactured using Thermoplastic Polyurethane (TPU), copolyester, polyamide, and carbon-based PLA, each exhibiting application-dependent mechanical and electrical characteristics [189, 190].

FDM Equipment Used in this Thesis

In this work, the dielectric materials will be the PLA and the TPU for rigid and flexible structures respectively. For high-frequency applications, PLA has been thoroughly studied as a dielectric [191–193].

The Protopasta [194] and Electrifi [195], mentioned above, are used as conductive filaments.

The E3D ToolChanger, represented in Fig. 3.2, is the chosen 3D printer for these works; it is a middle-way alternative between an industrial and a hobbyist printer.

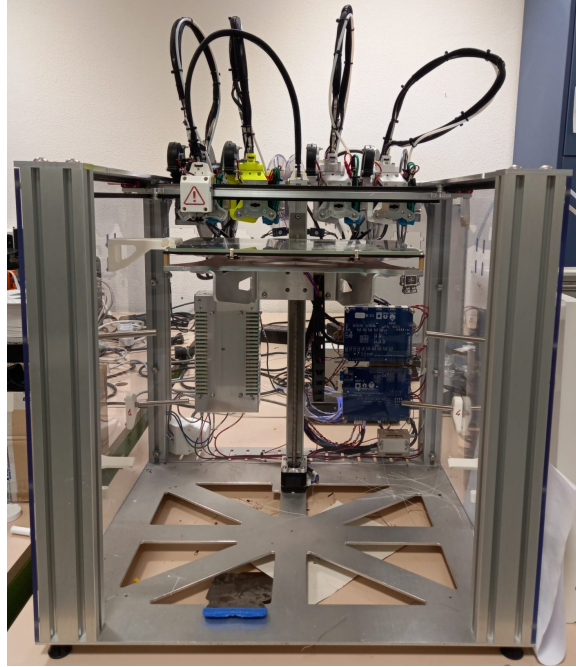


Figure 3.2. E3D ToolChanger 3D printer.

Its working principle is based on the use of a tool changer, which is a printing head switching among four extruders put in a specific docking position. The printer is fully customized in its hardware, through upgrades, and firmware, which is based on the Duet interface [196].

3.1.2 3D-printed Sensors

Activity with 3D printed sensors focuses on two sensors: a capacitive tri-axial accelerometer and a bend sensor. Accordingly with the thesis topic, microcontroller-oriented firmware was written and implemented for the bend sensor.

3D-printed accelerometers

The tri-axial capacitive accelerometer is based on an SLA-made structure implemented by Zega *et al.* [197].

The reference accelerometer is based on a central structure that supports six proof masses; each pair of moving masses refers to a separate axis. The central structure is connected to an external cubic-shaped frame through several folded springs. In a certain phase of the manufacture, a conductive bar is placed on each face of the cubic structure, to obtain pairs of differential capacitors on each axis, as reported in Fig. 3.3.

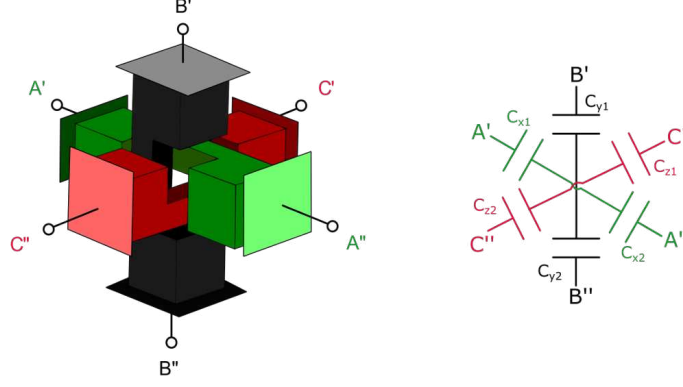


Figure 3.3. Working principle of the analyzed accelerometric sensor [198].

Primarily, the target of this work was to manufacture an FDM version of the reported sensor structure due to the easier process, lower cost, and lower time to market, with respect to the implementation by Zega *et al.* [197].

Firstly, the structure was designed on the Autodesk Fusion 360 software. Secondly, the design was meshed and sliced on the OrcaSlicer software [199], chosen due to its versatility. In this sensor, the conductive parts are manufactured with the Protopasta filament [194], the semi-rigid structure is printed with PLA, and supports are made with Polyvinyl Alcohol (PVA) — water-soluble filament — due to their position and, consequently, the effort required for their mechanical removal. Fig. 3.4 reports the 3D rendering of the described structure.

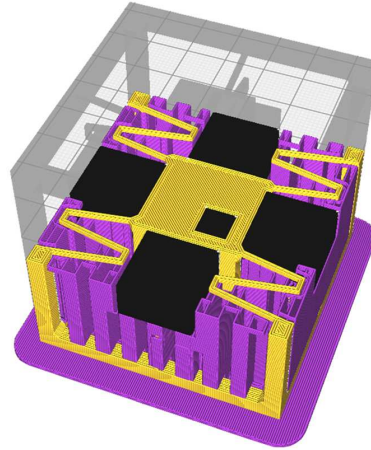


Figure 3.4. Rendering of the accelerometric sensor in the slicing phase [198].

The printing process takes around seven hours due to the mechanical complexity of the structure. After the printing, six copper bars are added to each face of the cubic structure. Finally, wires are soldered and glued to the conductive parts of the

structure to enable sensor reading. The sensor has then been reprinted by varying the internal infill to provide further characterization. The selected values for the infill are:

1. 100% infill
2. 50% infill
3. 10% infill

Sensors have been tested by using a shaker to provide different stimulation frequencies with both square and sinusoidal wave patterns. Additionally, to read its values properly, the sensor was equipped with a fully differential interface, introduced in Sec. 3.2.

As reported in Fig. 3.5, in each test session, the sensor has been stuck to the shaker with double-sided tape.

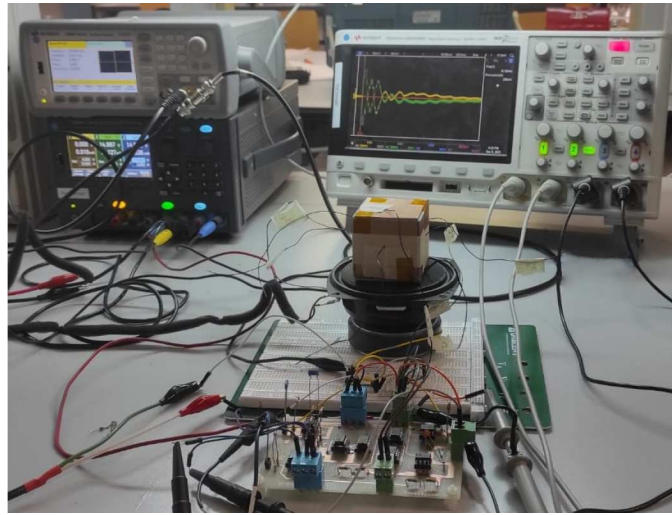


Figure 3.5. Test setup for the proposed FDM accelerometric sensor.

The three sensors have been subjected to three main tests:

1. the step response, performed with a square-wave-related movement of the shaker, so that the accelerometer is subjected to abrupt state changes;
2. the frequency response, obtained through a shaker sinusoidal movement, where frequency is varied to construct the spectrum;
3. the input-output (I/O) characterization, where a sinusoidal signal is applied to the sensor and, according to the g-value the output voltage is read on the interface;

Fig. 3.6 reports results from the step response of the three sensors.

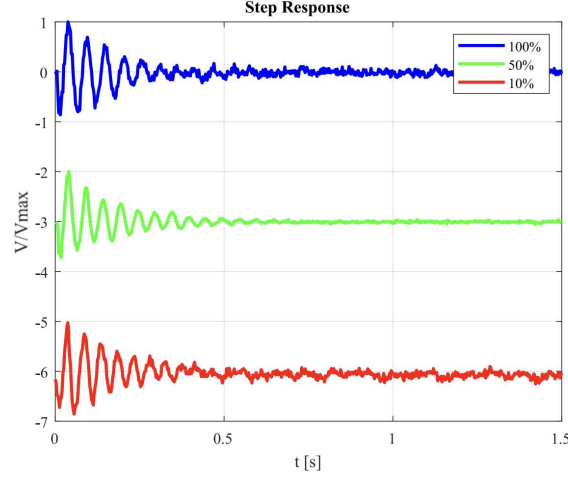


Figure 3.6. Step responses from the three FDM accelerometric sensors [198].

The normalized step responses exhibit two main differences among the sensors. First, the release time varies with infill density: the 100% infilled sensor responds more slowly due to its greater mass, while the 10% infilled sensor shows the fastest release. Second, the response-sinewave frequency — corresponding to the resonant frequency of the structure — increases as the sensor becomes lighter. Consequently, the fully infilled sensor has the lowest resonant frequency, and the 10% infilled sensor the highest.

Furthermore, Fig. 3.7, reports results related to the frequency domain.

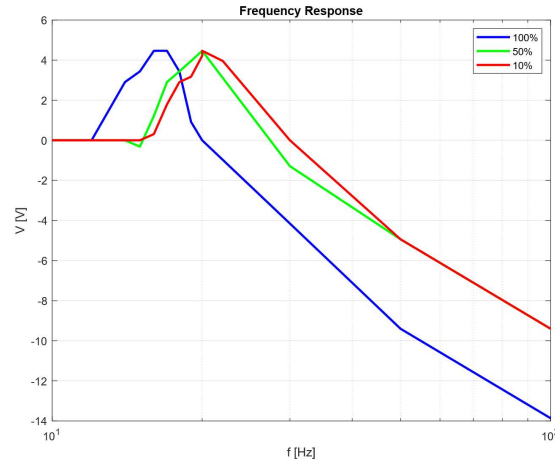


Figure 3.7. Frequency responses from the three FDM accelerometric sensors [198].

The results confirm the correlation between infill percentage and resonant frequency. All sensors exhibit a second-order system behavior, consistent with the spring-mass characteristics of accelerometers. The resonant frequencies measured were 17.7, 20.03, and 20.08 Hz for the 100%, 50%, and 10% infill sensors, respectively. These values are slightly influenced by the additional springiness introduced by the double-sided tape used for mounting.

Finally, Fig. 3.8 reports the I/O characteristics of the three sensors, neglecting the gravitational acceleration.

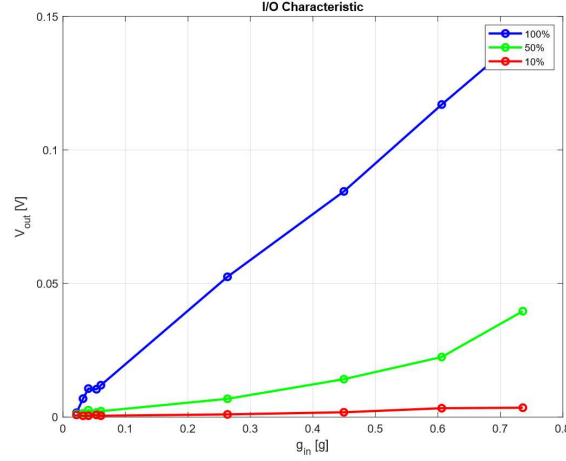


Figure 3.8. Input-output characteristics from the three FDM accelerometric sensors [198].

As shown in Fig. 3.8, sensor sensitivity increases with proof mass infill. Heavier masses exhibit a wider displacement under stimulation, which translates into larger variations in the shield distance of the differential capacitor pairs, resulting in higher output values. Conversely, lighter masses have a more limited movement range, yielding lower capacitance changes and reduced sensitivity. Linearity also improves with infill: the 100%-infilled sensor shows the best linearity due to its greater mechanical inertia, which stabilizes its motion and preserves the proportionality between input and output values.

These results are numerated and summarized in Tab. 3.1.

Parameter	10% infill	50% infill	100% infill
Sensitivity	0.005 V/g	0.045 V/g	0.2009 V/g
Resolution	100 mg	80 mg	20 mg
f_n	20.086 Hz	20.03 Hz	17.761 Hz
BW	16 Hz	15 Hz	10 Hz
Q	5.398	9.042	8.628
$lin_{err,max}$	26.6 %	19.4 %	4.8 %

Table 3.1. Summary performance table for the 3D-printed FDM accelerometric sensor.

This research has been published on the IEEE Access Journal [198].

3D-printed strain gauges

A second work on additive-manufactured sensors involves a 3D-printed resistive strain gauge. The purpose of the work is to design, manufacture, and characterize a 3D-printed strain gauge for its interfacing with a microcontroller. Originally, the sensor was conceived for healthcare use, e.g., a smart glove; however, these sensors can be used in a plethora of applications, such as dendrometry and a robotic gripping unit. The sensor, designed with Autodesk Fusion 360, has a multilayer structure composed of:

- TPU, used due to its flexibility, as the base;

- conductive traces in Electrifi material;
- TPU to cover the conductive traces;

For test-related reasons, two different shapes for the conductive traces are studied. Firstly, a wavy shape, reported in Fig. 3.9a and, secondly, a straight shape, reported in Fig. 3.9b.

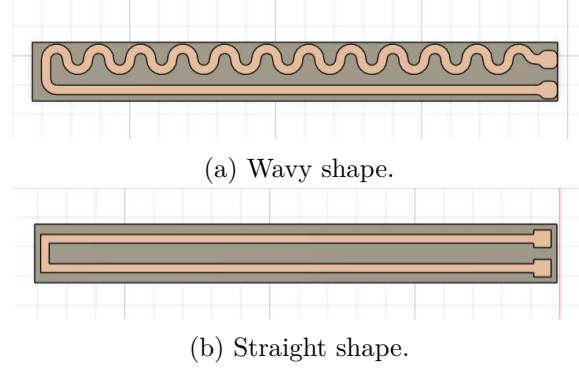


Figure 3.9. Conductive patterns used in the analysed strain gauge sensors.

This structure occupies a reduced area (90×10 millimeters), similar to other commercial sensors, and it guarantees mechanical robustness. In particular, the risk of conductive trace break is managed through the amount of TPU inserted under and above it. After the printing process, reported in Fig. 3.10a, sensors have been electrically interfaced by gluing a couple of wires on each sensor. Wires were glued using the MG-Chemicals 8331-D silver-based conductive glue. Sensors, after the post-printing setup, are reported in Fig. 3.10b.

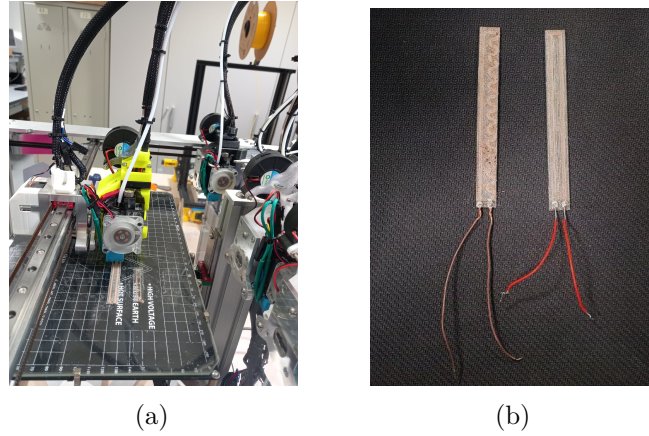


Figure 3.10. Strain gauges during (a) and after (b) the printing process [200].

After the sensor structure was set up, they were characterized. The characterization is based on resistance measurements after bending the sensors on several curved samples with different radii, as reported in Fig. 3.11.



Figure 3.11. Curved samples for the strain gauge characterization.

In this preliminary stage, the circuit interface is a simple voltage divider. As reported in Fig. 3.12, these two sensor structures respond very differently to the same bending radius; the radius reported in the image is 20 mm.

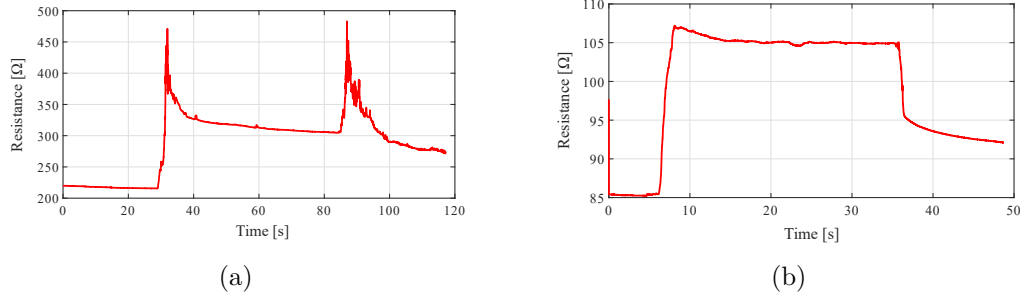


Figure 3.12. Resistance measurement for the wavy (a) and straight (b) sensors with 20 mm radius sample. [200].

In particular, the wavy sensor produces resistance peaks when its bending state changes. An exponential discharge of the resistance value follows these peaks. On the other hand, the straight sensor has a more stable trend when bent. The wavy sensor behavior is explained by the presence of non-bent segments into the pattern. Consequently, the mechanical tension that the sensor experiences during the bending is local. As a result, as some sensor spots are not bent, electrons experience a change in resistance only during the very first bending/unbending stages. After these first phases, they quickly adapt to the bent structure. This effect is amplified by the rounded shape of the waves. On the other hand, this phenomenon is not observed in the straight sensor, as the entire conductive trace is uniformly bent; hence, electrons will adapt to the new resistance much more slowly. From these observations, the straight sensor was chosen as the most suitable due to its stability during short bends. Consequently, the I/O characteristic has been extracted exclusively from the straight sensor. In this regard, the chosen sensor has been reprinted in three equal samples, and their I/O characteristics have been measured and reported in Fig. 3.13.

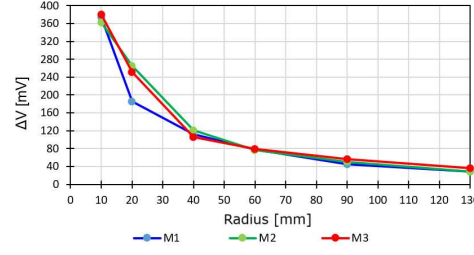


Figure 3.13. I/O characteristic of the straight FDM bend sensor. [200]

As shown in Fig. 3.13, the sensitivity has an inverse trend with respect to the radius: this means that the more the sensor is bent, the more it is sensitive. The average of the three sensor curves and the application of polynomial regression result in a fifth-order equation; the third, fourth, and fifth-order terms have relatively low influence, as reported below.

$$\Delta V = -9 \cdot 10^{-8} \cdot r^5 + 4 \cdot 10^{-5} \cdot r^4 - 0.0074 \cdot r^3 + 0.6466 \cdot r^2 - 28.61 \cdot r + 600.59 \quad (3.1)$$

In particular, Tab. 3.2 reports the error between this mathematical model and the actual measurements on the average curve.

R10	R20	R40	R60	R90	R130
5%	22%	4%	0.7%	3.2%	2%

Table 3.2. Percentage error of the mathematical regressed model with respect to the strain gauge I/O average characteristic.

Finally, the sensor has been shaped into a smart glove. In particular, firmware was written to capture and save the resistance change. For this purpose, the sensor was reprinted five times with hand finger-related lengths. The five sensors have been attached to a textile glove through Velcro®. Additionally, the interface circuit has been redesigned by including five high-performance Complementary Metal-Oxide Semiconductor (CMOS) operational amplifiers. For this purpose, three OPA2364, acting as output-stage buffers in the corresponding sensor PCB, have been chosen. The final PCB is reported in Fig. 3.14

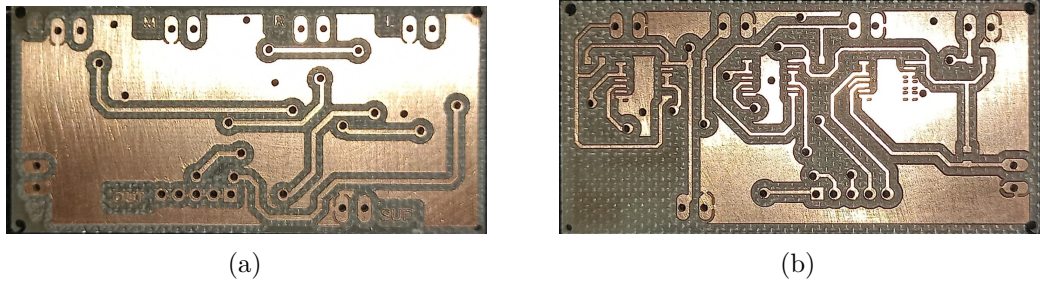


Figure 3.14. Top (a) and bottom (b) sides of the strain gauge circuit interface. [200].

A base was manufactured for the above-represented PCB, resulting in the final smart glove in Fig. 3.15.



Figure 3.15. Smart glove with bending sensors.

The cables from the smart glove were connected to an Arduino UNO board, whose firmware flow diagram, reported below (Fig. 3.16), performs the “Analog Read” function from each sensor to then arrange and send data through the serial port.

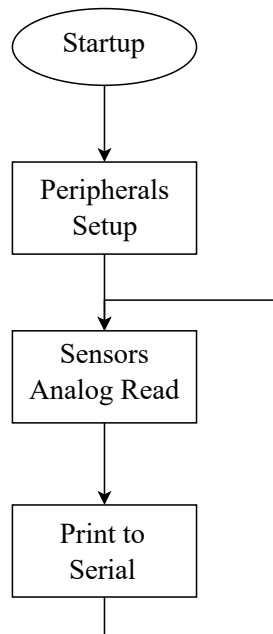


Figure 3.16. Flow diagram related to the smart glove firmware.

As a result, with the addition of this firmware, it has been possible to visualize the hand movements on the "Serial Plot" data visualizer tool. As reported in Fig. 3.17, the finger movements are clearly shown; in particular, a flexion-deflection of the thumb finger, and a deflection of the index finger, are distinguishable.

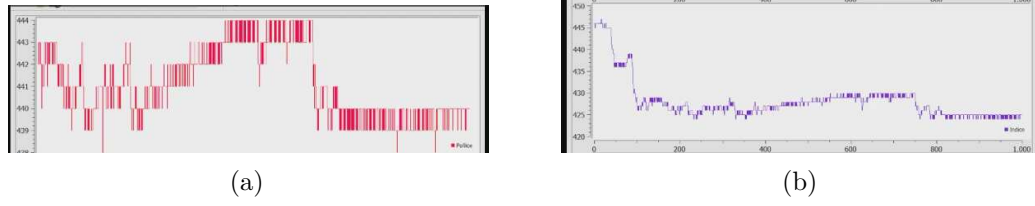


Figure 3.17. Movements of the thumb (a) and index (b) finger detected from the serial visualizer. [200].

3.1.3 3D-printed Circuits

This activity investigates second-order active and passive RC filters with fully embedded RC components. A preliminary analysis of material coefficients is required, specifically the resistivity ρ of the carbon-based Protopasta conductive filament and the relative dielectric constant ϵ_r of PLA. Based on these parameters, the circuits are fabricated and subsequently compared with LTSpice simulations that account for potential parasitic effects.

The Protopasta material resistivity ρ has been characterized by printing two types of blocks

- trace-related, having 5 mm height and 3 mm width, shown in Fig. 3.18a;
- resistance-related, having 0.4 mm height and 0.5 mm width, shown in Fig. 3.18b

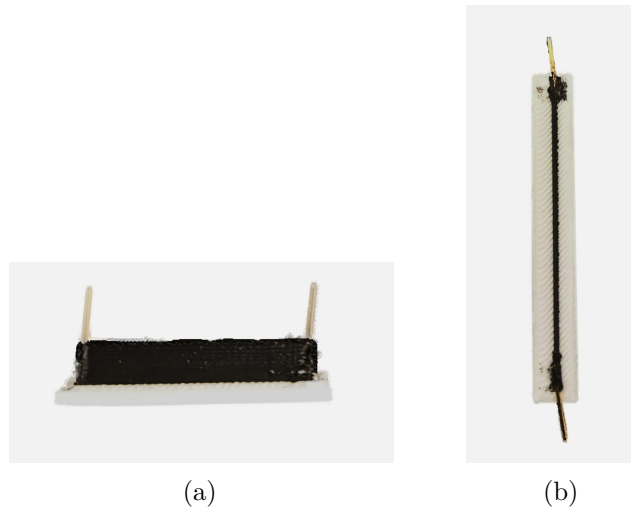


Figure 3.18. Protopasta-made samples for trace (a) and resistance (b) characterization.

As clear from Fig. 3.18, electrical contacts were melted inside the conductive blocks. The use of this method decreased the effective length of resistance block, while not significantly affecting the trace samples length. Resistivity measurements, performed with the Analog Discovery 2 instrument, showed the following results:

1. an average resistivity of $160 \Omega \cdot mm$ with 16% tolerance, for the traces;
2. an average resistivity of $65 \Omega \cdot mm$ with 6% tolerance, for the resistance paths.

Resistivity grows as the sample height — and thus the number of printed layers — increases, mainly due to micro air gaps created by imperfect interlayer adhesion. These gaps also raise the variability of resistivity, so thinner samples show more consistent values. As a result, thicker printed conductors exhibit higher resistivity and greater tolerance.

Given the extensive use of resistors in the circuits, the resistor resistivity ρ_{res} was further validated using samples of different lengths. Results from this test, for resistors, are shown in Tab. 3.3: R_{th} is the theoretical value calculated through Ohm's law, while R_{meas} represents the effective sample value.

N	Length [mm]	R_{th} [$k\Omega$]	R_{meas} [$k\Omega$]
1	10	4.47	4.38
2	21.25	5.52	9.3
3	32.5	12.03	14.22
4	43.75	17	19.14
5	55	26.1	24.06
6	66.25	29.1	28.98
7	77.5	36	33.91
8	88.75	49.4	38.83
9	100	50.5	43.75

Table 3.3. Values in the resistances counter-test.

As a result, the maximum relative error equals 10.84%.

The PLA filament was characterized using planar capacitors (1.5 mm thick, 600 mm² area), yielding an absolute permittivity of $2.6 \cdot 10^{-11}$ F/m and a relative permittivity of about 2.9. To verify this estimate, additional planar capacitors with different shield areas were fabricated (Fig. 3.19). Their theoretical capacitances, calculated from the known dimensions, were compared with the values measured on the printed samples.

The results, presented in Fig. 3.19, demonstrate a high degree of consistency again, with a maximum relative error of 1.55%.

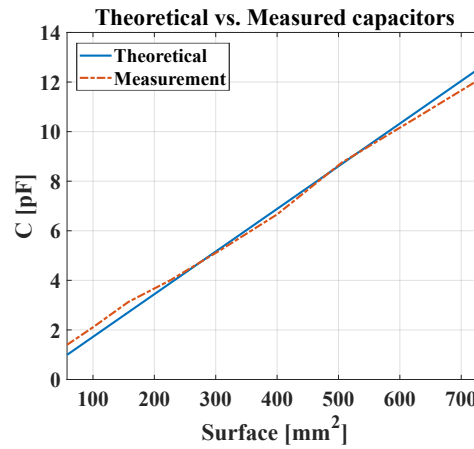


Figure 3.19. Comparison between the theoretical and measured capacitance of predefined capacitor samples [201].

For active filters designs, SPICE simulations were deployed to define the electrical

topology preliminarily and to determine the RC component values based on the required cut-off frequency and gain specifications. Consequently, the active filters adopted a Sallen-Key configuration, as shown in Fig. 3.20.

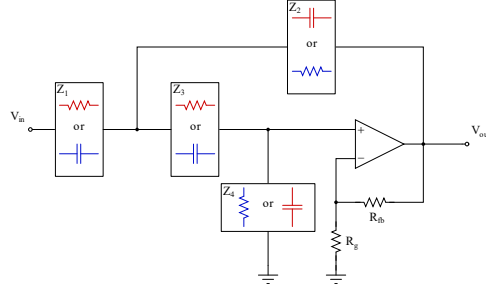


Figure 3.20. Schematics of the 3D-printed active filters.

In particular, in Fig. 3.20, red and blue components highlight the low-pass and high-pass structures respectively. As a matter of fact, this circuit is a low-pass or high-pass architecture combined with a non-inverting amplifier. For this reason, the (3.2) and the (3.3) express the cut-off frequency and the gain, respectively, of the analysed circuit structure.

$$f_{co} = \frac{1}{2\pi RC} \quad (3.2)$$

$$G_{DC,sk} = 1 + \frac{R_{fb}}{R_g} \quad (3.3)$$

The active circuit used as amplifier in these configurations is the LF411 integrated circuit.

Next, filters were designed on Autodesk Fusion 360. The sketch obtained from this step is then passed to the 3D printer slicer as an .stl file; the slicer, consequently, processes the design to generate ToolChanger-compatible g-code. Fig. 3.21 shows the printed version of the low-pass active filter.

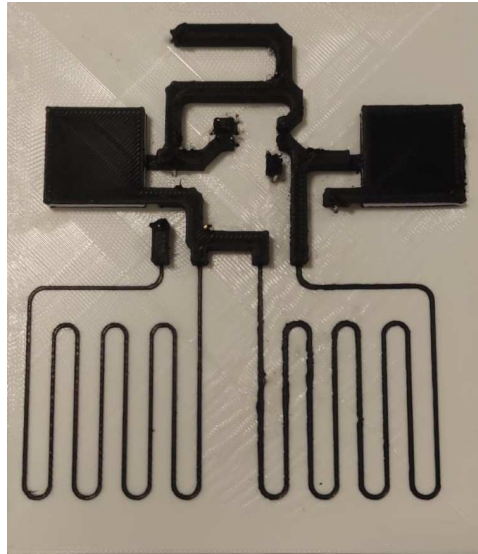


Figure 3.21. Printed active low-pass filter [201].

In this regard, several choices were necessary to minimize any parasitic components and ease the circuit shape, in particular:

1. a “wavy line” shape for resistors was employed to ensure high resistances and minimum area;
2. capacitors were printed vertically to avoid short circuits due to accidental conductive material deposition;
3. significant dimensions have been chosen for both the RC components to minimize the FDM-induced uncertainty.

Frequency sweeps have been performed for both high-pass and low-pass filter typologies. In Fig. 3.22, the ideal and measured trends for the low-pass filter are compared.

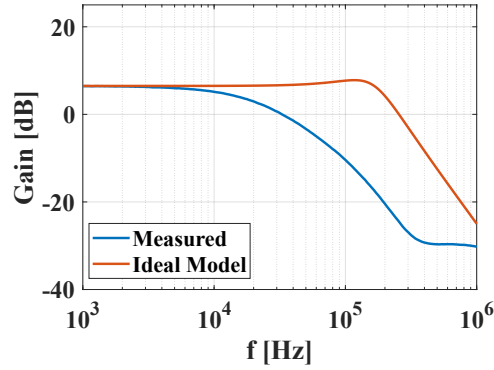


Figure 3.22. Results from the 3D-printed active low-pass filters with Sallen-Key topology. [202].

The measured DC gain (6.45) closely matches the theoretical value (6.5), with a deviation of only 0.7%. In contrast, the cut-off frequency drops from the ideal 209 kHz to 18 kHz, displaying a large deviation of approximately 91.4%. This mismatch indicates that parasitic effects — particularly inductance from long printed resistive paths and supply suppression caused by the resistive material — strongly affect the high-frequency behavior while leaving the low-frequency gain mostly unchanged. As a result, the measured filters exhibit a Bessel-like response without overshoot instead of the expected Chebyshev characteristic.

Similarly, results from the high-pass filter frequency sweep are reported in Fig. 3.23.

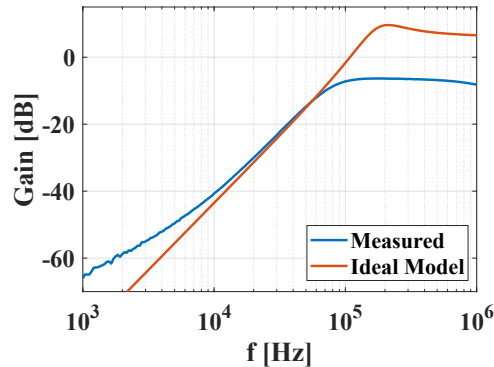


Figure 3.23. Results from the 3D-printed active high-pass filters with Sallen-Key topology. [202].

The high-pass filters diverge from the ideal behaviour due to parasitic effects and the high resistivity of the printed conductors. These issues cause gain drift at high frequencies, low-frequency noise, and slope distortion from capacitor mismatch. The response becomes Butterworth-like instead of Chebyshev, and the cutoff frequency drops from 125 kHz to 73.7 kHz ($\sim 41\%$ deviation). Although the gain magnitude remains fairly close to the ideal value, phase inversion and reduced bandwidth confirm a strong parasitic influence.

From the results of both the configurations,

- The gain difference between the low-pass and high-pass filters arises mainly from dominant resistive effects combined with parasitic capacitances, which strongly influence high-frequency behaviour.
- The cut-off frequency shows a larger relative error in the low-pass case because its grounded capacitors introduce a parasitic resistor–capacitor–ground path in protopasta, creating a zero-compensation effect that shifts the cut-off.

Low-pass filters were further analysed using a DoE-based model (Sec. 3.4) to predict cut-off frequency error as a function of RC component values. Resistors were swept from 80 to 160 k Ω and capacitors from 4.7 to 9.3 pF, with both ranges encoded as -1 and $+1$ for standardized DoE processing. Physical circuits were fabricated using all RC combinations, while parallel LTSpice simulations—including the previously identified parasitic effects—were performed to support the analysis. The combination of simulations and measurements are reported in Fig. 3.24.

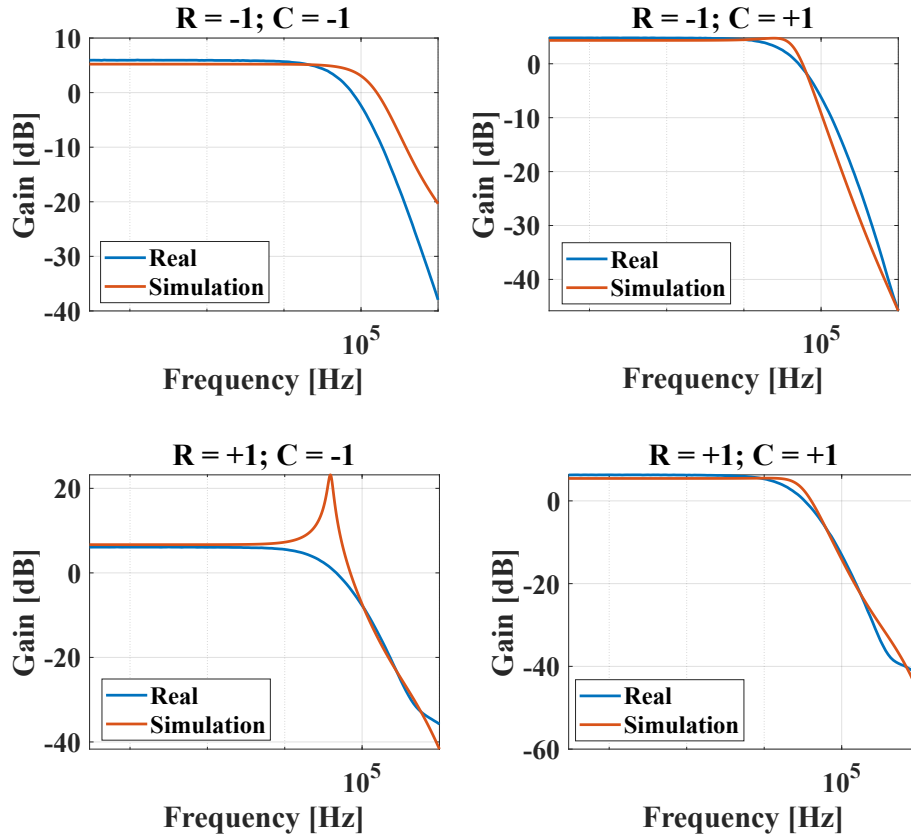


Figure 3.24. Results from frequency sweeps on the low-pass filters used for the extraction of the cut-off frequency error DoE-based model [201].

As a result, the following frequency errors were measured between theoretical and measured samples:

- 0.558 for $R=80 \text{ k}\Omega$ and $C=4.7 \text{ pF}$;
- 0.068 for $R=80 \text{ k}\Omega$ and $C=9.3 \text{ pF}$;
- 0.335 for $R=160 \text{ k}\Omega$ and $C=4.7 \text{ pF}$;
- 0.237 for $R=160 \text{ k}\Omega$ and $C=9.3 \text{ pF}$;

Imperfect modelling of parasitic effects may be one of the reasons behind these error values. The model extracted from these errors is reported in the (3.4)

$$err_f = 0.299 - 0.014 \cdot R - 0.147 \cdot C + 0.098 \cdot R \cdot C \quad (3.4)$$

This model exhibits a mean relative error of 0.299 for the cut-off frequency, highlighting the dominant role of the capacitor. Specifically, a full-range variation of the capacitance produces a 50% change in the error, whereas an equivalent variation in the resistance results in only a 5% change. Consequently, the reliability of these filters — interpreted as the cut-off frequency error minimization — is directly linked to the capacitor area, which was the only parameter modified in this study: larger areas lead to lower errors.

This work, and its model-based development, were published under conference papers [201, 202].

3.2 Circuits for Sensor Interfaces

In modern industrial systems, sensors play a pivotal role in automation, monitoring, and control by acquiring real-time data. However, their output signals often exhibit low amplitude or high noise sensitivity, requiring adaptation before digital processing. Interface circuits provide this adaptation by bridging the physical and digital domains through elements such as amplifiers, filters, and ADCs. Within a hardware-firmware architecture, they ensure reliable communication between the analogue front-end and the digital back-end. In industrial environments, this task becomes particularly challenging due to noise, electromagnetic interference, and long transmission paths; hence, proper interface design is essential to maintain signal integrity.

In this regard, sensor interfaces can be either addressed as single-ended or differential. As shown in Fig. 3.25, single-ended interfaces convert the sensor output to a voltage carried on a single wire or pattern with respect to ground.

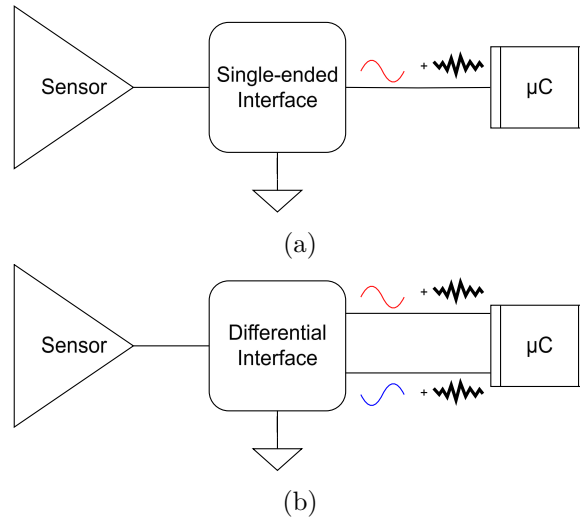


Figure 3.25. Distinction between single-ended (a) and differential (b) interface circuits.

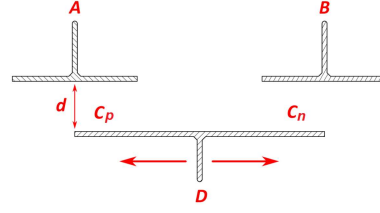
This approach is simple and allows direct measurement, but any noise on the line is unfiltered. Differential interfaces, in contrast, carry the voltage across a pair of symmetric wires and measure the difference, enabling effective cancellation of common-mode noise. Due to this advantage and the need to interface differential capacitive sensors, this thesis focuses on the design and fabrication of a differential interface.

Sensors, as well, have single and differential forms. A single sensor is manufactured and used as a single component in an electronic circuit; the differential sensor, on the other hand, is related to a pair of sensors, varying their value complementarily. Single sensors are simple, inexpensive, and require minimal readout circuitry, making them suitable for compact designs. However, they are more susceptible to noise, resulting in lower accuracy in harsh environments. Differential sensors, in contrast, offer higher noise immunity through common-mode rejection, which improves accuracy, stability, and signal quality. The trade-off is increased wiring complexity and size. Accordingly, differential sensors are preferred for small-signal detection [203]; in this work, a capacitive differential sensor is employed.

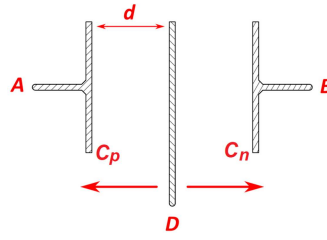
Considering the mathematical model of a capacitor, reported in (3.5), differential capacitive sensors are classified as linear or hyperbolic.

$$C = \varepsilon \frac{S}{d} \quad (3.5)$$

In this context, ε denotes the dielectric permittivity, S the shield surface, and d their separation. Linear capacitors, shown in Fig. 3.26a, exhibit a capacitance that varies linearly with the displacement of a central common shield, which alters the effective shield area. Hyperbolic capacitors, illustrated in Fig. 3.26b, instead display a hyperbolic capacitance variation, as the central shield movement changes the inter-shield distance.



(a) Linear differential capacitive sensors.



(b) Hyperbolic differential capacitive sensors.

Figure 3.26. Working principles of the differential capacitive sensors types.

When differential capacitive sensors are used, an input x parameter is defined to join both the C_p and C_n capacitors into one value.

$$x = \frac{C_p - C_n}{C_p + C_n} \quad (3.6)$$

While the C_p and C_n components are merged into the x parameter, their number varies around a common capacitance value, known as the baseline capacitance C_{bl} . As a result, C_p and C_n are expressed with respect to the x parameter and the C_{bl} capacitor, as reported in the (3.7).

$$C_{p,n} = \begin{cases} C_{bl} \cdot (1 \pm x) & \rightarrow \text{Linear Sensor} \\ \frac{C_{bl}}{1 \mp x} & \rightarrow \text{Hyperbolic Sensor} \end{cases} \quad (3.7)$$

Interestingly, the linearity and hyperbolicity of the sensors can be directly found in the $C_{p,n}$ - x parameter relationships in the related equations.

In this thesis, the design, characterization, and measurement of a fully differential interface for differential capacitive sensors are discussed. The proposed solution, shown in Fig. 3.27, employs an auto-balanced De Sauty bridge topology [204], comprising a differential capacitive sensor pair and a pair of VCCs. The VCCs

serves to track the sensor pair values via a negative feedback applied to their control voltages.

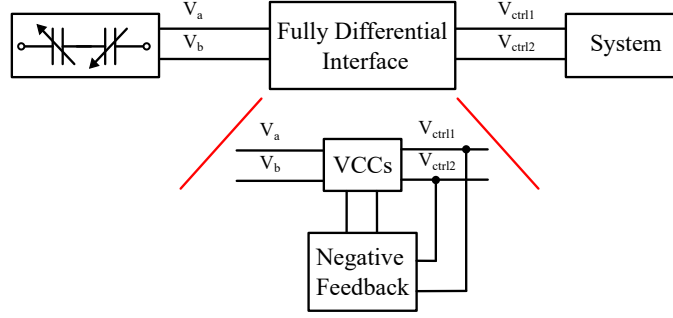


Figure 3.27. Graphical high-level representation of the proposed sensor interface [205].

As a matter of fact, the circuit is used as follows:

1. the configuration is defined and described;
2. the mathematical model is extracted through circuit analysis, with consideration of parasitic effects;
3. the manufacturing process is performed;
4. measurements and comparison with results from the ideal model are accomplished.

3.2.1 Circuit Description

As shown in Fig. 3.28, the fully capacitive bridge is singularly supplied with a sinusoidal signal v_s . The voltages at the bridge output nodes, v_a and v_b , constitute the negative feedback inputs, with V_A and V_B amplitudes, respectively.

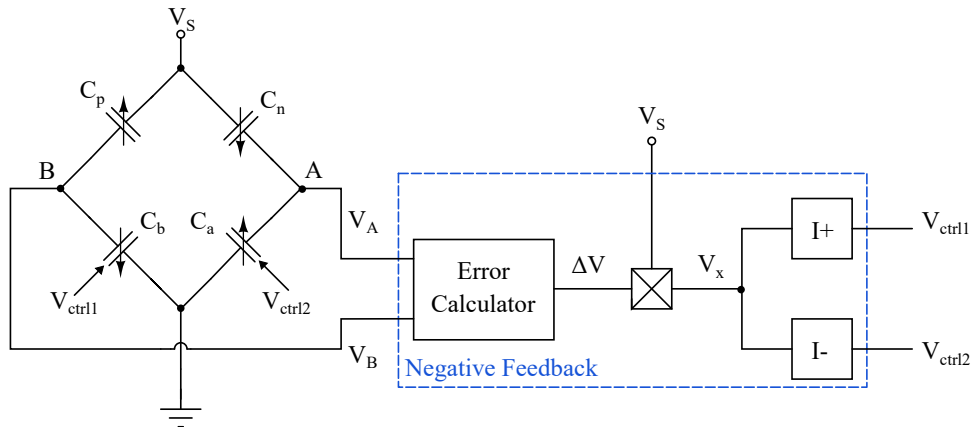


Figure 3.28. Schematic of the proposed interface circuit [205].

As a result, the mathematical relationship between the V_A and V_B voltages is expressed by the (3.8).

$$\begin{cases} V_A = V_S \cdot \frac{C_n}{C_a + C_p} \\ V_B = V_S \cdot \frac{C_p}{C_b + C_p} \end{cases} \quad (3.8)$$

As expressed by the (3.8), the VCCs values C_a, C_b , tend to the sensors capacitors values C_p, C_n , in steady state conditions. Additionally, in this specific application, the supply sinusoidal signal v_s has amplitude V_S and a 10 kHz frequency. The negative feedback is based on three main operations:

1. error calculation, implemented through an INA configuration with two internal stages;
2. error signal modulation through multiplication with the sinusoidal carrier v_s ;
3. demodulation of the modulated error signal through integration;

Firstly, the INA configuration block reported in fig. 3.29 is composed of two stages.

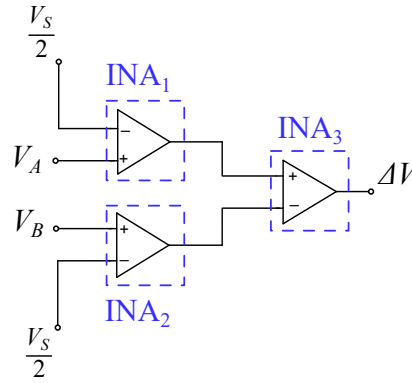


Figure 3.29. Internal structure of the “Error Calculator” block [205].

The sinusoidal signals v_A and v_B are first centered around known values by the first stage, which subtracts them by using INA_1 and INA_2 . These values have an amplitude of $V_S/2$. The second stage subtracts the outputs from the first stage through INA_3 , to obtain the Δv error voltage. As a result, the error voltage signal amplitude ΔV is expressed by the (3.9).

$$\Delta V = V_A - V_B = \left(\frac{C_n}{C_a + C_n} - \frac{C_p}{C_b + C_p} \right) \cdot V_S = H_{\text{bridge}} \cdot V_S \quad (3.9)$$

Due to the sinusoidal nature of the Δv signal, its spectrum content is characterized by a single spectral component at the carrier frequency, 10 kHz. Since the control voltages V_{ctrl1} and V_{ctrl2} acting on the couple of VCCs are DC, it is necessary to extract a zero-frequency component from the error Δv voltage. For this reason, the Δv signal is modulated through a multiplication with the v_s carrier sinusoidal signal, at the same frequency, obtaining the v_x signal. This operation, through the Werner rule and as reported in the (3.10), produces a signal with two harmonic components: a first at zero frequency, and a second at twice the signal frequency.

$$\begin{aligned} v_x(t) &= K_V \cdot \Delta v \cdot v_s(t) = K_V \cdot H_{\text{bridge}} \cdot V_S^2 \cdot \sin^2(\omega t) \\ &= K_V \cdot H_{\text{bridge}} \cdot V_S^2 [1 - \cos(2\omega t)] \end{aligned} \quad (3.10)$$

Specifically, the (3.10) contains the unitary K_V term, which provides the correct dimension into the equation. Once the error-related DC component was obtained,

the third and last step is to demodulate the v_x signal by filtering its zero-frequency component. This is performed by using an inverting and a non-inverting integrator, on the same v_x signal, to

- provide a fully integral regulation to the negative feedback, ensuring zero-convergence for the error signal;
- exploiting the frequency spectrum of the integrator, acting as a low-pass filter.

As a result, the control voltages V_{ctrl1} and V_{ctrl2} are defined in the (3.11).

$$\begin{cases} V_{ctrl1} = \int v_x(t) dt \\ V_{ctrl2} = -\int v_x(t) dt \end{cases} \quad (3.11)$$

For the sake of clarity, Fig. 3.30 provides a graphical representation of the modulation-demodulation procedure reported in this work.

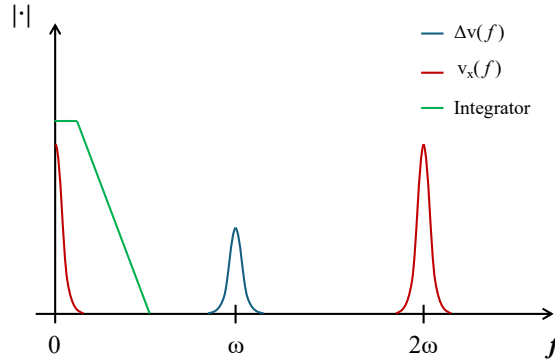


Figure 3.30. Representation of the modulation-demodulation procedure used in this work.

Finally, the V_{ctrl1} and V_{ctrl2} control voltages are connected to the VCCs. To implement the VCCs, the AD633 [206] integrated circuit was chosen. For this purpose, both the AD633 circuits are connected in Zhong configuration [207].

3.2.2 Mathematical Model

After circuit definition, its mathematical model has to be extracted. Parasitic effects, presented as parallel capacitors, have been considered to better assess the modelling. The complete H-bridge circuit is reported in fig. 3.31.

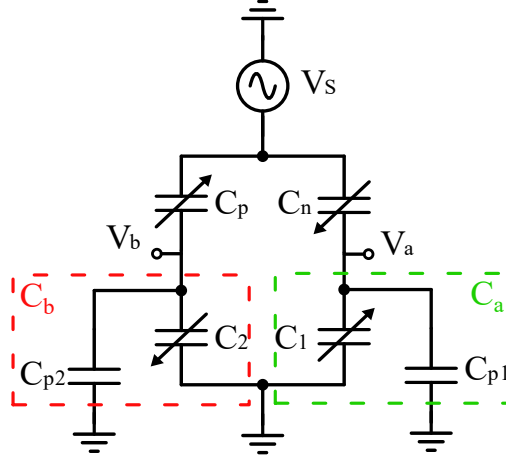


Figure 3.31. Interface H-bridge composed of sensor capacitors, VCCs, and parasitic capacitors [205].

After defining the circuit, its mathematical model has to be extracted. Parasitic effects, presented as parallel capacitors, have been considered to enhance the modelling result.

$$\begin{cases} C_a = C_1 + C_{p1} \\ C_b = C_2 - C_{p2} \end{cases} \quad (3.12)$$

The differential capacitive sensor structure changes the x -($C_p - C_n$) relationship [208]. Specifically, the x parameter is introduced to model the sensor in a $[-1, +1]$ range. Since in this work a linear differential sensor is considered, capacitors C_p and C_n are defined by the (3.13).

$$C_{p,n} = C_{bl}(1 \pm x) \quad (3.13)$$

where C_{bl} , known as the baseline capacitance, represents the sensors capacitors value when no input is applied ($x = 0$).

On the other hand, the VCCs are implemented using the AD633 circuit and connected in Zhong configuration [207]. As a result, the (3.14) expresses the mathematical relationship between the VCCs and their respective control voltages.

$$C_i = C_{fb} \left(1 - \frac{V_{ctrl,i}}{10 \text{ V}} \right) \quad i = 1, 2 \quad (3.14)$$

C_{fb} is the feedback capacitance used in the Zhong configuration.

By replacing the (3.12), (3.13), and (3.14) in the (3.9), the following (3.15) is obtained.

$$H_{L_{bridge}} = \frac{1}{a_1 + 1} - \frac{1}{a_2 + 1} \quad (3.15)$$

where

$$\begin{cases} a_1 = \frac{1 - \frac{V_{ctrl1}}{10}}{1 - x} + \frac{C_{p1}}{C_{bl}(1 - x)} \\ a_2 = \frac{1 - \frac{V_{ctrl2}}{10}}{1 + x} + \frac{C_{p2}}{C_{bl}(1 + x)} \end{cases} \quad (3.16)$$

hence, with the imposition of the fully differential condition $V_{ctrl1} = -V_{ctrl2}$ and the equilibrium condition $\Delta V = 0$, the circuit input-output characteristic, reported in

the (3.17), is obtained.

$$\begin{cases} V_{ctrl1} = 10x - \frac{5}{C_{bl}} [(C_{p2} - C_{p1}) - (C_{p1} + C_{p2})x] \\ V_{ctrl2} = -10x + \frac{5}{C_{bl}} [(C_{p2} - C_{p1}) - (C_{p1} + C_{p2})x] \end{cases} \quad (3.17)$$

The second term in both expressions of (3.17) depends on the baseline capacitance C_{bl} and the parasitic capacitors C_{p1} and C_{p2} . The baseline capacitor is inversely related to the control voltages, so increasing C_{bl} reduces both V_{ctrl1} and V_{ctrl2} . The parasitic capacitors also influence the magnitude of these control voltages and introduce an offset whenever $C_{p1} \neq C_{p2}$, regardless of the input parameter x .

3.2.3 Circuit Manufacturing

The circuit has been firstly designed through Autodesk Fusion360 software, using the accessory application for circuit design. The result, manufactured with CNC milling, is reported in Fig. 3.32

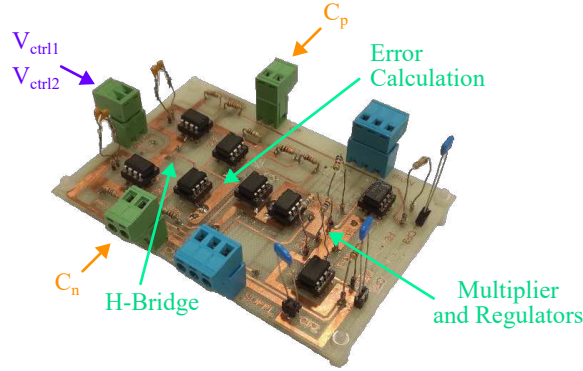


Figure 3.32. Final CNC-milled version of the presented circuit [205].

In the layout phase, symmetry has been maximized to minimize any parasitic effects. As indicated previously, AD633 Integrated Circuits (ICs) are used to implement multipliers and VCCs; the LF411 was used for the differential integral control; the INA128 was used for the error calculation circuit.

3.2.4 Measurements

The interface was subjected to static domain tests to extract and prove its transfer function. All the measurements have been performed with a linear differential capacitive sensor varying between 0 and 400 pF.

Static Domain Measurements

The first static domain measurement encompasses the circuit transfer function, defined as the relationship between the control voltages (V_{ctrl1} , V_{ctrl2}) and the input x parameter, as reported in (3.17). According to the formula, two linear trends, with positive and negative slopes, are expected. For this test, an arbitrary value was selected for the parasitic capacitances to achieve the best fit between theoretical and actual characteristics. In the resulting graph, as reported in fig. 3.33, a remarkable correlation can be noted.

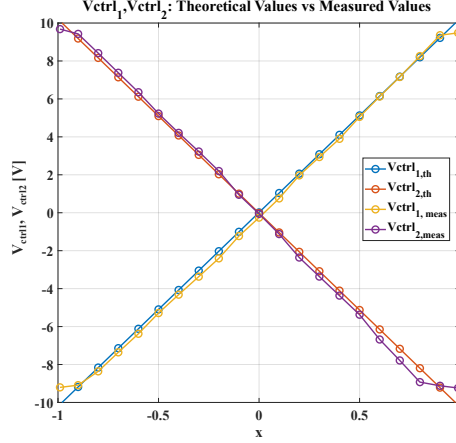


Figure 3.33. Circuit transfer function: theoretical and measured trends [205].

Theoretical and measured control-voltage trends match the expected linear but opposite behaviors, although both exhibit a parasitic-induced offset at $x = 0$. Measured values $V_{ctrl1} = -0.057$ V and $V_{ctrl2} = -0.256$ V yield a differential voltage of 0.2 V. Component nonidealities cause saturation when x lies roughly between -0.9 and 0.8 . The interface sensitivity, derived from the slope of ΔV_{ctrl} , is 102 mV/pF, and the match between theory and measurements is 9.3% for both control voltages.

To further analyze static performance, ΔV_{ctrl} was remeasured for different baseline capacitances C_{bl} , while keeping the VCC feedback capacitances constant. As shown in Table 3.4, modifying C_{bl} changes both sensitivity and the positions of the saturation points.

x	$V_{ctrl1} - V_{ctrl2}$ [V]		
	$C_{bl} = 20$ pF	$C_{bl} = 50$ pF	$C_{bl} = 200$ pF
-0.8	-12.9	-16.291	-16.31
-0.4	-8.86	-9.207	-8.313
-0.2	-4.1	-5.682	-4.625
0	-0.266	-0.387	-0.114
0.2	4.03	5.41	4.478
0.4	8.711	9.09	8.15
0.8	13.092	16.606	16.73

Table 3.4. Variation of the differential control voltage ΔV_{ctrl} with respect to input parameter x and circuit baseline capacitance C_{bl} .

Increasing the baseline capacitance shifts the saturation points of the transfer characteristic toward lower x values and reduces sensitivity. This occurs because the AD633-based VCC circuits exhibit a lower equivalent capacitance when the control voltage rises. Thus, when the baseline capacitance decreases, the auto-balancing loop raises the control voltages to compensate, eventually causing saturation near $x = -1$ and $x = +1$, where one sensor capacitor becomes small.

The transfer-function analysis, performed using varicap diodes to provide a controllable and narrow capacitance range, yields a resolution of approximately 20 fF with an error of 0.47 % error. As summarized in Table 3.5, the proposed interface outperforms existing solutions in resolution and error while maintaining comparable

sensitivity, with residual limitations attributed to PCB parasitic capacitances from the CNC-milled board.

Reference	Sensitivity [mV/pF]	Resolution	Error [%]	ΔC [pF]	Approach
This work	102	20 fF	0.47	400	C to V
[209]	N/A	500 pF	0.63	2000	C to digital
[210]	17.1	32.2 fF	N/A	27	C to frequency
[211]	222	2.5 aF	N/A	0.108	C to V
[212]	50	340 fF	0.8	280	C to V
[213]	175	N/A	0.13	20	C to digital

Table 3.5. Comparison of the proposed interface with other capacitive sensor interfaces in literature.

An additional, still important measurement performed in the static domain is the interface response to common-mode inputs. In particular, this analysis deepens the previous one by considering $x = 0$ and evaluating the effect on the differential control voltage ΔV_{ctrl} from changes in the C_{bl} baseline capacitance, and maintaining constant feedback and sensing capacitance values. Results, reported in Fig. 3.34, show a slow decrease of the differential control voltage when the C_{bl} capacitors are increased.

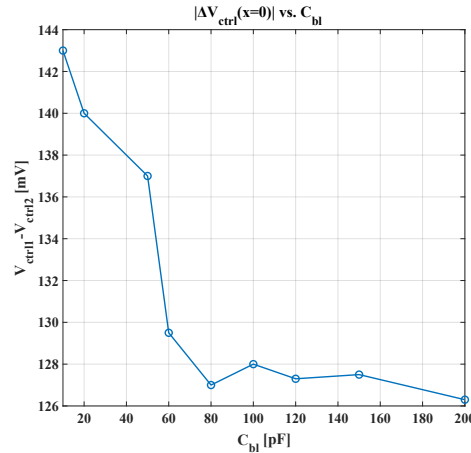


Figure 3.34. Response of the circuit interface to common-mode-related inputs [205].

In particular, by maintaining the VCCs feedback capacitors C_{fb} to 200 pF, the C_{bl} capacitance increase causes a convergence of the $(C_{bl} - C_{fb})$ difference to the zero helping, in turn, the convergence for the VCCs. Conversely, a decrease in the C_{bl} capacitance brings the baseline value farther from the VCCs feedback capacitors; hence, the differential control voltage increases, in this case, by 20 mV with respect to its minimum value. Additionally, the difference in terms of values between the C_{bl} capacitors significantly affects the offset measured with this interface. This problem was managed by simulating the baseline capacitors using discrete capacitors.

3.3 Circuits for Microelectronics

Microelectronics is essential in hardware-firmware systems for industrial applications, forming the core of microcontroller blocks and sensor interfaces. Proper microelectronic design ensures stable firmware execution with predictable timing, primarily relying on transistors while limiting the use of discrete components.

Circuits can be designed focusing on voltages or currents, resulting in Voltage-mode (VM) and Current-mode (CM) approaches. The VM approach is widely used in industrial electronics for its simplicity. In contrast, the CM approach is less common and more complex, but essential in microelectronics because transistors operate on currents. CM circuits were introduced in the late nineteenth century, briefly abandoned, and later reintroduced as technology enabled smaller and faster devices despite increased parasitic effects [214–218]. Modern CM circuits provide noise robustness, low-power operation [219], and high-speed performance [220, 221].

A key CM building block is the current conveyor, which was developed through three generations: First-Generation Current Conveyor (CCI), Second-Generation Current Conveyor (CCII), and Third-Generation Current Conveyor (CCIII). The CCII is most used due to its simplicity, internal buffers, and ability to implement common VM configurations with few components, while ensuring gain-bandwidth independence. However, CCII has low output impedance and is sensitive to non-idealities and parasitic effects, which limit its application range.

The VCII, as reported in fig. 3.35, is composed of three terminals:

- X** with high impedance, used as a voltage input node;
- Y** with low impedance, used as a current input node;
- Z** with low impedance, used as a voltage output node.

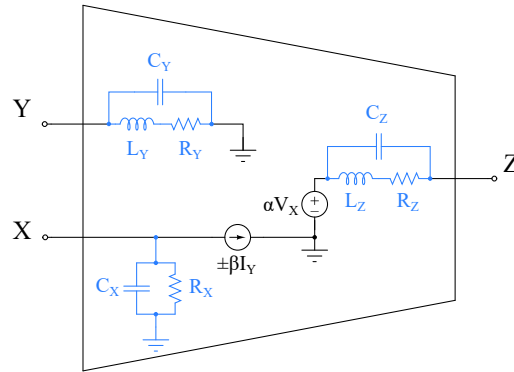


Figure 3.35. Equivalent circuit of the VCII [222].

The relationships between currents and voltages can be easily described with matrixes, as reported in the (3.18).

$$\begin{bmatrix} I_X \\ V_Y \\ V_Z \end{bmatrix} = \begin{bmatrix} \pm\beta & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \alpha & 0 \end{bmatrix} \cdot \begin{bmatrix} I_Y \\ V_X \\ I_Z \end{bmatrix} \quad (3.18)$$

The β parameter links currents between the X and Y terminals, while the α parameter links voltages between the X and Z terminals. Ideally, both parameters equal unity,

so the VCII behaves as a double buffer. In practice, parasitic effects — modeled in the frequency domain through RC or RLC networks — modify these relationships and reduce α and β below unity. As expressed in (3.19), the X terminal exhibits a resistive-capacitive trend, whereas the Y and Z terminals follow resistive-inductive-capacitive trends. Capacitive effects on the Y and Z terminals are usually neglected because they become relevant only at high frequencies.

$$\begin{bmatrix} I_X \\ V_Y \\ V_Z \end{bmatrix} = \begin{bmatrix} \pm\beta & \frac{1}{R_X} + sC_X & 0 \\ R_Y + sL_Y & 0 & 0 \\ 0 & \alpha & R_Z + sL_Z \end{bmatrix} \cdot \begin{bmatrix} I_Y \\ V_X \\ I_Z \end{bmatrix} \quad (3.19)$$

As a consequence,

1. the circuit structure directly affects the entity of these effects;
2. the buffer coefficients α and β depend on the circuit structure and define the quality of the VCII.

This chapter discusses a microelectronic circuit design centered on the VCII, aiming to achieve a rail-to-rail structure through power-supply pumping. The VCII is emphasized as its inherently current-mode architecture represents a natural evolution of transistor-level, current-regulated circuits. However, traditional supply-pumping techniques rely on voltage-mode charge pumps. The proposed solution, therefore, introduces a mixed current-/voltage-mode approach to implement a rail-to-rail circuit using the VCII.

3.3.1 The circuit blocks

This section deepens the circuits designed and used for this activity, with particular attention to the VCII and the charge pumps.

The VCII

The proposed VCII circuit, reported in fig. 3.36, features a ± 0.9 V non-pumped dual voltage supply for the input stages, on the X and Y terminals respectively; at the same time, the output stage on the Z terminal is supplied with the ± 1.45 V pumped dual supply voltage. Biasing currents, indicated as I_{B1} and I_{B2} equate $2 \mu\text{A}$.

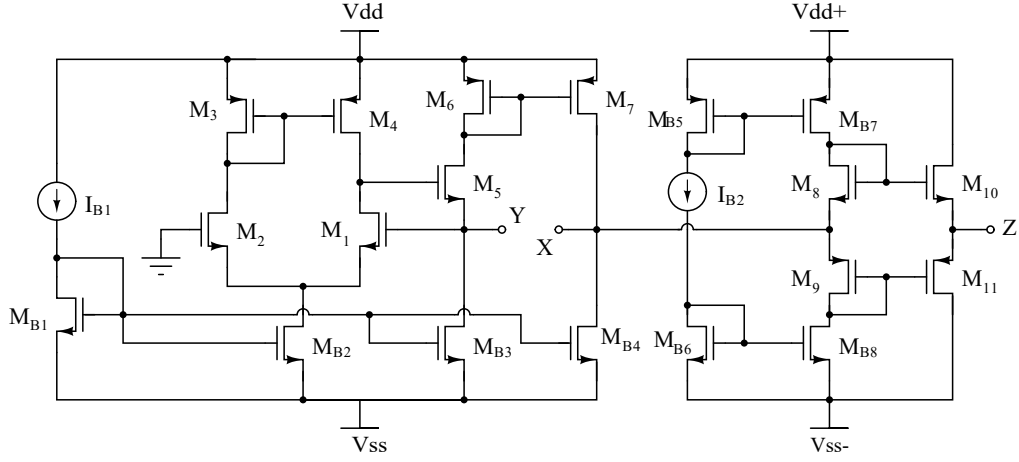


Figure 3.36. Proposed internal circuit for VCII [222].

An AB class output stage was used to implement the X-Z terminal; the source voltage at M_{10} and M_{11} transistors serves as output. In particular, the X-Y current buffer was implemented using a current mirror, with the M_6 and M_7 transistors. Most importantly, as reported above, the Y terminal must have the lowest impedance possible; hence, in this case, a controlled common-gate configuration was used. As a matter of fact, the (3.20) expresses the impedance on the Y terminal, Z_Y , in a steady state condition.

$$Z_Y \simeq \frac{1}{g_{m5} [1 + g_{m1} (r_{ds1} || r_{ds4})]} \quad (3.20)$$

Furthermore, the X terminal has a high impedance; hence, a simple common-source configuration was used. As a result, the impedance, reported in the (3.21), is the parallel equivalent resistance between the M_7 and M_{B4} drain terminals, and the equivalent series resistances calculated on the M_8 - M_{B7} and the M_9 - M_{B8} branches.

$$Z_X = r_{ds7} || r_{dsB4} || \left(\frac{1}{g_{m8}} + r_{dsB7} \right) || \left(\frac{1}{g_{m9}} + r_{dsB8} \right) \quad (3.21)$$

Finally, the low-Z terminal impedance, whose equation is reported in (3.22), is achieved through a common-drain configuration in the output stage of the mentioned class AB circuit.

$$Z_Z = \frac{1}{g_{m10}} || \frac{1}{g_{m11}} \quad (3.22)$$

The charge pumps

Since the VCII has a dual power supply, two charge pumps are needed. In particular, CP+s for positive power supply and CP-s for negative power supply are used. Fig.(3.37) reports both the structures used for the circuit implementation of the CP+ and CP-, respectively.

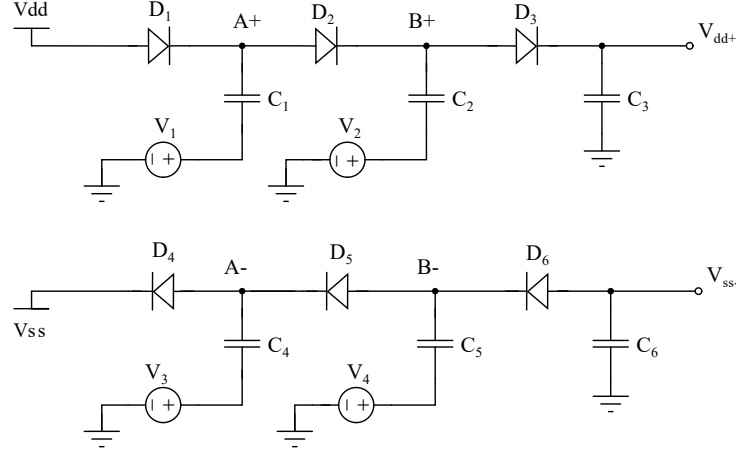


Figure 3.37. Circuits of the proposed positive charge pump (CP+), above, and negative charge pump (CP-), below [222].

The target is to boost the supply voltage with minimal power consumption. For this purpose, two three-stage charge pumps (CP+ and CP-) are used, each circuit triples the input voltage. Diodes are implemented with transistors in a diode configuration, where the bulk is tied to the drain to exploit the parasitic junction diode. Voltage is switched across C_1 - C_2 in the CP+ and across C_4 - C_5 in the CP-, using two phase-opposed square-wave signals per pump (V_1, V_2 for CP+ and V_3, V_4 for CP-). A third capacitor (C_3 or C_6) stores the pumped output voltage. Both charge pumps operate with the same principle, but opposite current direction, and they require three switching stages to reach steady-state conditions. Since the operation depends on voltages, these circuits are intended for high-load scenarios; with lower loads, higher current absorption causes a drop in the pumped output voltage and an increase in ripple.

3.3.2 Results

This section reports the results reached with this topology. In particular, sec. 3.3.2 shows results from the charge pump simulation and its effect in the VCII circuit; sec. 3.3.2 reports the extraction of the VCII α and β buffer parameters; and sec. 3.3.2 reports the extraction of terminal impedances for the VCII and its power consumption.

Charge pump results and output voltage dynamic

As shown in Fig. 3.38a, in charge pumps, ripples and output voltage drop are mainly influenced by stage capacitors and parasitic diodes, which require a horn-like transistor sizing using standard $0.18 \mu\text{m}$ CMOS technology. The capacitors are fixed at 100 nF and considered off-chip due to area constraints. Fig. 3.38a shows the pumped supply voltages for CP+ and CP- with the VCII as load (subfigure a) and the dynamic output change under resistive loads connected to the Z terminal (subfigure b). Pumped voltages reach $\pm 1.45 \text{ V}$ in about 1 ms . The lower-than-ideal 2.7 V value is due to the low equivalent load of the VCII, yet the output allows the dynamic extension beyond non-pumped supply limits. The resistive load variation demonstrates the effect on output dynamics: a $10 \text{ k}\Omega$ load limits dynamics to 94%,

representing the worst case, while a 100 k Ω load does not restrict output, representing the best case.

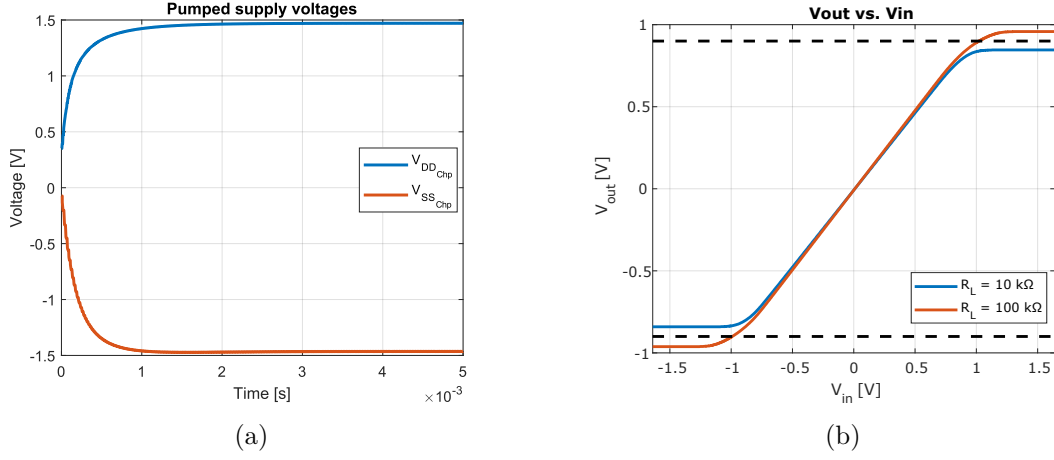


Figure 3.38. Pumped supply voltages (a) and VCII output voltage dynamic (b). [222]

α and β parameters extraction

The quality of the VCII depends on the α and β buffer parameters. Their behavior is analyzed over 1-100 MHz with loads from 10 k Ω to 1 M Ω . To evaluate α , a voltage source is applied to the X terminal and a load to Z, as in Fig. 3.39a, since α represents the voltage buffer between X and Z. To extract β , a current source is applied to Y and a load to X, as in Fig. 3.39b, reflecting the current buffer between X and Y.

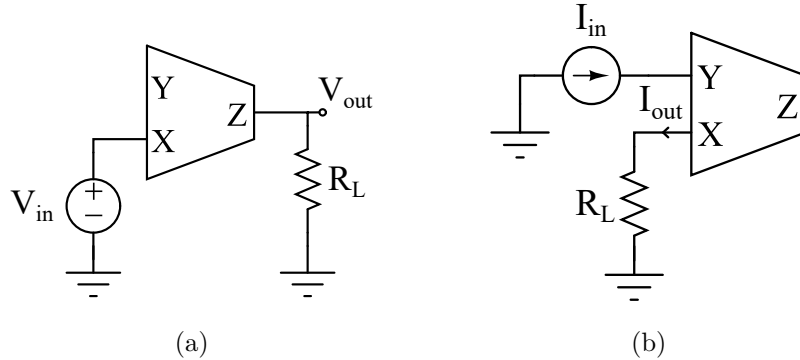


Figure 3.39. Circuit configuration for the α (a) and β (b) measurements. [222]

As shown in Fig. 3.40a, the α parameter exhibits a direct relationship with load: amplitude varies slightly from 0.94 to 0.97 (3.2%) and bandwidth increases from 191.5 to 281.7 MHz (14.2%), while Z-terminal equivalent impedance decreases. For the β parameter (Fig. 3.40b), amplitude and bandwidth decrease with load, dropping 62% and 90% respectively, due to the M_4 transistor operating in the linear region as drain-source voltage reduces.

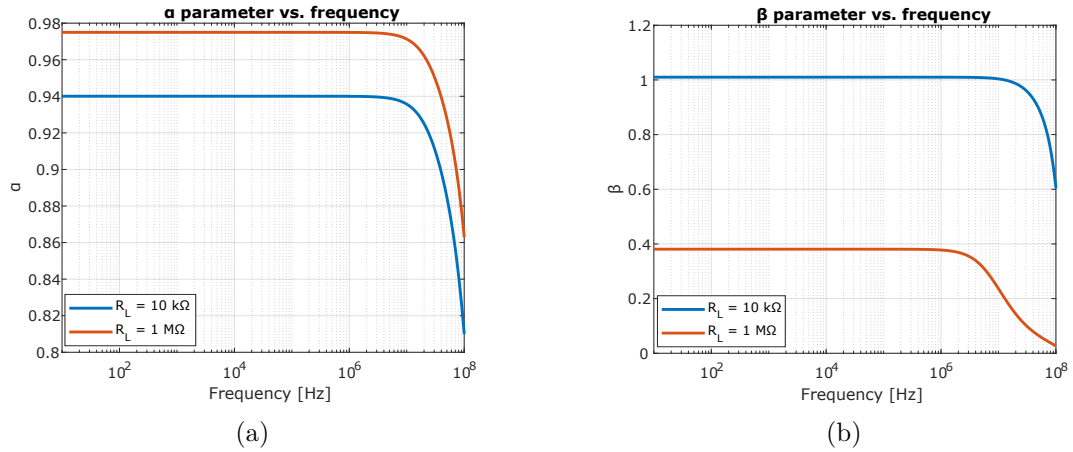
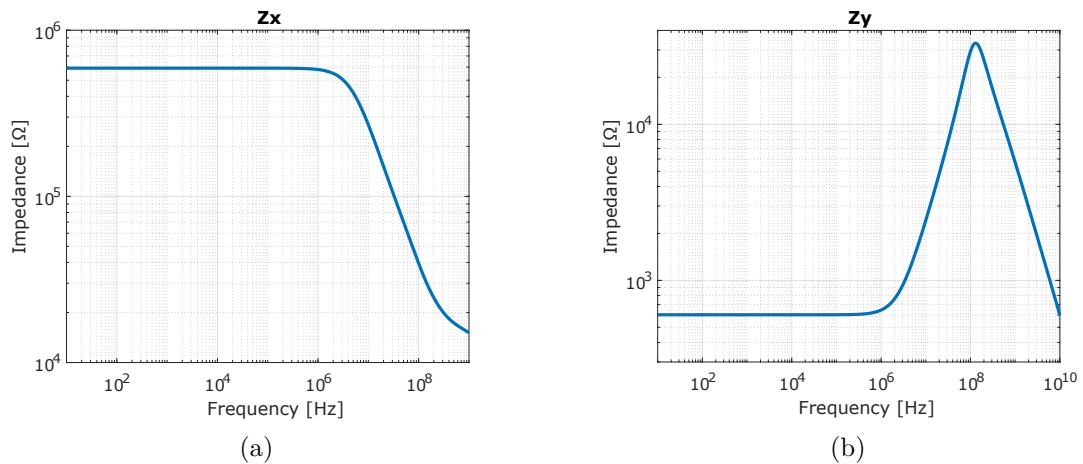


Figure 3.40. Frequency domain result for the α (a) and β (b) parameters.. [222]

VCII terminal impedances, transient test and power consumption extraction

In this section, the VCII terminal impedances are extracted and characterized, the whole circuit operation is shown through a transient test, and the power consumption is extracted.

The X terminal shows an equivalent resistance of 591 k Ω and capacitance of 50 fF, with signal distortion above 5.52 MHz (Fig. 3.41a). The Y terminal exhibits an RLC-like behaviour over 10 GHz: resistive up to 2 MHz with 602 Ω , inductive with 45 μH , and capacitive with 30 fF (Fig. 3.41b). Similarly, the Z terminal shows a resistive region of 370 Ω up to 1.63 MHz, an inductive region of 53 μH , and a capacitive region of 30 fF (Fig. 3.41c). Unlike the behavior at the Y terminal, the transition between inductive and capacitive regions includes a nearly constant section extending from 20 MHz to 2 GHz.



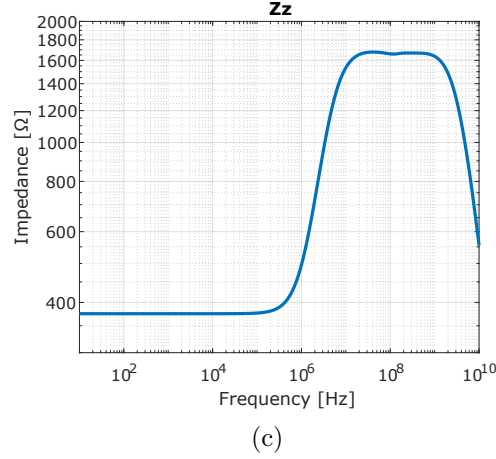


Figure 3.41. Frequency domain results for the X (a), Y (b), and Z (c) terminals. [222]

A transient test was performed on the full circuit by driving the X terminal with a voltage switching between 0.9 V and 0.1 V. As shown in Fig. 3.42, the outputs of the charge pumps exhibit only minor ripples. This behavior results from the low equivalent load presented by the VCII, due to the M_7 - M_{B4} branch structure (Fig. 3.36). The X-terminal voltage modulates the drain-source voltages of transistors, increasing current with higher voltages and decreasing it with lower voltages. The X-Z buffer effect is evident, as the Z-terminal voltage closely follows the X-terminal signal.

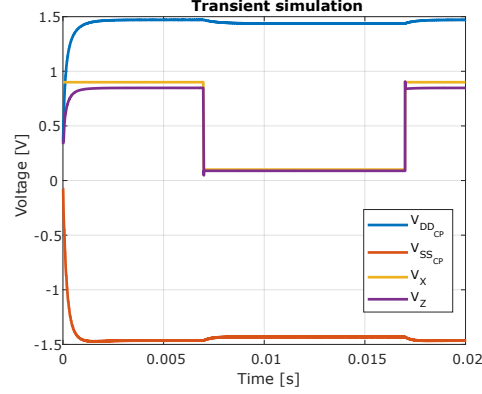


Figure 3.42. Time domain result for the transient test on the whole circuit [222].

The power consumption, extracted from the VCII in the transient test, is defined as the average power retrieved from both the positive and negative supply generators, V_{dd} and V_{ss} , equating $\pm 35 \mu\text{W}$. Furthermore, in the transient test, the charge pumps show a $\pm 3.4 \mu\text{W}$ power consumption.

3.4 DoE-based Mathematical Models

A mathematical model relates one or more input variables to one or more outputs. When derived through the DoE methodology, the model is linear in its coefficients. To construct it, input variables are first defined over specified minimum and maximum values, together with their corresponding outputs; the model is expected to provide reliable predictions within these ranges. For numerical consistency, all input intervals are normalized to $[-1, +1]$, where -1 and $+1$ correspond to the minimum and maximum input values, respectively. Subsequently, a screening procedure is applied to reduce the number of experimental executions. Screening imposes a functional constraint on one or more inputs—expressing a variable as a function of the others, as in (3.23)—and is particularly useful when dealing with a large number of parameters.

$$P_N = P_1 \cdot P_2 \cdot \dots \cdot P_{K-1} \quad (3.23)$$

where $P_1, \dots, P_K$ are input parameters. The reduction in the required number of samples depends directly on the number of imposed conditions and, in particular, on the number of inputs each condition involves. As shown in Fig. 3.43, for K input variables, applying a screening condition to M of them (with $M < K$) decreases the number of samples to 2^{K-M} . Consequently, the screening procedure reduces the experimental effort by a factor of 2^M .

		N# factors, k					
		3	4	5	6	7	8
N# runs	4	2_{III}^{3-1} $\pm C = AB$					
	8	2^3 full	2_{IV}^{4-1} $\pm D = ABC$	2_{III}^{5-2} $\pm D = AB$ $\pm E = AC$	2_{III}^{6-3} $\pm D = AB$ $\pm E = AC$ $\pm F = BC$	2_{III}^{7-4} $\pm D = AB$ $\pm E = AC$ $\pm F = BC$ $\pm G = ABC$	
	16	2^3 twice	2^4 full	2_V^{5-1} $\pm E = ABCD$	2_{IV}^{6-2} $\pm E = ABC$ $\pm F = ABD$	2_{IV}^{7-3} $\pm E = ABC$ $\pm F = ABD$ $\pm G = ACD$	2_{IV}^{8-4} $\pm E = ABC$ $\pm F = ABD$ $\pm G = ACD$ $\pm H = BCDE$
	32	2^3 4 times	2^4 twice	2^5 full	2_{VI}^{6-1} $\pm F = ABCDE$	2_{IV}^{7-2} $\pm F = ABC$ $\pm G = ABDE$	2_{IV}^{8-3} $\pm F = ABC$ $\pm G = ABD$ $\pm H = ACDE$
	64	2^3 8 times	2^4 4 times	2^5 twice	2^6 full	2_{VII}^{7-1} $\pm G = ABCDEF$	2_V^{8-2} $\pm G = ABCD$ $\pm H = ABEF$

Figure 3.43. DoE screening table [223].

While screening simplifies sample collection, it also reduces resolution and decreases the reliability of the resulting mathematical model. The extent of this degradation depends on the number of input variables that are screened. When many inputs are present, Pareto plots are first used to identify the most influential factors and limit the loss of resolution. Subsequently, linear regression is performed on these key inputs. In this context, the standard error and the t- and p-values play a central role, as the DoE-based model is linear in its coefficients. Notably, the p-value from Student's t-test is crucial, since it quantifies the probability of observing a non-zero coefficient under the null hypothesis that its true value is zero. Therefore, in DoE

analyses, the p-value is among the primary indicators considered after the standard error. The following subsection applies this DoE procedure to extract a mathematical model for an UWB pulse generator.

3.4.1 The Pulser

Several pulse-generating circuits can satisfy the high-voltage and fast-switching requirements of UWB systems. Among the most relevant topologies are the Vector Inversion Generator (VIG) [224], the Marx generator [225], and the Cockroft-Walton (CW) generator [226], each offering distinct mechanisms for voltage buildup, discharge, and energy delivery.

The VIG is a compact coaxial pulse generator optimized for sub-nanosecond operation thanks to its low parasitic inductance and enhanced impedance matching. Although no DoE-based models exist, analytical formulations such as those by Thrimawithana *et al.* [227] (see (3.24)) describe its voltage contributions from odd and even capacitor stages.

$$\begin{cases} I_L = \frac{N \cdot V_C}{R_L} + \sum_{n=1}^N \frac{V_{C(2n)}}{R_L} \\ V_{out} = N \cdot V_C + \sum_{n=1}^N V_{C(2n)} \end{cases} \quad (3.24)$$

The CW generator, consisting of diode-capacitor cascaded stages, is traditionally used for high-voltage DC generation but has applications in pulsed-power charging and triggering. While DoE modelling is absent, several works provide refined models that account for parasitic effects, including leakage inductances [228], dynamic behaviour [229], and behavioral or lumped-element formulations [230, 231].

The Marx generator remains one of the most widespread pulsed-power circuits due to its simplicity and scalability: capacitors are charged in parallel and discharged in series to produce high-voltage pulses. Its basic model is given by (3.25), but practical implementations vary significantly depending on physical layout and parasitic components [232–234]. Extensive research addresses parasitic inductances [235], diode and stage capacitance effects [236], as well as dynamic and high-speed operational modelling [237–239].

$$V_{out} = N \cdot V_{in} \quad (3.25)$$

Other noteworthy high-voltage pulse structures include the Blumlein line [240], capable of generating precisely shaped, impedance-matched pulses, and the Tesla coil [241], primarily resonant but still relevant in certain high-frequency high-voltage applications.

3.4.2 The Marx Generator

The Marx generator was selected as the candidate circuit because of its simple operating principle and high design flexibility. To align with modern trends — where solid-state switches such as SiC and GaN transistors are preferred — a modified solid-state Marx topology (Fig. 3.44) was adopted. Compared to traditional spark-gap implementations, this configuration offers precise control, higher repetition rates, and easier integration into compact systems, as spark-gap conduction relies on an inherently uncontrolled breakdown process.

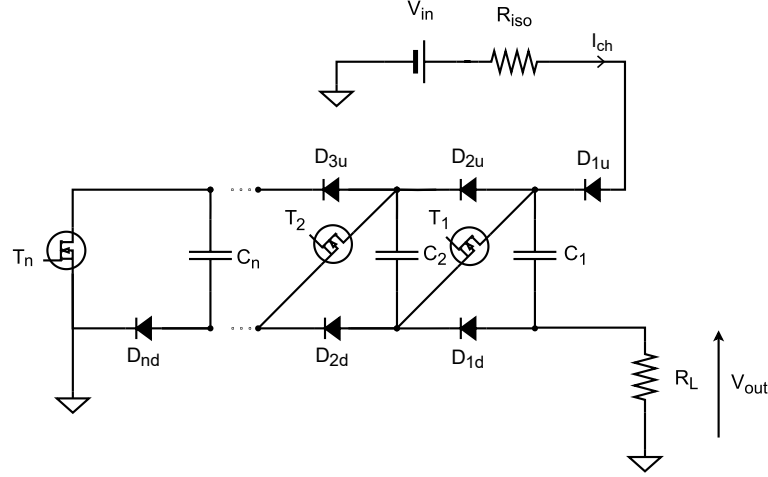


Figure 3.44. Schematic circuit of the used Marx generator.

The modified Marx generator operates in two phases: first, with all transistors open, the isolation resistor R_{iso} charges each stage capacitor; then, when the transistors switch on, the capacitors are placed in series, producing a negative output voltage peak as current flows through the load. The resulting output voltage follows the theoretical relation in (3.25). Compared to the classic topology, this configuration replaces inductors with diodes, improving simulation convergence. In this form, the circuit is suitable for a DoE-based extraction of its mathematical model, as detailed in Sec. 3.4.3.

3.4.3 Simulations

The simulated Marx generator enables circuit evaluation using near-real component and voltage values, providing early insight into its practical behavior. To better capture parasitic effects, an enhanced version of the topology from Sec. 3.4.2 is adopted (Fig. 3.45). This model includes branch parasitic inductances for both diode (L_{iu}, L_{id}) and capacitor (L_{Si}) paths, as well as series parasitic resistances R_S for the stage capacitors.

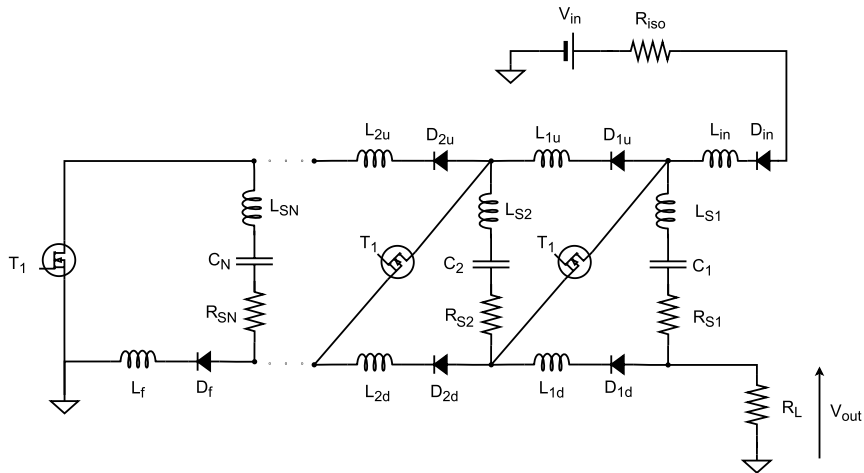


Figure 3.45. Schematic circuit of the used Marx generator with parasitic components.

As outlined in Sec. 3.4, the first step in extracting a mathematical model is defining

the input and output variables with their respective ranges. For this high-speed, high-voltage application, two output parameters are selected:

$V_{\text{out,min}}$ peak output voltage across R_L , used to uncover how V_{out} depends not only on V_{in} but also on other circuit parameters;

t_{fall} falling time of the output pulse (10%-90%), strongly influenced by circuit parasitics.

The corresponding input parameters are:

V_{in} input voltage in the $[400, 2000]$ V range;

N number of stages, in the $[3, 10]$ interval;

C stage capacitor value, in the $[100, 900]$ pF range;

R_{iso} insulation resistor, in the $[1, 6]$ $k\Omega$ range;

L_S parasitic series inductance per stage, in the $[0.1, 0.9]$ nH range;

R_S parasitic series resistance per stage, in the $[50, 1500]$ $m\Omega$ range.

According to Fig. 3.43, achieving maximum model resolution requires $2^6 = 64$ samples. Thus, a screening procedure is applied, based on the following generating conditions:

$$\begin{cases} L_S = N \cdot V_{\text{in}} \cdot R_{\text{iso}} \\ C = N \cdot V_{\text{in}} \cdot R_S \end{cases} \quad (3.26)$$

As shown in Fig. 3.43, applying the screening conditions reduces the required number of samples from 64 to 16, with only a negligible loss of resolution for this study. After converting each input variable to its $[-1, +1]$ normalized encoding, the resulting input combinations for all samples are listed in Tab. 3.6.

Sample	N	V_{in}	R_{iso}	R_S	L_S	C
1	-1	-1	-1	-1	-1	-1
2	1	-1	-1	-1	1	1
3	-1	1	-1	-1	1	1
4	1	1	-1	-1	-1	-1
5	-1	-1	1	-1	1	-1
6	1	-1	1	-1	-1	1
7	-1	1	1	-1	-1	1
8	1	1	1	-1	1	-1
9	-1	-1	-1	1	-1	1
10	1	-1	-1	1	1	-1
11	-1	1	-1	1	1	-1
12	1	1	-1	1	-1	1
13	-1	-1	1	1	1	1
14	1	-1	1	1	-1	-1
15	-1	1	1	1	-1	-1
16	1	1	1	1	1	1

Table 3.6. After-screening input combinations per sample in the simulated Marx topology.

The circuit was simulated using Simulation Program with Integrated Circuit Emphasis (SPICE) (Qorvo SPICE® and LTSPICE®) with real component models, specifically the Vishay VS-E5PX6012 diode and the GaN Systems GS66506T transistor. A 1 kHz control pulse with 0.001% duty cycle was applied to replicate the UWB operating conditions. Because branch inductances raise each stage to a higher-order network, convergence settings were carefully aligned across both simulators. After running all configurations in Tab. 3.6, the values of $V_{out,min}$ and the fall time were collected. Using R® and the pid library [223], Pareto plots were generated to determine the dominant input factors for each output, as illustrated in Fig. 3.46.

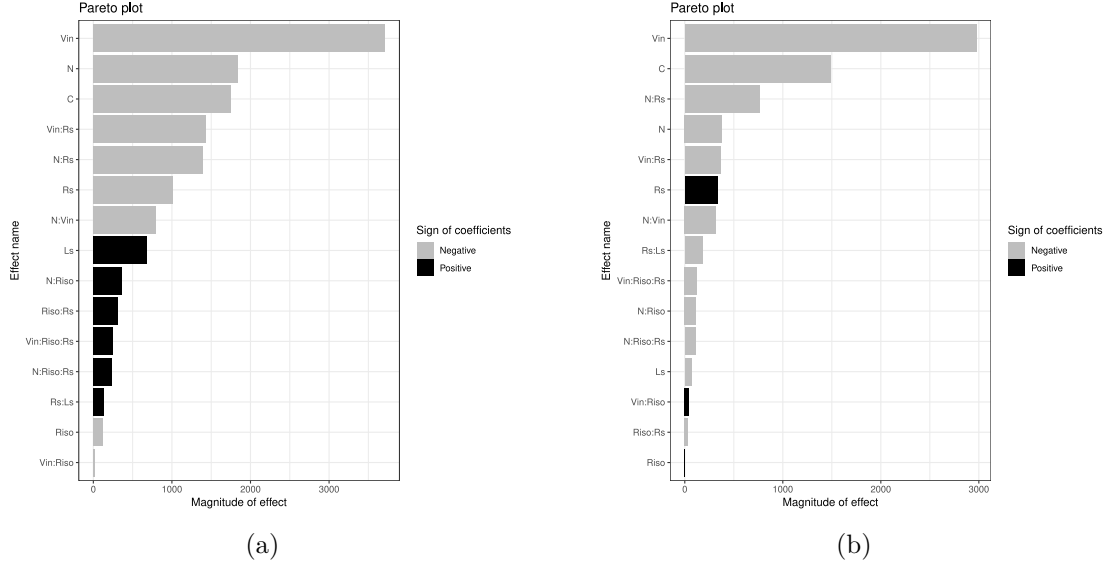


Figure 3.46. Pareto plots for the output voltage $V_{out,min}$ extracted from LTSPICE (a) and QSPICE (b).

Fig. 3.46 shows the variations in $V_{out,min}$ obtained from both simulators. In each case, the dominant factors influencing $V_{out,min}$ are the stage number N , input voltage V_{in} , and stage capacitor C , confirming that the selected input ranges capture the intrinsic circuit behavior in both simulations. This agreement also indicates that the screening process did not distort the underlying trends. Compared to the initial dataset, the second set yields less dispersed samples, improving model reliability. The standard errors are 324 for LTSPICE and 100.1 for QSPICE. Following Sec. 3.4, linear regression was applied using the most relevant inputs, producing the $V_{out,min}$ models in (3.27) and (3.28).

$$V_{out,min}^{LT} = -6035.9 - 3710.4 \cdot V_{in} - 1841.1 \cdot N - 1754.4 \cdot C - 789.8 \cdot V_{in} \cdot N - 1396.8 \cdot V_{in} \cdot C - 1426.5 \cdot N \cdot C - 1016.8 \cdot V_{in} \cdot N \cdot C \quad (3.27)$$

$$V_{out,min}^Q = -4970 - 2974.6 \cdot V_{in} - 379.9 \cdot N - 1489.2 \cdot C - 316.5 \cdot V_{in} \cdot N - 761.7 \cdot V_{in} \cdot C - 361.9 \cdot N \cdot C - 330.9 \cdot V_{in} \cdot N \cdot C \quad (3.28)$$

Equations (3.27) and (3.28) differ from the theoretical model in (3.25), reflecting the DoE philosophy, where all factors and interactions are empirically captured. Because mathematical models depend on the specific conditions under which the phenomenon

is analyzed, such deviations are expected. The two simulators also produce different coefficient dispersions, as seen in (3.27) and (3.28). In both models, the input voltage remains the dominant factor, balanced by the intercept terms. The LTSPICE model predicts overall higher output levels and shows greater sensitivity to the stage number, while the QSPICE model is slightly more sensitive to the capacitance C . Accordingly, QSPICE emphasizes capacitive effects, whereas Fig. 3.46 shows LTSPICE highlighting inductive effects. The corresponding t- and p-values for both models are listed in Tab. 3.7.

LTSpice			QSpice		
Parameter	t-value	p-value	Parameter	t-value	p-value
V_{in}	-11.451	$3.1 \cdot 10^{-6}$	V_{in}	-29.708	$1.8 \cdot 10^{-9}$
N	-5.682	$4.6 \cdot 10^{-4}$	N	-3.795	$5.3 \cdot 10^{-3}$
C	-5.414	$6.4 \cdot 10^{-4}$	C	-14.873	$4.1 \cdot 10^{-7}$
$N : V_{in}$	-2.437	$4.1 \cdot 10^{-2}$	$N : V_{in}$	-3.161	$1.3 \cdot 10^{-2}$
$C : V_{in}$	-4.311	$2.6 \cdot 10^{-3}$	$C : V_{in}$	-7.608	$6.3 \cdot 10^{-5}$
$N : C$	-4.402	$2.3 \cdot 10^{-3}$	$N : C$	-3.614	$6.8 \cdot 10^{-3}$
$N : V_{in} : C$	-3.138	$1.4 \cdot 10^{-2}$	$N : V_{in} : C$	-3.305	$1.1 \cdot 10^{-2}$

Table 3.7. t-values and p-values from the $V_{out,min}$ parameter in the simulated Marx circuit.

Both simulations show that the input voltage V_{in} and the stage capacitance C have the lowest p-values, confirming them as the most influential factors. In the QSPICE model, the stage number N and the C - V_{in} interaction also play a significant role. Because QSPICE is more sensitive to key inputs and produces results with lower dispersion, it is the more reliable simulator to analyze the output voltage of the circuit.

Secondly, the fall time t_{fall} output parameter is analyzed. Pareto plots extracted from the simulation results are reported in Fig. 3.47

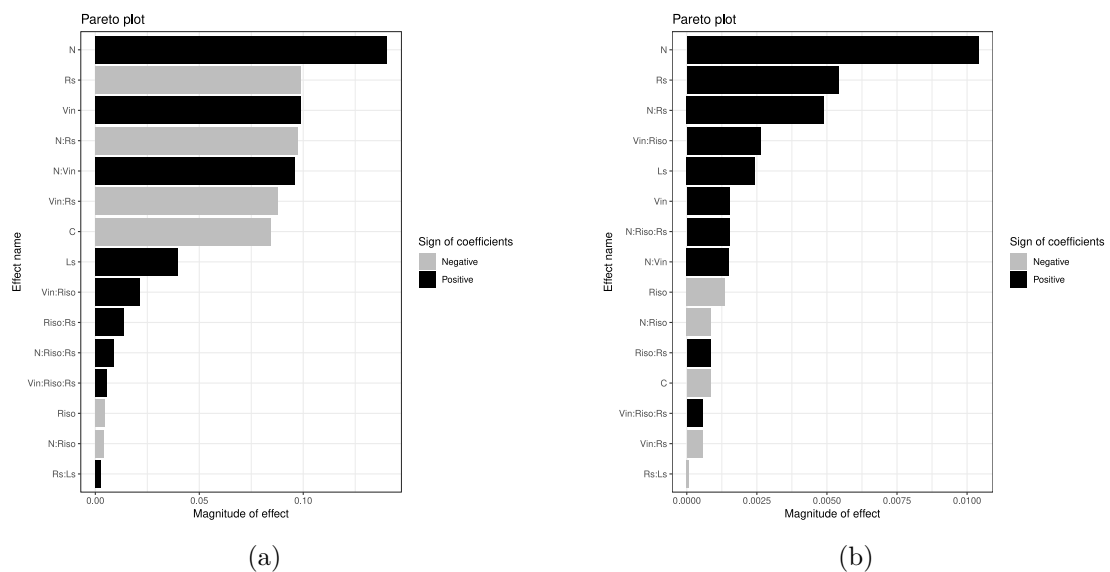


Figure 3.47. Pareto plots for the fall time t_{fall} extracted from LTSPICE (a) and QSPICE (b).

The Pareto charts in Fig. 3.47 show that the fall time t_{fall} mainly depends on the

stage number N and the stage resistor R_S in both simulations. A third influential factor appears in each model: the input voltage V_{in} in LTSPICE and the stage inductance L_S in QSPICE. The parameter R_S affects the two models with opposite signs: positive in QSPICE and negative in LTSPICE. In the LTSPICE model, R_S has a slightly stronger effect than the stage capacitor, but it was removed to keep the final model simple. The linear regression yields a standard error of 0.017 for LTSPICE and $1.187 \cdot 10^{-3}$ for QSPICE. The mathematical expressions for t_{fall} derived from both simulations are provided in (3.29) and (3.30).

$$t_{fall}^{LT} = 0.174 + 0.140 \cdot N - 0.099 \cdot R_S + 0.099 \cdot V_{in} - 0.097 \cdot N \cdot R_S + 0.096 \cdot N \cdot V_{in} - 0.088 \cdot R_S \cdot V_{in} - 0.085 \cdot N \cdot R_S \cdot V_{in} \quad (3.29)$$

$$t_{fall}^Q = 0.022 + 0.010 \cdot N + 0.005 \cdot R_S + 0.002 \cdot L_S + 0.005 \cdot N \cdot R_S + 0.003 \cdot N \cdot L_S - (8.2 \cdot 10^{-5}) \cdot R_S \cdot L_S + (5.7 \cdot 10^{-4}) \cdot N \cdot R_S \cdot L_S \quad (3.30)$$

The time-related output parameter mainly depends on the same two dominant factors, while a third factor changes between the two simulations and has a limited impact. The LTSPICE solver, which tends to emphasize inductive effects, likely explains why this simulator produces slightly larger average values than QSPICE. As a consequence, LTSPICE shows signals from the same circuit with slightly slower dynamics. Small discrepancies between the two sets of results may also arise from minor differences between the solvers. Therefore, although the two models can be compared regarding the influence of each parameter, they cannot be directly compared in magnitude. The t-values and p-values calculated with R, presented in Tab. 3.8, support this evaluation.

LTSpice			QSpice		
Parameter	t-value	p-value	Parameter	t-value	p-value
N	-8.169	$3.8 \cdot 10^{-5}$	N	8.76	$2.3 \cdot 10^{-5}$
R_S	-5.767	$4.2 \cdot 10^{-4}$	R_S	4.554	$1.9 \cdot 10^{-3}$
V_{in}	-5.755	$4.3 \cdot 10^{-4}$	L_S	2.049	$7.5 \cdot 10^{-2}$
$N : R_S$	-5.676	$4.7 \cdot 10^{-4}$	$N : R_S$	4.121	$3.3 \cdot 10^{-3}$
$N : V_{in}$	-5.590	$5.2 \cdot 10^{-4}$	$N : L_S$	2.207	$5.8 \cdot 10^{-2}$
$R_S : V_{in}$	-5.109	$9.2 \cdot 10^{-4}$	$R_S : L_S$	-0.069	$9.5 \cdot 10^{-1}$
$N : R_S : V_{in}$	-4.93	$1.1 \cdot 10^{-3}$	$N : R_S : L_S$	0.478	$6.5 \cdot 10^{-1}$

Table 3.8. t-values and p-values from the t_{fall} parameter in the simulated Marx circuit.

Tab. 3.8 shows that, in the LTSpice model, the parameter N has the strongest impact on the fall time, while R_S and V_{in} have similar and moderate influence. In the QSpice model, the N and R_S are the most influential parameters. These observations confirm that the fall time is strongly affected by the stage number, which is expected because a higher number of stages corresponds to more capacitors and a larger discharge time constant τ . Once again, QSpice proves to be more reliable for this analysis due to its smaller standard error and a larger set of relevant parameters.

Chapter 4

Conclusions

This doctoral thesis aimed to demonstrate the implementation possibilities of Hardware-Firmware (HW-FW) systems within industry-related contexts. Specifically, the work focuses on how the academic approach can be effectively ported to industrial applications. The research explored various applications, encompassing industrial instruments, wireless sensor networks, 3D-printed sensors, circuits, and their mathematical models. The flexible structure of HW-FW systems enables a broad spectrum of implementation methods and applications.

The main highlights of the thesis concern the crucial role of HW-FW systems in the evolving industrial landscape, spanning from Industry 4.0 to Industry 5.0. These systems are fundamental for several industrial necessities, including

- executing real-time control and processing essential for manufacturing workflows;
- enabling connectivity through field programmability and standard interfaces;
- supporting predictive maintenance and analytics via embedded diagnostic capabilities;
- guaranteeing system security

The design and implementation of HW-FW systems utilizing LoRa technology in three distinct industrial applications was documented in this work. In particular, a robust and energy-efficient Wireless Sensor Network (WSN) was developed to perform reliable environmental monitoring and geohazard assessment, and documented in Sec. 2.1. LoRa communication was selected for its long-range capabilities and low-power consumption, achieving an impressively low average current consumption in resting conditions for the smart nodes. The system proved effective in monitoring hydrogeological instability, recording inclination drifts, and measuring crack progression, incorporating external sensors such as crack meters and piezometers.

Additionally, a smart gas meter based on LoRa with a modular architecture was proposed and introduced in Sec. 2.2. The system structure encompasses a measurement block and a decoupled interfacing block, which communicate via Wireless Power Transfer (WPT). The design incorporated LoRaP2P for local neighbourhood communication and NB-IoT for internet transmission, ensuring operational flexibility adapted to specific network coverage scenarios. A critical feature was the integration

of blockchain technology (using Hyperledger Besu with Proof of Authority) to guarantee the transparency, integrity, and traceability of energy consumption records (“energy tokens”), thereby maximizing trust and accountability.

Finally, the Structural Health Monitoring (SHM) based on Spectral Analysis discussed in Sec. 2.3 introduced a redundant master-multi-slave system. Internal redundancy was enabled by utilizing two accelerometers (ADXL355 and ADXL345) and dual-protocol connectivity (RS-485 and LoRa). Experimental verification confirmed the ability of the system to detect structural modifications through changes in the time and frequency domain. Specifically, the system recorded a shift in the resonant frequency and an increase in the settling time following structural alteration.

Additionally, this work explored multidisciplinary research activities focused on advanced electronic design and modelling. These findings provide specific insights into how hardware components can be integrated with embedded control logic. Primarily, the work on 3D-Printed Sensors involved the FDM-manufacturing of a tri-axial accelerometer and a resistive strain gauge (bend sensor). The accelerator has been characterized under different infill values for its conductive proof masses. The bend sensor was characterized with a voltage-divider-based interface and then assembled into a smart glove connected to an Arduino UNO microcontroller. The customized firmware implemented basic logic to manage the analog reading of the sensor output and print the data via the serial port for real-time visualization of finger movements. This demonstrates how novel additive manufacturing hardware is directly linked to essential embedded firmware for functionality.

The design of a Fully Differential Analog Read-Out Circuit for differential capacitive sensors illustrates a specialized hardware approach tailored to maintain signal integrity. This analog circuit, using an auto-balanced De Sauty bridge with Voltage-Controlled Capacitors (VCCs), efficiently rejects common-mode noise. Achieving a satisfying resolution and error: this high-precision analog preprocessing shifts the complexity of noise handling away from the digital processing stage. This enables subsequent firmware routines to be simpler and dedicated primarily to data acquisition and high-level control.

Similarly, the design of a Rail-to-Rail Second-Generation Voltage Conveyor (VCII) emphasized achieving advanced performance metrics (noise robustness, wide dynamic range) through complex microelectronic hardware, including integrated charge pumps (CP+ and CP-). By addressing signal quality at the hardware level with specialized Current-mode (CM) circuits, the VCII architecture naturally simplifies the tasks required for a hypothetical control firmware.

Finally, the development of DoE-based Mathematical Models to analyze a high-voltage UWB pulse generator shows how complex, performance-critical relationships can be quantitatively captured. These empirical models, which capture the influence of parameters such as stage number and capacitance on pulse shape, represent knowledge that can be directly codified into firmware logic for precise real-time calibration or adaptive control of high-speed systems.

In conclusion, this thesis successfully achieved its purpose by exploring the HW-FW paradigm, choosing the LoRa technology as a starting point. The topics encountered in this work brought to a deeper understanding of the robustness and versatility of integrated HW-FW systems in industrial contexts. The findings highlight that well-engineered hardware absorbs system complexity, enabling the embedded firmware to focus on essential application logic, thereby maximizing the efficiency and reliability of the overall industrial solution.

Acronyms

HW Hardware

FW Firmware

AI Artificial Intelligence

LLM Large Language Models

LSTM Long Short-Term Memory

LoRa Long Range

SIL Software-In-the-Loop

HIL Hardware-In-the-Loop

DUT Device Under Test

IIoT Industrial Internet of Things

DDS Data Distribution Service

MQTT Message Queuing Telemetry Transport

CoAP Constrained Application Protocol

LoRaWAN Long Range Wide Area Network

EtherCAT Ethernet for Control Automation Technology

CIP Common Industrial Protocol

OPC UA Open Platform Communications Unified Architecture

AGV Automated Guided Vehicles

NB Narrow Band

LPWAN Low-Power Wide-Area Network

DSP Digital Signal Processors

MAC Multiply-and-Accumulate

FIR Finite Impulse Response

PLC Programmable Logic Controller

DAQ Data Acquisition

HDL Hardware Description Language

DAC Digital-to-Analog Converter

ADC Analog-to-Digital Converter

FPGA Field-Programmable Gate Array

MCU Microcontroller Unit

RAM Random Access Memory

SRAM Static RAM

PLL Phase-Locked Loops

GPIO General Purpose IO

SAR Successive Approximation Registers

MOSFET Metal-Oxide Semiconductor Field Effect Transistor

UART Universal Asynchronous Receiver-Transmitter

I²C Inter-Integrated Circuit

LCD Liquid Crystal Display

IR Infrared

GPIO General Purpose Input-Output

SPI Serial Peripheral Interface

USB Universal Serial Bus

ALU Arithmetic Logic Unit

ISR Interrupt Service Routine

FATFS Generic FAT Filesystem Module

CPU Central Processing Unit

DMA Direct Memory Access

WSN Wireless Sensor Network

CSS Chirp Spread Spectrum

IoT Internet of Things

GNSS Global Navigation Satellite System

VHR Very High-Resolution

TLS Terrestrial Laser Scanning

IRT Infrared Thermography

SAR Synthetic Aperture Radar

UAV Unmanned Aerial Vehicles

RSSI Received Signal Strength Integrity

MEMS Micro Electro-Mechanical Systems

OSI Open Systems Interconnection

GSM Global System for Mobile Communications

LTE Long-Term Evolution

IP Internet Protocol

HTTP Hypertext Transfer Protocol

UI User Interface

MQTT MQ Telemetry Transport

TCP Transmission Control Protocol

WPT Wireless Power Transfer

ALU Arithmetic Logic Unit

AMI Advanced Metering Infrastructures

SHM Structural Health Monitoring

CRC Cyclic Redundancy Check

FFT Fast Fourier Transform

GPS Global Positioning System

NMEA National Marine Electronics Association

TTF Time To First Fix

FFT Fast Fourier Transform

SPs Samples Per second

AM Additive Manufacturing

CNC Computer Numerical Control

SLA Stereolithography

AIP Aerosol Inkjet Printing

FDM Fused Deposition Modeling

DDM Direct Digital Manufacturing

SIW Substrate Integrated Waveguide

ABS Acrylonitrile Butadiene Styrene

UV Ultraviolet

HMI Human-Machine Interface

PLA Polylactic Acid

PCB Printed Circuit Boards

PVA Polyvinyl Alcohol

TPU Thermoplastic Polyurethane

VCC Voltage-Controlled Capacitor

INA Instrumentation Amplifier

IC Integrated Circuit

CMOS Complementary Metal-Oxide Semiconductor

JTAG Joint Test Action Group

CSS Chirp Spread Spectrum

VM Voltage-mode

CM Current-mode

CCI First-Generation Current Conveyor

CCII Second-Generation Current Conveyor

CCIII Third-Generation Current Conveyor

VCII Second-Generation Voltage Conveyor

CP+ positive charge pump

CP- negative charge pump

DoE Design of Experiments

UWB Ultrawide Band

VIG Vector Inversion Generator

CW Cockroft-Walton

SPICE Simulation Program with Integrated Circuit Emphasis

PoA Proof of Authority

SOS Sensor Observation Service

Bibliography

- [1] M. Gundall, D. Reti, and H. D. Schotten, “Application of virtualization technologies in novel industrial automation: Catalyst or show-stopper?,” in *2020 IEEE 18th International Conference on Industrial Informatics (INDIN)*, (Piscataway, NJ), pp. 283 – 290, IEEE, 2020.
- [2] T. P. Carvalho, F. A. A. M. N. Soares, R. Vita, R. da P. Francisco, J. P. Basto, and S. G. S. Alcalá, “A systematic literature review of machine learning methods applied to predictive maintenance,” *Computers & Industrial Engineering*, vol. 137, p. 106024, 2019.
- [3] M. Serror, S. Hack, M. Henze, M. Schuba, and K. Wehrle, “Challenges and opportunities in securing the industrial internet of things,” *IEEE Transactions on Industrial Informatics*, vol. 17, p. 2985–2996, May 2021.
- [4] G. H. Kruger, A. J. Shih, D. G. Hattingh, and T. I. van Niekerk, “Intelligent machine agent architecture for adaptive control optimization of manufacturing processes,” *Advanced Engineering Informatics*, vol. 25, no. 4, pp. 783–796, 2011. Special Section: Advances and Challenges in Computing in Civil and Building Engineering.
- [5] A. Novak, D. Bennett, and T. Kliestik, “Product decision-making information systems, real-time sensor networks, and artificial intelligence-driven big data analytics in sustainable industry 4.0,” *Economics, management and financial markets*, vol. 16, no. 2, pp. 62–72, 2021.
- [6] A. Bousdekis, G. Mentzas, D. Apostolou, and S. Wellsandt, “Evaluation of ai-based digital assistants in smart manufacturing,” in *Advances in Production Management Systems. Smart Manufacturing and Logistics Systems: Turning Ideas into Action* (D. Y. Kim, G. von Cieminski, and D. Romero, eds.), (Cham), pp. 503–510, Springer Nature Switzerland, 2022.
- [7] P. D. Marasco, “Navigating the complexity of touch,” *Science*, vol. 387, no. 6731, pp. 248–249, 2025.
- [8] O. Ott, L. Ralfs, and R. Weidner, “Framework for qualifying exoskeletons as adaptive support technology,” *Front Robot AI*, vol. 9, p. 951382, Dec. 2022.
- [9] A. Rejeb, K. Rejeb, J. G. Keogh, and E. Süle, “When industry 5.0 meets the circular economy: A systematic literature review,” *Circular Economy and Sustainability*, Apr 2025.

- [10] V. U. Somkuwar, R. Pal, A. Layak, S. Hukkeri, and S. Periyasamy, “Silk-based materials for sustainable soft electronics: structure, properties, and emerging applications,” *The Journal of The Textile Institute*, vol. 0, no. 0, pp. 1–16, 2025.
- [11] K.-H. Yu, T. Teng, S. H. Nah, H. Chai, Y. Zhi, K.-Y. Wang, Y. Chi, P. Psarras, M. Akbarzadeh, and S. Yang, “3d concrete printing of triply periodic minimum surfaces for enhanced carbon capture and storage,” *Advanced Functional Materials*, vol. n/a, no. n/a, p. 2509259, 2025.
- [12] A. Bousdekis, G. Mentzas, D. Apostolou, and S. Wellsandt, “Evaluation of ai-based digital assistants in smart manufacturing,” in *Advances in Production Management Systems. Smart Manufacturing and Logistics Systems: Turning Ideas into Action* (D. Y. Kim, G. von Cieminski, and D. Romero, eds.), (Cham), pp. 503–510, Springer Nature Switzerland, 2022.
- [13] A. Ahmadi, A. Cantini, V. G. Frías, and A. P. Staudacher, “The impact of labor flexibility on operational efficiency in industry 5.0: a systematic literature review,” *International Journal of Production Research*, vol. 0, no. 0, pp. 1–26, 2025.
- [14] X. Li, H. Shi, R. Xu, and W. Xu, “Ai awareness,” 2025.
- [15] G. Jin, X. Ni, K. Wei, J. Zhao, H. Zhang, and L. Jia, “Will the technological singularity come soon? modeling the dynamics of artificial intelligence development via multi-logistic growth process,” *Physica A: Statistical Mechanics and its Applications*, vol. 664, p. 130450, 2025.
- [16] F. Re Cecconi, A. Khodabakhshian, and L. Rampini, *Industry 5.0 in Construction: Towards a More Human-Centric and Ethical AI*, pp. 101–122. Cham: Springer Nature Switzerland, 2024.
- [17] L. Jena and B. Behera, “Environmental crisis and human well-being: A review,” vol. 66, pp. 561–574, 01 2017.
- [18] ISO/IEC JTC 1 Technical Committee, “ISO/IEC 20922:2016(en) - Information technology — Message Queuing Telemetry Transport (MQTT) v3.1.1,” 2016. Available at <https://www.iso.org/standard/69466.html>.
- [19] M. R. Palattella, R. Soua, A. Stemper, and T. Engel, “Aggregation of mqtt topics over integrated satellite-terrestrial networks,” *SIGMETRICS Perform. Eval. Rev.*, vol. 46, p. 96–97, Jan. 2019.
- [20] S. Katsikeas, K. Fysarakis, A. Miaoudakis, A. Van Bemten, I. Askoxylakis, I. Papaefstathiou, and A. Plemenos, “Lightweight & secure industrial iot communications via the mq telemetry transport protocol,” in *2017 IEEE Symposium on Computers and Communications (ISCC)*, pp. 1193–1200, 2017.
- [21] Q. Liu, J. Hou, Y. Yan, X. Wang, and X. Liu, “Research on cloud-edge interaction technology of power internet of things based on mqtt protocol,” *Journal of Physics: Conference Series*, vol. 2166, p. 012020, jan 2022.

- [22] J.-H. Guo, T.-Y. Lin, and K.-H. Hsia, “Web-based iot and robot scada using mqtt protocol,” *Journal of Robotics, Networking and Artificial Life*, vol. 9, no. 2, pp. 202–208, 2022.
- [23] C. B. Zach Shelby, Klaus Hartke, “RFC 7252, Constrained Application Protocol (CoAP),” June 2014. Available at <https://datatracker.ietf.org/doc/html/rfc7252>.
- [24] M. Iglesias-Urkia, A. Orive, A. Urbieto, and D. Casado-Mansilla, “Analysis of coap implementations for industrial internet of things: a survey,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, pp. 2505–2518, Jul 2019.
- [25] Object Management Group, “OMG Data Distribution Service (dds),” April 2015. Available at <https://www.omg.org/spec/DDS/1.4/PDF>.
- [26] LoRa Alliance, “TS001-1.0.4 LoRaWAN® L2 1.0.4 Specification,” 2020. Available at <https://resources.lora-alliance.org/technical-specifications/ts001-1-0-4-lorawan-l2-1-0-4-specification>.
- [27] P. J. Basford, F. M. J. Bulot, M. Apetroaie-Cristea, S. J. Cox, and S. J. Ossont, “Lorawan for smart city iot deployments: A long term evaluation,” *Sensors*, vol. 20, no. 3, 2020.
- [28] G.-C. Pătru, D.-C. Trancă, C.-M. Costea, D. Rosner, and R.-V. Rughiniș, “Lora based, low power remote monitoring and control solution for industry 4.0 factories and facilities,” in *2019 18th RoEduNet Conference: Networking in Education and Research (RoEduNet)*, pp. 1–6, 2019.
- [29] V. Pokhriyal, P. V, V. Kannan, D. K. Mahto, and V. Acharya, “Predictive maintenance for an industrial robotic arm using lora technology,” in *2024 IEEE 3rd International Conference on Computing and Machine Intelligence (ICMI)*, pp. 1–9, 2024.
- [30] R. F. Severino, A. Serrador, N. Datia, R. Campos-Rebelo, and J. Machado, “A low-cost solution for assets location tracking based on lorawan networks,” in *2023 IEEE 9th World Forum on Internet of Things (WF-IoT)*, pp. 1–8, 2023.
- [31] Texas Instruments Inc., “RS-232 Glossary and Selection Guide,” May 2023. Available at https://www.ti.com/lit/an/slla607/slla607.pdf?ts=1760103484726&ref_url=https%253A%252F%252Fwww.google.com%252F.
- [32] Texas Instruments Inc., “RS-485 Reference Guide,” 2014. Available at <https://web.archive.org/web/20180517101401/http://www.ti.com/lit/sg/slyt484a/slyt484a.pdf>.
- [33] Modbus Organization, “Modbus Application Protocol Specification - V1.1b3,” April 2012. Available at https://web.archive.org/web/20250825211935/https://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf.
- [34] G. Weigmann, Josef; Kilian, *Decentralization with PROFIBUS-DP: Architecture and Fundamentals, Configuration and Use with SIMATIC S7*. Publicis, 2000.

- [35] PROFIBUS Nutzerorganisation e.V., “PROFINET System Description Technology and Application,” October 2014. Available at <https://www.profibus.com/index.php?eID=dumpFile&t=f&f=51714&token=4ea5554cbb80a066e805a879116ead2a759c23c3>.
- [36] IEC Committee of Action, “IEC 61158: Industrial communication networks - Fieldbus specifications,” June 1999. Available at https://www.felser.ch/download/iec_61158_fieldbus.html.
- [37] S. Farné, E. Bassi, F. Benzi, and F. Compagnoni, “Iiot based efficiency monitoring of a gantry robot,” in *2016 IEEE 14th International Conference on Industrial Informatics (INDIN)*, pp. 714–719, 2016.
- [38] M. Paprocki and K. Erwiński, “Synchronization of electrical drives via ethercat fieldbus communication modules,” *Energies*, vol. 15, no. 2, 2022.
- [39] E. Romiti, J. Malzahn, N. Kashiri, F. Iacobelli, M. Ruzzon, A. Laurenzi, E. M. Hoffman, L. Muratore, A. Margan, L. Baccelliere, S. Cordasco, and N. Tsagarakis, “Toward a plug-and-work reconfigurable cobot,” *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 5, pp. 3219–3231, 2022.
- [40] M. Sarajchi and K. Sirlantzis, “Pediatric robotic lower-limb exoskeleton: An innovative design and kinematic analysis,” *IEEE Access*, vol. 11, pp. 115219–115230, 2023.
- [41] © Institute of Electrical and Electronic Engineers, EFTA 2001 By Paul Brooks European Marketing Manager, Logix/NetLinx Technology Adoption Rockwell Automation, “EtherNet/IP: Industrial Protocol White Paper,” October 2001. Available at https://literature.rockwellautomation.com/idc/groups/literature/documents/wp/enet-wp001_-en-p.pdf.
- [42] Comitato Elettrotecnico Italiano, “CEI EN 61131-9,” June 2014. Available at https://literature.rockwellautomation.com/idc/groups/literature/documents/wp/enet-wp001_-en-p.pdf.
- [43] OPC Foundation, “OPC UA Reference.” Available at <https://reference.opcfoundation.org/#>.
- [44] P. Varga, J. Peto, A. Franko, D. Balla, D. Haja, F. Janky, G. Soos, D. Ficzer, M. Maliosz, and L. Toka, “5g support for industrial iot applications— challenges, solutions, and research gaps,” *Sensors*, vol. 20, no. 3, 2020.
- [45] C/LAN/MAN - LAN/MAN Standards Committee, “IEEE Standard for Information Technology–Telecommunications and Information Exchange between Systems Local and Metropolitan Area Networks–Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 1: Enhancements for High-Efficiency WLAN,” May 2021. Available at <https://standards.ieee.org/ieee/802.11ax/7180/>.
- [46] R. K. Singh, P. P. Puluckul, R. Berkvens, and M. Weyn, “Energy consumption analysis of LPWAN technologies and lifetime estimation for IoT application,” *Sensors (Basel)*, vol. 20, Aug. 2020.

- [47] D. M. Hernandez, G. Peralta, L. Manero, R. Gomez, J. Bilbao, and C. Zubia, "Energy and coverage study of lpwan schemes for industry 4.0," in *2017 IEEE International Workshop of Electronics, Control, Measurement, Signals and their Application to Mechatronics (ECMSM)*, pp. 1–6, 2017.
- [48] F. Formosa, M. Malgeri, and M. Vigo, *A LPWAN Case Study for Asset Tracking*, pp. 63–81. Cham: Springer International Publishing, 2021.
- [49] M. Islam, H. M. M. Jamil, S. A. Pranto, R. K. Das, A. Amin, and A. Khan, "Future industrial applications: Exploring lpwan-driven iot protocols," *Sensors*, vol. 24, no. 8, 2024.
- [50] J. Jose and T. Sasipraba, "Indoor air quality monitors using iot sensors and lpwan," in *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, pp. 633–637, 2019.
- [51] S. Duangsuwan, A. Takarn, R. Nujankaew, and P. Jamjareegulgarn, "A study of air pollution smart sensors lpwan via nb-iot for thailand smart cities 4.0," in *2018 10th International Conference on Knowledge and Smart Technology (KST)*, pp. 206–209, 2018.
- [52] A. Upadhyay, S. Mukhuty, V. Kumar, and Y. Kazancoglu, "Blockchain technology and the circular economy: Implications for sustainability and social responsibility," *Journal of Cleaner Production*, vol. 293, p. 126130, 2021.
- [53] ethereum.org, "Ethereum Whitepaper," August 2025. Available at <https://ethereum.org/whitepaper>.
- [54] Hyperledger, "A Blockchain Platform for the Enterprise - Hyperledger Fabric," 2023. Available at <https://hyperledger-fabric.readthedocs.io/en/release-2.5/>.
- [55] IOTA Stiftung, "IOTA Documentation," 2025. Available at <https://docs.iota.org/>.
- [56] ETSI Group, "ETSI GS ZSM 001, Zero-touch network and Service Management (ZSM) - Requirements based on documented scenarios," October 2019. Available at https://www.etsi.org/deliver/etsi_gs/ZSM/001_099/001/01.01.01_60/gs_ZSM001v010101p.pdf.
- [57] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, "Zero Trust Architecture," August 2020. Available at <https://nvlpubs.nist.gov/nistpubs/specialpublications/NIST.SP.800-207.pdf>.
- [58] A. Mohseni Ejiyeh, "Real-time lightweight cloud-based access control for wearable iot devices: A zero trust protocol," in *Proceedings of the First International Workshop on Security and Privacy of Sensing Systems*, SensorsS&P, (New York, NY, USA), pp. 22–29, Association for Computing Machinery, 2023.
- [59] I. Ahmed, T. Nahar, S. S. Urmi, and K. A. Taher, "Protection of sensitive data in zero trust model," in *Proceedings of the International Conference on Computing Advancements*, ICCA '20, (New York, NY, USA), Association for Computing Machinery, 2020.

- [60] S. Ilić, A. Jevtić, S. Stanković, and G. Ristić, “Floating-gate mos transistor with dynamic biasing as a radiation sensor,” *Sensors*, vol. 20, no. 11, 2020.
- [61] X. Wang, Y. Wang, Q. Lin, and X. Yang, “Assessing global landslide casualty risk under moderate climate change based on multiple gcm projections,” *International Journal of Disaster Risk Science*, vol. 14, pp. 751–767, Oct 2023.
- [62] M. Farvacque, C. Corona, J. Lopez-Saez, R. Mainieri, M. Stoffel, F. Bourrier, N. Eckert, and D. Toe, “Estimating rockfall release frequency from blocks deposited in protection barriers, growth disturbances in trees, and trajectory simulations,” *Landslides*, vol. 19, pp. 7–18, Jan 2022.
- [63] J. Corominas, N. Lantada, M. A. Núñez-Andrés, and O. C. Mavrouli, “Fatal non-seismic rockfalls in spain,” *Geoenvironmental Disasters*, vol. 12, p. 17, May 2025.
- [64] P. Bertolo, C. Oggeri, and D. Peila, “Full-scale testing of draped nets for rock fall protection,” *Canadian Geotechnical Journal*, vol. 46, no. 3, pp. 306–317, 2009.
- [65] A. Volkwein, K. Schellenberg, V. Labiouse, F. Agliardi, F. Berger, F. Bourrier, L. K. A. Dorren, W. Gerber, and M. Jaboyedoff, “Rockfall characterisation and structural protection – a review,” *Natural Hazards and Earth System Sciences*, vol. 11, no. 9, pp. 2617–2651, 2011.
- [66] S. Lambert and F. Bourrier, “Design of rockfall protection embankments: A review,” *Engineering Geology*, vol. 154, pp. 77–88, 2013.
- [67] D. Peila, S. Pelizza, and F. Sassudelli, “Evaluation of behaviour of rockfall restraining nets by full scale tests,” *Rock Mechanics and Rock Engineering*, vol. 31, pp. 1–24, Feb 1998.
- [68] F. Vagnon, A. Segalini, and A. M. Ferrero, “Studies of flexible barriers under debris flow impact: An application to an alpine basin,” *Procedia Earth and Planetary Science*, vol. 15, pp. 165–172, 2015. World Multidisciplinary Earth Sciences Symposium, WMESS 2015.
- [69] A. Volkwein, W. Gerber, J. Klette, and G. Spescha, “Review of approval of flexible rockfall protection systems according to etag 027,” *Geosciences*, vol. 9, no. 1, 2019.
- [70] T. Fernandez-Steege, C. Arnhardt, K. Walter, S. ha, F. Niemeyer, B. Nakaten, S. Homfeld, K. Asch, R. Azzam, R. Bill, and H. Ritter, “Slews - a prototype system for flexible real time monitoring of landslides using an open spatial data infrastructure and wireless sensor networks,” *Geotechnol. Sci. Rep.*, vol. 13, 01 2009.
- [71] M. Thakur, R. K. Dhiman, S. Singh, N. Aggarwal, and F. Bourrier, “Development of rockfall early warning system (rews) for prediction of rockfall hazard in manikaran, nw himalaya, india,” *Journal of Geological Society of India*, vol. 101, p. 437–447, Apr. 2025.

- [72] E. Ghazanfari, S.-U. Yoon, Y. Dong, X. Li, C. Medina, D. Seserko, L. Cheng, T. Yun, and S. Pamukcu, "Subsurface geo-event monitoring using wireless sensor networks," *Geotechnical Special Publication*, pp. 1732–1742, 03 2011.
- [73] M. R. James and S. Robson, "Straightforward reconstruction of 3d surfaces and topography with a camera: Accuracy and geoscience application," *Journal of Geophysical Research: Earth Surface*, vol. 117, no. F3, 2012.
- [74] F. Ardizzone, G. Basile, M. Cardinali, N. Casagli, S. D. Conte, C. D. Ventisette, F. Fiorucci, F. Garfagnoli, G. Gigli, F. Guzzetti, G. Iovine, A. Mondini, S. Moretti, M. Panebianco, F. Raspini, P. Reichenbach, M. Rossi, L. Tanteri, and O. Terranova, "Landslide inventory map for the briga and the giampilieri catchments, ne sicily, italy," *Journal of Maps*, vol. 8, no. 2, pp. 176–180, 2012.
- [75] L. Spampinato, S. Calvari, C. Oppenheimer, and E. Boschi, "Volcano surveillance using infrared cameras," *Earth-Science Reviews*, vol. 106, no. 1, pp. 63–91, 2011.
- [76] M. Barla and F. Antolini, "An integrated methodology for landslides' early warning systems," *Landslides*, vol. 13, pp. 215–228, Apr 2016.
- [77] A. M. Fanos, B. Pradhan, S. Mansor, Z. M. Yusoff, and A. F. b. Abdullah, "A hybrid model using machine learning methods and gis for potential rockfall source identification from airborne laser scanning data," *Landslides*, vol. 15, pp. 1833–1850, Sep 2018.
- [78] M. Erdelj, M. Król, and E. Natalizio, "Wireless sensor networks and multi-uav systems for natural disaster management," *Computer Networks*, vol. 124, pp. 72–86, 2017.
- [79] L. Weidner and G. Walton, "Monitoring the effects of slope hazard mitigation and weather on rockfall along a colorado highway using terrestrial laser scanning," *Remote Sensing*, vol. 13, no. 22, 2021.
- [80] F. Ravet, F. Briffod, A. Goy, and E. Rochat, "Mitigation of geohazard risk along transportation infrastructures with optical fiber distributed sensing," *Journal of Civil Structural Health Monitoring*, vol. 11, pp. 967–988, Sep 2021.
- [81] A. Sheth, K. Tejaswi, P. Mehta, C. Parekh, R. Bansal, S. Merchant, T. Singh, U. Desai, C. Thekkath, and K. Toyama, "Senslide : a sensor network based landslide prediction system," pp. 280–281, 01 2005.
- [82] M. Ragnoli, M. Scarsella, A. Leoni, G. Ferri, and V. Stornelli, "Wireless sensor network-based rockfall and landslide monitoring systems: A review," *Sensors*, vol. 23, no. 16, 2023.
- [83] H. Wang, X. Tuo, D. Xi, L. Fan, Z. Zhang, G. Zhang, and S. Hao, "Research on one zicm2410-based wireless sensor network for landslide monitoring," in *2011 7th International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 1–4, 2011.

- [84] M. Janeras, J.-A. Jara, M. J. Royán, J.-M. Vilaplana, A. Aguasca, X. Fàbregas, J. A. Gili, and P. Buxó, “Multi-technique approach to rockfall monitoring in the montserrat massif (catalonia, ne spain),” *Engineering Geology*, vol. 219, pp. 4–20, 2017. Special Issue on Monitoring, physical and numerical modeling of landslides and debris flows.
- [85] A. Giorgetti, M. Lucchi, E. Tavelli, M. Barla, G. Gigli, N. Casagli, M. Chiani, and D. Dardari, “A robust wireless sensor network for landslide risk analysis: System design, deployment, and field testing,” *IEEE Sensors Journal*, vol. 16, no. 16, pp. 6374–6386, 2016.
- [86] J. Singh, R. Kaur, and D. Singh, “Energy harvesting in wireless sensor networks: A taxonomic survey,” *International Journal of Energy Research*, vol. 45, no. 1, pp. 118–140, 2021.
- [87] Semtech Corp., “LoRa and LoRaWAN - an1200.86,” 2024. Available at <https://www.semtech.com/uploads/technology/LoRa/lora-and-lorawan.pdf>.
- [88] R. Fekih Romdhane, Y. Lami, D. Genon-Catalot, N. Fourty, A. Lagrèze, D. Jongmans, and L. Baillet, “Wireless sensors network for landslides prevention,” in *2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*, pp. 222–227, 2017.
- [89] M. Muladi, I. M. Wirawan, D. Lestari, S. Choirul, E. Raka, A. Leksono, F. Qodri, and S. Prasetyo, “Chirpstack-based lorawan platform for land-sliding monitoring system,” *Instrumentation Mesure Métrologie*, vol. 24, pp. 73–79, 02 2025.
- [90] K.-H. Tseng, M.-Y. Chung, L.-H. Chen, and Y.-W. Huang, “Implementation of composite lpwan on the slope disaster prevention monitoring system,” *IEEE Sensors Journal*, vol. 22, no. 3, pp. 2658–2671, 2022.
- [91] C. Wang, W. Guo, K. Yang, X. Wang, and Q. Meng, “Real-time monitoring system of landslide based on lora architecture,” *Frontiers in Earth Science*, vol. Volume 10 - 2022, 2022.
- [92] M. M. Timbul Sigiuro, I. P. Manalu, G. I. Wowiling, S. M. Silalahi, E. S. Sinambela, and F. Simatupang, “Improving road safety in landslide prone areas: Real-time lora-based landslide detection and warning system,” in *2023 29th International Conference on Telecommunications (ICT)*, pp. 1–5, 2023.
- [93] I. W. D. Pancane, I. P. G. A. Putra, I. W. Gede Suarjana, and A. Supranartha, “Lora network design for monitoring landslides on 150 kv delivery tower,” in *2024 10th International Conference on Smart Computing and Communication (ICSCC)*, pp. 153–158, 2024.
- [94] OpenJS Foundation, “Node-RED - low-code programming for event driven applications.,” 253. Available at <https://nodered.org/>.

- [95] P. Esposito, M. Ragnoli, G. Barile, A. Leoni, G. Ferri, and V. Stornelli, "A multi-level lora application for underwood monitoring," in *2023 16th International Conference on Advanced Technologies, Systems and Services in Telecommunications (TELSIKS)*, pp. 251–254, 2023.
- [96] M. Ragnoli, A. Leoni, G. Barile, G. Ferri, and V. Stornelli, "Lora-based wireless sensors network for rockfall and landslide monitoring: A case study in pantelleria island with portable lorawan access," *Journal of Low Power Electronics and Applications*, vol. 12, no. 3, 2022.
- [97] M. Ragnoli, P. Esposito, V. Stornelli, G. Barile, E. D. Santis, and N. Sciarra, "A lora-based wireless sensor network monitoring system for urban areas subjected to landslide," in *Proceedings of the 2023 8th International Conference on Cloud Computing and Internet of Things, CCIOT '23*, (New York, NY, USA), p. 91–97, Association for Computing Machinery, 2023.
- [98] M. Ragnoli, P. Esposito, V. Stornelli, G. Ferri, A. Leoni, and N. Sciarra, "Rock masses crack monitoring using a lora-based wireless sensor network," in *2023 8th International Conference on Smart and Sustainable Technologies (SpliTech)*, pp. 1–4, 2023.
- [99] P. Esposito, G. Ferri, and V. Stornelli, "Monitoring systems for a volcanic area subjected to rockfall: Case study in the island of ischia," in *2024 11th International Conference on Electrical, Electronic and Computing Engineering (IcETran)*, pp. 1–5, 2024.
- [100] M. Ragnoli, P. Esposito, N. Sciarra, G. Ferri, and V. Stornelli, "Landslide monitoring system for inhabited areas: a case study in central italy," in *2024 IEEE Sensors Applications Symposium (SAS)*, pp. 1–5, 2024.
- [101] V. Stornelli, M. Ragnoli, P. Esposito, N. Sciarra, G. Barile, and E. De Santis, "Rockfall monitoring using multi-technological wireless sensor network," in *2023 58th International Scientific Conference on Information, Communication and Energy Systems and Technologies (ICEST)*, pp. 3–10, 2023.
- [102] M. Ragnoli, G. Barile, A. Leoni, S. Ricci, and V. Stornelli, "Energy-optimized monitoring system for underground cave environments based on long preamble lora," *IEEE Access*, vol. 13, pp. 24717–24735, 2025.
- [103] L. Montuori, M. Alcázar-Ortega, C. Vargas-Salgado, and I. Mansó-Borràs, "Enabling the natural gas system as smart infrastructure: Metering technologies for customer applications," in *2020 Global Congress on Electrical Engineering (GC-ElecEng)*, pp. 96–100, 2020.
- [104] R. Moletta and G. Albagnac, "A gas meter for low rates of gas flow: Application to the methane fermentation," *Biotechnology Letters*, vol. 4, pp. 319–322, May 1982.
- [105] F. Cascetta and P. Vigo, "The future domestic gas meter: Review of current developments," *Measurement*, vol. 13, no. 2, pp. 129–145, 1994.
- [106] S. Feng, F. Farha, Q. Li, Y. Wan, Y. Xu, T. Zhang, and H. Ning, "Review on smart gas sensing technology," *Sensors*, vol. 19, no. 17, 2019.

- [107] M. C. A. Prabowo, S. S. Hidayat, and F. Luthfi, “Low cost wireless sensor network for smart gas metering using antares iot platform,” in *2020 International Conference on Applied Science and Technology (iCAST)*, pp. 175–180, 2020.
- [108] Z. Wang, C. Hu, D. Zheng, and X. Chen, “Ultralow-power sensing framework for internet of things: A smart gas meter as a case,” *IEEE Internet of Things Journal*, vol. 9, no. 10, pp. 7533–7544, 2022.
- [109] Wireless Power Consortium, “Wireless Power Consortium, Qi specifications,” 2023. Available at <https://www.wirelesspowerconsortium.com/knowledge-base/specifications/>.
- [110] L. Chiominto, G. D’Emilia, P. Esposito, G. Ferri, E. Natale, D. Polverini, P. Spinozzi, V. Stornelli, and L. Chiavaroli, “A new and smart gas meter with blockchain validation for distributed management of energy tokens,” *Eng*, vol. 6, no. 11, 2025.
- [111] H. Y. Özderya, “SerialPlot - Realtime Plotting Software,” 2025. Available at <https://hackaday.io/project/5334-serialplot-realtime-plotting-software>.
- [112] N. Novas, J. Gázquez, J. MacLennan, R. García, M. Fernández-Ros, and F. Manzano-Agugliaro, “A real-time underground environment monitoring system for sustainable tourism of caves,” *Journal of Cleaner Production*, vol. 142, pp. 2707–2721, 2017.
- [113] G. Barile, A. Leoni, L. Pantoli, and V. Stornelli, “Real-time autonomous system for structural and environmental monitoring of dynamic events,” *Electronics*, vol. 7, no. 12, 2018.
- [114] M. Muttillio, L. D. Battista, T. de Rubeis, and I. Nardi, “Structural health continuous monitoring of buildings – a modal parameters identification system,” in *2019 4th International Conference on Smart and Sustainable Technologies (SpliTech)*, pp. 1–4, 2019.
- [115] L. Pantoli, M. Muttillio, G. Ferri, V. Stornelli, R. Alaggio, D. Vettori, L. Chinzari, and F. Chinzari, “Electronic system for structural and environmental building monitoring,” in *Sensors* (B. And, F. Baldini, C. Di Natale, V. Ferrari, V. Marletta, G. Marrazza, V. Militello, G. Miolo, M. Rossi, L. Scalise, and P. Siciliano, eds.), (Cham), pp. 481–488, Springer International Publishing, 2019.
- [116] T. de Rubeis, M. Muttillio, I. Nardi, L. Pantoli, V. Stornelli, and D. Ambrosini, “Integrated measuring and control system for thermal analysis of buildings components in hot box experiments,” *Energies*, vol. 12, no. 11, 2019.
- [117] M. Ragnoli, D. Colaiuda, A. Leoni, G. Ferri, G. Barile, M. Rotilio, E. Laurini, P. De Berardinis, and V. Stornelli, “A lorawan multi-technological architecture for construction site monitoring,” *Sensors*, vol. 22, no. 22, 2022.
- [118] M. Lake, J. Wernette, T. Khan, and A. Abdelgawad, “Design of a structural health monitoring and notification system,” in *2017 IEEE International Conference on Electro Information Technology (EIT)*, pp. 321–323, 2017.

- [119] A. Fernández-Navamuel, D. Zamora-Sánchez, T. Varona-Poncela, C. Jiménez-Fernández, J. Díez-Hernández, D. García-Sánchez, and D. Pardo, “Vibration-based shm strategy for a real time alert system with damage location and quantification,” in *European Workshop on Structural Health Monitoring* (P. Rizzo and A. Milazzo, eds.), (Cham), pp. 245–255, Springer International Publishing, 2021.
- [120] G. F. Gomes, Y. A. D. Mendéz, P. da Silva Lopes Alexandrino, S. S. da Cunha, and A. C. Ancelotti, “The use of intelligent computational tools for damage detection and identification with an emphasis on composites – a review,” *Composite Structures*, vol. 196, pp. 44–54, 2018.
- [121] T. Ooijevaar, *Vibration based structural health monitoring of composite skin-stiffener structures*. Phd thesis - research ut, graduation ut, University of Twente, Netherlands, Mar. 2014.
- [122] H.-N. Li, L. Ren, Z.-G. Jia, T.-H. Yi, and D.-S. Li, “State-of-the-art in structural health monitoring of large and complex civil infrastructures,” *Journal of Civil Structural Health Monitoring*, vol. 6, pp. 3–16, Feb 2016.
- [123] A. A. Mufti, “Structural health monitoring of innovative canadian civil engineering structures,” *Structural Health Monitoring*, vol. 1, no. 1, pp. 89–103, 2002.
- [124] J. Ko and Y. Ni, “Technology developments in structural health monitoring of large-scale bridges,” *Engineering Structures*, vol. 27, no. 12, pp. 1715–1725, 2005. SEMC 2004 Structural Health Monitoring, Damage Detection and Long-Term Performance.
- [125] M. Muttillio, V. Stornelli, R. Alaggio, R. Paolucci, L. Di Battista, T. de Rubeis, and G. Ferri, “Structural health monitoring: An iot sensor system for structural damage indicator evaluation,” *Sensors*, vol. 20, no. 17, 2020.
- [126] R. Paolucci, M. Muttillio, M. Di Luzio, R. Alaggio, and G. Ferri, “Electronic sensory system for structural health monitoring applications,” in *2020 5th International Conference on Smart and Sustainable Technologies (SpliTech)*, pp. 1–5, 2020.
- [127] S. Gopalakrishnan, M. Ruzzene, and S. Hanagud, “Computational techniques for structural health monitoring,” 01 2011.
- [128] C. Gentile, A. Ruccolo, and F. Canali, “Long-term monitoring for the condition-based structural maintenance of the milan cathedral,” *Construction and Building Materials*, vol. 228, p. 117101, 2019.
- [129] Y. Zhang, M. Kurata, and J. P. Lynch, “Long-term modal analysis of wireless structural monitoring data from a suspension bridge under varying environmental and operational conditions: System design and automated modal analysis,” *Journal of Engineering Mechanics*, vol. 143, no. 4, p. 04016124, 2017.
- [130] A. Aloisio, L. D. Battista, R. Alaggio, E. Antonacci, and M. Fragiaco, “Assessment of structural interventions using bayesian updating and subspace-based fault detection methods: the case study of s. maria di collemaggio

- basilica, l'aquila, italy," *Structure and Infrastructure Engineering*, vol. 17, no. 2, pp. 141–155, 2021.
- [131] A. T. Eshghi, S. Lee, H. Jung, and P. Wang, "Design of structural monitoring sensor network using surrogate modeling of stochastic sensor signal," *Mechanical Systems and Signal Processing*, vol. 133, p. 106280, 2019.
 - [132] Z. Liu, W. Liu, N. Li, Y. Yu, X. Mao, and Z. Yang, "Design of wireless synchronous structural monitoring system for large bridges," in *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2017* (J. P. Lynch, ed.), vol. 10168, p. 101683A, International Society for Optics and Photonics, SPIE, 2017.
 - [133] S. Valenti, M. Conti, P. Pierleoni, L. Zappelli, A. Belli, F. Gara, S. Carbonari, and M. Regni, "A low cost wireless sensor node for building monitoring," in *2018 IEEE Workshop on Environmental, Energy, and Structural Monitoring Systems (EESMS)*, pp. 1–6, 2018.
 - [134] A. B. Noel, A. Abdaoui, T. Elfouly, M. H. Ahmed, A. Badawy, and M. S. Shehata, "Structural health monitoring using wireless sensor networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1403–1423, 2017.
 - [135] J. Lee, I. Khan, S. Choi, and Y.-W. Kwon, "A smart iot device for detecting and responding to earthquakes," *Electronics*, vol. 8, no. 12, 2019.
 - [136] I. Khan, S. Choi, and Y.-W. Kwon, "Earthquake detection in a static and dynamic environment using supervised machine learning and a novel feature extraction method," *Sensors*, vol. 20, no. 3, 2020.
 - [137] P. Pierleoni, M. Conti, A. Belli, L. Palma, L. Incipini, L. Sabbatini, S. Valenti, M. Mercuri, and R. Concetti, "Iot solution based on mqtt protocol for real-time building monitoring," in *2019 IEEE 23rd International Symposium on Consumer Technologies (ISCT)*, pp. 57–62, 2019.
 - [138] J. Lifshitz and A. Rotem, "Determination of reinforcement unbonding of composites by a vibration technique," *Journal of Composite Materials*, vol. 3, no. 3, pp. 412–423, 1969.
 - [139] Y. Fu, J. Liu, Z. Wei, and Z. Lu, "A two-step approach for damage identification in plates," *Journal of Vibration and Control*, vol. 22, no. 13, pp. 3018–3031, 2016.
 - [140] J.-T. Kim, Y.-S. Ryu, H.-M. Cho, and N. Stubbs, "Damage identification in beam-type structures: frequency-based method vs mode-shape-based method," *Engineering Structures*, vol. 25, no. 1, pp. 57–67, 2003.
 - [141] G. F. Gomes, J. Antonio Carlos Ancelotti, and J. Sebastião Simões da Cunha, "Residual stress prediction in porous cfrp using artificial neural networks," *Composites: Mechanics, Computations, Applications: An International Journal*, vol. 9, no. 1, pp. 27–40, 2018.

- [142] S.-H. Jeong, W.-S. Jang, J.-W. Nam, H. An, and D.-J. Kim, “Development of a structural monitoring system for cable bridges by using seismic accelerometers,” *Applied Sciences*, vol. 10, no. 2, 2020.
- [143] A. Girolami, D. Brunelli, and L. Benini, “Low-cost and distributed health monitoring system for critical buildings,” in *2017 IEEE Workshop on Environmental, Energy, and Structural Monitoring Systems (EESMS)*, pp. 1–6, 2017.
- [144] M. Maizuar, L. Zhang, S. Miramini, P. Mendis, and C. Duffield, “Structural health monitoring of bridges using advanced non-destructive testing technique,” in *ACMSM25* (C. M. Wang, J. C. Ho, and S. Kitipornchai, eds.), (Singapore), pp. 963–972, Springer Singapore, 2020.
- [145] F. Lo Iacono, G. Navarra, and M. Oliva, “Structural monitoring of “himera” viaduct by low-cost mems sensors: characterization and preliminary results,” *Meccanica*, vol. 52, pp. 3221–3236, Oct 2017.
- [146] M. Dumont and D. Wolf, “Usage of mems capacitive acceleration sensors for structural monitoring,” in *Dynamics of Civil Structures, Volume 2* (S. Pakzad, ed.), (Cham), pp. 77–89, Springer International Publishing, 2019.
- [147] P. Cawley and R. D. Adams, “The location of defects in structures from measurements of natural frequencies,” *The Journal of Strain Analysis for Engineering Design*, vol. 14, no. 2, pp. 49–57, 1979.
- [148] A. Aloisio, L. Di Battista, R. Alaggio, and M. Fragiaco, “Sensitivity analysis of subspace-based damage indicators under changes in ambient excitation covariance, severity and location of damage,” *Engineering Structures*, vol. 208, p. 110235, 2020.
- [149] A. Leoni, G. Barile, P. Esposito, R. Paolucci, V. Stornelli, and G. Ferri, “A master multi-slave system based on structural modal frequencies,” *Electronics*, vol. 12, no. 15, 2023.
- [150] Espressif Systems Co. Ltd., “ESP32-WROOM-32 microcontroller unit - datasheet,” 2025. Available at https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf.
- [151] Adafruit Industries LLC, “FGPMMOPA6H GPS - datasheet,” 2011. available at <https://www.digikey.com/htmldatasheets/production/1797888/0/0/1/fgpmmopa6h.html>.
- [152] Analog Devices, Inc., “ADXL354/ADXL355 accelerometer - datasheet,” 2025. Available at https://www.analog.com/media/en/technical-documentation/data-sheets/adxl354_adxl355.pdf.
- [153] Analog Devices, Inc., “ADXL345 accelerometer - datasheet,” 2022. Available at <https://www.analog.com/media/en/technical-documentation/data-sheets/adxl345.pdf>.

- [154] Integrated Silicon Solution Inc., “IS37SML01G1 IS38SML01G1 NAND memory - datasheet,” 2020. Available at <https://www.issi.com/WW/pdf/37-38SML01G1.pdf>.
- [155] Analog Devices, Inc., “RS485 transceiver - datasheet,” 2024. Available at <https://www.analog.com/media/en/technical-documentation/datasheets/MAX13487E-MAX13488E.pdf>.
- [156] Shenzhen Hope Microelectronics Co., Ltd., “RFM95W/96W/98W LoRa transceiver - datasheet,” 2018. Available at <https://cdn.sparkfun.com/assets/a/9/6/1/0/RFM95W-V2.0.pdf>.
- [157] Bel - power, protect, connect, “VX78-500 regulator - datasheet.” Available at <https://www.belfuse.com/products/power-supplies/dc-dc-converters/vx78-500-series>.
- [158] Texas Instruments Inc., “BQ2407x battery charger - datasheet,” 2024. Available at https://www.ti.com/lit/ds/symlink/hd3ss3411-q1.pdf?ts=1752596376474&ref_url=https%253A%252F%252Fwww.ti.com%252Fsitesearch%252Fen-us%252Fdocs%252Funiversalsearch.tsp%253FlangPref%253Den-US%2526nr%253D30%2526searchTerm%253D1+channel+16gbps+2%253A1+mux.
- [159] Texas Instruments Inc., “LM1117 regulator - datasheet,” 2023. Available at https://www.ti.com/lit/ds/symlink/lm1117.pdf?ts=1746370689135&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FLM1117%253Futm_source%253Dgoogle%2526utm_medium%253Dcpc%2526utm_campaign%253Dapp-null-null-GPN_EN-cpc-pf-google-ww_en_cons%2526utm_content%253DLM1117%2526ds_k%253DLM1117+Datasheet%2526DCM%253Dyes%2526gclidsrc%253Daw.ds%2526gad_source%253D1%2526gad_campaignid%253D1767856010%2526gclid%253DCj0KCQjwoNzABhDbARIsALfY8VPqZYCbX0caUPFtcpWgCUSIQaLJpCEn-au2vTKSHEGxXHWU2i1Dc_waAoXeEALw_wcB.
- [160] M. Guerra, C. Volpone, L. Galantucci, and G. Percoco, “Photogrammetric measurements of 3d printed microfluidic devices,” *Additive Manufacturing*, vol. 21, pp. 53–62, 2018.
- [161] W. Xu, S. Huang, D. Han, Z. Zhang, Y. Gao, P. Feng, and D. Zhang, “Toward automated construction: The design-to-printing workflow for a robotic in-situ 3d printed house,” *Case Studies in Construction Materials*, vol. 17, p. e01442, 2022.
- [162] A. Le-Bail, B. C. Maniglia, and P. Le-Bail, “Recent advances and future perspective in additive manufacturing of foods based on 3d printing,” *Current Opinion in Food Science*, vol. 35, pp. 54–64, 2020. Food Engineering & Processing.
- [163] A. Simonovic, E. Rohwer, and T. Stander, “Sla-printed k-band waveguide components using tollens reaction silver plating,” *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 13, no. 2, pp. 230–239, 2023.

- [164] Y. Li, J. Li, M. Zhang, H. Wang, J. Xu, and S. Xiao, “A monolithic stereolithography 3-d printed ka-band spherical resonator bandpass filter,” in *2018 IEEE Radio and Wireless Symposium (RWS)*, pp. 56–59, 2018.
- [165] R. Walczak and K. Adamski, “Inkjet 3d printing of microfluidic structures—on the selection of the printer towards printing your own microfluidic chips,” *Journal of Micromechanics and Microengineering*, vol. 25, p. 085013, jul 2015.
- [166] S. Magdassi, M. Grouchko, O. Berezin, and A. Kamyshny, “Triggering the sintering of silver nanoparticles at room temperature,” *ACS Nano*, vol. 4, pp. 1943–1948, Apr 2010.
- [167] D. J. Roach, C. Roberts, J. Wong, X. Kuang, J. Kovitz, Q. Zhang, T. G. Spence, and H. J. Qi, “Surface modification of fused filament fabrication (fff) 3d printed substrates by inkjet printing polyimide for printed electronics,” *Additive Manufacturing*, vol. 36, p. 101544, 2020.
- [168] A. Roshanghias, M. Krivec, and M. Baumgart, “Sintering strategies for inkjet printed metallic traces in 3d printed electronics,” *Flexible and Printed Electronics*, vol. 2, p. 045002, oct 2017.
- [169] A. H. Espera, J. R. C. Dizon, Q. Chen, and R. C. Advincula, “3d-printing and advanced manufacturing for electronics,” *Progress in Additive Manufacturing*, vol. 4, pp. 245–267, Sep 2019.
- [170] M. R. Khosravani and T. Reinicke, “3d-printed sensors: Current progress and future challenges,” *Sensors and Actuators A: Physical*, vol. 305, p. 111916, 2020.
- [171] M. K. Idris, J. Qiu, G. W. Melenka, and G. Grau, “Printing electronics directly onto carbon fiber composites: unmanned aerial vehicle (uav) wings with integrated heater for de-icing,” *Engineering Research Express*, vol. 2, p. 025022, may 2020.
- [172] D. Espalin, D. W. Muse, E. MacDonald, and R. B. Wicker, “3d printing multifunctionality: structures with electronics,” *The International Journal of Advanced Manufacturing Technology*, vol. 72, pp. 963–978, May 2014.
- [173] D. J. Kang, L. González-García, and T. Kraus, “Soft electronics by inkjet printing metal inks on porous substrates,” *Flexible and Printed Electronics*, vol. 7, p. 033001, aug 2022.
- [174] M. S. Mannoer, Z. Jiang, T. James, Y. L. Kong, K. A. Malatesta, W. O. Soboyejo, N. Verma, D. H. Gracias, and M. C. McAlpine, “3d printed bionic ears,” *Nano Letters*, vol. 13, pp. 2634–2639, Jun 2013.
- [175] J. J. Adams, E. B. Duoss, T. F. Malkowski, M. J. Motala, B. Y. Ahn, R. G. Nuzzo, J. T. Bernhard, and J. A. Lewis, “Conformal printing of electrically small antennas on three-dimensional surfaces,” *Advanced Materials*, vol. 23, no. 11, pp. 1335–1340, 2011.

- [176] S. Vella, C. Smithson, K. Halfyard, E. Shen, and M. Chrétien, “Integrated capacitive sensor devices aerosol jet printed on 3d objects,” *Flexible and Printed Electronics*, vol. 4, p. 045005, dec 2019.
- [177] E. Macdonald, R. Salas, D. Espalin, M. Perez, E. Aguilera, D. Muse, and R. B. Wicker, “3d printing for the rapid prototyping of structural electronics,” *IEEE Access*, vol. 2, pp. 234–242, 2014.
- [178] ISO/ASTM International, “ISO/ASTM 52900:2021 Additive Manufacturing - General Principles - Fundamentals and Vocabulary,” 2021. Available at <https://www.iso.org/obp/ui/#iso:std:iso-astm:52900:ed-2:v1:en>.
- [179] J.-H. Cho, K.-Y. Park, C.-M. Lim, and H.-W. Son, “Design and implementation of an x-band horn antenna with a metamaterial lens using 3d printing technology,” *IEEE Access*, vol. 12, pp. 17773–17781, 2024.
- [180] K. H. Church, N. B. Crane, P. I. Deffenbaugh, T. P. Ketterl, C. G. Neff, P. B. Nesbitt, J. T. Nussbaum, C. Perkowski, H. Tsang, J. Castro, J. Wang, and T. M. Weller, “Multimaterial and multilayer direct digital manufacturing of 3-d structural microwave electronics,” *Proceedings of the IEEE*, vol. 105, no. 4, pp. 688–701, 2017.
- [181] N. ‘A Alias, S. M. Shaharum, and M. S. A. Karim, “Design of a 3d printed uwb antenna for a low-cost wireless heart and respiration rate monitoring,” *IOP Conference Series: Materials Science and Engineering*, vol. 917, p. 012041, sep 2020.
- [182] M. Bozzi, C. Tomassoni, L. Perregini, R. Bahr, and M. Tentzeris, “Additive manufacturing of substrate integrated waveguide components,” in *2016 IEEE MTT-S International Microwave Workshop Series on Advanced Materials and Processes for RF and THz Applications (IMWS-AMP)*, pp. 1–4, 2016.
- [183] I. Piekarz, J. Sorocki, K. Wincza, S. Gruszczynski, and J. Papapolymerou, “Suspended microstrip low-pass filter realized using fdm type 3d printing with conductive copper-based filament,” in *2018 IEEE 68th Electronic Components and Technology Conference (ECTC)*, pp. 2470–2476, 2018.
- [184] S. J. Leigh, R. J. Bradley, C. P. Purssell, D. R. Billson, and D. A. Hutchins, “A simple, low-cost conductive composite material for 3d printing of electronic sensors,” *PLOS ONE*, vol. 7, pp. 1–6, 11 2012.
- [185] J. S. Park, T. Kim, and W. S. Kim, “Conductive cellulose composites with low percolation threshold for 3d printed electronics,” *Scientific Reports*, vol. 7, p. 3246, Jun 2017.
- [186] T. Salo, M. Luukko, A. Halme, and J. Vanhala, “Additive manufactured compliant surface reflectance sensor,” in *2023 IEEE International Conference on Flexible and Printable Sensors and Systems (FLEPS)*, pp. 1–4, 2023.
- [187] M. Arh and J. Slavič, “Single-process 3d-printed triaxial accelerometer,” *Advanced Materials Technologies*, vol. 7, no. 7, p. 2101321, 2022.

- [188] R. Chirila, M. Ntagios, and R. Dahiya, “3d printed wearable exoskeleton human-machine interfacing device,” in *2020 IEEE SENSORS*, pp. 1–4, 2020.
- [189] T. Salo, A. Halme, J. Lahtinen, and J. Vanhala, “Enhanced stretchable electronics made by fused-filament fabrication,” *Flexible and Printed Electronics*, vol. 5, p. 045001, oct 2020.
- [190] N. Lazarus and S. S. Bedair, “Creating 3d printed sensor systems with conductive composites,” *Smart Materials and Structures*, vol. 30, p. 015020, dec 2020.
- [191] C. Dichtl, P. Sippel, and S. Krohns, “Dielectric properties of 3d printed polylactic acid,” *Advances in Materials Science and Engineering*, vol. 2017, no. 1, p. 6913835, 2017.
- [192] E. Huber, M. Mirzaee, J. Bjorgaard, M. Hoyack, S. Noghanian, and I. Chang, “Dielectric property measurement of pla,” in *2016 IEEE International Conference on Electro Information Technology (EIT)*, pp. 0788–0792, 2016.
- [193] G. Boussatour, P.-Y. Cresson, B. Genestie, N. Joly, and T. Lasri, “Dielectric characterization of polylactic acid substrate in the frequency band 0.5–67 ghz,” *IEEE Microwave and Wireless Components Letters*, vol. 28, no. 5, pp. 374–376, 2018.
- [194] Protoplant, makers of Protopasta, “Protopasta Conductive PLA,” 2025. Available at <https://proto-pasta.com/pages/conductive-pla>.
- [195] Multi3D, “Electrifi - Highly Conductive 3D Printing Filament,” 2025. Available at <https://www.multi3d11c.com/>.
- [196] Duet3D Limited, “Duet3D,” 2025. Available at <https://www.duet3d.com/>.
- [197] V. Zega, M. Invernizzi, R. Bernasconi, F. Cuneo, G. Langfelder, L. Magagnin, M. Levi, and A. Corigliano, “The first 3d-printed and wet-metallized three-axis accelerometer with differential capacitive sensing,” *IEEE Sensors Journal*, vol. 19, no. 20, pp. 9131–9138, 2019.
- [198] G. Barile, P. Esposito, V. Stornelli, and G. Ferri, “Development and analysis of a multimaterial fdm 3d printed capacitive accelerometer,” *IEEE Access*, vol. 11, pp. 40175–40181, 2023.
- [199] Orca Slicer, “OrcaSlicer - 3D Printing Software,” 2025. Available at <https://orca-slicer.com/>.
- [200] L. Persi, *Progettazione e sviluppo di un sistema di sensori stampato in 3D per il monitoraggio dei movimenti delle dita di una mano*. PhD thesis, Università degli Studi dell’Aquila, 2024. Available at <https://tesionline.univaq.it/handle/20.500.12319/19275?mode=simple>.
- [201] P. Esposito, G. Barile, V. Stornelli, and G. Ferri, “A model for performance evaluation of low-frequency 3d-printed electronic filters,” in *2024 9th International Conference on Smart and Sustainable Technologies (SpliTech)*, pp. 1–5, 2024.

- [202] G. Barile, P. Esposito, A. Possemato, V. Stornelli, and G. Ferri, “Design and manufacture of 3d-printed circuit boards,” in *Proceedings of SIE 2023* (C. Ciofi and E. Limiti, eds.), (Cham), pp. 402–413, Springer Nature Switzerland, 2024.
- [203] R. Puers, “Capacitive sensors: When and how to use them,” *Sensors and Actuators A: Physical*, vol. 37-38, pp. 93–105, 1993. Proceedings of Eurosensors VI.
- [204] G. Barile, G. Ferri, and V. Stornelli, *Electronic Interfaces for Differential Capacitive Sensors*. River Publishers, 2020.
- [205] G. Barile, P. Esposito, V. Stornelli, and G. Ferri, “A fully differential analog read-out circuit for differential capacitive sensors,” *IEEE Sensors Journal*, vol. 25, no. 5, pp. 7763–7770, 2025.
- [206] Analog Devices, Inc., “AD633 analog multiplier - datasheet,” 2015. Available at <https://www.analog.com/media/en/technical-documentation/data-sheets/ad633.pdf>.
- [207] G.-Q. Zhong, R. Bargar, and K. S. Halle, “Circuits for voltage tuning the parameters of chua’s circuit: Experimental application for musical signal generation,” *Journal of the Franklin Institute*, vol. 331, no. 6, pp. 743–784, 1994.
- [208] A. De Marcellis, C. Reig, and M.-D. Cubells-Beltrán, “A capacitance-to-time converter-based electronic interface for differential capacitive sensors,” *Electronics*, vol. 8, no. 1, 2019.
- [209] J. A. Hidalgo-López, “A simple digital readout circuit for differential resistive or capacitive sensors,” *Measurement Science and Technology*, vol. 34, p. 085117, may 2023.
- [210] M. S. Arefin, J.-M. Redouté, and M. R. Yuce, “A mems interface ic with low-power and wide-range frequency-to-voltage converter for biomedical applications,” *IEEE Transactions on Biomedical Circuits and Systems*, vol. 10, no. 2, pp. 455–466, 2016.
- [211] D. Kong and Q. Duan, “A pseudo-differential capacitive mems accelerometer analog front-end design,” *Journal of Physics: Conference Series*, vol. 2477, p. 012033, apr 2023.
- [212] A. Depari, E. Sisinni, A. Flammini, G. Ferri, V. Stornelli, G. Barile, and F. R. Parente, “Autobalancing analog front end for full-range differential capacitive sensing,” *IEEE Transactions on Instrumentation and Measurement*, vol. 67, no. 4, pp. 885–893, 2018.
- [213] X. Li, P. Wang, G. Li, and Y. Zhang, “Design of a novel self-test-on-chip interface asic for capacitive accelerometers,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 70, no. 7, pp. 2834–2843, 2023.
- [214] J. Wang, H. S.-h. Chung, and R. T.-h. Li, “Characterization and experimental assessment of the effects of parasitic elements on the mosfet switching performance,” *IEEE Transactions on Power Electronics*, vol. 28, no. 1, pp. 573–590, 2013.

- [215] M. Rodder, "Effects of channel width and parasitic bipolar action on p-type poly-si resistor characteristics," *IEEE Electron Device Letters*, vol. 12, no. 5, pp. 241–243, 1991.
- [216] J. Tanaka, T. Toyabe, H. Matsuo, S. Ihara, H. Masuda, and F. Otsuka, "Reverse short-channel effect of threshold voltage in locos parasitic mosfets," *Solid-State Electronics*, vol. 38, no. 3, pp. 567–572, 1995.
- [217] J. Raja, K. Jang, C. P. T. Nguyen, N. Balaji, S. Chatterjee, and J. Yi, "Drain-induced barrier lowering and parasitic resistance induced instabilities in short-channel insnzn tfts," *IEEE Electron Device Letters*, vol. 35, no. 7, pp. 756–758, 2014.
- [218] S. Luan and G. W. Neudeck, "An experimental study of the source/drain parasitic resistance effects in amorphous silicon thin film transistors," *Journal of Applied Physics*, vol. 72, pp. 766–772, 07 1992.
- [219] M. Scarsella, G. Barile, V. Stornelli, L. Safari, and G. Ferri, "A survey on current-mode interfaces for bio signals and sensors," *Sensors*, vol. 23, no. 6, 2023.
- [220] S. Jamuar and S. Rajput, "Low-voltage current-mode analog circuit structures and their applications," in *15th Annual IEEE International ASIC/SOC Conference*, pp. 477–478, 2002.
- [221] A. I. Ozkan, B. Metin, N. Herencsar, and O. Cicekoglu, "Mos-only current-mode analog signal processing functional cores," in *2019 IEEE 62nd International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 917–920, 2019.
- [222] P. Esposito, G. Barile, G. Ferri, and V. Stornelli, "A rail-to-rail second generation voltage conveyor for low-power applications," in *2024 19th Conference on Ph.D Research in Microelectronics and Electronics (PRIME)*, pp. 1–5, 2024.
- [223] K. G. Dunn, "Process improvement using data, release 46e2b8," Feb 2025.
- [224] R. Fitch and V. Howell, "Novel principle of transient high-voltage generation," *Proceedings of the Institution of Electrical Engineers*, vol. 111, pp. 849–855, 1964.
- [225] G. Bauer and H. Heinrich, "A low-impedance high-voltage nanosecond pulser," *Journal of Physics E: Scientific Instruments*, vol. 1, p. 688, jun 1968.
- [226] J. D. Cockcroft, E. T. S. Walton, and E. Rutherford, "Experiments with high velocity positive ions.—(i) further developments in the method of obtaining high velocity positive ions," *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, vol. 136, no. 830, pp. 619–630, 1932.
- [227] D. J. Thrimawithana and U. K. Madawala, "A generalized mathematical model for vector inversion generators," in *2008 34th Annual Conference of IEEE Industrial Electronics*, pp. 429–434, 2008.

- [228] T.-L. Su, Y.-M. Zhang, S.-W. Chen, Y.-T. Liu, H.-Y. Lv, and J.-T. Liu, “A 600kv 15ma cockcroft–walton high-voltage power supply with high stability and low-ripple voltage,” *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 560, no. 2, pp. 613–616, 2006.
- [229] M. Rentzsch, F. Gleisberg, H. Guldner, F. Benecke, and C. Ditmanson, “Closed analytical model of a 20 kv output voltage, 800 w output power series-parallel-resonant converter with walton cockroft multiplier,” in *2008 IEEE Power Electronics Specialists Conference*, pp. 1923–1929, 2008.
- [230] A. Rajaei, I. Dehghanian, M. Shahparasti, and E. Poursmaeil, “Behavioral switching model for current-fed cockcroft–walton voltage multiplier,” *Journal of Power Electronics*, vol. 20, pp. 365–375, Mar. 2020.
- [231] S.-H. Son, T.-H. Kim, C.-H. Kwon, S.-R. Jang, C.-H. Yu, and H.-S. Kim, “Lumped parameter model of cockcroft-walton voltage multiplier in resonant converters,” in *2023 11th International Conference on Power Electronics and ECCE Asia (ICPE 2023 - ECCE Asia)*, pp. 2681–2687, 2023.
- [232] R. Risch and J. Biela, “Solid-state marx generator vs. linear transformer driver: Comparison of parasitics and pulse waveforms for nanosecond pulsers,” in *2021 IEEE Pulsed Power Conference (PPC)*, pp. 1–12, 2021.
- [233] A. Rajabi-Nezhad, A. A. Razi-Kazemi, M. Soheyli, and F. Malekipour, “Assessment of different compact marx generator structures on the voltage profile,” *IEEE Transactions on Plasma Science*, vol. 48, no. 1, pp. 189–195, 2020.
- [234] H. R. Mirzaei, F. Bayat, and K. Miralikhani, “A semi-analytic approach for determining marx generator optimum setup during power transformers factory test,” *IEEE Transactions on Power Delivery*, vol. 36, no. 1, pp. 10–18, 2021.
- [235] H. Jiang, J. Zhou, Y. Huang, Y. Zeng, J. Humphries, Q. Xu, C. Song, L. Xing, Z. Fei, and J. Nalborczyk, “Design of a marx generator for hemp filter evaluation taking account of parasitic effect of components,” in *2016 International Symposium on Electromagnetic Compatibility - EMC EUROPE*, pp. 839–843, 2016.
- [236] H. Canacsinh, L. M. Redondo, F. F. Silva, and E. Schamiloglu, “Modeling of a solid-state marx generator with parasitic capacitances for optimization studies,” in *2011 IEEE Pulsed Power Conference*, pp. 1422–1427, 2011.
- [237] L. Lamy Rocha, H. Canacsinh, J. F. Silva, L. M. Redondo, and T. Luciano, “Modeling marx generators for maximum pulse repetition rate estimation,” in *2017 IEEE 21st International Conference on Pulsed Power (PPC)*, pp. 1–4, 2017.
- [238] H. Canacsinh, L. M. Redondo, J. F. Silva, and E. Schamiloglu, “Solid-state bipolar marx modulator modeling,” *IEEE Transactions on Plasma Science*, vol. 42, no. 10, pp. 3048–3056, 2014.

- [239] P. Gjurovski, L. Hüssen, and R. Negra, “Utilizing a high-speed marx supply modulator in an amplitude-modulated mixerless transmitter for 64 qam,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 69, no. 1, pp. 1063–1071, 2021.
- [240] D. Senthil Kumar, S. Khandekar, S. Nekkenti, M. Kumar Das, U. Shanmuganathan, and S. K. Nayak, “Design, development and characterization of blumlein pulse forming line based pulsed power system for high power microwave source research,” *Next Energy*, vol. 3, p. 100110, 2024.
- [241] A. Haddad and D. Warne, *Advances in High Voltage Engineering*. The Institution of Engineering and Technology, 2004.

Tesi redatta con il cofinanziamento dell'Unione europea-*NextGeneration EU*
Piano Nazionale di Ripresa e Resilienza Missione 4, Componente 2 –
Investimento 3.3. del PNRR – “Dottorati innovativi che rispondono ai fabbisogni di innovazione delle
imprese” – CUP E11I22000210001