
Enviromental monitoring at the Montefiore institute thanks to a LoRa network

Auteur : Leroy, Zoé

Promoteur(s) : Redouté, Jean-Michel

Faculté : Faculté des Sciences appliquées

Diplôme : Master : ingénieur civil électricien, à finalité spécialisée en "signal processing and intelligent robotics"

Année académique : 2022-2023

URI/URL : <http://hdl.handle.net/2268.2/17690>

Avertissement à l'attention des usagers :

Tous les documents placés en accès ouvert sur le site le site MatheO sont protégés par le droit d'auteur. Conformément aux principes énoncés par la "Budapest Open Access Initiative"(BOAI, 2002), l'utilisateur du site peut lire, télécharger, copier, transmettre, imprimer, chercher ou faire un lien vers le texte intégral de ces documents, les disséquer pour les indexer, s'en servir de données pour un logiciel, ou s'en servir à toute autre fin légale (ou prévue par la réglementation relative au droit d'auteur). Toute utilisation du document à des fins commerciales est strictement interdite.

Par ailleurs, l'utilisateur s'engage à respecter les droits moraux de l'auteur, principalement le droit à l'intégrité de l'oeuvre et le droit de paternité et ce dans toute utilisation que l'utilisateur entreprend. Ainsi, à titre d'exemple, lorsqu'il reproduira un document par extrait ou dans son intégralité, l'utilisateur citera de manière complète les sources telles que mentionnées ci-dessus. Toute utilisation non explicitement autorisée ci-avant (telle que par exemple, la modification du document ou son résumé) nécessite l'autorisation préalable et expresse des auteurs ou de leurs ayants droit.



UNIVERSITY OF LIÈGE

SCHOOL OF ENGINEERING AND COMPUTER SCIENCE

**Environmental monitoring at the Montefiore Institute
thanks to a LoRa network**

*Master's thesis completed in order to obtain the degree of
Master of Science in Electrical Engineering*



Author:
Zoé LEROY

Supervisor:
Pr. Jean-Michel REDOUTÉ

Academic year 2022-2023

UNIVERSITY OF LIÈGE

School of Engineering and Computer Science

**Environmental monitoring at the Montefiore Institute
thanks to a LoRa network**

Zoé LEROY

Supervised by Pr. Jean-Michel REDOUTÉ

Academic year 2022-2023

Abstract

In a world where everything becomes automated and digital, the development of connectivity technologies is a key element. In this context, the emergence of wireless communication devices revolutionizes the possibilities that are offered, especially for environmental monitoring. These devices are part of a wider active research topic called Internet Of Things (IoT).

This work aligns with this overall context. It aims at creating an autonomous network communicating via a LoRa (long-range) wireless protocol to monitor indoor environmental parameters. In particular, the goal is to create, from scratch, several standalone nodes that sense their environment and transmit the obtained data to a central gateway. This gateway finally shares the obtained data with a network for user display.

The thesis begins with the development of a simplified network. This first step aims at developing the main elements that can then be used in the global network. In this first part, several sensors, interfaced with a development board, transmit data via wireless communication to the gateway. This part also includes a complete description of the choices that have been made regarding communication protocols and sensors.

The work then focuses on the creation of autonomous nodes. This aspect is related to the autonomous supply voltage of the nodes via indoor solar panels as well as the creation of the node antenna for data transfer.

The remaining part of this thesis is dedicated to the description of the nodes. In particular, this section explains how they have been designed and implemented, both from a software and hardware point of view. The final section then describes how the gateway sends data to the created network.

Acknowledgment

I would like to first thank Pr. Jean-Michel Redouté, for his support and punctual advice during the project.

Then, I also want to express my gratitude to Mr. Morgan Diepart for his follow-up and expertise. His help in the practical development of this work was a key element. In the same way, I want to extend my gratitude to Mr. Philippe Laurent for his valuable advice.

I also want to thank Mr. Samuel Dricot for his time assembling the boards. I am also thankful to Mr. Gabriel Digregorio for the review of this document. More generally, I thank the entire Microsys laboratory team, for all the equipment and the good atmosphere.

Finally, I would like to thank my parents and friends for their encouragement and support.

To all of them, Thank you!

Contents

1	Introduction and project description	1
1.1	Internet of Things	1
1.2	Project description	2
2	Technology choices and state-of-the-art	5
2.1	Communication technology	5
2.1.1	Overview of communication technologies	5
2.1.2	LoRa technology description	9
2.1.3	From LoRa to LoRaWAN	12
2.2	Microcontroller selection	13
2.3	Gateway	14
3	A simplified network	17
3.1	LoRa communication implementation	17
3.1.1	Microcontroller	17
3.1.2	Gateway implementaion	21
3.2	Communication test	22
3.3	Sensors selection and implementation	23
3.3.1	Temperature and relative humidity sensor	25
3.3.2	Ambient light sensor	28
3.3.3	CO ₂ level sensor	30
3.3.4	Sound volume sensor	32
3.3.5	Conclusion and general comment	33
3.4	Conclusion	34
4	Energy management	35
4.1	Energy supply devices	35
4.1.1	Energy storage	35
4.1.2	Selection of the supply voltage	36
4.1.3	Energy manager	37
4.1.4	Schematic and connections	40
4.1.5	Indoor solar panels	41
4.2	Energy requirement estimation	46
4.3	Communication scheduling	46
5	Communication design: Antenna	47
5.1	Slight overview of antenna technology	47
5.2	Impedance matching	48
5.3	Creation of an antenna on a PCB	49
5.3.1	Antennas types	49
5.3.2	PCB antennas	50
5.3.3	PCB design	51
5.4	Performance analysis	54
5.4.1	VNA calibration	54

5.4.2	Track impedance	55
5.4.3	Meander monopole antenna	56
5.4.4	Compact helical antenna	57
6	Standalone sensing nodes	59
6.1	Supply voltage	59
6.1.1	Connections	59
6.1.2	Battery and backup	61
6.1.3	Stable supply voltage: LC filter design	62
6.2	Microcontroller peripherals	63
6.2.1	Crystal	63
6.2.2	J-link	64
6.2.3	Antenna connection	64
6.3	Final PCB design	66
6.4	Microncontroller implementation	67
6.5	Standalone node testing	68
7	Database and user interface	69
7.1	From the gateway to usable data	69
7.1.1	Identification	69
7.2	User interface	70
8	Conclusion and further work	71
8.1	Conclusion	71
8.2	Further works	72
	Appendices	73
A	Energy management	75
A.1	Energy management chip <i>e-pease</i>	75
B	Smith chart	77
B.1	Smith chart presentation	77
B.2	Using the Smith chart for impedance matching	78
C	Antenna design	81
C.1	Antenna selection	81
C.2	Radiation pattern	81
C.3	Antennas test PCB	84
D	Standalone PCB	85
D.1	MCU pin assignment	85
D.2	Standalone PCB bill of materials	85

List of Figures

1.1	IoT network architecture ¹ . [1]	1
2.1	Up-chirp and down-chirp illustration [2].	10
2.2	LoRa technology unmodulated signal and modulated signal [2].	10
2.3	Message transfer in explicit or implicit mode [3]	10
2.4	LoRa packet example [2].	11
2.5	LoRaWAN architecture [1].	12
2.6	LoRaWAN system diagram [4].	12
2.7	STM32WL55JC1 <i>Nucleo</i> board [5].	14
2.8	Packages of the STM32WL55 chip [6].	14
2.9	TTGO by LILYGO gateway.	14
3.1	Simplified network diagram.	17
3.2	LPTIM1 clock frequency selection.	20
3.3	Assignment of the gateway's MCU pins.	22
3.4	Range test of the simplified network at the Montefiore Institute. red = no reception, orange = regular reception, and green = systematic reception x = fixed position of the node	23
3.5	Design of the test PCB for the sensors.	25
3.6	Temperature and humidity sensor (HS3003) schematic.	26
3.7	Temperature and humidity sensor (HS3003) data fetching protocol.	26
3.8	Ambient light sensor (OPT3001) schematic.	28
3.9	Comparison between the ambient light sensor and the lux meter (493 vs 482.56).	30
3.10	Illustration of the wires making shadows on the ambient light sensor.	30
3.11	CO ₂ level sensor (SCD40) footprint.	31
3.12	CO ₂ level sensor (SCD40) schematic.	31
3.13	Sound volume sensor (T3902) schematic.	33
3.14	Sound volume sensor (T3902) pins.	33
3.15	Sound volume sensor (T3902) data format.	33
4.1	Energy management diagram.	35
4.2	Energy management chip (<i>e-peas</i> AEM10941) schematic.	40
4.3	PowerFilm solar panel, current with respect to the voltage under 200 and 1000 <i>lx</i> .	42
4.4	Panasonic solar panel, current as a function of the voltage, under 200 <i>lx</i> .	43
4.5	Anysolar solar panel, current a function of the voltage, under the sun.	43
4.6	The panel test setup.	44
4.7	Evolution of the voltage of the supercapacitors with respect to time.	44
4.8	Evolution of the energy harvested by panels with respect to time.	45
4.9	Evolution of the energy per unit area harvested by panels with respect to time.	45
4.10	Series, parallel and series-parallel connection of solar panels.	45
5.1	Antenna connection diagram.	47
5.2	Dipole antenna [7].	48
5.3	Monopole antenna [7].	48

5.4	RF transmission line representation.	48
5.5	Meander monopole antenna design.	50
5.6	Compact helical antenna design.	50
5.7	Meander monopole antenna schematic.	51
5.8	Compact helical antenna schematic.	51
5.9	Sectional view of a microstrip.	52
5.10	PCB stack-up [8].	53
5.11	Design of the test PCB for the antennas.	54
5.12	Calibration kit.	55
5.13	Impedance of an open circuit after calibration of the VNA.	55
5.14	Impedance of a short circuit after calibration of the VNA.	55
5.15	Impedance of a 50 Ω load after calibration of the VNA.	55
5.16	Impedance of the 607 mm long track in the Smith chart	55
5.17	Meander monopole antenna S11 with no cut in the antenna.	56
5.18	Meander monopole antenna S11 with a cut that shortens the antenna by 16 mm.	56
5.19	Theoretical meander antenna cut.	57
5.20	Practical meander antenna cut.	57
5.21	Compact helical antenna S11 with no cut in the antenna.	57
5.22	Compact helical antenna S11 with 5 lines cut off.	58
5.23	Compact helical antenna S11 with 2 lines scratch off.	58
6.1	Standalone sensing node diagram.	59
6.2	Power scheme of STM32WL MCUs [6].	60
6.3	Battery level feedback: schematic.	61
6.4	<i>E-peas</i> output voltage with no load.	62
6.5	<i>E-peas</i> output voltage with a 1200 Ω load (AC coupling mode).	62
6.6	<i>E-peas</i> output voltage with a 120 Ω load (AC coupling mode).	62
6.7	HSE and LSE crystals: reference.	63
6.8	ST-link: symbol and useful pins.	64
6.9	Antenna schematic for low-power transmission and reception modes.	65
6.10	Standalone node: final PCB design. <i>KiCad</i> visualization (left), blank PCB (center), PCB with soldered components (right)	67
7.1	Network diagram.	69
7.2	Example of the data display thanks to the dashboard with fake data.	70
A.1	Simplified schematic view of the <i>e-peas</i> chip AEM10941	75
A.2	Solar panel testing setup thanks to the <i>e-peas</i> evaluation board.	75
B.1	Smith chart for inductance representation (left) and admittance representation (right).	77
B.2	Representation of the motion of a point on the Smith chart depending on the added component (C or L) in series or in parallel.	79
B.3	Example of impedance matching when the point is on the left of the Smith chart.	79
B.4	Example of impedance matching when the point is on the right of the Smith chart.	80
C.1	<i>Texas Instruments</i> antenna selection quick guide.	81
C.2	Axis orientation [9].	82
C.3	Meander monopole antenna, radiation patterns. Left: theta = 0, phi = 0; Right: theta = 180, phi = 0	82
C.4	Meander monopole antenna, radiation patterns. Left: theta = 90, phi = 0; Right: theta = 90, phi = 180	82
C.5	Meander monopole antenna, radiation patterns. Left: theta = 90, phi = 270; Right: theta = 90, phi = 90	83
C.6	Compact helical antenna, radiation patterns. Left: theta = 0, phi = 0; Right: theta = 180, phi = 0	83

C.7	Compact helical antenna, radiation patterns. Left: $\theta = 90$, $\phi = 0$; Right: $\theta = 90$, $\phi = 180$	83
C.8	Compact helical antenna, radiation patterns. Left: $\theta = 90$, $\phi = 270$; Right: $\theta = 90$, $\phi = 90$	84
C.9	Antenna test setup thanks to the VNA.	84

List of Tables

2.1	Communication protocols: frequency, bandwidth, and modulation comparison.	6
2.2	Communication protocols: data rate, payload, and messages limitation comparison. . .	7
2.3	Communication protocols: urban and rural ranges comparison.	8
2.4	Communication protocols: TX and RX power consumption comparison.	8
2.5	Communication protocols: security comparison.	9
2.6	Communication protocols: summary. v = good characteristics, \pm = medium interest, x = less accurate for the application	9
2.7	Microcontroller characteristics.	13
3.1	LoRa related pins.	21
3.2	Temperature and relative humidity sensors comparison.	25
3.3	MCU pin assignment for the temperature and humidity sensor (HS3003).	27
3.4	Ambient light sensors comparison.	28
3.5	MCU pin assignment for the ambient light sensor (OPT3001).	29
3.6	CO ₂ level sensors comparison.	30
3.7	Sound volume sensor characteristics.	32
3.8	MCU pin assignation for the microphone (T3902).	33
4.1	Comparison between batteries and supercapacitors.	36
4.2	Maximum transmit output power depending on the supply voltage.	37
4.3	Current consumption depending on the power output and the supply voltage.	37
4.4	Voltage range and maximum current outputted by HVOUT and LVOUT.	39
4.5	Threshold voltage values with respect to the value of CFG[2:0].	39
4.6	Indoor solar panels technical comparison.	41
4.7	Indoor solar panels: theoretical performances comparison.	42
4.8	Energy requirement estimation.	46
5.1	Antenna types comparison.	50
5.2	Meander monopole antenna and compact helical antenna characteristics [10, 11]. . . .	50
6.1	Battery level feedback: pin assignment.	61
6.2	HSE and LSE crystals: reference.	63
6.3	ST-link: pin assignment.	64
6.4	Antenna mode configuration depending on the CTRL_antenna pin value.	64
D.1	STM32WLE5C8 pin assignment	85
D.2	Bill Of Materials (BOM) for a standalone sensing node	86

Chapter 1

Introduction and project description

In today's rapidly evolving world, the degree of connectivity continues to surge. Children are almost born with a phone in their hands. Laptops are everywhere in amphitheaters. Athletes rely on connected watches to follow their performances. Inspired by this trend, people began to question why objects and homes couldn't also be connected. This is the beginning of IoT - Internet of Things.

Based on this concept, the objective of this project is to create an autonomous network to monitor the evolving environmental parameters. Nodes will be embedded sensors and will send the data acquired thanks to the wireless LoRa communication protocol. The data acquired will give information about human presence or not. It gives a better idea of how rooms are occupied. This application will be particularized at the Montefiore Institute (or B28). This building is owned by ULiege University. However, the design will be adaptable to any environment.

1.1 Internet of Things

The application of the Internet of Things can be extended to any object. Either the software and sensors are already integrated into the object, or they are added afterward. Those objects can be watches, but also cars, security cameras, healthcare monitors... The data collected is then sent to a network. This data transfer may be wireless or not depending on the application and the communication protocol. The structure of an IoT application is based on a network. The latter is composed of nodes (the "objects") that send data to gateways. A gateway generally receives data from multiple nodes. The gateways themselves transfer the data to a network server. Therefore, they can be accessed from a phone or any other device. A network structure is illustrated in Figure 1.1.

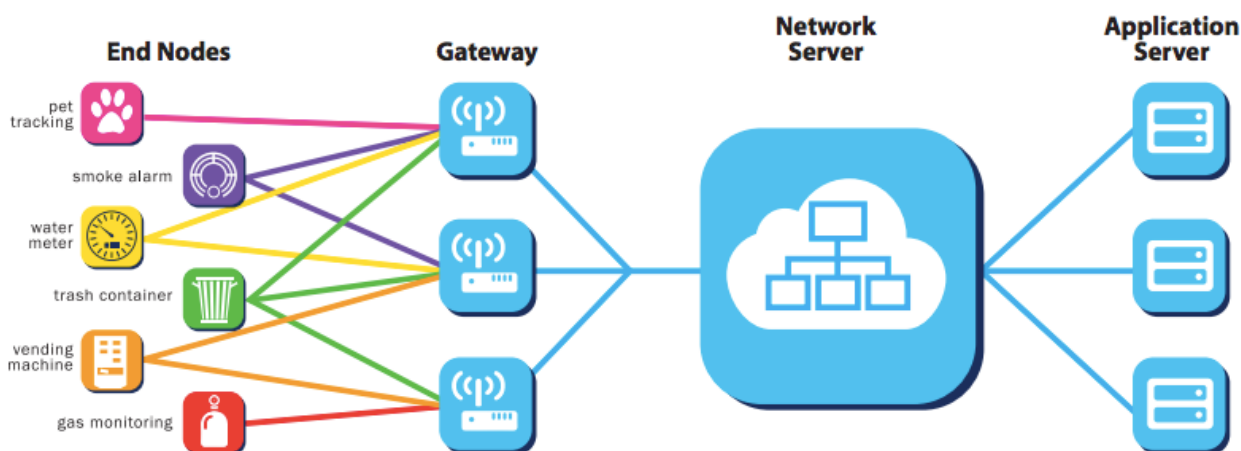


Figure 1.1: IoT network architecture¹. [1]

¹In this work, image references are provided in the general bibliography appended at the end of this document.

A particular type of network is the LPWAN. It stands for Low Power Wide Area Network. Because they cover wide areas, they are also wireless networks. The second part is the "low power" characteristic. In other words, nodes ideally have low energy consumption required for sending data and therefore are really easy to maintain. Low power depends on the modulation techniques but also on the way the network is used. For instance, is it required to stay connected or not,... The challenge is to combine wide area with low power. Nodes are usually battery-operated. In that case, for the battery to last as long as possible, the application must be as low power as possible. Sometimes the system is also powered by solar panels and is autonomous if their consumption does not exceed the production. And finally, nodes can be plugged into the main supply which has no interest in the low-power part. Those advantages make LPWANs very popular in a various range of applications. There are multiple types of LPWANs. They mainly differentiate due to the communication technology they use. Various of those technologies will be discussed in Section 2.1.

1.2 Project description

The goal of this thesis is to create a low-power wide area network in a building and in particular at the Montefiore Institute. The principle is to place a node in every classroom in order to detect whether the room is occupied or not. This realization presents an opportunity for optimization and resource allocation. For instance, if it determines that a room is almost never used, it is a sign that its dedication must be reevaluated and upgraded: maybe for a student group looking for additional space or an assistant requiring a dedicated room.

In order to achieve this goal, several aspects have to be investigated. This work tries to cover those different aspects, from node creation to gateway data processing via radio frequency (RF) communication.

The first part of the work deals with node creation. It describes how to detect occupancy in practice. Five monitoring parameters have been first selected. The idea was then to combine the values of the different data to ensure reliable detection. Finally, each node includes sensors for the temperature, relative humidity, and ambient light level. A CO₂ sensor was under consideration but is dismissed due to a too-high current, and high energy consumption.

The data provided by those different sensors then have to be sent to a central gateway. In the first step, a simplified network is built. The data is collected from the sensors. Which are connected and powered by a computer that acquires the data. It makes it really easy to use. The board also has a LoRa module integrated. The radio frequency signal is then transferred thanks to the integrated antenna. This simplified network is described, in detail, in Chapter 3.

The data is acquired by a gateway. Unlike the nodes, the gateway is more flexible in terms of location. The only constraint is that it must be sufficiently close to the nodes to correctly receive the messages. In practice, placing the gateway at the center of the nodes it covers is a good practice. Due to that flexibility, the gateway can be placed near the main supply and therefore does not need to be low energy.

The gateway in this application has a WiFi connection which is used to transmit the data to a database. The database is handled by a *Raspberry Pi*. This computer hosts a web server for data visualization.

Even if this configuration is interesting and may work correctly, using a *Nucleo* board for each node is not an option as it will be explained later on in this work. In a second approach, a custom board will be designed for the microcontroller and the rest of the system. This comes with several constraints among which is the power supply which is therefore not ensured by a computer as previously. The node, in this second version, is also designed to be energy self-sufficient. To achieve this objective, indoor solar panels are added to each node. A power management module is also used for good management of the energy provided by the photovoltaic cell. The module charges a supercapacitor, and, thanks to the energy stored, it outputs a tunable supply voltage for the node. The practical design of those

Nucleo-free nodes is detailed in Chapter 6.

The last building block between a functional node and a correct gateway is radio frequency communication. This wireless data transfer requires an antenna. Although it was embedded in the *Nucleo* board in the first approach, the antenna is not provided in the standalone node described here above. To ensure data transfer, a new antenna needs to be designed on every created node. This antenna design comes with significant requirements both in terms of hardware and software. Details about antenna design are given in Chapter 5 of this work.

In a few words, this work consists in designing a network, from scratch. It deals with every step of the design by starting with node creation (software, sensor choice) using an MCU development board. It then provides an overview of energy management for low-power nodes. The remaining part of the work is dedicated to the creation of nodes that get rid of the development board (antenna, MCU supply, etc.). The node must be as cheap and as small as possible. This work is finally concluded by regrouping the different elements into a single functional node that interacts in a built network.

Chapter 2

Technology choices and state-of-the-art

Designing an LPWAN implies several choices among which the communication protocol, the MCU, and the gateway used. This chapter aims at providing information to the user about the choices that have been made in this project. One will compare the different possibilities that exist nowadays and the complete process to find a good combination.

2.1 Communication technology

This section deals with the choice of the wireless communication protocol. Firstly, an overview to present multiple existing technologies. Then a more detailed analysis of the selected protocol.

2.1.1 Overview of communication technologies

To select a suitable low-power and wide-area technology, 6 protocols are compared. Other technologies exist, such as ANT, Z-wave, Zigbee RF4CE, IrDA,... it is therefore not an exhaustive list. Firstly, they are briefly presented, then they will be deeply compared on various parameters [1, 12, 13, 14, 15, 16, 17, 18].

- **LTE-M:**

LTE-M stands for Long-Term Evolution Machine. It is derived from the LTE standard for 4G technology as developed by the 3GPP (3rd Generation Partnership Project) organization. The technology is thus licensed by 3GPP which makes it more costly. On the other hand, the technology is based on the cellular existing network. It allows the reuse of the LTE installed base. Furthermore, it benefits from the security of mobile networks, such as authentication. The LTE-M modems follow the LTE architecture and are therefore complex. The major characteristic of LTE-M is its really high data rate which is linked with high power consumption. It is used for high-risk systems and systems with video images.

- **NB-IoT:**

NB-IoT stands for Narrow Band Internet of Things. It is a subset of the LTE standard with a limited bandwidth. As a consequence, it also uses the cellular network. However, the LTE architecture has few changes to limit the bandwidth which makes modems less complex. On the other hand, the availability in rural areas is compromised and the security is a bit decreased but it is still recommended for critical networks. It is overall less complete than LTE-M but uses less power.

- **BLE:**

BLE stands for Bluetooth Low Energy. Transmits data in the 2.4 GHz frequency band. It is an alternative to Bluetooth technology for low-power applications. Together, they are designed to satisfy the needs of every developer. It uses 40 channels (2 MHz spacing) compare to the 79 channels of the classic Bluetooth technology (1 MHz spacing). However, it does not really covers wide areas, no more than a hundred meters.

- **Zigbee:**

The Zigbee name refers to the waggle dance of honey bees after their return to the beehive. Zigbee is less expensive than previous technologies. It uses a 128-bit symmetric encryption key for security. Zigbee follows the IEEE standard 802.15.4 which defines the physical layer (or layer 1) and media access control (MAC layer) of the RF package. It is medium-range and therefore not wide area (hundred meters).

- **SigFox:**

Sigfox is using the free unlicensed spectrum with an ultra-narrow band (100 Hz). The number of messages per day and their size are limited. It takes around 10 ms to send a bit. It uses the DBPSK (differential binary phase-shift keying) and the GFSK (Gaussian frequency shift keying) depending on the direction of the transmission. The network is deployed in 72 countries. Due to financial issues, it has recently been repurchased by *UnaBiz* (04/22).

- **LoRa:**

The LoRa stands for "Long Range". This protocol defines the physical layer as well as the modulation. It uses a chirp spread spectrum modulation. It may be associated with the LoRaWAN protocol which specifies the network's architecture. Furthermore, LoRa transceivers are exclusively manufactured by *Semtech*. Those can however be integrated into any device. LoRaWAN networks can be public (costly) or private. It does provides an encryption technology for security.

Frequency, bandwidth, and modulation technology

The first points of comparison are the frequency, the bandwidth, and the modulation technique. Those values are presented in Table 2.1.

	LTE-M		NB-IoT	BLE
Frequency	800-900 MHz		800-900 MHz	2.4 GHz
Bandwidth	1.08 MHz		180-200 kHz	<100 kHz
Modulation	QPSK		OFDM (downlink) and SC-FDMA (uplink)	GFSK
	Zigbee		Sigfox	LoRa
Frequency	868 MHz	2.4 GHz	868 MHz	868 MHz
Bandwidth	2 MHz	16 · 2 MHz	100 Hz	125 or 250 kHz
Modulation	BPSK	OQPSK	BPSK (uplink) and GFSK (downlink)	CCS

Table 2.1: Communication protocols: frequency, bandwidth, and modulation comparison.

As mentioned previously LTE-M and NB-IoT use the same cellular licensed frequencies. The others use either 868 MHz or 2.4 GHz. Those are unlicensed frequency bands. In particular, they use the industrial, scientific, and medical (ISM) radio band. Using unlicensed bands makes the application cheaper, which is an advantage. However, European restrictions state that an application cannot occupy the public bands more than 1% of the time.

On one hand LoRa, Sigfox, and Zigbee can work at 868 MHz. This frequency is actually available in Europe while the corresponding in the USA is 915 MHz. On the other hand, BLE and Zigbee work at 2.4 GHz. This is not exhaustive, for instance, Lora can work at mainly any available frequency, but the value given is compatible with the LoRaWAN protocol.

The nodes and the gateway are meant to be placed in a building. Unfortunately, the walls in the buildings are very thick and frequent. The signal is therefore prone to reflection and attenuation. One might prefer the lower frequency for better penetration of the walls. In fact, having a lower frequency is the same as having a bigger wavelength. Therefore the signal becomes harder to stop. Furthermore, it is less sensitive to reflection. Notice that the penetration and reflection are also dependent on the

walls' materials and thickness as well.

The following parameter is the bandwidth. They range from narrow (100 *Hz* for Sigfox) to wide (>1 *MHz* for LTE-M and Zigbee). Having a wide band allows the rejection of narrow noise sources. The transmission is therefore more robust.

The last parameter is the modulation technology. Modulation is the process of encoding data (digital in this case) onto a carrier signal. The modulation techniques are all different and it is not relevant to develop them all. Notice that NB-IoT and Sigfox do not handle the same way uplink communications and downlink communications. From the perspective of the node, uplink is a synonym of TX. The data is transferred from the node to the gateway (up in the hierarchy). In the same way, downlink is equivalent to RX.

Data rate, payload, and messages limitation

Table 2.2 resumes all the data-related values. The first value is the data rate, in other words, the speed at which the data is transferred. The second parameter is the max payload. It is the maximum number of bytes that a message can handle. And finally, the last parameter is the maximum number of messages one can send or receive in one day.

	LTE-M	NB-IoT	BLE
Data rate	1 <i>Mbps</i>	60-200 <i>kbps</i>	1 <i>Mbps</i>
Max payload in bytes	500	1600	255
Max nb msg/day	-	-	-
	Zigbee	Sigfox	LoRa
Data rate	20 <i>kbps</i> (at 868 <i>MHz</i>) 250 <i>kbps</i> (at 2.4 <i>GHz</i>)	100-600 <i>bps</i>	50 <i>kbps</i>
Max payload in bytes	100-120	12 (up) and 8 (down)	255
Max nb msg/day	-	140 (up) and 4 (down)	-

Table 2.2: Communication protocols: data rate, payload, and messages limitation comparison.

Looking at the application, messages are small, and just a few of them will be transferred each day. However, the goal is not to make restrictive choices to let place for future improvements of the network. Therefore having a correct speed transfer and no restriction on the maximum number of messages is preferred.

Data rates are expressed in [bps] which stands for bits per second. The standout value is Sigfox's data rate which is the lowest. This is correlated with the fact that it is also the narrowest bandwidth. However, it still needs to respect the 1% ISM bands usage restriction. Therefore it is more restricted in the number of messages and the size of the message. The data rates of the other technologies are a few orders of magnitude bigger. It is in *kbps* and even in *Mbps*, which is not restrictive at all. Notice that the better the data rate, the lower the time needed to send a message.

The following parameter is the payload size. The lowest by far is once again Sigfox. The payload size is 12 bytes uplink and 8 bytes. This is limiting for the application. In fact, the data from the sensors are sent in one unique message for simplicity and power savings. Concerning the other technologies, their payload max sizes have the same magnitude order which is way sufficient.

Finally, Sigfox is the only technology with a hard limit on the number of messages.

Urban and rural ranges

Table 2.3 provides a comparison of the operating ranges. One notices that the rural range is always greater than the urban range. In fact, in an urban environment, there are way more obstacles for the signal which is more likely to be reflected and attenuated. Notice that the environment in the Montefiore Institute is worse than an urban environment. In fact, it consists of multiple thick walls. Therefore, the practical result will be even worse than the theoretical one presented in the Table. For the application, 500 *m* is a minimum to hope to cover all of the Montefiore Institute. Knowing the conditions of the walls and that other buildings are nearby, one will take the biggest range. This way, if later, they are included in the project, it is not a problem.

	LTE-M	NB-IoT	BLE
Range (rural)	10 <i>km</i>	10 <i>km</i>	100 <i>m</i>
Range (urban)	1 <i>km</i>	1 <i>km</i>	100 <i>m</i>
	Zigbee	Sigfox	LoRa
Range (rural)	100 <i>m</i>	40 <i>km</i>	20 <i>km</i>
Range (urban)	100 <i>m</i>	10 <i>km</i>	5 <i>km</i>

Table 2.3: Communication protocols: urban and rural ranges comparison.

In particular, Zigbee and BLE technologies are not wide enough. Better but not the best are NB-IoT and LTE-M. And finally, LoRa and Sigfox can send messages on distances up to kilometers.

Power consumption

The nodes are autonomous in energy, and RF communication is a big part of the energy consumption. All the technologies presented are low power, however, some are better than others. For a constant voltage, the greater the current, the higher the energy consumption. To link the current with the energy, the following equations are used $E = P \cdot t = \int U \cdot I \cdot dt$. With P the power, U the voltage, I the current, and t the time. Table 2.4 shows the current consumption in TX or RX mode for a 3.3 *V* supply voltage. Caution when looking at the values that the current also depends on the output power of the signal when transmitted. The higher the transmitted power, the higher the energy consumption will be.

	LTE-M	NB-IoT	BLE
TX Power Consumption	380 <i>mA</i>	74-200 <i>mA</i>	10 - 30 <i>mA</i>
RX Power Consumption	53.33 <i>mA</i>	46 <i>mA</i>	10 - 30 <i>mA</i>
	Zigbee	Sigfox	LoRa
TX Power Consumption	33-135 <i>mA</i>	10-50 <i>mA</i>	28 <i>mA</i>
RX Power Consumption	17-31 <i>mA</i>	10 <i>mA</i>	10.5 <i>mA</i>

Table 2.4: Communication protocols: TX and RX power consumption comparison.

One directly notices that RX values are always lower than TX values. Except for BLE which has the same current in both cases. This is not a surprise, as transmitting a signal requires sending it with a lot of power for it to be best transmitted. Whereas the received signal has already lost part of its power.

LTE-M is the worst regarding energy consumption. This is correlated with the fact that it has a greater data rate of 1 *Mbps*. Then NB-IoT and Zigbee follow. And finally, the current consumption of Sigfox BLE and LoRa is of the same order of magnitude.

Price and security

In terms of price, NB-IoT and LTE-M are the more costly because they use licensed bands. While the other technologies use unlicensed bands, they are not free either. In fact, one needs to pay a public network subscription. Notice that LoRa proposes to create your own private network, which does not require any subscription.

The networks also provide security for the data. The security protocol is given in Table 2.5. The standard 3GPP describes a security architecture and procedure. Every technology provides a security mechanism if needed.

	LTE-M	NB-IoT	BLE
Security	3GPP	3GPP	encryption
	Zigbee	Sigfox	LoRa
Security	encryption and authentication	AES 128b	if LoRaWAN: AES 128b

Table 2.5: Communication protocols: security comparison.

Conclusion

Table 2.6 resumes all the data collected. Regarding all the criteria, the LoRa technology is selected. This technology, in fact, has a range of up to 5 *km* in a rural environment. Its consumption is in the tens of *mA* order. LoRa uses a frequency of 868 *MHz* and has no big restriction on data transfer. And finally, the security is not defined by the LoRa protocol but by the optional LoRaWAN protocol. The following section provides a better description of the LoRa technology and explains what LoRaWAN adds to the existing protocol.

	LTE-M	NB-IoT	BLE	Zigbee	Sigfox	LoRa
Frequency	v	v	x	v x	v	v
Data	v	v	v	v	x	v
Range	±	±	x	x	v	v
Power consumption	x	x	v	x	v	v
Security	v	v	v	v	v	±

Table 2.6: Communication protocols: summary.

v = good characteristics, ± = medium interest, x = less accurate for the application

2.1.2 LoRa technology description

Data transfer:

LoRa technology is a proprietary LPWAN technology developed by *Semtech Corp.* This means that any LoRa transceiver module has a *Semtech* chip embedded, whatever the module supplier. LoRa technology uses a chirp spread spectrum modulation (CSSM) to transmit data. Needless to say, it is based on a spread spectrum technique. This method spread the signal in the frequency domain thanks to a big bandwidth. In practice, the application uses LoRa at $f_{center} = 868.000 \text{ MHz}$ and a bandwidth $BW = 125 \text{ kHz}$ ($= f_{high} - f_{low}$).

The "C" of CSSM refers to "chirp". A chirp is a sweep in frequency on the bandwidth. Basic chirp is a ramp: from f_{low} to f_{high} for up-chirp, and from f_{high} to f_{low} for down-chirp. An illustration is shown in Figure 2.1.

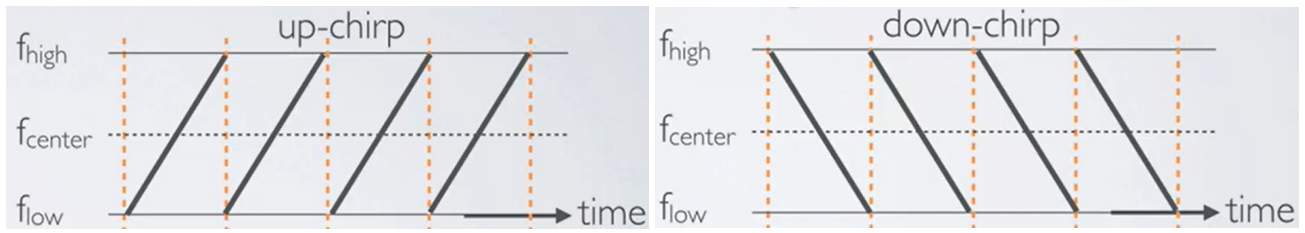


Figure 2.1: Up-chirp and down-chirp illustration [2].

Chips carrying data are cyclically shifted base chirps. The information carried depends on the shift value. The spreading factor SF states how many bits of information are contained in a symbol. In other words how many different values the shift may take. It varies from 7 to 12. In the application, it is set to 7. The symbol rate R_s can be computed thanks to the following formula:

$$R_s = \frac{BW}{2^{SF}} = \frac{125000}{2^7} \approx 1000 \text{ [symbols/s]}$$

With BW the bandwidth. Notice that for a higher spreading factor value, the transmission time for a symbol increases. Figure 2.2 shows an example of modulated signal, based on basic up-chirps.

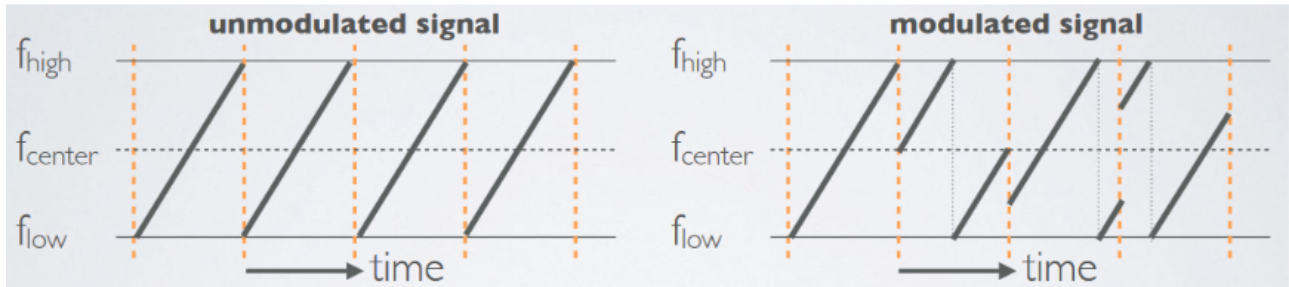


Figure 2.2: LoRa technology unmodulated signal and modulated signal [2].

Complete transmission message

When a message is transmitted, it follows a predetermined packet mode. This makes it possible for the message to be correctly decoded. The packet mode can be explicit or implicit. In the application, it is in explicit mode. The two formats comparison are shown in Figure 2.3 and are presented below.

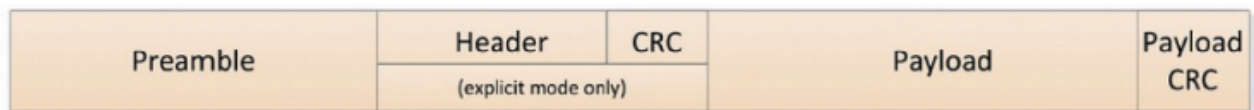


Figure 2.3: Message transfer in explicit or implicit mode [3]

- A packet always starts with a preamble consisting of 8 basic up-chirps, followed by 2 down chirps before the modulated signal. This makes it possible for the decoder to know when a message starts. Furthermore, it has a synchronization purpose. It enables the decoder to align by adjusting the time at which it considers a symbol start. This is possible because the duration of each symbol is constant and well-known.
- In the explicit mode, it contains a header with its CRC (cyclic redundancy check). The header contains information about the payload length, the code rate of the payload, and a possible CRC at the end of the packet. And the CRC is some additional bits for the correction of possible errors. And header and its CRC are coded with a coding rate of 4/8. The coding rate is also an error correction mechanism.

- The payload is the actual useful data, it has a maximum of 255 bytes. It can be coded with various coding rate values (4/5, 4/6, 4/7, or 4/8). For this application, the coding rate is set to 4/5, the default LoRa value.
- And finally an optional CRC can be added. Its coding rate is the same as the header coding rate.

A message encoded following the LoRa protocol is illustrated in Figure 2.4. One recognizes the characteristics 8 up-chirps followed by the 2 down-chirps at the beginning of the message. Then the signal is modulated.

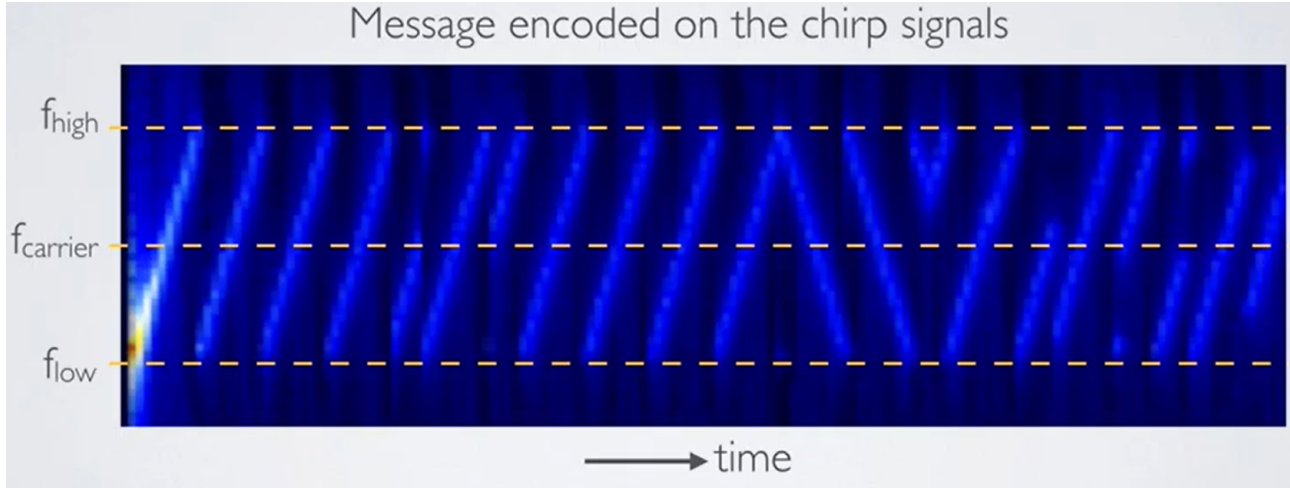


Figure 2.4: LoRa packet example [2].

Time on air

For given values of the spreading factor (SF), coding rate (CR), and signal bandwidth (BW) the total on-the-air transmission time of a LoRa packet can be calculated. The time on air of a LoRa packet has 2 main components: the preamble time and the payload time: $T_{packet} = T_{pr} + T_{pa}$.

First, from the symbol rate, one can compute the time of one symbol T_s .

$$T_s = \frac{1}{R_s} = 1 \text{ ms}$$

Then the preamble time on air is computed thanks to the number of symbols in the preamble $n_{pr} = 8$.

$$T_{pr} = (n_{pr} + 4.25) \cdot T_s = 12.25 \text{ ms}$$

Concerning the payload, one first computes the number of symbols required depending on the number of bytes in the payload.

$$\begin{aligned} n_{pa} &= 8 + \max\left(\text{ceil}\left[\left[\frac{8 \cdot PL - 4 \cdot SF + 28 + 16 \cdot CRC - 20 \cdot IH}{4 \cdot SF}\right] \cdot (CR + 4)\right], 0\right) \\ &= 8 + \max\left(\text{ceil}\left[\left[\frac{8PL - 4 \cdot 7 + 28 + 16 \cdot 4/8 - 20 \cdot 0}{4 \cdot 7}\right] \cdot (4/5 + 4)\right], 0\right) \\ &= 8 + \max\left(\text{ceil}\left[\left[\frac{8PL - 12}{28}\right] \cdot (4/5 + 4)\right], 0\right) \\ &= 8 + \max\left(\text{ceil}\left[\left[\frac{2PL - 3}{7}\right] \cdot 4.8\right], 0\right) \end{aligned}$$

With,

- PL: the number of bytes in the payload
- SF = 7: the spreading factor
- IH = 0: states if the header is disabled (0 = header, 1 = no header)
- CR = 4/8: the coding rate

And finally, the time on air of the payload symbols is computed as follows.

$$T_{pa} = n_{pa} \cdot T_s$$

2.1.3 From LoRa to LoRaWAN

The LoRaWAN protocol uses the LoRa physical layer description. It describes the MAC (media access control) layer on top of the LoRa modulation as shown in Figure 2.5

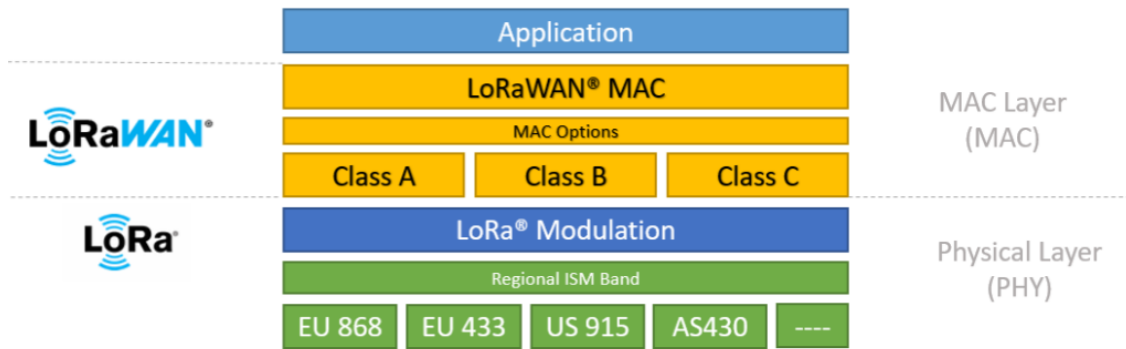


Figure 2.5: LoRaWAN architecture [1].

LoRaWAN defines 3 classes. Class A devices support RX and TX communications. The node can send messages at any time, after uplink transmission, two reception windows are opened by the node. Class B and Class C are extensions of Class A.

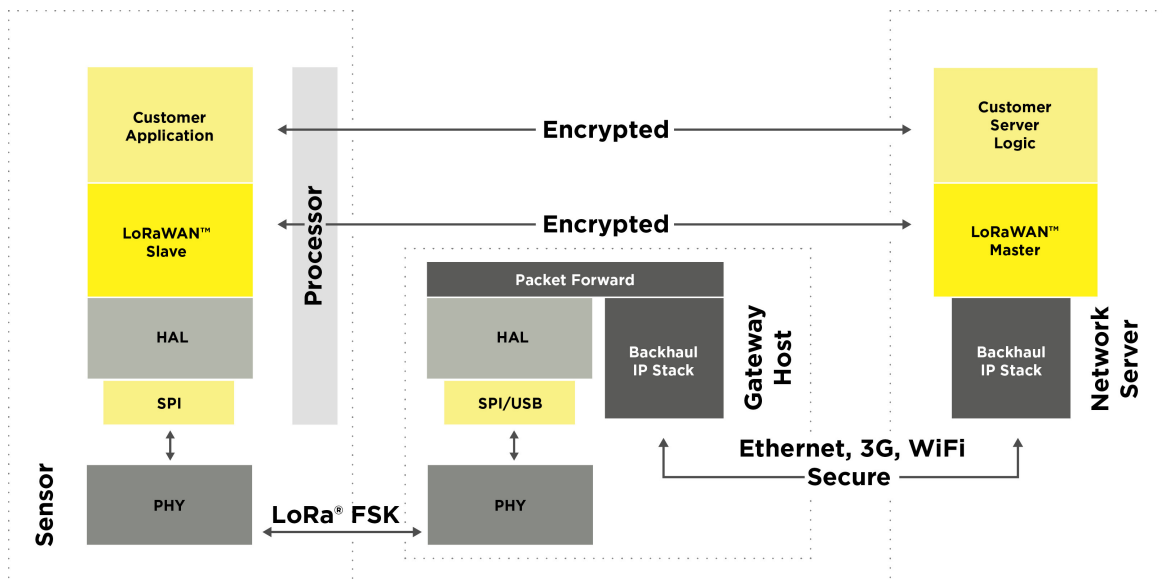


Figure 2.6: LoRaWAN system diagram [4].

Figure 2.6 shows the entire LoRaWAN communication. One notices that the scheme is already more complex. To use the technology, a LoRaWAN network server in addition to the gateway is required. The most known one is The Things Network. But others such as LORIoT or ResIoT exist. Generally, they are free for a limited number of gateways and nodes. Thanks to the network server, the data encryption and device authentication are directly done.

By simplicity, the application will use the **LoRa protocol** and not the LoRaWAN protocol. A private network will be created for the gateway to send the data to. One has to notice that the data is thus not encoded when send. The data is not critical, therefore, it is not an issue. If it becomes an issue, one must integrate it into the LoRa payload construction, or it is still possible with some slight changes to transition to a LoRaWAN network.

2.2 Microcontroller selection

In the previous section, the LoRa technology is selected as the communication protocol. Therefore, the following step is to find the correct hardware for the application. The node needs to be able to transmit and receive in LoRa.

To do so, either each node has a microcontroller with an embedded LoRa chip. Or each node has a microcontroller and a separate LoRa module. Table 2.7 has a proposition for each solution. Prices, in this table, and every other table, are the selling prices on *Mouser* in June 2023. Notice that those prices vary considerably.

Manufacturer	Type	Device name	Price [€]	Vdd [V]	Memory [kb]		Clock frequency
					flash	SRAM	
<i>STM</i>	chip	STM32WL55JC16	12.64	1.8-3.6	256	64	32 MHz
	chip	STM32WLE5C8	8.69	1.8-3.6	256	64	32 MHz
	board	<i>Nucleo</i> -STM32WL55J1	43.58				
<i>Arduino</i>	board	MKR WAN 1300	39.96	5	256	32	48 MHz
		SAMD21 μ C and CMWX1ZZABZ LoRa module					

Table 2.7: Microcontroller characteristics.

The first proposition is from *STM* (or *STMicroelectronics*). It has a series of MCU with a LoRa *Semtech* device integrated. Furthermore, this MCU is available on a *Nucleo* board which is represented in Figure 2.7. This is really convenient for the development process. LEDs and buttons are already integrated, easy access to the pins thanks to the connectors, and a SMA-connected antenna.

The second proposition is the *Arduino* board. The board embedded a MCU chip and a separate LoRa module. This solution enables a large combination between different Lora modules and MCUs. However, having only one chip for both the core and the LoRa module is really convenient. Moreover, STM32 MCUs are easily coded in *C* thanks to the STM32CubeIDE. It is also a well-known performant microcontroller family already used by Microsys lab, it is, therefore, the chosen solution.

It exists in 3 different packages: UFQFPN48, WLCSP59, and UFBGA73. Those packages are illustrated in Figure 2.8. One notices pins under the chip for the two last types. They are therefore way harder to solder. The UFQPN48 package is selected. Notice that it is a bit less compact, but the real drawback is that not all pins are accessible. Only 48 pins compared to the 73 pins with the UFBGA73 package.

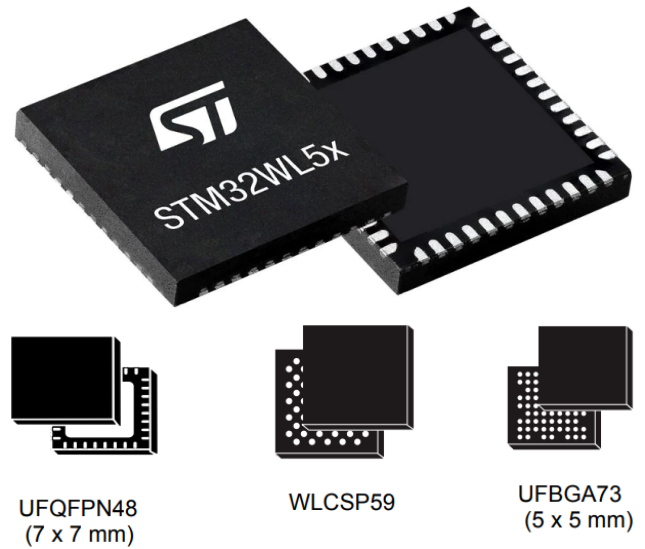
Figure 2.7: STM32WL55JC1 *Nucleo* board [5].

Figure 2.8: Packages of the STM32WL55 chip [6].

The microcontroller chip embedded in the STM32 *Nucleo* (STM32WL55JC16) has 2 cores. This is not needed for the application, one core is sufficient. This chip is part of the STM32WLx5xx series. The STM32WLE5C8 is a single-core microcontroller, specifically designed for low-power wireless applications (indicated through the "E"). Hopefully, it has the same API and the same footprint as the STM32WL55C16 which makes them easily interchangeable with little code adaptation. Therefore, the application is first developed on the *Nucleo* board. And after the nodes are embedded a single-core low-power microcontroller.

2.3 Gateway

The main hardware for the node being selected, one must find a gateway. The gateway is a TTGO by LILYGO, it is represented in Figure 2.9. It is a development card containing an ESP32 microcontroller. In addition to the microcontroller, it contains an SX1276 LoRa module.



Figure 2.9: TTGO by LILYGO gateway.

The module also handles WiFi and Bluetooth wireless protocol. This is important as it will communicate with the server using WiFi. In addition, it contains an OLED screen if one wants to display the data as it comes.

The module is supplied either by a USB port or a battery. The gateway does not need to be low power, as one assumes it can be connected to the mains due to its high versatility in location.

The application uses one gateway, however, if the scope increases and one wants more area to be covered, multiple gateways can be deployed. They will therefore all communicate with the same network. Caution must be taken when the same package is received by multiple gateways to avoid duplicates.

Chapter 3

A simplified network

This section presents a simplified network realization. Its diagram is illustrated in Figure 3.1. In this network, the *Nucleo* board is used and connected to a laptop. The board microcontroller code is generated with the *STM32CubeIDE*. The MCU reads data from sensors, then the microcontroller sends those data in a LoRa message. And finally, the message is received and demodulated by the gateway.

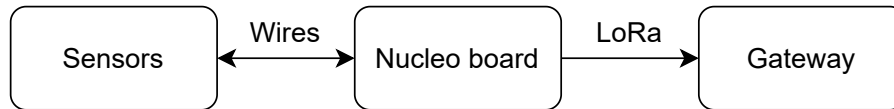


Figure 3.1: Simplified network diagram.

The first step is to perform a first LoRa communication. Then the gateway will receive and decode the packet. When the board and the gateway are able to communicate, the maximum range of the network will be tested by trying communication between a fixed gateway and a board moving in the Montefiore corridors.

And finally, one wants to communicate useful data to the gateway. Therefore, sensors are selected. Their communication protocols are studied and then executed thanks to the microcontroller. And eventually, the environmental parameters are retrieved.

During the year, codes were placed on the ULiege GitLab. The latter is a platform for collaborative version control, continuous integration, and deployment. It allows for a backup on the web and to have all the history with the different versions. In addition, it allows multiple people to work on the same project. Furthermore, it is a public page so they are available to everyone.

3.1 LoRa communication implementation

For communication to be successful, it must have a sender and a receiver. The first part of this section will focus on the MCU as the sender and the second part on the gateway as the receiver.

3.1.1 Microcontroller

STM32CubeIDE

The MCU code is written in *C* with the help of the *STM32CubeIDE*. This IDE is a really powerful tool. It requires a project creation to handle the implementation. The *Nucleo* board or the MCU must be specified during the creating phase. This allows the IDE to generate codes that the target microcontroller understands. The major strength of the IDE is the *.ioc* file. In this configuration file, one can modify the settings, add additional peripherals and select their configuration, and modify the clocks... Then, the IDE will generate codes from those configurations. The initialization will be

handled, as well as the setting of the parameters... Notice that at every moment, one can change the configuration and regenerate the code, without erasing the code added, if it is correctly placed between the `/* USER CODE BEGIN */` and `/* USER CODE END */` indications. This IDE is based on the open-source Eclipse IDE and is used for the presence of tools to speed up development and the examples provided..

PuTTY

The *Nucleo* board also uses USART2 to communicate with the laptop thanks to a COM port. This data can be processed by an application such as *PuTTY*.

To allow the usage of the `printf()` function. The `<stdio.h>` file is included in the project. Then one simply adds a function that sends the data contained in the variable `ch` using the USART2. This is done thanks to the `HAL_UART_Transmit()` function. The function is named `__io_putchar(int ch)`.

Project creation

A project is created for the simplified network application. The correct *Nucleo* board is selected. During the creation phase, one can choose to use only one of the two cores in the microcontroller. The application clearly does not need two cores for the few tasks it executes. Therefore only the M4 core is enabled and M0+ is disabled. It is not possible to use only M0+. This will also allow a less complicated code. In fact, dealing with communication between both cores is not straightforward.

To handle LoRa communication, the `SUBGHZ_PHY` middleware is activated in the `.ioc` file. In the configuration file, LoRa is indicated as modem use. And in the same sub-file, all the middleware settings such as the spreading factor, the code rate, and the bandwidth,... can be modified.

The project generation is very powerful, but it creates a lot of files to be understood. In general, there is one main file and one file per peripheral, with their ".h" file associated. The file associated with the LoRa communication is in the `SubGHz_Phy` folder.

Sequencer

The project as generated uses a sequencer at its core. Tasks with their priorities can be added to the sequencer which will execute them by decreasing orders of priority. It acts as a Round Robin. If the sequencer has no task left to execute, the MCU enters idle mode until a new task is set. The sequencer file is in the `Utilities` folder.

To add a task, the following steps must be followed:

1. First, one needs to associate a task number with the task name. To do so, in the `utilities_def.h` file, one finds the enum `CFG_SEQ_Task_Id_t`. An enum type def takes multiple entrances. Based on the order of the value in the enum, each value is associated with a number. One needs to add the task name `Task_Name` to the list.
2. The `stm32_seq.h` and `utilities_def.h` files must be included in any files using the sequencer. In this implementation, this is the `subghz_phy_app` file.
3. The next step is to declare a task, to "register" it to the sequencer. To do so, one calls the following function:

```
UTIL_SEQ_RegTask((1 << Task_Name), UTIL_SEQ_RFU, Fct_name);
```

The first argument is a 1 shifted to the left a number of times depending on the value associated with *Task_Name*. It enables the sequencer to identify a task when called. The second parameter *UTIL_SEQ_RFU* is set to 0 and informs the sequencer that all tasks registered shall be considered. And finally, the last parameter is the name of the function which must be called when the task is executed. In the practical implementation, tasks are declared at the initialization phase in the *SubghzApp_Init()* function.

4. The next step is to call the sequencer, for it to execute the tasks if needed. To do so, the following function must be called in the main while loop:

```
UTIL_SEQ_Run(UTIL_SEQ_DEFAULT)
```

5. Finally, when needed, a task is set for execution by the sequencer thanks to the following function:

```
UTIL_SEQ_SetTask((1 << Task_Name), Task_Priority);
```

The first parameter act as an identifier and the second parameter specifies the task priority.

Task

A task is created to send data. The task name is *CFG_SEQ_Task_Send* and the function to be called is *Send_data*. This function sends the data thanks to a LoRa communication. The data to be sent are placed in a buffer of *uint8_t* types, then the following function is called with the buffer and the size of the buffer to send.

```
Radio.Send(transmitBuffer, Buffer_size);
```

This is the only line of code for the LoRa transmission. It illustrates the power of the HAL functions and the configuration file.

Button

For the first test, the task for sending a LoRa package is set whenever the user pushes the first button. It triggers an interrupt. When the interrupt is triggered, the corresponding callback function is called. And in that function, the task is set.

Buttons are provided on the *Nucleo*, as well as LEDs and connectors,... To help with the configuration, STM provides BSP (board support packages). It is a folder containing files defining functions related to the features on the board. For instance, it configures the microcontroller outputs to correctly placed the antenna in emission mode. It also provides a function to change the configuration for reception. One can choose to use them or not in the configuration file. For the simplified network they are used as they simplify the code.

Timer

In practice, pushing on the button every time one wants to send a message is not practical at all. Messages should be sent automatically. To do so, a timer is set to periodically test if a message must be sent. In the simplified version, the message is always sent. To test the application range, it is easier to set a low period. Therefore, the first message-sending schedule consists in sending a message every 5 seconds.

To do so, one needs to configure a timer. Unfortunately, regular timers do not run when in idle mode, while the sequencer set the MCU in idle mode if it has no tasks to execute. Hopefully, low-power timers (*lptim*) do work in idle mode. Therefore, one needs to enable a low-power timer on the configuration

page. The timer interrupts must be enabled as well. One also needs to set the clock in the clock configuration setup. In this application it has been chosen that the clock for the low power timer selected (lptim1) would be LSE which is a 32.768 kHz crystal clock. In the .ioc file, there are a "Clock Configuration" section to select the frequency of the timer. This is illustrated in Figure 3.2.

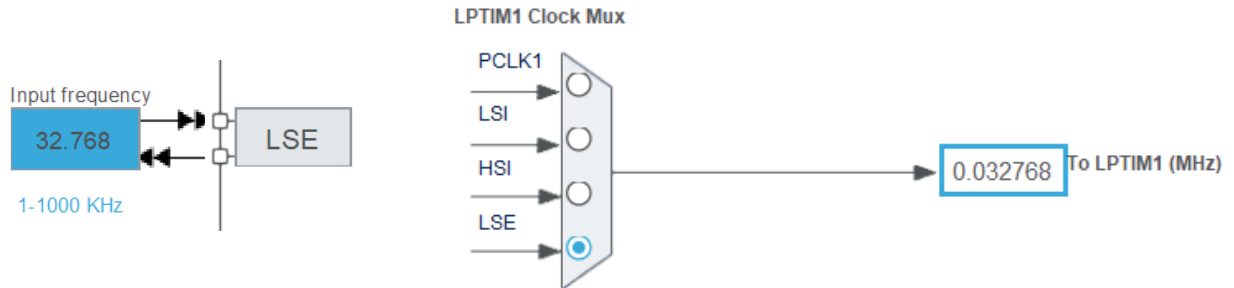


Figure 3.2: LPTIM1 clock frequency selection.

The next step is to set the clock prescaler (*pre*) and the repetition counter (*rep_cnt*). The prescaler divides the clock frequency before it enters the timer. Concerning the repetition counter, its value is decreased by 1 each time an LPTIM counter overflow event occurs. When it is zero, the time elapsed counter increases by 1.

6 modes are available, and the timeout one is selected. To initialize the timer, the following function is called:

```
HAL_LPTIM_TimeOut_Start_IT(&hlptim1, (uint32_t) Period, (uint32_t) Timeout)
```

The IT indicated it enable the interrupts. The first parameter identifies the timer. The second parameter is the period which indicates the auto-reload value. It takes a value between 0x0001 and 0xffff = 65535. and the last parameter is the timeout. They can both take values between 0 and 65535. It enables the MCU to wake up from low-power modes. If the time elapsed counter reaches the timeout value, the following callback is called:

```
HAL_LPTIM_CompareMatchCallback(LPTIM_HandleTypeDef * hlptim)
```

In another part of the code, it is declared as "weak" and may therefore be overwritten. This is done in the *main.c* file. In this function, the sending data task is set. And it will be executed by the sequencer whenever the processor is ready.

The callback function is called at a frequency that can compute as follow:

$$f_i = LSE \cdot \frac{1}{pre} \cdot \frac{1}{1 + rep_cnt} \cdot \frac{1}{1 + Timeout}$$

Caution that the prescaler must be a power of 2 comprised between 1 and 128 because it fits on 3 bits. On the other hand, the repetition counter can take any value between 0 and 255. In practice, $pre = 32$ which indicates that the repetition counter is decreased with a frequency of approximately 1 kHz . The repetition counter is set to 1 and the timeout to 254 for the task to be set every 4 seconds.

Conclusion

Thanks to *STM32CubeIDE* and HAL functions, not a lot of code is needed to have a functional application. One however needs to correctly understand each setting and the hierarchy in the generated files.

3.1.2 Gateway implementaion

This section will focus on the gateway code to handle a LoRa reception. The code is written in *C++* in the visual studio code application using the platerm.io add-on. It can be flashed into the gateway MCU easily thanks to the USB port.

Platform.io

The platform.io extension is added to the application. The project configuration is stored in the *platform.ini* file. It contains information on the development environment. Including

- the board: ttgo-lora32-v2
- the platform: espressif32
- the framework: *arduino*
- the monitor speed: 115200
- the libraries dependency

Setting the framework to *Arduino* means that the main file follows the *Arduino* framework. In other words, the *main.cpp* file contains a *setup()* function and a *loop()* function. As the names suggest, the *setup()* function is called one time in the beginning, then the *loop()* function is called endlessly in a loop.

The monitor speed is used by the serial monitor handle by *Platerformio* to communicate with the gateway thanks to a VCOM port.

And finally, one big strength of *Platerformio* is the multiple public libraries available. One simply adds the name of the library in the libraries dependency field and every function declared in those libraries can be used.

LoRa module

Packages received by the gateway are decoded in the LoRa module. However, this module is not integrated into the microcontroller. Therefore, the data must be exchanged between the two chips. The LoRa module has an SPI communication interface. For the communication to succeed, one must first know which pin can handle the SPI communication in the gateway. To do so, one looks at the pins functions provided in Figure 3.3. The pin connections may be verified in the gateway schematics.

Table 3.1 resumes the pin of the gateway microcontroller connected to the LoRa module.

Pin number	5	18	19	23	26	27
Function	SCLK	CS	MISO	RST	DIO / IRQ	MOSI

Table 3.1: LoRa related pins.

For a spi communication, 4 pins are required:

- a pin for the clock (SCLK)
- a pin for the data in one way, master output, and slave input (MOSI)
- a pin for the data in the other way, from the slave to the master (MISO)
- a pin for the chip selection. If pulled up, the module is disconnected. It allows multiple devices on the same SPI module. Each device has its own CS pin.

Finally, the RST pin is to make a reset of the module and the DIO or IRQ is the interrupt request pin.

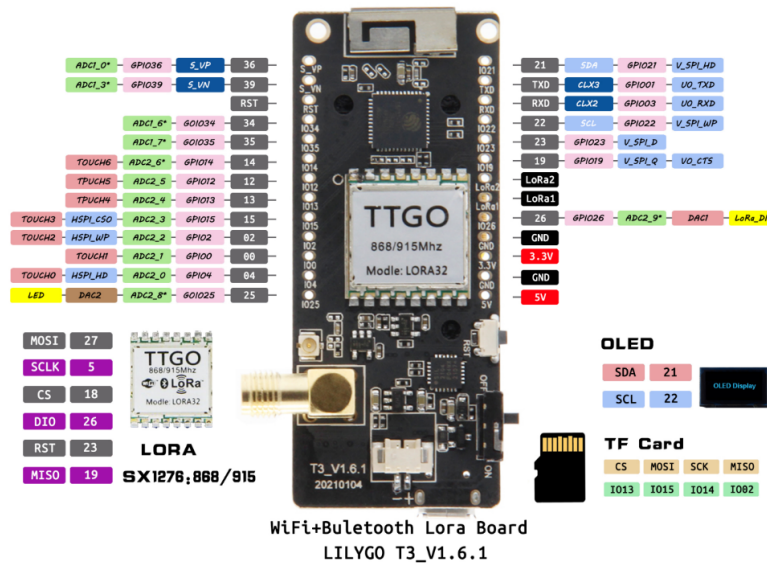


Figure 3.3: Assignment of the gateway’s MCU pins.

Implementation

The implementation uses the *LoRa* library, written by Sandeep Mistry. This allows using simple functions to handle the LoRa reception.

- First the SPI communication is started. `SPI.begin(SCLK, MISO, MOSI, CS)`
- The IRQ, RST, and CS pins are communicated to the LoRa library `LoRa.setPins(CS, RST, IRQ)`
- Starting the Lora module `LoRa.begin(LORA_freq)` with `LORA_freq = 868 MHz`. Those 3 first steps are done in the setup part.
- Then in the `loop()` function, the `LoRa.parsePacket()` function is called endlessly. The output is the size of the LoRa package received. If it is not zero then data is received.

OLED screen

To visualize the data, one is interested in using the OLED screen of the gateway. To do so, *adafruit* libraries are added. The screen communicates in I²C and its address is 0x3c. The screen is first declared `display(128, 64, &Wire, -1)`. The first 2 parameters are the screen size. The *Wire* function is used to start the communication thanks to the following commands:

```
Wire.begin(OLED_SDA, OLED_SCL);
display.begin(SSD1306_SWITCHCAPVCC, 0x3c);
```

Then one can write on the OLED screen (`display.println("text")`), multiple parameters as the text size, the text color, the location of the text,.. may be chosen. And finally, the scene can be cleared (`display.clearDisplay()`).

3.2 Communication test

Having established a LoRa communication from the node to the gateway, it is time to test the range capacity of the communication. To do such a test, the node is left at a fixed place while the gateway is moving across the corridors of the Montefiore Institute. In the final application, it is the opposite,

the gateway is fixed, and the node can be placed anywhere around it. However, the gateway displays a message if it correctly receives the message. For the test, the node sends a message every second. If the message is correctly displayed on the screen of the gateway, the area is accessible thanks to LoRa communication.

For the test, the node is left in the Microsys laboratory. The gateway moves in the corridors, powered by a portable battery. The results are shown in Figure 3.4. The cross represents the location of the node. The green represents the location at which the gateway always receives the message. The orange areas are zones where the gateway receives sometimes the message and sometimes does not. And finally, in red zones, no message is received. Tests were made in two different settings. One with the antenna toward the center of the building and when with the antenna facing out (up in the Figure). The results were really similar, with a slight advantage when the antenna faced the center of the building.

Based on the graph, the range of the application is approximately a hundred meters. This is lower than the theoretical range. It was expected because of the multiple walls. For the Montefiore Institute, if the gateway is placed at a more centered place, it will cover all the rooms.

Notice that the parameters of the LoRa transfer are important as well. In particular, the spreading factor influences the range. When SF is higher, the data rate is lower. And a slower data rate implies a higher energy and thus a better range for the data set. It is therefore a tradeoff between the greater range and the lower energy consumption. In this application, SF = 7, the lower value (it goes up to 12).

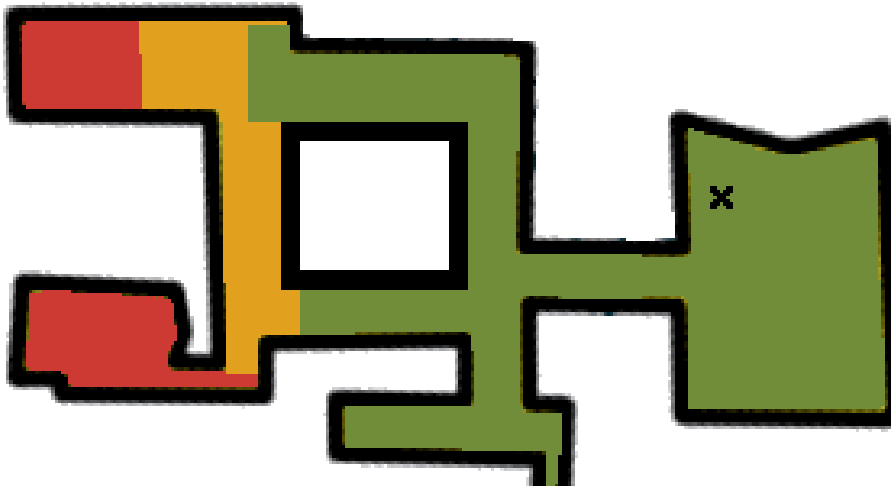


Figure 3.4: Range test of the simplified network at the Montefiore Institute.
 red = no reception, orange = regular reception, and green = systematic reception
 x = fixed position of the node

3.3 Sensors selection and implementation

The goal is to monitor the environment thanks to sensors. At first, it has been chosen to monitor 5 parameters. Those will be the data sent from the node to the gateway. This choice is not exhaustive, and if new parameters want to be monitored, one can simply add a sensor to the final product. Those parameters are interesting on their own, and together they should be able to detect human presence in the room:

- **Temperature:** Monitoring the temperature is really interesting to detect if a room is overheated or underheated. It can detect if a heater was been left on when it should not. Furthermore, temperature variations are correlated to human presence. In fact, each person increases the ambient temperature only by breathing. However, it is important, when processing the data, to look at the variation of temperature rather than the absolute temperature value. In fact, the

heaters might be on or off, the season, the presence of the sun, the windows or the door open.. all those factors influence the temperature as well.

- **Relative humidity:** Human comfort depends on the humidity level. Control of the relative humidity may also be required for some lab experiments. Furthermore, similarly to temperature, relative humidity is correlated with the presence of people in the room. It increases due to the water present in the air being exhaled by humans while breathing.
- **Ambient light level:** Having an idea of the illuminance present in the room is a real advantage. One can easily know which room is lit up. It reduces the work of the care keeper who verifies each evening that the lights are correctly powered down. Furthermore, the value is correlated with human presence, as, typically, the lights are on only if the room is occupied.
- **CO₂ level:** Monitoring the CO₂ level indicates if the rooms are well aerated or not. This is also one obvious parameter for human detection. Human breathing in a room rejects CO₂, therefore, having an increase in the CO₂ level is correlated with people being in the room.
- **Noise** thanks to a microphone: Knowing the level of noise in a room is a good indication of whether or not there are people chatting in the room, or if a course is given. The noise level overcoming a threshold, the room is certainly occupied. However, no noise does not mean anyone in the room.

The microcontroller can handle voltages from 1.8V to 3.3V, and the final supply voltage value is not decided yet. In fact, having a higher voltage consumes more energy but the LoRa transmission power is impacted by it. To be adjustable, the selected sensor should handle at least the same range.

It is a low-power application, therefore the current consumed by the sensor in active mode must be as low as possible. And when not used, the sensor supply voltage will be disabled by the MCU, in order to limit to 0 its consumption.

The following sections will present all the sensors selected. For each sensor, it will start with a comparison with other existing sensors, then the connections of the sensor will be presented in schematics. Then the communication protocol will be presented and then implemented. And finally, sensors are tested. All the functions implemented are in the *sensors.c* and *sensors.h* files.

To test the sensors, a test PCB was designed. It is represented in Figure 3.5. In fact, the sensors cannot be tested on a breadboard. On the board, each sensor has its own connections. The ground is common but the supply voltages are uncoupled. This PCB has been created with a double ground plane stitched via multiple vias. In practice, that means that a ground plane has been created on both layers and both planes are regularly connected with each other. This technique reduces the potential interference that might occur between planes (main network, common mode, ...). It basically shields the signals.

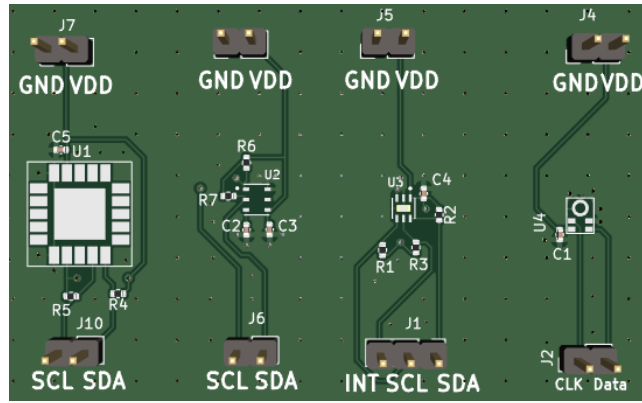


Figure 3.5: Design of the test PCB for the sensors.

3.3.1 Temperature and relative humidity sensor

The first sensor is for the relative humidity and the temperature. In fact, humidity sensors nearly always monitor the temperature as well. This is because the relative humidity HR varies with the temperature T , it is defined as the partial pressure of water vapor P_{vap} in air divided by the saturation vapor pressure $P_{sat}(T)$ of water at that temperature.

$$\%HR = \frac{P_{vap}}{P_{sat}(T)} \cdot 100$$

Table 3.2 compares different sensors. The HS3003 is chosen because of its supply voltage range and its low current consumption.

	Communication protocol	Vdd [V]	Idd [μ A]			Range	Resolution	Accuracy
			Measure	Average				
SHT21S	SDM signal (sigma delta)	2.1 - 3.6	160	-	%HR	[0,100]	0.04	± 2
					$^{\circ}$ C	[-40,125]	0.01	± 0.3
SHT3x-DIS	I ² C	2.15 - 5.5	600	1.7	%HR	[0,100]	0.01	± 2
					$^{\circ}$ C	[-40,125]	0.01	± 0.2
HS3003	I ² C	1.8 - 3.3	-	24.4	%HR	[0,100]	0.01	± 2.5
					$^{\circ}$ C	[0,70]	0.015	± 0.25

Table 3.2: Temperature and relative humidity sensors comparison.

Schematic

This section presents the sensor connections. Those are mandatory for it to work correctly. The MCU is responsible for powering or not the sensor. To do so, the supply voltage (VDD) of the sensor is connected to a pin of the MCU (HS3003_EN). If the pin is high, the sensor is powered on, otherwise, it is off. This is possible because the sensor does not need a lot of current. If too much current is needed the pin could be damaged. For extra protection, a resistor (R202) is placed between the MCU pin and the sensor, it limits the current spikes and helps improving the EMC behavior by reducing potential ringing. Caution must be taken that the voltage at the resistor terminals must be low for the sensor to be correctly powered. Figure 3.7 shows the full sensor connections.

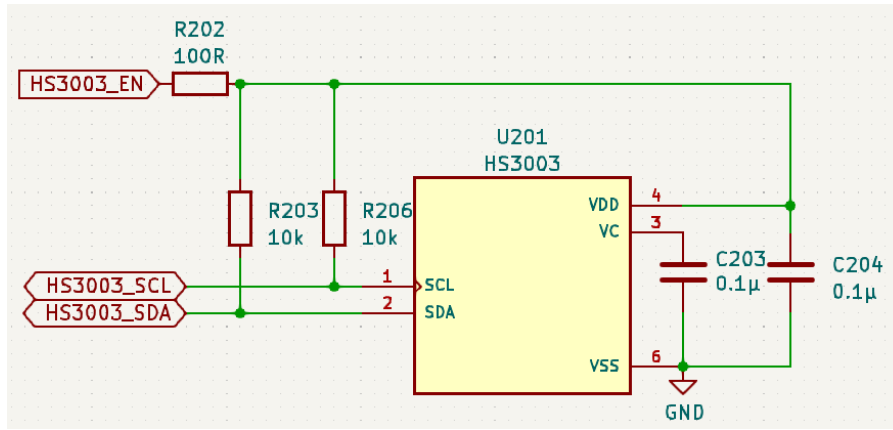


Figure 3.6: Temperature and humidity sensor (HS3003) schematic.

The sensor has 6 pins.

- Pin 1 and pin 2 are respectively SCL and SDA. They are connected to pins of the STM32 that support the I²C protocol. Furthermore, pull-up resistors are placed between the lines and VDD.
- Pin 3 is connected to the ground through a decoupling capacitor as requested by the datasheet
- Pin 4 is connected to VDD and also has a decoupling capacitor. Such capacitors decouple the AC signal from a DC signal. They must be placed as close as possible to the pin to avoid new AC noise on the wire between the capacitor and the pin.
- Pin 6 is connected to the ground and finally, pin 5 is not connected.

Communication protocol

The sensor communicates using the I²C protocol. This protocol works with 2 lines, SDA and SCL. The master initializes the communication by generating the clock on SCL and pulling down SDA while SCL is high. Then the master sends the address of the device of 7 bits, followed by a read (=1) or write (=0) bit on the SDA line. It indicates the data transfer direction. All transfers consist of 8 bits and a response bit: 0 for Acknowledge (ACK) or 1 for NACK. After the ACK is received, another data byte can be transferred. If there is no acknowledgment, the communication is stopped.

The HS3003 device address is 0x44. The master is the microcontroller embedded in the *Nucleo* board. For the master to acquire a measurement, it must start with a measurement request (MR). This is done by sending the device address followed by 0 (= write). And then stop the communication without writing anything.

Then the master must wait until the end of the measurement cycle. When it is the case, it can fetch the data (DF). The master starts a communication with the read bit. The device sends an acknowledgment and starts sending bytes as long as the master acknowledges it. The first 2 bytes [13-0] contain the humidity data and the following 2 bytes [15-2] contain the temperature data on 14 bits. This sequence is represented in Figure 3.7. The two first bits of the communication are the status bits. It must be 00 for valid data. And the last 2 bits must be ignored.

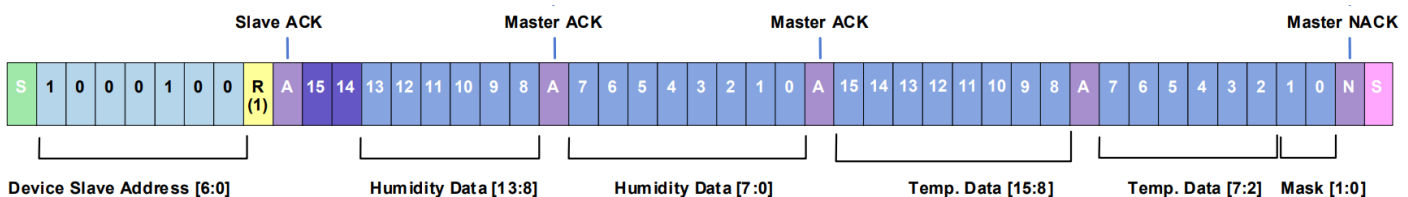


Figure 3.7: Temperature and humidity sensor (HS3003) data fetching protocol.

To have the final value, the data needs to be processed following the requirement of the datasheet:

$$Humidity[\%HR] = \left(\frac{Humidity[13 : 0]}{2^{14} - 1} \right) \cdot 100$$

$$Temperature[^\circ C] = \left(\frac{Temperature[15 : 2]}{2^{14} - 1} \right) \cdot 165 - 40$$

The measurement time depends on the resolution of the data. In this application, the highest resolution is used (14 bits) which implies 33.90 ms. This time can be reduced by changing the resolution value in the non-volatile memory. To access that memory, the master must do it within a small period of 10ms after applying power to the sensor. The master sends the address with a write bit, followed by 3 bytes: 0xA00000. This command takes 120µs to process, then the master sends a "write" request, followed by the register address. Caution that only the resolution bits can be changed.

Implementation

The first step is to select the pins of the microcontroller. The enable signal can be on any pin, while the others must be on pins supporting the I²C protocol. Keep in mind that not all pins will be available on the STM32 chip compared to the *Nucleo* board. Table 3.3 resumes the pin assignment. The correspondence between the STM32 pins and the *Nucleo* connectors number can be found in the user manual UM2592 available on *STMicroelectronics* website.

The enable pin is simply an output of the MCU, while the other pins are connected to the first I²C module.

Label name	HS3003_EN	HS3003_SCL	HS3003_SDA
STM32 pin	PA15	PB8	PB7
GPIO mode	Output	I2C1_SCL	I2C1_SDA
Nucleo pin	CN7 - 17	CN9 - 6	CN9 - 1

Table 3.3: MCU pin assignment for the temperature and humidity sensor (HS3003).

Thanks to the HAL functions, it is really easy to implement.

- To power up or down the sensor, the pin is SET (= 1) or RESET (= 0) thanks to the function:

```
HAL_GPIO_WritePin(HS3003_EN_GPIO_Port, HS3003_EN_Pin, GPIO_PIN_(RE)SET)
```

- Starting an I²C write operation is done in one line. The data to be sent is in a buffer, and the number of bytes to be sent is specified by the *data_size* parameter.

```
HAL_I2C_Master_Transmit(&hi2c1, HS3003_ADDR, data, data_size, HAL_MAX_DELAY)
```

- And finally a read operation is done in one line as well and is very similar to the transmit function. The data retrieved is put in the buffer *buf*. The MCU will acknowledge *buf_size* bytes before sending a STOP signal.

```
HAL_I2C_Master_Receive(&hi2c1, HS3003_ADDR, buf, buf_size, HAL_MAX_DELAY)
```

Test

Finally, the sensor is tested thanks to the test PCB. The pins are correctly connected to the *Nucleo* board. The voltage of the board is 3.3V. For the first test, when the user pushes the button, the data is retrieved by the MCU. This test was not conclusive. The temperature value was correct, but the

humidity value was not. It kept a constant value of around 75%.

One possible explanation is the sensor being dried out due to the soldering process. The sensor must be exposed at the least amount of time to high temperatures. In addition, rehydration conditions are recommended after soldering. Otherwise, there might be an initial offset. This hopefully disappears after some time. It is also possible, that it was damaged due to fabrication or manipulation.

A baking and rehydration process is described in the datasheet. However, letting the entire PCB through this process could have damaged the CO₂ sensor also on the board. Therefore, a new sensor was soldered on a second PCB. The soldering process and manipulation were theoretically the same. However, with the same code and the same external settings, the value of the relative humidity was correct. The values are verified by comparing them to the values received by other sensors of the same type in the lab (BMP388).

3.3.2 Ambient light sensor

The following sensor will evaluate the ambient light level. To give an order of magnitude of values to be retrieved, the illuminance of an office has typically values between 200 to 700 *lx*. A comparison of different sensors is done in Table 3.4

	Communication protocol	Vdd [V]	Idd [A]	Range [<i>lx</i>]	Resolution [<i>lx</i>]	Relative Accuracy
TSL2520	I ² C	1.8	195	-	0.001	
BH1750FVI-TR	I ² C	2.4 - 3.6	120	1 - 65535	1	± 20%
OPT4001	I ² C	1.6 - 3.6	30	312.5 μ - 83 k	312.5 μ	0.4 <i>lx</i>
OPT3001DNPR	I ² C	1.6 - 3.6	3.7	0.01 - 83 k	0.01	0.2%

Table 3.4: Ambient light sensors comparison.

Based on the voltage range and the current consumed, the best are OPT4001 and OPT3001 sensors. However, the OPT4001 sensor has a PicoStar-4 package and the actual sensing part is situated on the same side as the pads. Therefore, the PCB must have a hole at this position. However, even with this hole, the PCB is so thick that it interferes with the measure. To resolve this problem, one may think of a flex PCB, but it is really expensive. As a consequence, the OPT3001 sensor is chosen.

Schematic

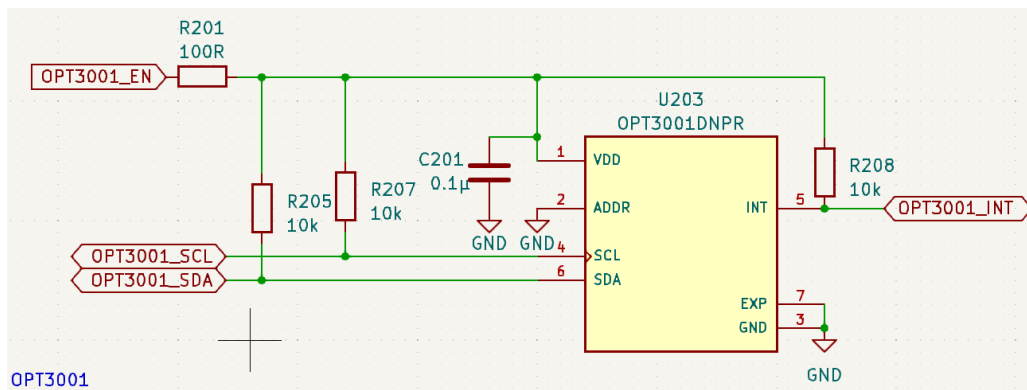


Figure 3.8: Ambient light sensor (OPT3001) schematic.

The sensor has 6 pins and an additional exposed pad. This sensor also communicates in I²C and therefore has similar connections as the HS3003 sensor. The light sensor is activated by a pin of the microcontroller, and this pin is protected by R201. Pins 4 and 6 are respectively SCL and SDA. They

are connected to the microcontroller, and they have 2 pull-up resistors. In addition, a third line is present, the "INT". This line is useful to interrupt a processor. It controls a high-limit and low-limit range for the light level. This range can be tuned by the user to suit the application. If the condition is not met, a fault event occurs. And if the number of such events exceeds the number in the fault count register, then an interrupt is triggered by the sensor. This feature is not used, but the pin is still connected, in case it is needed in the future.

The exposed pad and pin 3 are both connected to the ground. And finally, pin 2 is also connected to the ground. This pin is responsible for the address of the component, required for the communication protocol. It sets the 2 LSB of the address by either being connected to ground, VDD, SDA, or SCL. This allows 4 of the same sensor to be connected to the same I²C module, without clashing each other.

Communication protocol

The sensor uses the I²C protocol. This protocol has already been explained for the humidity and relative humidity sensor. The behavior is the same for this application. However, asking for data is not done exactly the same way.

The microcontroller starts with a transmit command. The first byte (after the address of the device) is the address of the configuration register. In this register, the value 0xd2, and 0x10 is written. This command will ask for a single-shot measure, with 100 ms conversion time, and an automatic choice of the range.

Then the microcontroller waits the time of the measure (200ms for safety), before looking at the result. To look at the value of one register, the microcontroller, sends the target register address, then receives 2 bytes. before looking at the result, one might first look at bit 7 of the configuration register to be sure that the conversion is ready. If the data is ready, the values of the result register (0x00) are retrieved.

The result is stored on 16 bits. Bits 15:12 is the exponent (E) and bits 11:0 is the fractional result (R). The final value is computed thanks to the following formula:

$$Illuminance = 0.01 \cdot 2^E \cdot R$$

Implementation

Table 3.5 resumes the pin assignment on the microcontroller. The sensor is connected to the second I²C module. It could have been connected to I2C1 (which is the module handling the temperature and humidity sensor). Doing so would have reduced the number of pins required for the implementation. This could have been done because the I²C protocol uses addressing.

Label name	OPT3001_EN	OPT3001_SCL	OPT3001_SDA	OPT3001_INT
STM32 pin	PB4	P12	PA11	PB12
GPIO mode	Output	I2C2_SCL	I2C2_SDA	/ (Input)
Nucleo pin	CN8 - 4	CN5 - 10	CN5 - 9	CN9 - 3

Table 3.5: MCU pin assignment for the ambient light sensor (OPT3001).

Concerning the actual codes, it uses the same functions as the previous sensor.

Test

The sensor was tested thanks to the PCB and the *Nucleo* board. The sensor acts correctly by increasing the number of lux when there is more light. Results are compared to the data given by a lux-meter

which acts as a reference. However, the lux meter is consistently higher (about 5% error). An example is given in Figure 3.9. The sensor is placed really close to the lux meter. This difference is partially due to wires making a shadow on the sensor. Those are the wires to connect the PCB board and the sensors to the *Nucleo* board and thus the MCU. By pushing them on the side, the value is already better. In the final design, the sensor must be **as far as possible from every shadow source**. The advantage is that no wires will be required for the final PCB.



Figure 3.9: Comparison between the ambient light sensor and the lux meter (493 vs 482.56).

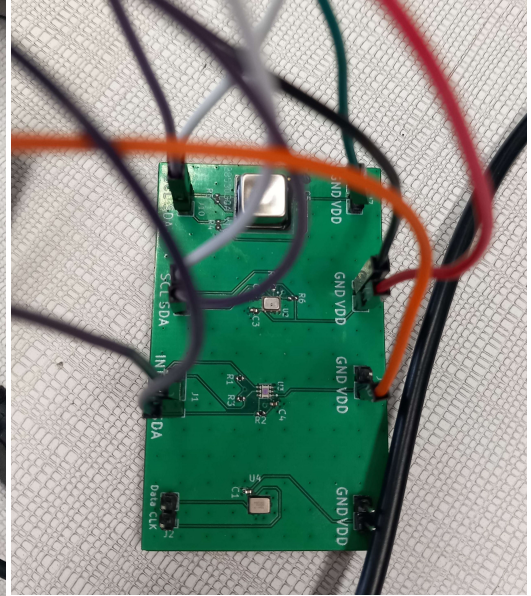


Figure 3.10: Illustration of the wires making shadows on the ambient light sensor.

3.3.3 CO₂ level sensor

The next sensor considered is the CO₂ sensor. Normal concentration in outside air is around 400 ppm. Table 3.6 presents some CO₂ sensors. One sees that those sensors are costly and tend to consume a lot of current. This is not in phase with the low-power application and affordable nodes (for easy multiplication) philosophy.

	Price [€]	Vdd [V]	I _{dd} [mA] (at 3.3 V)	Range [ppm]	Wake-up time	Accuracy	
			Peak	Average			
SCD40	37.56	2.4 - 5.5	175	0.45	0 - 40000	1 s	± (50 ppm + 5%) 400 - 2000 ppm
SCD41	50.53	2.4 - 5.5	175	0.45	0 - 40000	1 s	± (40 ppm + 5%) 400 - 5000 ppm
APC1001J	50.06	4.5 - 5.5	120	10	400 - 65000	3 min	-

Table 3.6: CO₂ level sensors comparison.

The APC1001J sensor provides the ppm in CO₂ equivalent. It is therefore a multi-gas sensor and not only a CO₂ detection. A big disadvantage is the start-up time of 3 minutes. The SCD4x series seems a better solution with a 1000 ms wake-up time. A compromise between price, accuracy, wake-up time,... Caution, that none of the CO₂ sensors accepts a 1.8 V voltage.

In addition to the CO₂ level, the products presented in Table 3.6 also give the temperature and relative humidity levels as well. However, due to the high current, if the sensor were to be used, it would be less frequent, when the temperature and relative humidity shows an increase for example. When people

seem to be detected by the low-power sensor, the CO₂ sensor would validate the result. Therefore, one cannot get rid of the humidity and temperature sensor. However, the temperature and humidity values are still retrieved to further validate the values of the HS3003 sensor.

Schematic

The sensor has 20 pins and an additional exposed pad. They are represented in Figure 3.11. One notices that there are a lot of "DNC" (do not connect). For the other pins, their connection is shown in Figure 3.12. Due to current requirements, the alimentation of the sensor cannot be connected to a pin of the MCU. It must be directly connected to the VDD source. It is the case for pins 7 and 19 (7DD and VDDH respectively). The "GND"s are obviously connected to the ground.

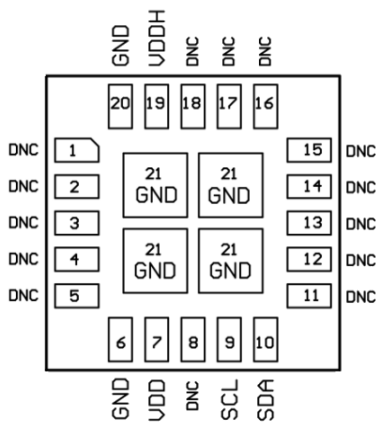


Figure 3.11: CO₂ level sensor (SCD40) footprint.

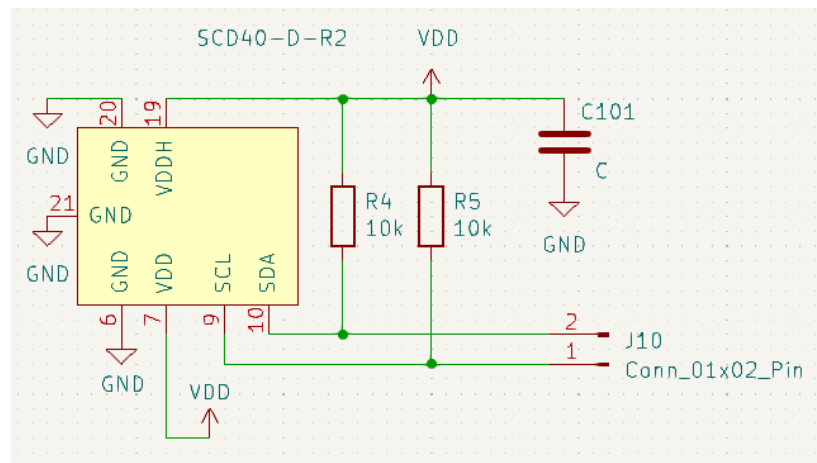


Figure 3.12: CO₂ level sensor (SCD40) schematic.

The sensor uses the I²C protocol, therefore, SDA and SLC are connected to the MCU. For the first design, there are connected to pins and thanks to wires connected to the MCU. The two pull-up resistors should not be forgotten. And finally a decoupling capacitor between the sensor and the ground.

Communication protocol

The SCD40 does not provide a single-shot measurement mode (it is only available in the SCD41 version). The trick is to start a periodic measurement, wait for the first measure to be ready, retrieve it, and stop the measures. This technique requires waiting for 5s between the moment a measurement is requested and the moment the measurement is ready.

The sensor uses commands on 2 bytes for the I²C protocol.

1. Firstly, the MCU sends the command 0x21b1 to start a periodic measurement
2. Then MCU waits for the measurement, which takes 5 s. The latter is the period of the periodic measurement.
3. The MCU asks for the ready status data to ensure that the data is ready. To do so, the command 0xe4b8 is sent. The ready status data is received by the MCU after 1 ms.
4. When the data is ready, MCU want to read the measurements. This is done thanks to the command 0xec05 which asks the device to send them.
5. The device needs 1 ms to process the command, then, the microcontroller can receive them by starting an I²C receive process.

The data are received in 6 bytes. The 2 first bytes are the CO₂ value C and the third byte is the corresponding CRC (to be sure the data was read correctly). The two following are the temperature value T followed by the CRC. And finally 2 bytes for the humidity H and the corresponding CRC. To obtain the final values, those parameters are processed as follows:

$$\begin{aligned} CO_2 [ppm] &= C \\ Temperature [^{\circ}C] &= -45 + 175 \cdot \left(\frac{T}{2^{16}}\right) \\ Humidity [\%HR] &= 100 \cdot \left(\frac{H}{2^{16}}\right) \end{aligned}$$

Test

And finally, the sensor is tested. The temperature and relative humidity values were compared to the values found by HS300S. The values are a perfect match down to the hundredth.

However, the CO₂ value always equals 0... The device is able to perform a self-test. Therefore, the MCU sends the command 0x3639. After 1 second, the result is sent, and if it is non-zero, the sensor has detected a malfunction. The test was done and a malfunction was indeed found. The device unfortunately does not specify which type of malfunction.

A new sensor could have been soldered on another PCB, as it was done for the HS3003 sensor. However, due to its high current consumption, a CO₂ sensor is already not the best option. And unfortunately, chosen supply voltage module can only handle 80 mA as output current, and the CO₂ sensor has current peaks up to 205 mA. Therefore, the sensor cannot be used in the final design. Furthermore, the supply voltage for the CO₂ sensor is at least 2.4 V which does not let good flexibility for the MCU supply voltage.

3.3.4 Sound volume sensor

This sensor is a microphone. The T3902 of TDK is selected. It is a MEMS microphone and its characteristics are provided in Table 3.7. It has a wide supply voltage range.

	Communication protocol	Vdd [V]	Idd [A]	Clock [kHz]
T3902	PDM	1.65-3.63	185	400 - 800

Table 3.7: Sound volume sensor characteristics.

Schematic

As for the light level sensor and the temperature and relative humidity sensor, the microphone is supplied by the MCU. The resistor R204 is placed in series to protect the pin from too much current. A decoupling capacitor is placed next to the power pin.

In addition to the power pin, the sensor has 3 pins and an exposed pad. The schematic is presented in Figure 3.13. The exposed pad and the low-power pins are connected to the ground. The *Select* pin enables a second microphone on the same line. This feature is not used here and the pin is connected to the ground as well. Finally, one pin is for the clock of the data transfer. The clock must be generated by the MCU. The useful information passes on the *data* line.

The sensor footprint was not available on *KiCad* (electronic design automation (EDA) suite for schematic and PCB layout). Therefore it was designed on the footprint editor of *KiCad* following the requirement of the datasheet. The footprint shape is represented in Figure 3.14

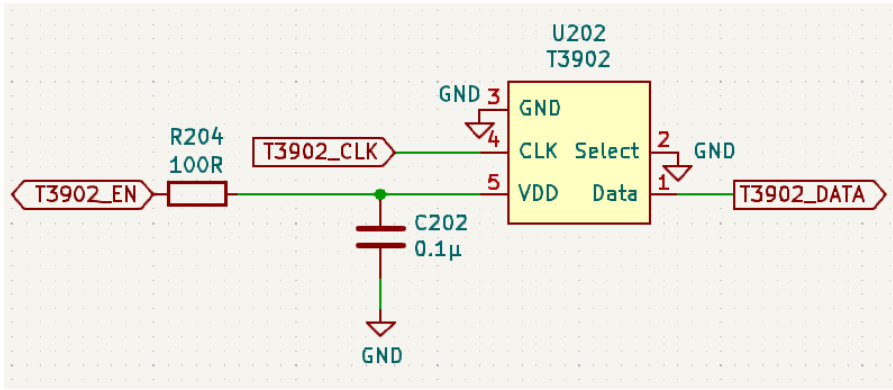


Figure 3.13: Sound volume sensor (T3902) schematic.

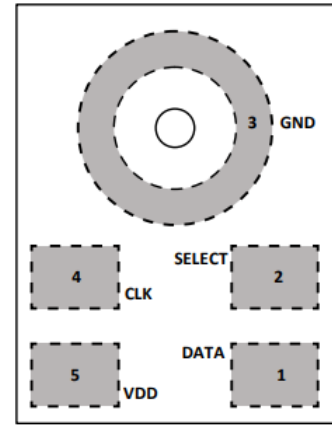


Figure 3.14: Sound volume sensor (T3902) pins.

Communication protocol

The device sends data whenever the MCU generates a clock at the supported frequency. The format is represented in Figure 3.15. At each time clock, the data line is V_{DD} or 0. The data is transmitted in the PDM (Pulse Density Modulation format). That means that the density of the pulse indicates the signal amplitude. A high density of V_{DD} values indicates a high amplitude signal. And high density of 0 values indicates a high negative amplitude signal.

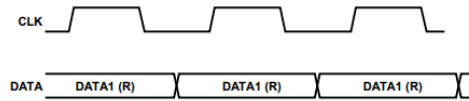


Figure 3.15: Sound volume sensor (T3902) data format.

Implementation

Because the data is in the PDM format, the values must be acquired for multiple clock steps. To have a correct estimation, it is better to listen for a few seconds. To process the data, one could count the time between each rising edge (or falling edge) of the data. The data processing implementation has not been done due to a lack of time. However, the sensor is still put on the final PCB for it to be done later without needing a new PCB design. The pin assignation is resumed in Table 3.8.

Label name	T3902_EN	T3902_DATA	T3902_CLK
STM32 pin	PA9	PA10	PA8
GPIO mode	Output	Input	SPI2_SCK
Nucelo pin	CN5 - 2	CN8 - 3	CN10 - 16

Table 3.8: MCU pin assignation for the microphone (T3902).

3.3.5 Conclusion and general comment

The sensors are powered by pins of the MCU. To protect the pin, a resistor is placed in series. By testing the sensors when supplied with 1.8V, the light sensor did not work well. However, when removing the resistor it works correctly. The voltage drop at the resistor was too high. And reducing the resistor value will allow a too much current for the pin. Therefore another solution needs to be found if one wants to protect the MCU pins.

Out of the 4 selected sensors, 1 is dismissed and a second is embedded but the data retrieved is not processed. However, the data provided by the 2 remaining sensors covers 3 useful environmental parameters: the temperature, the relative humidity, and the light level.

3.4 Conclusion

This section presented a simplified network. The network consists of a node communicating with sensors and sending those data to a gateway.

The *Nucleo* board was really easy to use because the supply voltage came from the laptop. However, the final node should be self-sufficient in energy. Therefore, the following section will tackle the energy management problem.

The board also came with its antenna. In the final product, the antenna will directly be printed on the PCB. The design of the antenna is done in Chapter 5.

Finally the final node will be created on a PCB embedding the sensors and the code developed in this section, the energy management solution developed in the following chapter, and the antenna design in Chapter 5.

Chapter 4

Energy management

The previous section presented a simplified network. In that network, the node was a nucleo board supplied by a computer. This is not as simple for the final node. The node should be self-sufficient in energy. The first consideration was to take sensors with low consumption. In this section, the interest is placed on the power source of the microcontroller.

The easiest way to harvest energy is to use solar panels. However, the nodes are placed indoors away from direct sunlight. Hopefully, indoor solar panels exist. They obviously will harvest less power. Three different types will be tested and compared.

The supply voltage is provided from a chip integrating the energy management circuit for low-power applications. It extracts DC power from solar panels to simultaneously store energy in a rechargeable element and supply the microcontroller with a tunable voltage. A section will then discuss whether to use a supercapacitor or a battery as a storage element.

The supply voltage for the microcontroller will be fixed regarding the LoRa module requirement for power and current. The diagram of the whole energy management figure is represented in Figure 4.1.

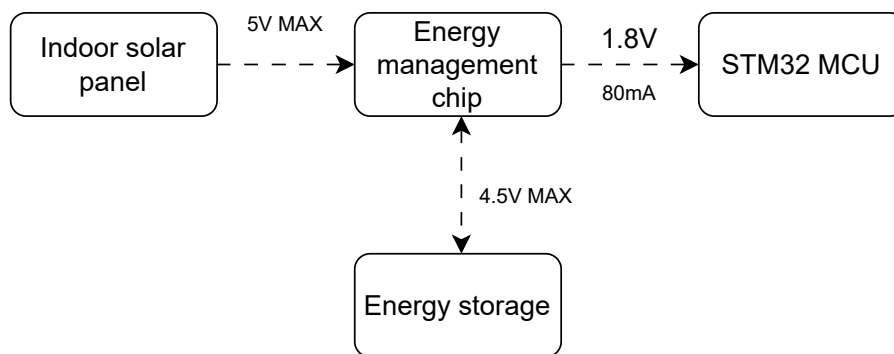


Figure 4.1: Energy management diagram.

4.1 Energy supply devices

This first section is interested in which hardware should be used for energy storage, energy management chip, and indoor solar panels

4.1.1 Energy storage

Two big types of energy storage solutions: supercapacitors and batteries.

Supercapacitors are capacitors with a high capacitance value. They are based either on Electric Double-Layer Capacitor (EDLC) layout or on electrochemical pseudocapacitances. It features 2 electrodes coated in a carbon-based porous material. The electrodes are separated by an electrolyte that is itself divided by a membrane. The energy is stored and released thanks to the physical absorption and desorption of ions in the electrolyte contained between its electrodes. This enables a better energy density than a simple capacitor, but a lower efficiency for charge and discharge cycle (0.9 against 0.95). Another big difference with a classical capacitor is that supercapacitors support only low voltages (around 3 V). The energy of a supercapacitor is unsurprisingly the same as the energy stored in a capacitor. It can be computed thanks to the following formula:

$$E = \frac{1}{2} \cdot C \cdot V^2$$

Where C is the capacitance and V is the voltage at the terminals of the capacitor.

Table 4.1 makes a comparison between batteries and supercapacitors. The first big difference is the energy density. Batteries have a really big density compared to supercapacitors (about a factor 100). And the second main difference is the power, the speed at which it can discharge or charge. In this category, the supercapacitors are way better. Those two points show that supercapacitors are well-suited for low-power applications. In fact, low-power applications do not use a lot of energy and therefore low energy density is not a problem. However, in such applications, generally, one task is done periodically, and the rest of the time the consumption is almost zero. For example, sending a packet thanks to the LoRa protocol. Those tasks require power delivered in short bursts so need fast discharge and thus high specific power.

In addition, supercapacitors are more durable and therefore lower maintenance than batteries. And finally, they are safer to be placed in every room of the Montefiore Institute.

Feature	Batteries	Supercapacitors
Energy density	Big (100-240 Wh/kg)	Lower (1-5 Wh/kg)
Specific power	Lower (1000-3000 W/kg)	Big (10000+ W/kg)
Charge/Discharge efficiency	70 - 85 %	85 - 98 %
Cycle Life	Deteriorates over time (500+)	More durable (100000+)
Process	Chemical	Physical
Safety	Depends on the technology	High (do not explode or catch on fire)

Table 4.1: Comparison between batteries and supercapacitors.

The application will use DSF205Q3R0 supercapacitors (2 F, 3 V, and only 1.22€). The number of Farad required for the application will be discussed later. By putting supercapacitors in parallel, the capacitance can be multiplied. And if a higher voltage is needed, capacitors can be put in series. Notice that putting 2 identical capacitors in series, reduce the total capacitance by a factor 2. Therefore, to have 2 F, and 6 V, 4 supercapacitors are required. When putting them in series, caution to add a balun at the junction for the capacitors to be balanced. Otherwise, the current must be exactly the same.

4.1.2 Selection of the supply voltage

Before developing the energy management chip and its parameters, one should select the supply voltage of the MCU. The range of selection is 1.8 to 3.3 V. In practice, this choice is not independent of the energy management solution. Therefore one should know that the chosen chip can output a **maximum of 80 mA** whatever the voltage.

The LoRa module is the element which is the most susceptible to be affected by the voltage level. In fact, the maximum transmit output power of the Sub-GHz radio depends on the voltage supply.

This information is provided in the datasheet and reported in Table 4.2. It is not a surprise that the maximum transmit output power decreases with the supply voltage. The power is expressed in dBm . The following formula link the power expressed on dBm to the power expressed in mW .

$$x[dBm] = 10 \cdot \log_{10} \frac{P[mW]}{1[mW]}$$

V_{DDPA} supply (V)	Maximum transmit output power (dBm)
3.3	+ 22
2.7	+ 20
2.4	+ 19
1.8	+ 16

Table 4.2: Maximum transmit output power depending on the supply voltage.

Another essential parameter is the power consumption based on the voltage and the output power. Table 4.3 reports the data available in the datasheet. It only gives the value for 3.3 V or 1.8 V input voltage. One sees that for high power, the current always exceeds the limit of 80 mA (due to the energy management chip). An exception for +17 dBm and +14 dBm in optimal settings, but such settings cannot be guaranteed and some security must be taken. Optimal settings are described in the application note AN5457. This will be described better in Section 6.2.3

Therefore, the subGHz module will be used in low-power mode. This mode is optimized for 14 dBm. The current consumption is more reasonable. Choosing 3.3 V or 1.8 V results in the same power consumption ($V \cdot I = \text{constant}$). This is certainly generalized for any supply voltage between 1.8 V and 3.3 V. Finally, **1.8 V is chosen** as it is easier to obtain thanks to the energy management chip. One may expect lower loss thanks to a lower voltage.

Frequency band (MHz)	PA match (conditions)	Power output ⁽¹⁾	Typ	Unit
868 to 915	Low power (optimized for 14 dBm)	+14 dBm, $V_{DDRF} = 3.3$ V	23.5	mA
		+10 dBm, $V_{DDRF} = 3.3$ V	17.5	
		+14 dBm, $V_{DDRF} = 1.8$ V	41.5	
		+10 dBm, $V_{DDRF} = 1.8$ V	28.5	
	Low power (optimal settings) ⁽²⁾	+15 dBm, $V_{DDRF} = 3.3$ V	25.5	
		+10 dBm, $V_{DDRF} = 3.3$ V	15	
		+15 dBm, $V_{DDRF} = 1.8$ V	51	
		+10 dBm, $V_{DDRF} = 1.8$ V	25	
868 to 915	High power (optimized for 22 dBm)	+22 dBm, $V_{DDRF} = 3.3$ V	120	
		+20 dBm, $V_{DDRF} = 3.3$ V	107.5	
		+17 dBm, $V_{DDRF} = 3.3$ V	98	
		+14 dBm, $V_{DDRF} = 3.3$ V	92	
	High power (optimal settings)	+20 dBm, $V_{DDRF} = 3.3$ V	92.5	
		+17 dBm, $V_{DDRF} = 3.3$ V	58	
		+14 dBm, $V_{DDRF} = 3.3$ V	45.5	

Table 4.3: Current consumption depending on the power output and the supply voltage.

4.1.3 Energy manager

In this section, the utility and the functioning of an energy management chip will be discussed. Such chip extracts DC power from solar panels to simultaneously store energy in a rechargeable element and

supply the MCU with a DC voltage.

The chip used as an energy manager is the AEM10941 from *e-peas* [19]. It is really practical because it comes as a chip for cheap (5.22 €). A block diagram of the chip is presented in the appendix. And it also comes on an evaluation board (EVK10941) for the development phase. Jumpers on the evaluation board enable the selection of the correct configurations. The module will be used in Section 4.1.5 to test the solar panels. This section will focus on the chip, but other than the final schematic, the module has the same characteristics as the chip.

The *e-peas* module is not the only chip of this type available on the market, there is for example the EM850x chip from *EM Microelectronic*. However, the regular dealers do not have it or only sell it in bulk.

The *e-peas* chip extracts DC power from solar panels. However, when selecting the solar cells, the solar cell's voltage cannot exceed 5 V. The power is stored in a storage element, whose voltage cannot exceed 4.5 V. When enabled, the chip supplies two independent regulated voltages. An optional battery can be added to take over when the storage element is not charged enough.

Converters

The chip consists of an ultra-low-power boost converter followed by a buck convert in cascade to supply the voltage regulators. They have a high-power efficiency.

Theory of operations

When all nodes are discharged and no energy comes from the panel, the chip is in *deep sleep mode*. As soon as the required cold voltage (380 mV by default) and 3 μ W are available from the solar panel, the chip enters *Wake up mode*. The voltage of the buck converter (V_{buck}) rise up to 2.2 V and the voltage of the boost converter V_{boost} rises up to the over-charged voltage V_{ovch} . Then the storage element is charged thanks to the boost converter. When the charge of the supercapacitor reaches the "charge ready" voltage V_{chrdy} , the chip enters *normal mode*.

In normal mode, the supply voltages are enabled. All the power not consumed by the load is stored in the storage element. However, its voltage cannot exceed the over-charged voltage V_{ovch} . When it reaches it, the internal logic prevents it from further increasing. On the other hand, if its voltage decreases under the over-discharged voltage V_{ovdis} , the chips enter *Shutdown mode*. It waits for 600 ms and if the voltage has still not reached V_{chrdy} , the circuit returns to *deep sleep mode*.

Note that $V_{ovdis} < V_{chrdy} < V_{ovch}$. Their values are tunable.

Supply Voltage enable

The low-voltage supply and high-voltage supply are driven by highly-efficient LDO (low dropout) regulators for low noise and high stability. The LDO for HVOUT is placed directly behind the boost converter. While the buck converter is placed between the boost and the LDO for LVOUT. The outputs can be disabled, either both or only one, thanks to ENHV and ENLV pins (connected to V_{buck} or to ground).

The range of values that the output voltages can take are resumed in Table 4.4. Note that V_{batt} is the voltage of the storage element. Either of the output voltages can handle 1.8 V which is great. However, the maximum current is not the same. The LoRA module needs more than 20 mA. Therefore, LVOUT

is disabled and HVOUT will power the MCU with 1.8 V. The voltage of the supercapacitor must be at least 2.1 V.

	Vmin [V]	Vmax [V]	Imax [mA]
HVOUT	1.8	$V_{batt} - 0.3$	80
LVOUT	1.2	1.8	20

Table 4.4: Voltage range and maximum current outputted by HVOUT and LVOUT.

Configuration

Thanks to the 3 configuration pins, V_{ovch} , V_{chrdy} , and V_{ovdis} can be set. The typical usage is to use supercapacitors or batteries. The 3 possibilities for the supercapacitor are presented in Table 4.5.

CFG			Threshold voltages [V]			LDO		Typical use	$(V_{ovch}^2 - V_{ovdis}^2)$	
2	1	0	V_{ovch}	V_{chrdy}	V_{ovdis}	V_{hv}	V_{lv}			
1	0	0	2.7	2.3	2.2	1.8	1.2	single-cell supercapacitor	2.45	
0	1	1	4.55	3.67	2.8	2.5	1.8	dual-cell supercapacitor	12.86	
0	1	0	4.5	3.92	3.6	3.3	1.8	dual-cell supercapacitor	7.29	
0	0	0	custom mode							
			4.5	2.25	2.2	1.8	1.8	-	15.41	

Table 4.5: Threshold voltage values with respect to the value of CFG[2:0].

The fourth possibility is to make custom mode values. This is possible by adding 4 resistors as a voltage divider on the SET_OVDIS , SET_CHRDY , and SET_OVCH pins. However, some conditions must be respected, which does not let a big range for improvements.

$$\begin{aligned}
 V_{chrdy} + 0.05 &\leq V_{ovch} \leq 4.5 \\
 V_{ovdis} + 0.05 &\leq V_{chrdy} \leq V_{ovch} - 0.05 \\
 2.2 &\leq V_{ovdis} \\
 V_{hv} &\leq V_{ovdis} - 0.3
 \end{aligned}$$

To have the highest range of usable voltage supply, one wants the highest V_{ovch} value and the lowest V_{ovdis} and V_{chrdy} values. The maximum usable energy stored by the supercapacitor is $\frac{1}{2} \cdot C \cdot (V_{ovch}^2 - V_{ovdis}^2)$ assuming no charge from the solar panels (either energy used really quick or during the night). Notice that a dual-cell supercapacitor is 2 single-cell supercapacitors. The *e-peas* module has a balun pin to connect the connection of both capacitors in order to balance the internal voltage to avoid damaging the super-capacitor because of excessive voltage on one cell. It compensates for any mismatch.

The application needing 1.8 V capable of outputting 80 mA, the "100" CFG configuration is chosen. If the stored energy is not enough, one could consider the custom mode, using a dual capacitor cell as shown in the table. Notice that to have the same capacitance values, it requires 4 single-cell capacitors. It also requires adding 6 resistors on the PCB.

Maximum power point tracking

Two configuration pins (SELMPP[1:0]) allows selecting the MPP (maximum power point) ratio. The MPP ratio is defined as $MPP_{ratio} = \frac{V_{mpp}}{V_{oc}}$. Where V_{oc} is the open circuit voltage of the power source, and V_{mpp} is the voltage at which the solar panel has the highest power. The ratio can be chosen between 4 values: {70%, 75%, 85%, 90% }.

The boost converter is regulated thanks to an internal MPP tracking module. The voltage across the solar panels is continuously compared to V_{mpp} . If the voltage is too high, the boost converter is switched on and otherwise, it is switched off. This allows for the best power transfer.

4.1.4 Schematic and connections

This section will focus on the electrical connection of the *e-peas* chip. Figure 4.2 shows the complete schematic.

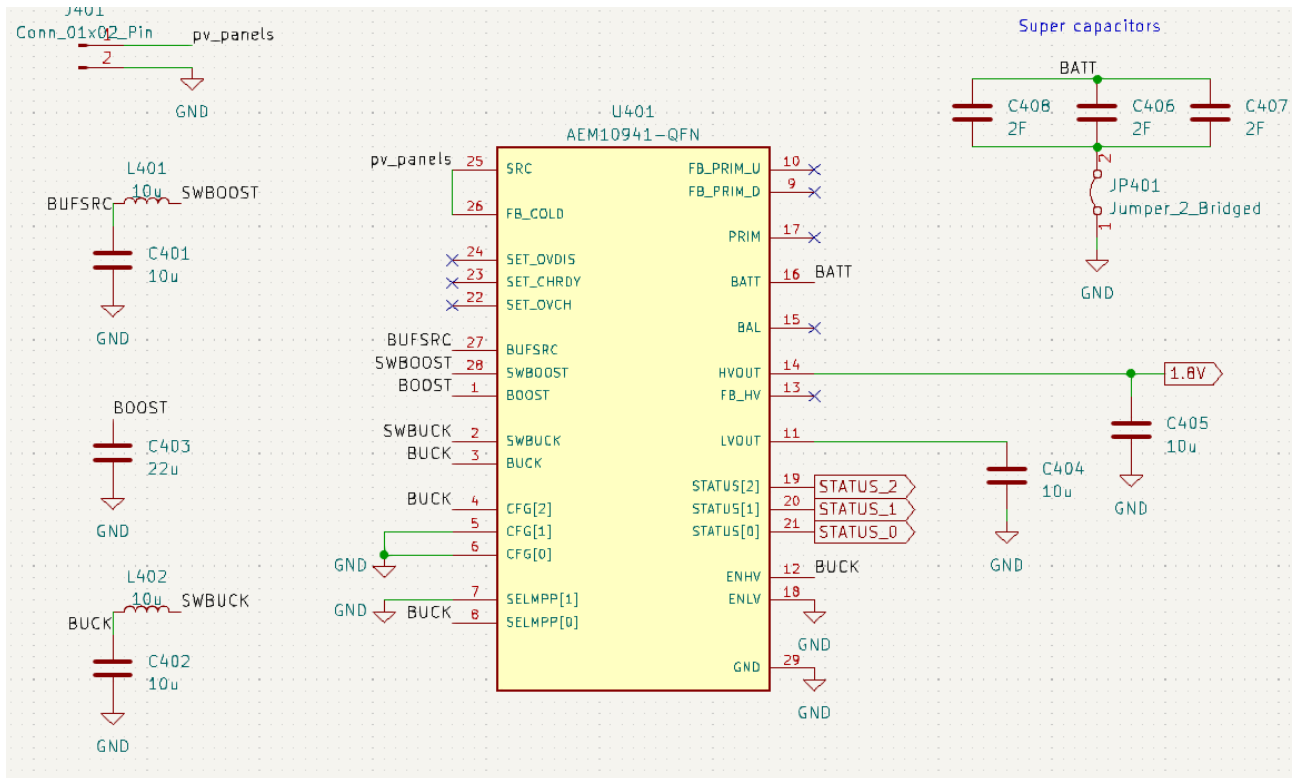


Figure 4.2: Energy management chip (*e-peas* AEM10941) schematic.

For the chip to work correctly, inductors and capacitors must be added. Those are essential elements for both converters. The value of those components is specified in the *e-peas* datasheet. It concern L401, L402, C401, C403, and C402. In addition, C404 and C408 are also capacitors recommended by the datasheet at the output of the LDO to stabilize the voltage.

The low voltage output is disabled while the high voltage output is enabled thanks to the ENLV and ENHV pins respectively. The ground is considered as a digital zero. And the value of the BUCK pin is considered as a digital 1. On the CFG pins, the value 100 is connected. It is not the custom mode, therefore, SET_OVDIS, SET_CHRDY, SET_OVCH, and FB_HV pins are left floating. And finally, the value 01 is selected in the SELMPP pins to have a ratio of 75%. This value is discussed in Section 4.1.5.

The storage element is composed of multiple supercapacitors in series which are connected to the BATT pin. A jumper is placed for an ampermeter to be used. They are single-cell supercapacitors so the BAL pin responsible for the balun is left floating. In addition, there is no backup battery so the PRIM pin is left floating as well.

On the SRC pin, the photovoltaic panels are connected. In the schematic, it is connected to a terminal block to be able to control the orientation of the panel.

The 3 status pins are connected to the microcontroller. Those pins carried out information about the mode the chip is currently in and notified when the supply voltage is too low. And the voltage at HVOUT is the supply voltage of the microcontroller.

4.1.5 Indoor solar panels

In this section, the focus is made on the solar cells selection. Solar cells can be classified based on their material. Two big types stand out. Amorphous (or thin film) and Crystalline solar cells.

Amorphous solar panels are based on a glass-like solid surface (the substrate) on which an extremely thin layer of non-crystalline semiconducting materials is deposited. That material exhibits a rectifying behavior and is suitable for photovoltaic applications. Examples of the material are copper indium gallium selenide (CIGS), cadmium telluride (CdTe), or hydrogenated amorphous silicon (a-Si:H). They are really cheap and have a greater tolerance to shade thanks to the absorption of a wider band of the light spectrum. However, it does not have the highest efficiency ($\approx 10\%$).

On the other hand, crystalline is the older, classic technology. Within this type, they are 2 categories: monocrystalline and polycrystalline. The difference is during the purification of silicon. For polycrystalline, when the material passes from the liquid state to the solid state, several grains (pure crystals) form and clump together (forming grain boundaries between the grains). While for monocrystalline, the molten silicon solidifies by epitaxy, causing the initial crystal to grow larger and larger, resulting in silicon formed from a single crystal. They are both found on rigid panels, but monocrystalline is the more efficient ($\approx 20\%$ against $\approx 16\%$). They must be placed as perpendicularly as possible to the light source and do not react well to shadow.

The nodes will be placed in a building so a classic solar panel will not be well suited. One must search for solar panels suitable for indoor applications. Three solar cells are selected. They are presented in Tables 4.6 and 4.7.

Supplier	Name	Technology	Price [€]	Area[mm ²]	$V_{oc,max}$
Power Film	LL200-2.4-37	Amorphous	4.85	54 x 36.5	3.7 V
Panasonic	AM-1454CA	Amorphous	3.76	41.6 x 26.3	2.5 V
Anysolar	KXOB201K04F	Monocrystalline	6.48	42 x 23	2.76 V

Table 4.6: Indoor solar panels technical comparison.

The first two use amorphous technology while the last uses monocrystalline cells. One notices that in fact, the monocrystalline technology is more expensive. Regarding the area, one wants to minimize it. In fact, the node must be as small as possible to be unnoticed in the room it is placed in. However, one also wants to maximize the energy harvesting which requires the maximum area. The limit was set to 60 x 60 mm².

A second important criterion is the maximum open circuit voltage ($V_{oc,max}$) which is the upper limit of the voltages on the solar panel terminals. As seen in the previous section, this voltage cannot exceed 5 V.

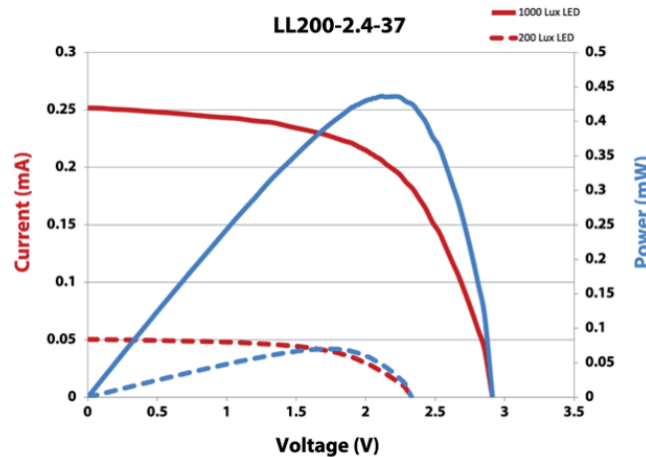
In Table 4.7 their theoretical performances are compared. However, the values are not given in the same conditions, which limits the comparison. But it, however, gives an idea of the order of magnitude of the power to expect.

	Illuminance [lx]	V_{op}	I_{op}	$P = V_{op} \cdot I_{op}$	Power/Area	MPPT ratio
PowerFilm LL200-2.4-37	200	1.6 V	43 μA	0.068 mW	3.45 $\mu W/cm^2$	$\approx \frac{1.8}{2.3} = 78\%$
	1000	2.1V	214 μA	0.449 mW	22.78 $\mu W/cm^2$	$\approx \frac{2.3}{2.9} = 79\%$
Panasonic AM-1454CA	200	1.5 V	33 μA	0.049 mW	4.48 $\mu W/cm^2$	$\approx \frac{0.5 \cdot 4}{0.63 \cdot 4} = 79\%$
Anysolar KXOB201K04F	1 sun \approx $10^5 lx$ at noon	2.23 V	78.7 mA	175.6 mW	18.18 mW/cm ²	$\approx \frac{2.23}{2.76} = 80\%$

Table 4.7: Indoor solar panels: theoretical performances comparison.

PowerFilm panel

For the PowerFilm panel, values are given for 200 lx and for 1000 lx . Notice that 200 lx is in the lower range for a well-illuminated room, but 1000 lx is higher than the upper range (around 700-800 lx). V_{op} and I_{op} are respectively the operational voltage and the operational current. They are a point of the V-I curve that makes the link between them, given in the datasheets. In Figure 4.3 one sees in red the current as a function of the voltage and in blue the power. The power is simply the product of I and V. The dashed line is for 200 lx and the solid line is for 1000 lx .

Figure 4.3: PowerFilm solar panel, current with respect to the voltage under 200 and 1000 lx .

Firstly, one sees that whatever the illuminance, the curves are similar within a scale factor. In fact, the higher the illuminance, the higher the voltage and current. And therefore the higher the power. The best operation point is the MPP which is the point with the maximal power. This point depends on the illuminance. The points in the Table are not the MPPs but they are relatively close.

The maximal power harvestable corresponds to the largest rectangle that can be put under the red curve. For the *e-peas* chip, the MPP ratio must be computed. It is the ratio between the voltage at MPP and the open circuit voltage. It is comprised between 0 and 1 by definition and it is generally between 70% and 95% in practice.

$$MPP_{ratio} = \frac{V_{MPP}}{V_{oc}}$$

One notices that the MPP_{ratio} stays relatively constant with respect to the illuminance. The closest allowed value of the *e-peas* module is 75%. Ideally, the curve looks like a step. In practice, it is curvier, the operating current is lower than the short circuit current and the operating voltage is lower than

the open circuit current. This is characterized by the fill factor (FF) which is the ratio between the rectangular area and the full area underneath the curve.

The power harvested is really low as expected by the low illuminance available.

Panasonic panel

Concerning the Panasonic panels, they are amorphous panels with a glass substrate. The chosen model has 4 cells in series, each of them measuring approximately 2.7 cm^2 . The V-I characteristics of a cell at 200 lx are shown in Figure 4.4. The values in the table are not at the MPP either. Looking at the 2 measurements at 200 lx , the Panasonic has a better power production when reported in the same area. However, they have the same order of magnitude.

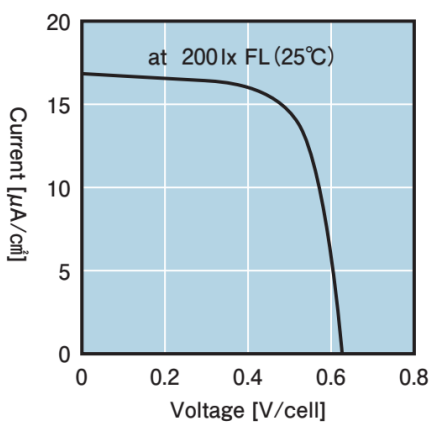


Figure 4.4: Panasonic solar panel, current as a function of the voltage, under 200 lx .

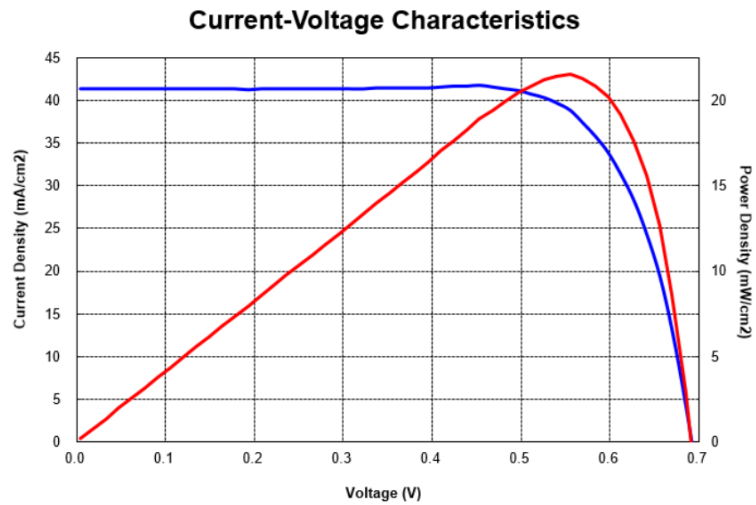


Figure 4.5: Anysolar solar panel, current a function of the voltage, under the sun.

Anysolar panel

Finally, the Anysolar panel. While the previous panels were designed for indoor applications, those panels are not designed especially for indoor. But are functional in low-light conditions as well. They have a high efficiency of 25%. Test conditions are done under the sunlight. It is therefore not representative of the power harvested by a node when placed in a classroom. However, it gives an order of magnitude for the MPP_{ratio} . The V-I curve is given in Figure 4.5. Notice that under the sunlight, the panel produces 18.18 mW/cm^2 at the operating point. As expected, it is really big compared to $\approx 4 \text{ μW/cm}^2$ generated by the 2 other panels under 200 lx .

Experiments

To better compare the panel and to have a better idea of the energy one can harvest in one day, one must test them in similar conditions. The tests are made thanks to the *e-peas* module. For each panel, the same operations are followed.

1. Reset the *e-peas* module by connecting each reset pin to GND. This will discharge all the capacitors present in the module.
2. Select the correct configuration:
 - Single-cell capacitor

- 75% MPP
 - Both output voltages (HVOOUT and LVOOUT) disable
3. Connect the supercapacitor as the storage element
 4. Connect the solar panel as the source

The setup is placed in a controlled area of 650 lx . A picture of the setup is represented in Figure 4.6. The area is protected from any other light sources as sunshine. Between each experience, the lamp is not moved.

Each panel is placed for 9 hours, and each hour, the voltage across the capacitor is reported as represented in Figure 4.7. Nine hours represents the number of hours a room is expected to be illuminated if it is used all day.



Figure 4.6: The panel test setup.

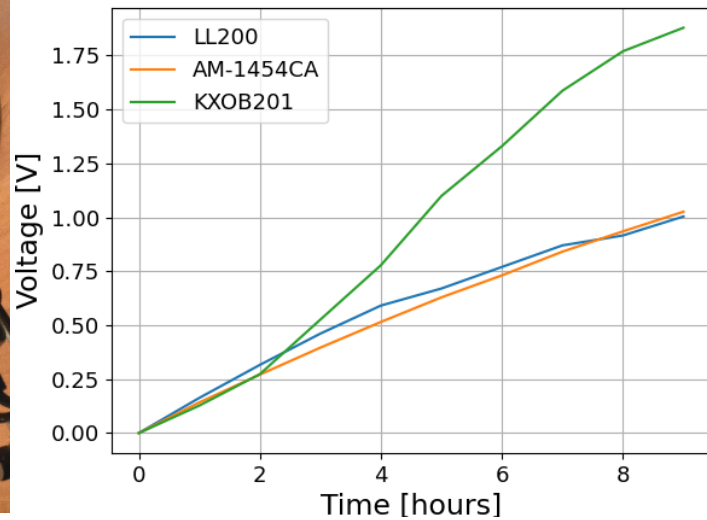


Figure 4.7: Evolution of the voltage of the supercapacitors with respect to time.

One sees that the voltage increases with time. The real interesting value is the energy harvested. Knowing that the supercapacitor has a capacitance of 2 F , the energy is easily computed thanks to the following equation:

$$E = \frac{1}{2} \cdot C \cdot V^2$$

The energy with respect to time is reported in Figure 4.8. The panels have different sizes, therefore, for a better comparison, the energy has to be reported per unit area [cm^2]. The Panasonic and the Anysolar panels have approximately the same area, while the Power Film panel is almost 2 times bigger in area.

In the beginning, less energy is collected, this is because the buck and the boost converters embedded in the *e-peas* module must be charged and that consumes energy. Then the energy brought for each hour is more or less constant. In fact, it follows a line.

The best panel is by far the Anysolar one. After 9h, it collected 3.5 J while the other only collected 1 J .

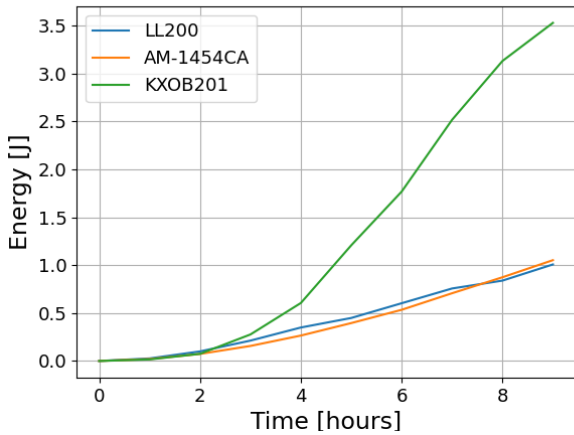


Figure 4.8: Evolution of the energy harvested by panels with respect to time.

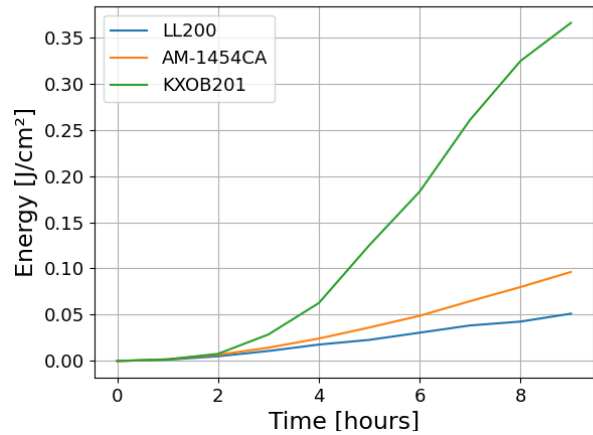


Figure 4.9: Evolution of the energy per unit area harvested by panels with respect to time.

If the energy collected is insufficient, placing multiple panels may be considered. Panels may be placed in series, in parallel, or in a combination of series and parallel. It is illustrated in Figure 4.10. When panels are in series, the voltages add up, and the current stays the same. Notice that the panels are never exactly the same and are illuminated the same way. Therefore, the final current is the lower current when considering each panel individually. This means that if a panel is not well exposed, the entire set will be inefficient. Furthermore, in that configuration, the voltages are summed, and the *e-peas* module only accepts a voltage of 5 V. Panels in series are therefore not suitable for this application. For the same reasons, a combination of series and parallel is not suitable either.

On the other hand, if the panels are set in parallel, the voltage stays constant, and the current sums up. This is the solution if multiple panels are needed to increase the energy potential. To know if it is needed, an energy requirement estimation is done.

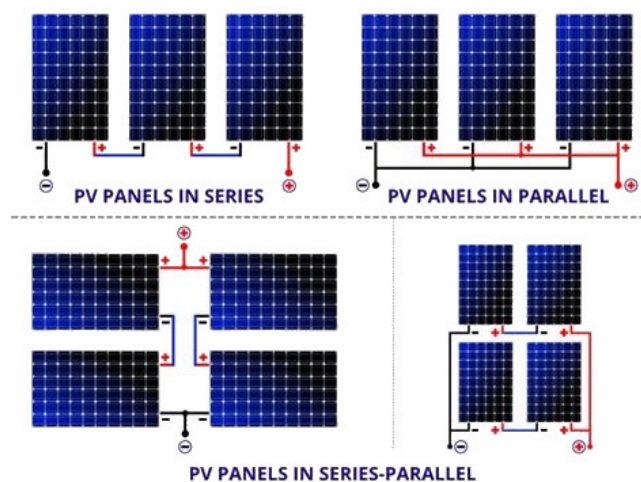


Figure 4.10: Series, parallel and series-parallel connection of solar panels.

Notice that each of the three suppliers offers the panels in multiple size declination by playing with cells in series and in parallel.

4.2 Energy requirement estimation

In this section, the energy needed for one LoRa package transmission will be estimated. Energy is needed to acquire data from the sensor and then send the package. Table 4.8 is a rough estimation of the energy needed to send one packet. Values are found in the datasheets. The duration of the LoRa transmission is a rough approximation combining the time of flight computed for a payload of 8 bytes, the time for the module to wake up, ... Because not a lot of bytes are sent, the data transfer is really quick.

Notice that the estimation is lower than the actual consumption because it does not take into account the energy used by the microcontroller. For the i²c communication with the sensors or the SPI communication with the LoRa module for instance.

On the final node, by comparing the energy in the supercapacitor before and after transmission, this value can be better estimated.

	Vdd [V]	I	P = U I	t	E = P t
Temperature and humidity sensor	1.8	24.4 μA	44 μW	41 ms	1.8 μJ
Ambient light sensor	1.8	3.7 · 2 μA	13.3 μW	201 ms	2.7 μJ
Microphone	1.8	185 μA	333 μW	≈ 2 s	0.67 mJ
LoRa transmission	1.8	41.5 mA	75 mW	≈ 200 ms	15 mJ
Total					15.7 mJ

Table 4.8: Energy requirement estimation.

4.3 Communication scheduling

In ideal conditions, the *e-pease* module acquires up to 3 J in 9 h. However, this is almost optimal conditions. The panel is directly illuminated with 600 lx. In practice, being flat on a desk is 400, and laying on the wall is around 200 lx. Furthermore, the light is not systematically on every day. Therefore, as a first approximation, the node sends data every hour for 12 h hours (8h - 20h) and then does not transmit during the night. Therefore the solar panel must at least harvest 0.2 J. This schedule may be updated when more data about the energy harvest in practice is available. And when the energy required for transmission is estimated thanks to the final node.

Chapter 5

Communication design: Antenna

The node will send data to the gateway. However, to send data, an antenna is required. This section will focus on the choice and design of a PCB antenna. The antenna should be a good transmitter and receiver at 868 MHz. First, the basics of antennas are presented, then the impedance matching issue is considered and solved thanks to the Smith charts. And finally, 2 different antennas are designed and then tested on a PCB.

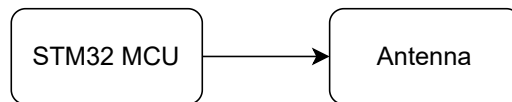


Figure 5.1: Antenna connection diagram.

5.1 Slight overview of antenna technology

An antenna is a configuration consisting of conducting components, typically wires or a collection of wires. It is utilized for the conversion of high-frequency electrical currents into electromagnetic waves and vice versa. In other words, it is a transducer converting current waves into electromagnetic (EM) waves.

The ideal length of an antenna depends on the wavelength of the incoming signal. The wavelength depends on the frequency of the radio signal and the speed of the wave. Radio wave moves at the light speed c and LoRa communication is done at 868 MHz. Therefore, one can compute the wavelength λ .

$$\lambda = \frac{c}{f} = \frac{3 \cdot 10^8 [m/s^2]}{868 \cdot 10^6 [Hz]} = 0.345 m = 34.5 cm$$

The ideal length of an antenna height is a quarter of the wavelength $\frac{\lambda}{4} = \frac{34.5}{4} = 8.6 cm$ at 868 MHz [20]. Therefore, the mechanical length is inversely proportional to the numerical value of the frequency. In practice, the length is greater because of the difference between the physical length and the electrical length (due to the dielectric constant of the substrate). For isotropic antennas, if they are good transmitters, they are also good receivers.

In this part, 2 antennas will be presented: the dipole and the monopole antenna. They are both widely used technologies.

A dipole is a basic antenna with 2 arms. Each arm is $\lambda/4$ long. It is symmetrically fed at the middle. The voltage is technically equal to zero at the center and at its highest (in absolute value) at the edges. On the other side, the current is zero at the edges and at its highest in the middle.

When a signal arrives on an ideal dipole antenna, it travels down the first arm until the end of the other arm.

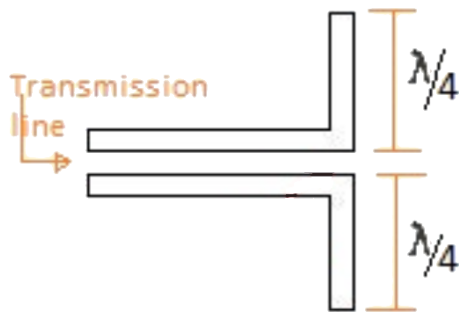


Figure 5.2: Dipole antenna [7].

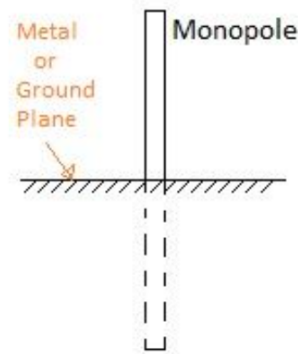


Figure 5.3: Monopole antenna [7].

A monopole is one arm of a dipole over a copper ground plane which serves as the second arm. The single arm has an image through a metal or ground plane. It, therefore, acts as a dipole. It is smaller and typically used for PCB antenna

5.2 Impedance matching

Importance of impedance matching:

When an antenna sends a signal, one wants the receptor to receive the maximum power from the original signal. This section will focus on how to achieve this objective.

The antenna sending the signal is called the emitter. And the antenna receiving the signal is the receptor. An RF line is modeled in Figure 5.4. If the system is not well designed, part of the incident wave will be reflected at the load and it does not work in optimal conditions. This will limit the power transfer.

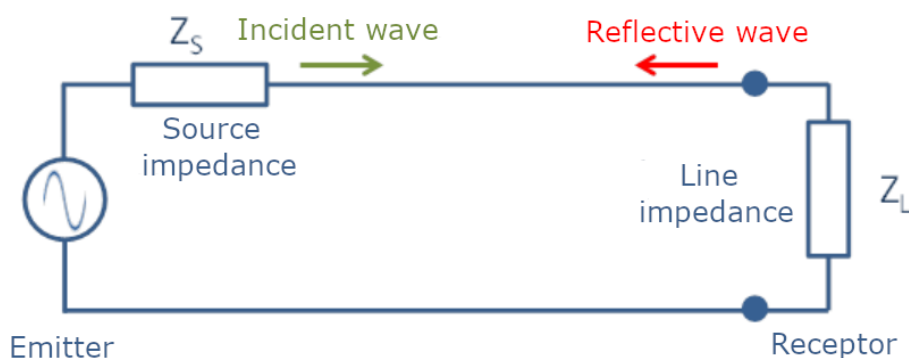


Figure 5.4: RF transmission line representation.

To transfer maximum power, the source impedance and the load impedance must be matched. There exist two types of matching: conjugate matching and reflectionless matching. In conjugate matching, the load impedances should be the complex conjugate of each. By doing so, the maximum power is transferred. While in reflectionless matching, the two impedances are equal, this ensures no reflection of the wave.

If the impedance to match has no imaginary part, the two matching amounts to the same thing. On the other side, when it has an imaginary part, counterintuitively, no reflection does not mean the best

power transfer.

Notice that the impedance before matching is generally complex and thus depends on the frequency. The matching must be done at the application target frequency. Matching is generally done thanks to a pi-network. This is a passive element in series (inductor or capacitor) surrounded on either side by a passive element (respectively capacitor or inductor) in parallel. A famous tool for impedance matching is the Smith chart. Its functioning is described in the Appendix.

Practical consideration

In this application, the receptor is the antenna of the gateway. It is designed to have an impedance of 50Ω . This is generally the case for every antenna. The target impedance is therefore not complex.

To match the impedance of the emitter, one must start by estimating its current impedance. This can be done thanks to a VNA (vector network analysis) [21].

If the initial design of the antenna does not have an impedance of 50Ω , matching is required. However, the signal does not only pass through the antenna, it also passes through a track between the module and the antenna. This track must be matched to the 50Ω as well.

5.3 Creation of an antenna on a PCB

This section aims at selecting the antenna for the node. Parameters to keep in mind when choosing an antenna are the size, the cost, and the performance. The antenna will be placed on the node PCB. This section starts with a comparison of the antenna type. Then 2 PCB antennas are presented. They are drawn, printed, and tested on a PCB.

5.3.1 Antennas types

When designing an antenna to connect to a PCB, they are 3 main types:

- Whip antennas:
They are external antennas with a connector, SMA for instance. If size and cost are not a concern, whip antennas are a great solution. They can have really good performances. The cheapest version consists of a wire of length $\lambda/4$ connected at one extremity to the ground plane of the PCB.
- PCB antennas:
An antenna is created thanks to a trace of length $\lambda/4$ on the PCB. It is connected to the feeder (RF track) and a ground plane to reflect the second part of the antenna. In this setting, the antenna is not exposed to the external world. This technique is really low cost, but it takes more space than a wire. However it is printed directly on a PCB, therefore it is more robust. It is really easy to integrate, good for single-band applications and has perfect repeatability. On the other side, it needs to be simulated to correctly design it and then be tested on a prototype.
- Chip antenna:
This is a compact and miniaturized antenna with a connector. Both the antenna and the conductor are assembled in a small IC package, so it becomes very advantageous. It has a medium cost and performance.

	Whip	PCB	Chip
Size	○ ○ ○	● ○ ○	● ● ●
Cost	● ○ ○	● ● ●	● ● ○
Performance	● ● ●	● ● ○	● ○ ○
	worst: ○ ○ ○, best: ● ● ●		

Table 5.1: Antenna types comparison.

The final choice fell on the PCB antenna because they are the cheapest while keeping good performances. They are also really easy to integrate. To avoid part of the design and simulation process, antennas will be based on existing and well-known good designs.

5.3.2 PCB antennas

Designing a performing antenna is not straightforward. Texas Instruments provided an antenna selection quick guide which presents about twenty antennas. One must select the correct frequency, in other words, 868 MHz. Based on the efficiency and size, the two best antennas have been retained: the meander monopole antenna and the compact helical antenna. Those 2 antennas must be connected to the feeder. Table 5.2 compares them on several comparison points. Another comparison point is the radiation patterns. They are provided in the appendix. They are very similar for both antennas in any direction.

	Frequency [MHz]	Bandwidth [MHz]	Efficiency	Dimensions [mm ²]
Meander monopole	868	88	98%	38 x 25
Compact helical	868	37	63%	19 x 22

Table 5.2: Meander monopole antenna and compact helical antenna characteristics [10, 11].

Meander monopole antenna

The meander monopole is the most efficient (with 98 %). It is the recommended antenna whenever the space is available. It also has a bigger bandwidth which lets more margin for variation in the frequency. It is therefore more robust.

A meander antenna is designed by folding the conductors back and forth so that the antenna can be made shorter. This design results in a smaller compact size but with the loss of radiation resistance, efficiency, and a decrease in bandwidth compared to a straight antenna. It is represented in Figure 5.5.

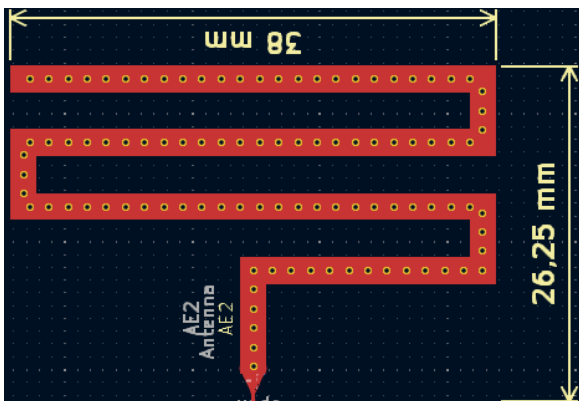


Figure 5.5: Meander monopole antenna design.

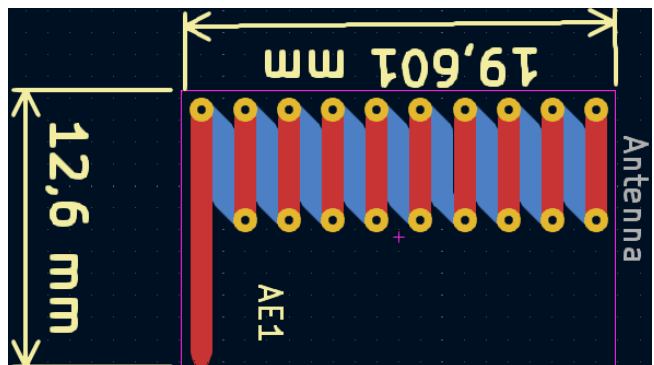


Figure 5.6: Compact helical antenna design.

Impedance is close to 50 Ω even without matching but it is still better to have external components for

impedance matching. The design note (DN024) already compute matching component values. Notice, due to physical imperfections, the matching might need to be slightly adapted.

Compact helical antenna

On the other side, the helical antenna is less efficient, but it is way smaller. It uses a quarter of the area of the meander monopole antenna. It is the second recommended antenna. The design is shown in Figure 5.6. By itself, the antenna does not match 50Ω at 868 MHz . Hopefully, the design manual also provides a matching network.

5.3.3 PCB design

Now that the 2 antennas have been selected, they must be tested. To do so, a PCB is designed. This section details all the considerations and precautions for the design. The PCB is a 4 layers PCB which embedded both antennas and connectors to test them.

Schematics

To test correctly the antennas, one must connect the pi-matching network to the antenna. The values for both designs are found in the design note. They are represented in Figure 5.7 for the meander monopole antenna and in Figure 5.8. In addition to the antenna and the pi-matching, a connector is needed to test the antenna. A simple pin will completely ruin the matching. Therefore a u.FL connector is used. It is a widely used coaxial connector. It has a high-frequency performance up to 6 GHz . And most important, has a characteristic impedance of 50Ω . Thanks to this connector, the antenna will be connected to a VNA for testing.

One might notice a difference between the 2 schematics. For the helical antenna, there are 2 connectors. Each one can be connected to the antenna thanks to a 0Ω resistor. This is to test the impact of a longer path between the antenna and the connector.

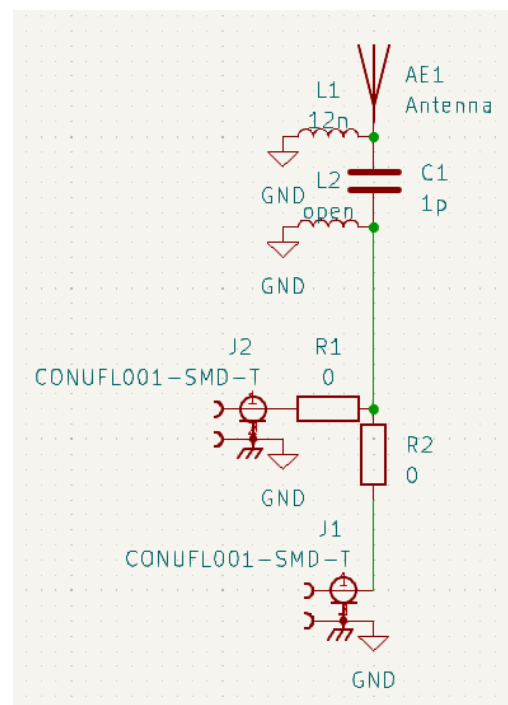
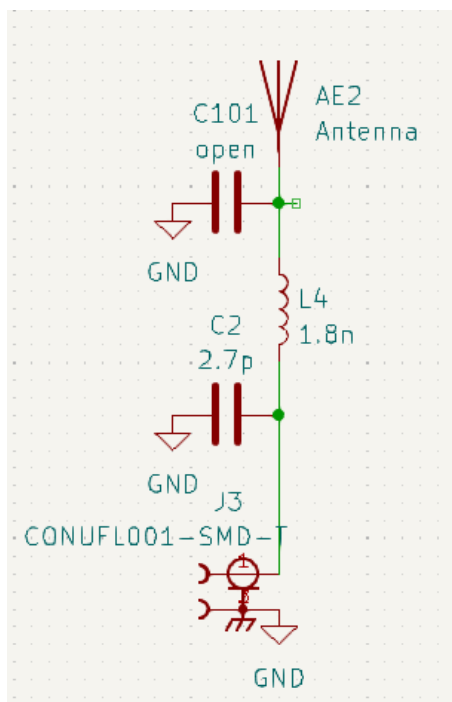


Figure 5.7: Meander monopole antenna schematic. Figure 5.8: Compact helical antenna schematic.

General consideration

For the antennas to correctly work, they must be isolated from the rest of the circuit. That means that the area they cover, and its surroundings, must be blank of anything. No components, no tracks, and no ground plane. In addition, for the arm of the monopole antenna to correctly reflects, a ground copper plane is placed at the base of the antenna. The ground plane stretches in the opposite direction. The ground plane must at least be on the top and bottom layers. In practice, layers 2 and 3 also have ground planes. Obviously, those zones are interrupted for the tracks to pass. And multiple vias are used to avoid a piece of the ground plane between two tracks to form an antenna.

For the signal to be correctly transmitted, the antennas must match each other, but the transmission line must match as well. In fact, the signal output by the LoRa module must be transmitted to the antenna alimentation thanks to tracks. One must therefore design the line to have an impedance $Z_L = 50 \Omega$. The width of the trace w is the unknown of the problem.

The transmission line is a copper trace on the PCB on top of a copper ground plane with a dielectric between them. This is called a microstrip and is represented in Figure 5.9. Finding the adequate length of the copper trace can be done thanks to the following approximations. It provides a solution within a 10 % error. L_{unit} is the inductance by unit of length and C_{unit} the capacitance by unit of length.

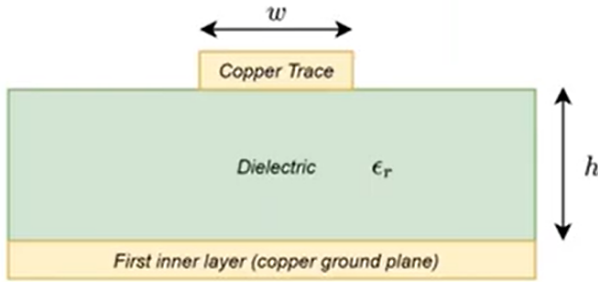


Figure 5.9: Sectional view of a microstrip.

$$Z_L \approx \sqrt{\frac{L_{unit}}{C_{unit}}} \quad (5.1)$$

$$C_{unit} \approx \frac{\varepsilon_0 \cdot \varepsilon_r \cdot (w + 2h)}{h} \quad (5.2)$$

$$L_{unit} \approx \frac{1}{\nu^2 \cdot C_{unit}} \quad (5.3)$$

$$\Rightarrow w \approx h \cdot \left(\frac{1}{Z_L \cdot \nu \cdot \varepsilon_0 \cdot \varepsilon_r} - 2 \right) \quad (5.4)$$

With,

- ε_0 : permittivity of the air
- ε_r : relative permittivity of dielectric material
- $\nu = \frac{c}{\sqrt{\varepsilon_r}}$: the speed of light in the dielectric (with c the speed of light in the air)
- h : height between the outer layer and the ground layer

The first 3 equations are found theoretically and combining them gives an estimation of the trace width. Equation 5.1 is the impedance of the line, ignoring the copper trace resistance. And equation 5.2 is the capacitance of a parallel plate capacitor. The term $2f$ stands for fringing because the trace has a finite width w while the ground plane underneath is considered infinite.

For the final Equation 5.4, one sees that the width is proportional to the distance between the trace and the first copper layer. In fact, if the ground is closer the parasitic capacitor is smaller. The thinner the line the easier to put on a PCB. Therefore, the final PCB is chosen to have a 1.6 mm thickness and 4 layers. This enables $h = 0.2104 \text{ mm}$ instead of 1.53 for a classical 2 layers PCB. The PCB dielectric is prepreg 7628 which has a dielectric constant $\varepsilon_r = 4.6$.

Layer	Material Type	Thickness	
Layer	Copper	0.035mm	
Prepreg	7628*1	0.2104mm	
inner Layer	Copper	0.0152mm	1.1mm (with copper core)
Core>	Core	1.065mm	
inner Layer	Copper	0.0152mm	
Prepreg	7628*1	0.2104mm	
Layer	Copper	0.035mm	

Figure 5.10: PCB stack-up [8].

The approximation gives a width $w = 0.32 \text{ mm}$. This value can be compared to impedance calculators. The *JLCP* one finds a width of 0.349. And *Mantaro* calculator finds 0.341. Therefore tracks by which passes the RF signals have a width of 0.3 mm. The resulting impedance will be slightly to big.

Theoretically, the track width influencing the power transfer is an important phenomenon when the track is at least $\lambda/10$ long. It is therefore non-negligible for tacks of 3.45 cm. In the application, tracks are smaller, but it is still a good practice to take them into account.

To limit even more the parasitic capacitances, RF lines are surrounded by vias connected to the ground. This creates a shield with the copper plane already present on layers 2, 3, and 4.

Meander monopole antenna

The trace of the meander monopole antenna is on the top layer of the PCB and also on the bottom layer. Having the two sides, compare with only one side, optimize the antenna's performance thanks to a lower resistive loss and a slightly wider bandwidth. The two tracks are regularly connected using vias. The antenna must be isolated from the rest of the circuit and must have a ground plan at the bottom part for the monopole conductor to be reflected.

The footprint of the antenna was made on *KiCad* strictly following the documentation dimensions. The antenna is actually designed too long. The last straight conductor should be cut in the middle for the antenna to work at its best at 868 MHz. Having it too long ensure that the antenna is sufficiently long even with imperfection. The antenna can then be shortened to better match the frequency afterward. This will be developed in the following section.

Compact helical antenna

This second antenna follows a helical design. The trace starts on the top layer, starts "half a circle" with one line, changes layer thanks to a via to the bottom layer, and then a line comes back to the same vertical line and a slight offset of the horizontal line. This is repeated multiple times. The design strictly respects the data provided in the application report. The footprint was actually already present in *KiCad* libraries. As for the previous antenna, it needs to be connected to the feeder.

Final PCB

The final PCB is presented in Figure 5.11. The lighter central stripe is the ground plane. On the edges, both antennas are well isolated. The other noticeable element is the long trace of 607 mm at the center of the PCB. It has a connector on both sides which allows testing the impedance of the line, which should be 50Ω at 868 MHz.

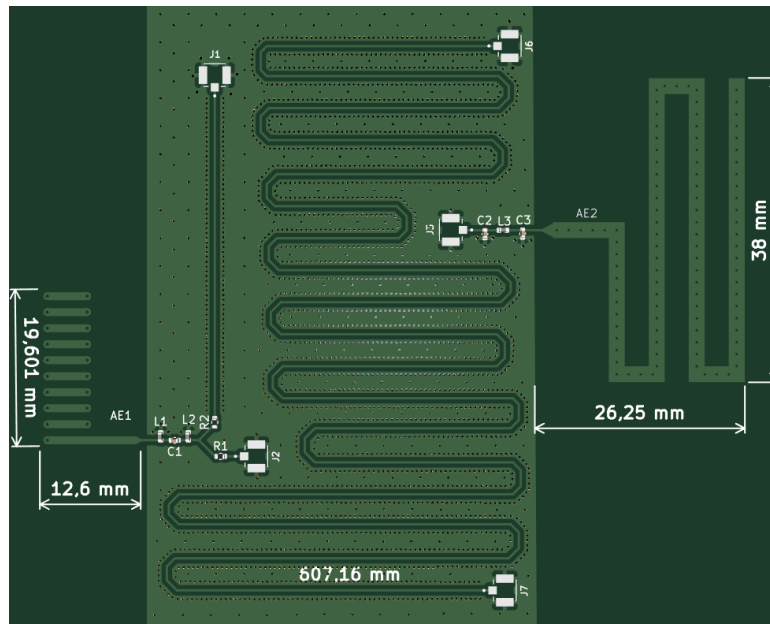


Figure 5.11: Design of the test PCB for the antennas.

5.4 Performance analysis

Testing antennas can be done thanks to a VNA. A VNA is a vector network analyzer. It measures the magnitude and phase of multiple frequencies. To do so, it characterizes the components thanks to swept-frequency and swept-power test signals. It can compute the impedance of the network as well, based on those two results. It represents it on a Smith chart. It also measures the incident, reflected, and transmitted waves that travel along transmission lines. The system is represented thanks to the scattering parameters (S-parameters) and standing wave ratio (SWR). The reflected waves are due to the mismatch. A VNA is used for antenna testing but can also be used for filter design among many other features.

The S-parameters characterize the input-output relation between 2 ports. SWR is defined as $\frac{1 + |\tau|}{1 - |\tau|}$ with τ the reflection coefficient. The lower the SWR the better. The reflection is therefore characterized by S11, S22, and SWR. While S21 and S12 characterize the transmission.

Caution when testing the antennas not to put a hand on the antenna or to put it too close to the electrical circuit (laid flat on the tiro-clas table for instance). Otherwise, they act as pseudo-antenna and interfere with the PCB antenna.

5.4.1 VNA calibration

Before using the VNA on the antennas, it must be calibrated. In fact, to connect an antenna to the VNA, multiple connectors and wires must be used. Those elements are not perfect and change the measure impedance. The VNA proposes a calibration utility to compensate for the line impedance and characteristics.

To do so, one needs a calibration kit. On this kit, the connector is connected on an open circuit, then a short circuit, then a 50Ω circuit. The kit used is presented in Figure 5.12. For each configuration, the impedance after calibration is represented on the Smith diagram by the VNA. Those are represented respectively in Figures 5.13, 5.14, and 5.15. The calibration was correctly effectuated. In fact, for an open circuit and whatever the frequency, the impedance is infinite. For the short circuit, the impedance is zero, and for the load, the points are centered at the 1 impedance value.

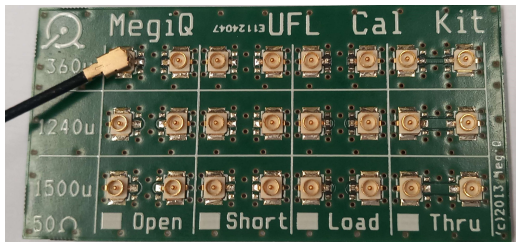


Figure 5.12: Calibration kit.

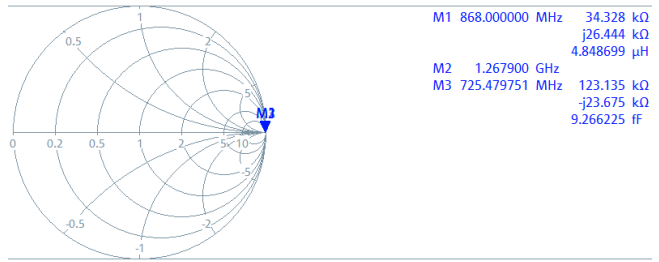


Figure 5.13: Impedance of an open circuit after calibration of the VNA.

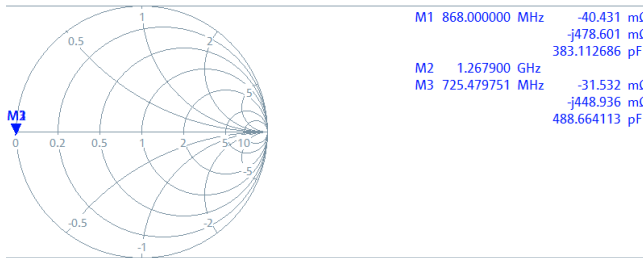


Figure 5.14: Impedance of a short circuit after calibration of the VNA.

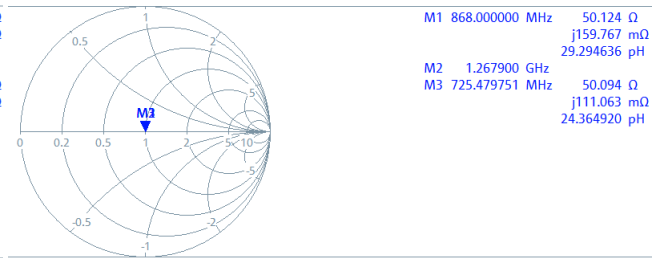


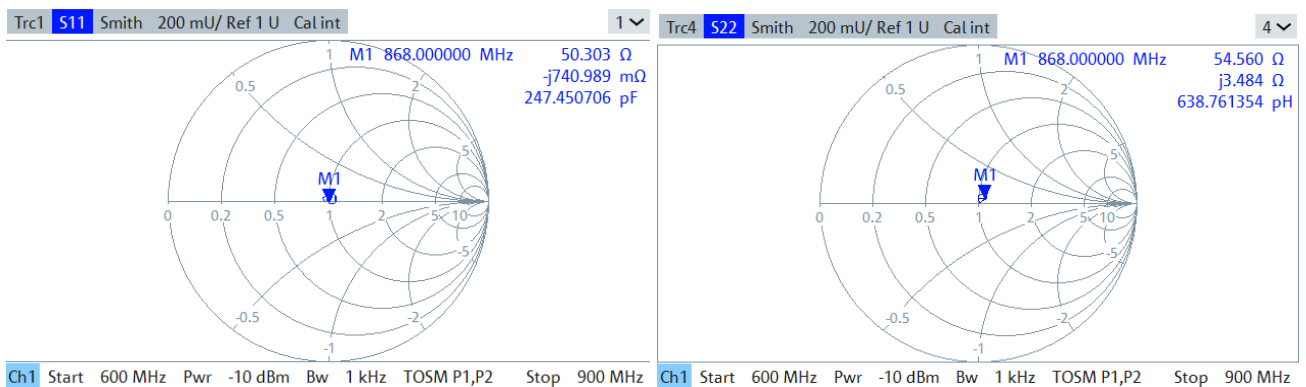
Figure 5.15: Impedance of a 50 Ω load after calibration of the VNA.

5.4.2 Track impedance

Before testing the antenna, the impedance of the long track of 607 mm is tested. To do so, both ends of the line are connected to the VNA thanks to the u.FL connector. The impedance view from each connector is represented in Figures 5.16a and 5.16b. It corresponds to the S11 and S22 parameters on Smith charts.

The impedance at 868 MHz are respectively $50.3 + j0.74 \Omega$ and $54.56 + j3.48 \Omega$. First notice that they are really close to the 50 Ω target impedance at the center of the chart. However, they are not exactly the same, this may be caused by a poorer mechanical connection of the u.FL connector either in the calibration phase or in the test phase.

Furthermore, one notices that the impedance has almost no variation with the frequency. This is an expected observation as the track width in the approximation does not vary with the frequency. The VNA tested frequency from 600 to 900 MHz.



(a) First side (J7 connector on the PCB).

(b) Other side (J8 connector on the PCB).

Figure 5.16: Impedance of the 607 mm long track in the Smith chart

5.4.3 Meander monopole antenna

The antennas can now be tested. Starting with the meander monopole antenna. The target frequency is 868 MHz . Therefore the span of frequency is set from 650 MHz to 950 MHz .

For the first test, the S11 parameter is shown in Figure 5.17. The antenna radiates the best at 805 MHz . The frequency is too low. This means that the antenna is too long. In fact, the electrical length of the antenna works the best at $\lambda/4$. The wavelength is inversely proportional to the frequency, therefore an increase in frequency results in a decrease in the wavelength and thus, of the antenna. Notice that due to the meandering and parasitic resistance, the physical length is longer than the electrical one.

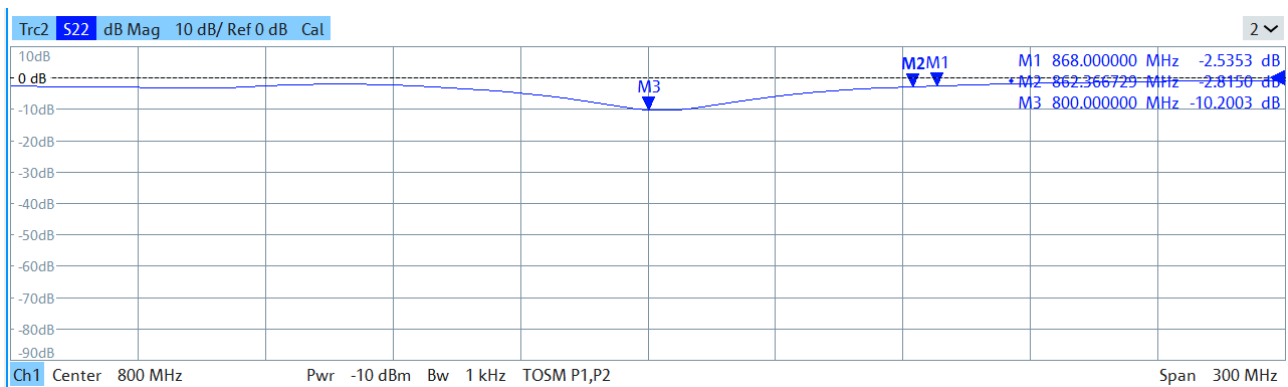


Figure 5.17: Meander monopole antenna S11 with no cut in the antenna.

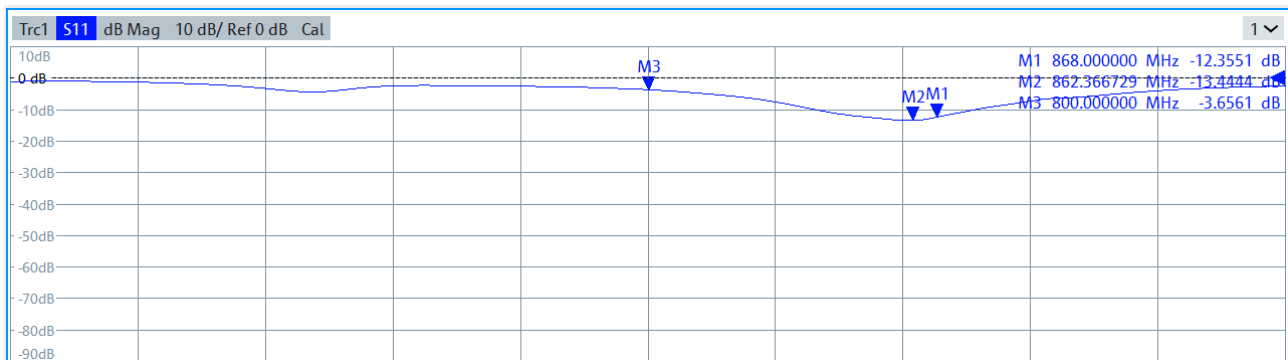


Figure 5.18: Meander monopole antenna S11 with a cut that shortens the antenna by 16 mm .

To match the radiation frequency to 868 MHz , the antenna is carefully cut and tested until they match. Caution not to cut too much at a time because there's no turning back. The final S11 signal is shown in Figure 5.18. About 16 mm is cut off. The impedance of the network at 868 MHz is $77.89 + j22.08 \Omega$. This will enhance reflections. The antenna has -12.35 dB of reflected wave which is a correct amount. For even more performance, the matching could be adapted a bit. However, this antenna is fully functional.

Needing to cut the antenna for it to work at the correct frequency is actually not a surprise. The datasheet provides the theoretical location of the cut in Figure 5.19. This is really close to the practical cut presented in Figure 5.20. Notice that it is only a cut, the unused copper trace is still present.

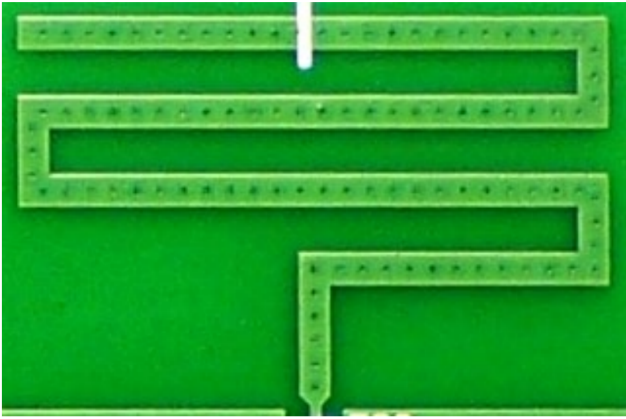


Figure 5.19: Theoretical meander antenna cut.



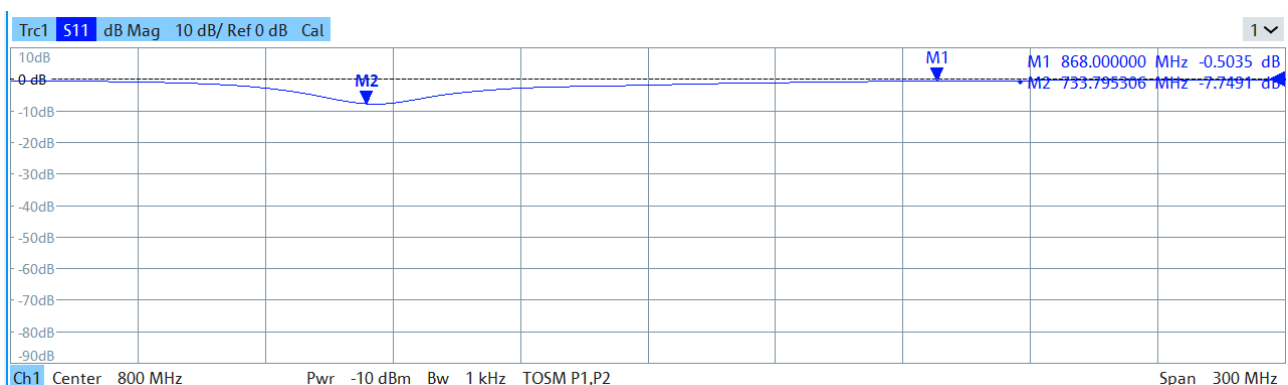
Figure 5.20: Practical meander antenna cut.

5.4.4 Compact helical antenna

The second antenna to test is the helical antenna. It is tested the same way. The parameter of interest is S_{11} as well. On the first try, the radiation frequency was too low, around 725 MHz , as shown in Figure 5.21. Therefore, the antenna is carefully cut again. However, the frequency moves slowly, after 5 lines cut off, the frequency is still 830 MHz as shown in Figure 5.22. This might indicate a too-high interference of the cut copper lines

The process is started again with a new PCB and instead of cutting the trace, it is entirely scratched out. Therefore, no interaction with the useless copper. In this configuration, after 1 line scratched out is a frequency of 762 MHz . However, after 2 lines, the frequency is already at 940 MHz as shown in Figure 5.23. Such a difference was not expected.

Due to a lack of time, the antenna is not further tested. In fact, the meander monopole antenna already worked really well. If the size of the final node is really too big, one might consider testing again this antenna which would need only a quarter of the area of the monopole antenna.

Figure 5.21: Compact helical antenna S_{11} with no cut in the antenna.

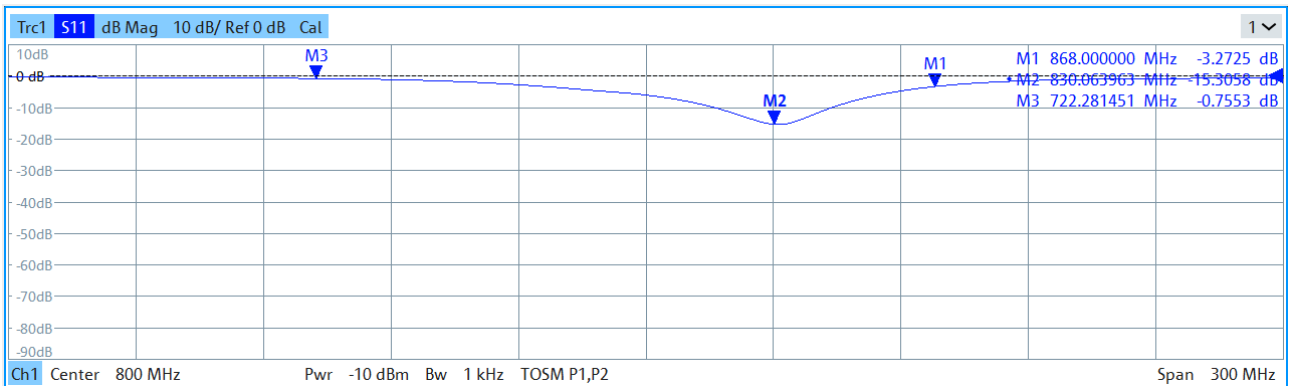


Figure 5.22: Compact helical antenna S11 with 5 lines cut off.

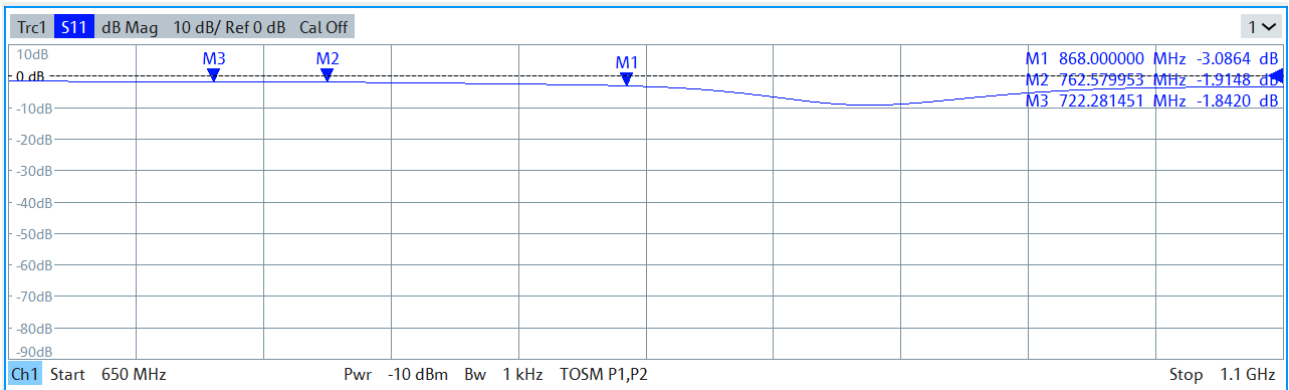


Figure 5.23: Compact helical antenna S11 with 2 lines scratch off.

Chapter 6

Standalone sensing nodes

In this chapter, all the elements detailed in the previous chapters are put together to create a standalone node. This node communicates with sensors (chapter 3), is autonomous in energy (chapter 4), and communicates to a gateway using the LoRa protocol thanks to an antenna (chapter 5). The diagram of the standalone node is presented in Figure 6.1.

For the *Nucleo* board to be replaced, all the other elements present on the board must be replaced. The supply voltage must be correctly connected. Elements such as crystals for the clocks and a port for programming the microcontroller need to be added. The antenna also needs to be carefully connected. And finally, a node prototype is created thanks to a PCB board.

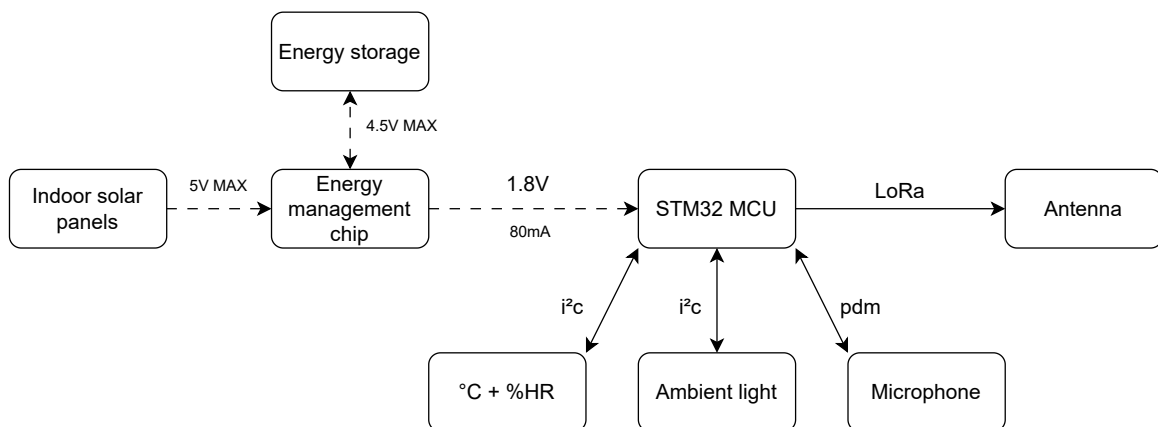


Figure 6.1: Standalone sensing node diagram.

6.1 Supply voltage

This first section manages everything related to the supply voltage provided by the *e-peas* module. Starting by dealing with the connection of the multiple power pins of the MCU. Then measure the supercapacitors' voltage to know how much energy is left. And finally, filter the noise in the supply voltage.

6.1.1 Connections

Connecting the supply voltage to the MCUs is actually not trivial. They are indeed multiple power pins which are not especially connected. Figure 6.2 shows the complete power scheme as presented in the MCU's datasheet. The different uses of the different pins are detailed below.

- V_{DD} : It is the main supply voltage. It supplies the entire core. It is connected to $HVOUT = 1.8 V$ the output of the *e-peas* module.

- V_{SS} : It is the low voltage reference, it is connected to the ground. The exposed pad is also connected to the ground.
- V_{DDA} : This voltage is responsible for the ADC module. If the voltage is 0, the ADC module is disabled. However, this module will be used to measure the supercapacitor voltage level. Therefore V_{DDA} is also connected to V_{DD} . For the UFQFPN48 package, V_{REF+} is internally connected to V_{DDA} .
- V_{BAT} : This pin allows for an external battery to be a backup if the main supply voltage is not supplied anymore. When the MCU relies on V_{BAT} , only part of it works. This feature is not used because adding a battery to the node is costly and takes place. Therefore, V_{BAT} is connected to the ground.
- V_{DDRF} : This voltage supplies the SubGhz radio which is responsible for the LoRa communication. It allows the LoRa module to be connected to a different supply voltage. In this application, it is connected to V_{DD} .
- V_{FBSMPS} and V_{LXSMPS} must be connected to 1.55 V. This voltage is supplied thanks to the SMPS which is a DC/DC converter. V_{DDPA} is connected to the same pin.

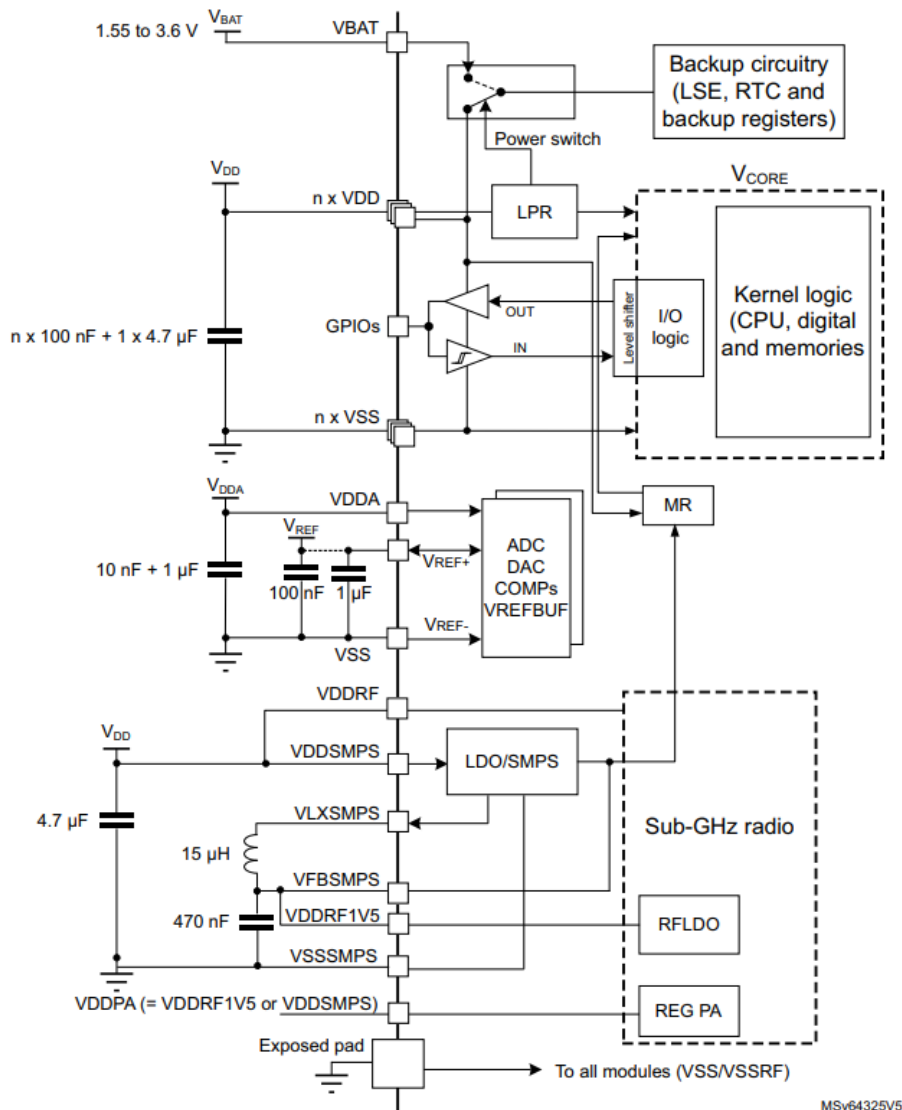


Figure 6.2: Power scheme of STM32WL MCUs [6].

All the passive components suggested on the datasheet are used in the final schematic. Those components are mainly decoupling capacitors. Each power supply pair has at least one. They are placed as close as possible to the MCU pins for maximal efficiency.

6.1.2 Battery and backup

The supply voltage V_{DD} is provided by the *e-peas* module. The latter store's energy in supercapacitors. If those capacitors are too much discharged, the output supply voltage HVOUT is disabled and the MCU is no longer supplied. Therefore, before sending a LoRa message, one wants to verify the supercapacitor voltage to decide whether to send it or not.

To do so, the ADC module is used. The voltage across the supercapacitor can exceed 1.8 V (up to 3 V) while the ADC only goes until 1.8 V. Therefore a voltage divider splits the voltage by 2. To do so, 2 resistors of 100 k Ω are used. The voltage divider is shown in Figure 6.3. However, it is unfortunate to lose power due to the current passing through the voltage divider. To resolve that issue, the MCU only connects the voltage divider to the ground when needed. This is done thanks to the BATT_FB_EN pin. It is left floating by default and whenever a measure is needed it connects it to the ground. For the pin to be floating, it is configured as an input. Table 6.1 resumes the pin used and associated them to a pin of the MCU.

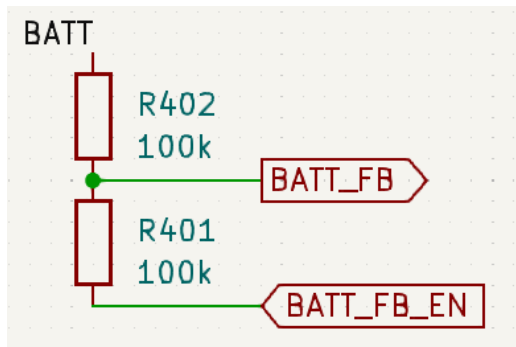


Figure 6.3: Battery level feedback: schematic.

Label	BATT_FB	BATT_FB_EN
GPIO mode	ADC_IN4	Output (when measuring) Input (floating pin)
STM32 pin	PB2	PB0

Table 6.1: Battery level feedback: pin assignment.

In the MCU implementation, the following steps are followed to perform an ADC measure:

1. In order to connect the voltage divider to the ground and allows the current to flow, the BATT_FB_EN pin is set as an output, and outputs 0.
2. The ADC module is started. Caution that there are multiple ADC input pins, the one to start is input 4, in order to have the voltage of the BATT_FB pin.

HAL_ADC_Start(&hadc)

3. Then one must wait until the conversion is completed. This is done with the following blocking function:

HAL_ADC_PollForConversion(&hadc, timeout)

4. When the data is ready, it is retrieved thanks to the following function:

HAL_ADC_GetValue(&hadc)

5. When the data is retrieved, the ADC module is stopped.

HAL_ADC_Stop(&hadc)

6. And finally the BATT_FB_EN pin is turned back in input mode.

6.1.3 Stable supply voltage: LC filter design

The supply voltage must be as constant as possible. Especially the reference voltage of the ADC module. To ensure it, decoupling capacitors were already placed. To have an even more stable voltage, one might be interested in the output voltage of the *e-peas* module to see the variation.

The voltage is observed in multiple situations. First when no current is drawn. In this case, the module output is 1.8 V with a high voltage noise as represented in Figure 6.4. Decoupling capacitors are there to suppress that noise.



Figure 6.4: *E-peas* output voltage with no load.

Things are slightly different when current is drawn. For more readability and as the noise is the interesting part, the oscilloscope is configured in AC mode, for the DC value to be removed. Figures 6.5 and 6.6 show the output when a resistance is placed between the voltage and the ground. The resistance values are $1200\ \Omega$ and $120\ \Omega$ which corresponds to a current of respectively 1.5 mA and 15 mA . One clearly sees noise with a sawtooth shape at a lower frequency.



Figure 6.5: *E-peas* output voltage with a $1200\ \Omega$ load (AC coupling mode).



Figure 6.6: *E-peas* output voltage with a $120\ \Omega$ load (AC coupling mode).

The first difference between the 2 graphs is the voltage peak to peak. Its value is 80 mV for $1200\ \Omega$ and 40 mV for $120\ \Omega$. The other difference is the frequency of that noise. For $1200\ \Omega$, it is every $100\ \mu\text{s}$ which corresponds to a frequency of 10 kHz . While for $120\ \Omega$ it is every $10\ \mu\text{s}$ which corresponds to a frequency of 100 kHz .

To stabilize the voltage of the ADC module, a low-pass filter is added between V_{DDA} and the output of the *e-peas* module. The filter is an LC filter to avoid putting a resistance and having ohmic loss.

The disadvantage is that inductors take way smaller values than resistors. The cutoff frequency f_c can therefore not be as small.

$$f_c = \frac{1}{2\pi\sqrt{LC}}$$

Values are chosen for the frequency to be smaller than 10 kHz while keeping reasonable values for the capacitor and the inductor.

$$L = 100\mu H$$

$$C = 22\mu F$$

$$f_c = 3390 Hz$$

6.2 Microcontroller peripherals

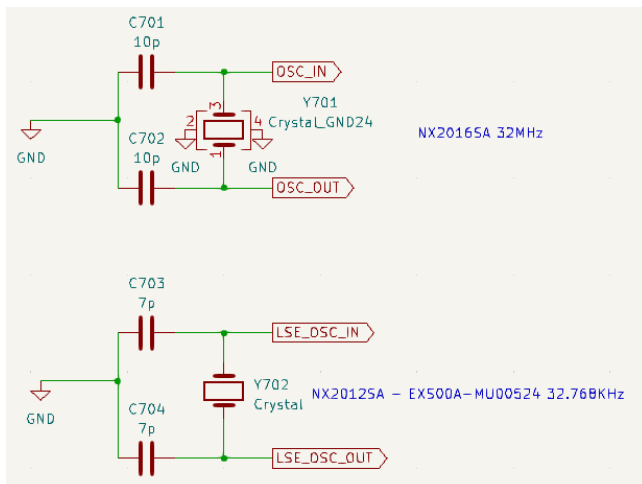
6.2.1 Crystal

For the MCU to correctly work, it uses 2 external clocks: HSE (high-speed external oscillator) and LSE (low-speed external oscillator). HSE is the main clock at 32 MHz . While LSE is a slower clock at 32.768 kHz . Those were previously integrated on the *Nucleo* board and must be replaced. Elements that provide a clock are quartz crystals or TXCO.

When an electric field is applied to a quartz crystal, it starts vibrating at a very specific frequency. This is because quartz is a piezoelectric material that can generate an electric charge when mechanically deformed. In practical circuits, two capacitors are generally used to create oscillators. The size of the crystal and the temperature around it can affect the frequency at which it vibrates.

The clock can also be generated by a TCXO (temperature-compensated crystal oscillator). It is based on a quartz oscillator as well. The main difference, and advantage, with respect to the classical crystal oscillator is (as the name suggests) a better constant behavior with regard to temperature. To obtain such results, it however needs a supply voltage.

TCXO is often used as HSE while LSE is provided by a crystal. In this application, no need for really high precision in time. In addition, the LoRa protocol is tolerant concerning frequency precision. Therefore, they will both be crystals. In addition, it limits the electrical energy consumption. The electrical schematic is represented in Figure 6.7. The chosen references are recommended in the application note AN2867 which is an oscillator design guide for STM32 microcontrollers, among others. They are presented in Table 6.2.



	Reference	Frequency	Price
HSE	NX2016SA	32 MHz	0.55 €
LSE	NX2012	32.768 kHz	0.72 €

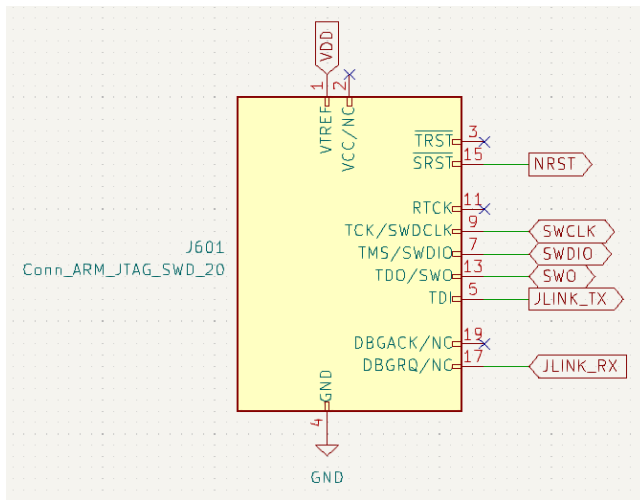
Table 6.2: HSE and LSE crystals: reference.

Figure 6.7: HSE and LSE crystals: reference.

6.2.2 J-link

To program the MCU, a J-link is used. This device is debug probes used in the industry for debugging and programming MCU. It has a 20-pin connector, however not all the pins are connected. Therefore, only the useful ones are represented in the symbol in Figure 6.8.

For the device to program the MCU, an interface must be chosen. The two most popular interfaces are JTAG (Joint Test Action Group) and SWD (Serial Wire Debug). In this application, the SWD is chosen. 3 pins are necessary for that communication. First *SWCLK*, the clock signal is generated by the computer. Then *SWDIO*, a bi-directional data line on a single pin. And *NRST*, which is connected to the reset pin of the MCU. This allows the ST-link to reset the MCU. It is not the only way to reset the MCU. Another way is thanks to a reset button placed on the PCB. An optional pin is *SWO* (serial wire output), which retrieves the data outputted from a pin of the MCU.



	SWCLK	SWDIO
GPIO mode	DEBUG_SWCLK	DEBUG_SWDIO
STM32 pin	PA14	PA13
	nRESET	SWO
GPIO mode	-	DEBUG_SWO
STM32 pin	NRST	PB3
	JLINK_TX	JLINK_RX
GPIO mode	USART2_TX	USART2_RX
STM32 pin	PA2	PA3

Table 6.3: ST-link: pin assignment.

Figure 6.8: ST-link: symbol and useful pins.

JLINK_TX and JLINK_RX are 2 other pins used for an additional optional feature. In fact, the connector supports a virtual COM port (VCOM). It uses 2 lines, one for each direction: TX and RX. They are connected to the USART2 ports of the MCU. Those pins allow keeping using *PuTTY* to see the *printf()* values for instance as explained in section 3.1.1. The only difference is the COM number of the device. It is really useful in the development phase.

6.2.3 Antenna connection

There are 3 modes of usage of the radio module. The sending mode, either low power or high power, and the receiving mode. The high power mode is proscribed because of its too high current needs. There are still 2 modes left. In the application, only the sending mode is required. However, for future improvements, receiving instruction or data from the gateway is a real opportunity. To allow those 2 modes, an RF switch is used. It is controlled by a pin of the MCU (PA7) in output mode, represented by the label *CTRL_antenna* on the schematic presented in Figure 6.9. It is connected in sending mode by default. Table 6.4

RF switch BGS12P2L6	
CTRL_antenna pin	Antenna mode
0 (reset)	Transmission
1 (set)	Reception

Table 6.4: Antenna mode configuration depending on the CTRL_antenna pin value.

On the MCU, the *RFO_HP* pin is responsible for the high-power transmission. The *RFO_LP* pin is responsible for the low-power transmission. And finally, *RFI_N* and *RFI_P* pins are responsible for the reception. Caution must be taken that the signal is a differential signal. To connect the pins, one should follow the RF matching network design guide provided in the application note AN5457. Starting with the connection from the low power output pin *RFO_LP* until the switch. When the signal is at the RF switch, one finds the pi matching for the antenna impedance before reaching the antenna as discussed in Chapter 5.

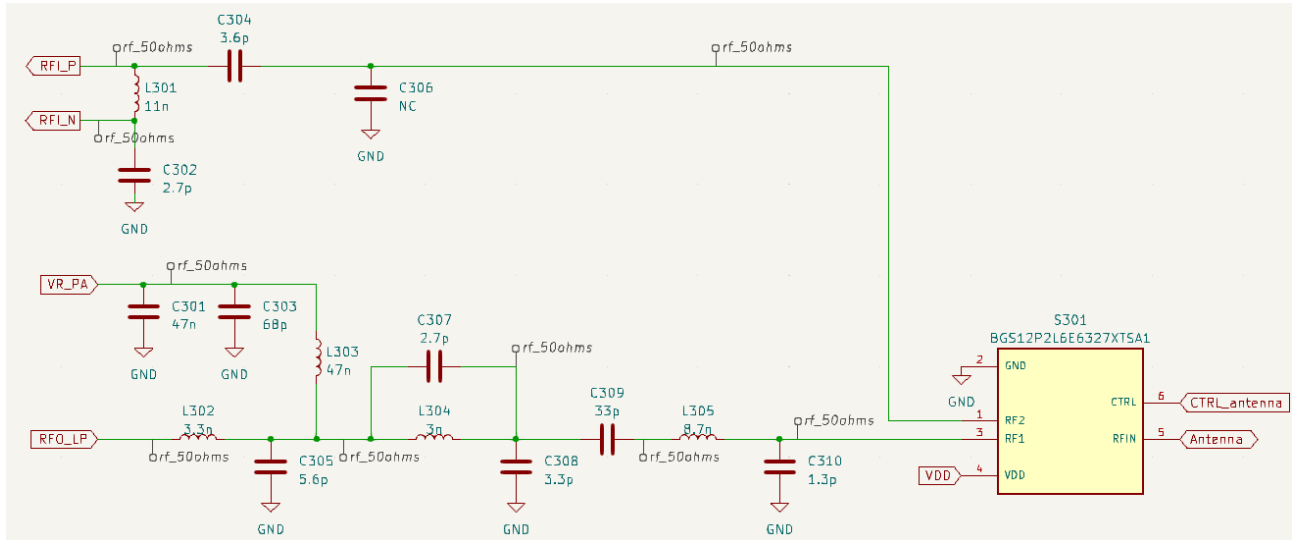


Figure 6.9: Antenna schematic for low-power transmission and reception modes.

Low power transmission

The following describes each block of the output matching and filtering network.

- VR_PA biasing and filtering: C301, C303 and L303 creates a low pass filter for *VR_PA*. It does not let the AC signal from *RFO_HP* go back to the DC source. The inductor however let a bit of RF signal passing but their energy is absorbed by the capacitor.
- PA (power amplifier) output matching: L302 and C305 are used to match the impedance of the power amplifier embedded in the MCU. Its impedance is $15.27 \Omega + j1.27 \Omega$ when the transmit output power is 14 dBm. For optimality, the circuit must have $15.27 \Omega - j1.27 \Omega$ because to transfer maximum power, the source and the load impedance should be the complex conjugate of each other.
- PA output filtering: the amplifier is non-linear. L304 and C307 are placed to filter the second harmonic ($2 \cdot 868 \text{ MHz}$) of the main signal frequency. It is a notch filter or band-stop filter.
- Matching: C308 is added to compensate for the mismatch due to the notch filter.
- Low pass frequency: C308, L305, and C310 create a low pass frequency with a cutoff frequency greater than 900 MHz .
- And finally, C309 simply blocks the DC signal. It is a high pass filter.

High power transmission

The high-power output pin *RFO_HP* will not be used. However, it is not impossible that a signal is transmitted on that pin, non-intentionally. Therefore to protect the MCU, a resistor connect the pin to the ground to dissipate the energy if this occurs. The resistor is 50Ω to match the LoRa module

impedance and to avoid for the signal to reflect.

If the pin would have to be used, the same procedure as for the low power transmission would have to be followed.

Reception

Regarding the reception part, it is less complex. The DC signal is filtered thanks to the C304 capacitor. The C306 capacitor is not used, it could have been needed for the RF switch but this is not the case.

The input voltage is single-ended and must be converted into a differential signal, not referenced to the ground anymore. Therefore, the voltage on RFI_P must be equal to minus RFI_N thanks to C302 and L301. This relation links the value of the capacitor with the value of the inductor. This circuit is called a balun.

The inductor is also used to match the $50\ \Omega$ of the low-noise amplifier (LNA) integrated into the LoRa module. This fixes the value of the inductor and by transitivity, the value of the capacitor.

Final components value

The *Nucleo* board embeds the same output matching and filtering network for transmission and matching network for reception. The component values are generally the same as the ones found in the design note. However, sometimes they are slight changes. In this case, the values on the *Nucleo* board are used. In fact, the *Nucleo* already showed that it is fully functional. Those values are indicated in Figure 6.9.

6.3 Final PCB design

Now that every element has been discussed, it is time to design the final PCB prototype. The full schematic can be found in the appendices.

Multiple precautions have been taken while designing the PCB.

- The decoupling capacitors are as close as possible to the corresponding elements.
- For performance concerns, the sensors are as far away as possible. The light sensor is on the opposite side of the board with respect to the massive st-link connector to avoid any shading. In addition, the humidity and temperature sensor is isolated from any chip that could heat up and influence the measurements.
- For the best RF transmission, several points of attention:
 - The RF path is as straightforward as possible
 - The trace has the correct width for impedance matching and layer 2 is a ground plan. Only interrupted for a few vias.
 - The RF path is surrounded by vias connected to the ground as a shield.
 - When traces needed to cross each other, one of them passes on layer 4 thanks to vias. In addition, layer 3 is used for the supply voltage tracks.
 - A ground plane zone is also put on layers 1, 3, and 4 in addition to layer 2. Those layers are regularly connected thanks to vias.
- Because it is a prototype, arrangements are made for easier access to signals and protection against a deficient chip.

- The board is normally powered by the *e-peas* module or it can also be alimented by an external generator thanks to 2 connectors (J102). Jumpers JP101 and JP102 are responsible for that choice.
 - The board has a lot of test points (ground, supply voltage, *e-peas* output voltage,...).
 - To measure the current flowing, a jumper is added in series to the supercapacitors.
 - A u.FL connector is added to test the antenna if needed.
- A reset button is always a good practice.
 - For size concerns, the majority of the passive components are in the 0402 format.
 - All the components are on the same side.

The final PCB is represented in Figure 6.10. The final PCB has a size of $71.8 \times 40.7 \text{ mm}^2$ and a price of 51.78 €. The entire BOM can be found in the appendix. Prices are from *Mouser* in June 2023. Knowing the quick price variation, this might be changed.

If the size is a concern, it can be made small by removing the jumpers and test points. If it is still not sufficient, one might place components on both sides of the PCB. However, this becomes harder during the soldering phase. Notice that the majority of the area is taken by the antenna and the ST-link connector.

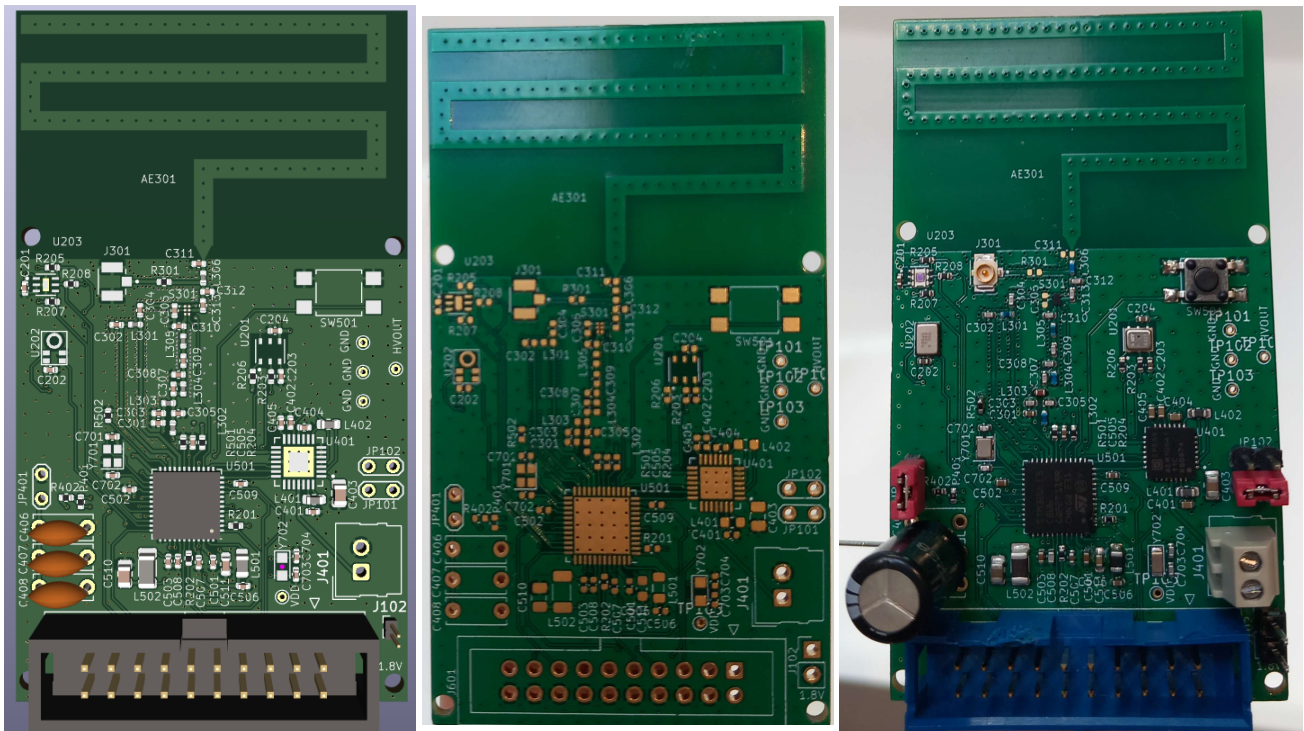


Figure 6.10: Standalone node: final PCB design.

KiCad visualization (left), blank PCB (center), PCB with soldered components (right)

6.4 Microncontroller implementation

To test the node, the MCU must be programmed. To do so, a new project is created on the STM32CubeIDE program. Caution must be taken that the MCU to select is the STM32WLE5C8. This is the low-energy version of the MCU embedded in the *Nucleo* board. Hopefully, they are really close to one another and support the same HAL functions. Therefore, the functions implemented are simply copy-pasted at the correct location.

The configuration page is configured exactly the same way as for the *Nucleo* project. With one exception. In the "SubGHz_Phy middleware" tab of the SUBGHZ_PHY configuration panel, one parameter is "Select radio Driver". In the *Nucleo* project, it was set to "Bsp via extSettings". BSP stands for board support packages and is specific to the *Nucleo* board. One, therefore, selects "User board".

The BSP was useful for the LEDs and buttons present on the board, but also to configure the antenna mode thanks to the RF switch. This was handled in the *radio_board_if.c* file. In that file, instead of calling the BSP functions, they must be rewritten. In practice, the only function in which one needs to add code is the *RBI_ConfigRFSwitch(RBI_Switch_TypeDefConfig)*. Depending on the mode of the antenna, the CTRL_antenna pin must be set or reset.

6.5 Standalone node testing

Finally, the node is tested. For the test, the code is modified for messages to be sent every 5 seconds.

First, the power management part is tested on its own. On startup by ensuring that the storage element (supercapacitor) is correctly connected. Then one connects the pv panel to the terminal block and ensures that it is illuminated. The node is left for a moment, and eventually, the supercapacitor charges sufficiently for the chip to output the target voltage. Testing with a voltmeter, the output voltage is in fact 1.8 V.

Then the node is tested with the generator. First, the voltage is set at 1.8 V. Thanks to the VCOM, one can verify which step was correctly executed. When a message is requested, the data is correctly acquired from the sensors. However, the node resets itself when it tries to send the LoRa package. By increasing slightly the voltage to 1.83 V, the LoRa package is correctly sent.

Because of the voltage loss between the *e-peas* chip output and the LoRa module, the output of the *e-peas* module is not sufficient LoRa transmission. The design must therefore be adapted for the output voltage of the *e-peas* module to be increased. Hopefully, the node is designed to be supplied thanks to any voltage in the range 1.8 to 3.3 V.

The output voltage is determined by the configuration pins CFG[2:0]. Multiple choices are available among which:

- Use a custom configuration (CFG = 000) for any voltage between 1.8 to 3.3 V. Using at least 2 V to be sure it will work. This required 6 additional resistors on the design.
- Use the dual supercapacitor mode (CFG = 010) to have 3.3 V which supplies the whole node
- Use the dual supercapacitor mode (CFG = 010) to have 3.3 V for the LoRa module, and 1.8V for the rest of the board.

The easiest way is to use the dual supercapacitor mode with a high 3.3 V supplying the node. Keeping the same transmit power for the LoRa module, the application should roughly consume the same energy. However, this requires putting 2 supercapacitors in series and connecting their junction to the balun pin of the *e-peas* chip which takes space.

Chapter 7

Database and user interface

Before starting this chapter, I would like to acknowledge the important contributions of Morgan Diepart to this part.

Previous chapters ensure that the gateway receives correctly some useful data. This chapter aims at transferring that data into a network. This network is able to receive data from multiple gateways. And eventually, it makes the data available through a website. The diagram is shown in Figure 7.1.

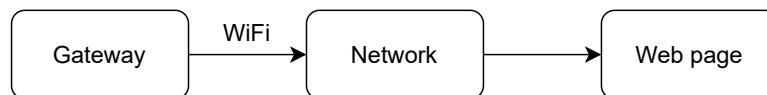


Figure 7.1: Network diagram.

7.1 From the gateway to usable data

WiFi connection

The network is materialized by a *Raspberry Pi* board. The data is transmitted from the gateway to the *Raspberry Pi* thanks to WiFi communication. To do so, the gateway first needs to be connected to the Uliege WiFi. This can be done thanks to its MAC number. It can be acquired thanks to the `esp_efuse_mac_get_default(uint8_t * mac)` function. When the MAC number is registered, the gateway can connect to the university's network.

The *Raspberry Pi* is obviously also connected to the WiFi to retrieve the data.

7.1.1 Identification

The MAC number is also chosen to be the identification number of gateways. It allows the *Raspberry Pi* to identify which gateway sends the data and therefore, it allows multiple gateways to be used.

In order to use multiple nodes, one must find a way to identify them. To do so, in every package send by a node, the first byte is its identifier. To do so, a number is assigned to each node. It is changed manually before programming the node. For better robustness, the number could be put in the OTP (one time programmable) memory which as its name indicates, can only be programmed a limited amount of times. This memory is not modified when the MCU is programmed.

Another solution is to use the unique ID of the MCU. It is available thanks to the `HAL_GetUIDw0()`, `HAL_GetUIDw1()`, and `HAL_GetUIDw2()` functions and is 96 bits longer. This is really long to be send at every LoRa communication.

API

To communicate with the *Raspberry Pi*, an API is created. When a gateway is physically added, it also needs to be added to the software thanks to the API. One can specify the location in which it is placed. It also mentioned the time this gateway is active. The API also allows adding nodes and specifying the room in which they are placed.

The gateway can send data to the *Raspberry Pi* thanks to the `esp_http_client_set_post_field()` function. The data must be in a .json format. The node id is a string, then the time, then the gateway id, then the temperature in float, then the relative humidity in float, then the luminosity in int, and finally, the sound volume in float.

Database

The *Raspberry Pi* stores the data in a database. If it receives the same data from multiple within a short time, it considers them as duplicates and processes only one of them. This can occur if a node is placed between 2 gateways that both receive the LoRa packet.

7.2 User interface

The data is now stored in the *Raspberry PI*. The data is still not available to the user. To do so, the *Raspberry Pi* puts the data on a web page. A really easy and convenient to process the data is to print them on a graph. The interface is done thanks to *Grafana*. The latter allows the presentation of data on graphs on a dashboard. The dashboard will automatically reload itself. The interface is available at <http://msys-rpi-lora.mont.priv:8000/d/dd955c83-9ace-4d9c-82b5-e031f6b2892e/b28?from=now-7d&to=now&orgId=2&kiosk>

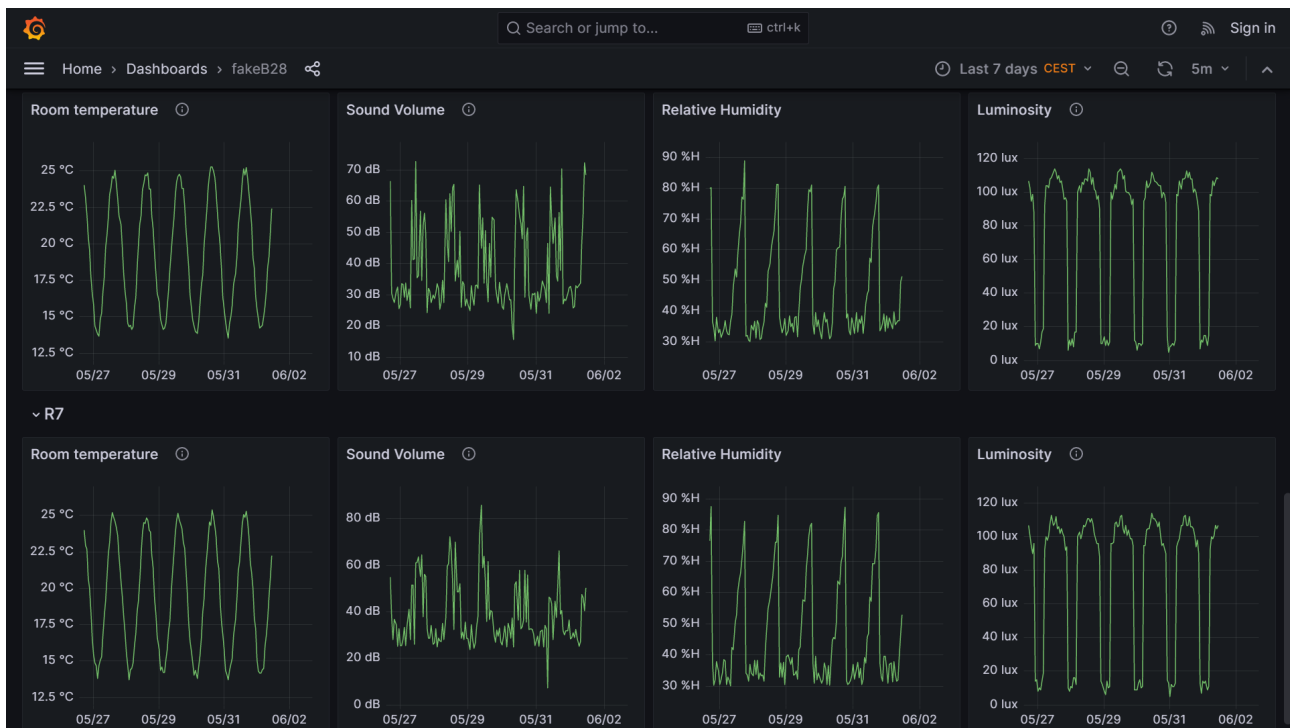


Figure 7.2: Example of the data display thanks to the dashboard with fake data.

The interface for fake data is shown in Figure 7.2. A dashboard is created for each building. Within a dashboard, it exists a line for each room. And on each line, the evolution of each parameter is shown on a graph. The time for which the data is required is tunable. The url refreshes itself periodically. This is perfect to show on a screen.

Chapter 8

Conclusion and further work

8.1 Conclusion

In conclusion, this master's thesis has developed a standalone node monitoring the temperature, relative humidity, and ambient light level. During every step, one kept in mind to have a node as cheap, as small, and as low-power as possible. The node is then integrated into a network. This network is capable of handling multiple gateways which themselves can handle multiple nodes.

First, to select the technology of communication between the node and the gateway, different LPWAN technologies are studied to finally select the LoRa communication protocol for its long-range and low current consumption. Based on this choice, a gateway and an MCU for the node are selected. Both support bidirectional LoRa communication.

Then, thanks to a *Nucleo* board embedding the selected node MCU, the development of a simplified network was possible. For this network, 4 sensors are selected and tested thanks to a test PCB. The data retrieved by the *Nucleo* board is then sent in a LoRa package to the gateway. The gateway then displays the message on its OLED screen. Finally, because of its high consumption and high peak current, the CO₂ sensor is dismissed. The 3 remaining sensors are compatible with a voltage range of 1.8 V to 3.3 V, allowing them to function whatever the MCU supply voltage.

The *Nucleo* board enabled experimenting with LoRa communication. However, it will not be in the final node and each element it embeds must find an analog in the final node PCB design. For instance, the MCU of the node (STM32WLE5) is the low-power version of the MCU embedded in the board (STM32WL55).

The first big issue is to supply the node since the *Nucleo* board was connected to a computer. The power management is done thanks to an *e-peas* chip. It acquires energy thanks to indoor solar panels and stores it in super-capacitors. It then outputs a tunable voltage with a maximal current of 80 mA. It has been decided to use a 1.8 V supply voltage. It is tested thanks to an *e-peas* module.

In addition to the power management, the node needs an antenna. Multiple possibilities were compared, and finally, a meander monopole antenna is printed on the PCB. It is a cheap solution with high efficiency of transmission. Caution was made to match the impedance of both the antenna and the tracks for the maximum power transmission. The antenna is tested using a VNA.

To completely mimic the *Nucleo* board, caution was made to correctly connect the supply voltage to the MCU. In addition, a node need crystals to manage the clock signal. Those were added to the node PCB design. Caution was made for the connection of the antenna to the MCU. The track must be matched, and the signal filtered for more precision. Furthermore, an IDC connector is used for the MCU to be debugged and programmed thanks to an ST-Link EDU. The first prototype of the node can now be assembled. It works perfectly when supplied with at least 1.83V for the generator. The

energy management module on its own also works as expected. It correctly outputs 1.8V thanks to the supercapacitors and the solar panel. The output voltage of the *e-peas* is therefore slightly too low. Because of the loss between the output of the *e-peas* chip and the LoRa module in the MCU, the voltage must be increased thanks to another configuration setup of the *e-peas* chip.

And finally, to complete the LPWAN application, the data received by the gateway is sent to a *Raspberry Pi* module thanks to WiFi. The module stores them on a database and displays them thanks to *grafana* on a web page. This allows anybody to access the data easily and visualize them on graphs.

8.2 Further works

The node is able to send useful data (temperature, relative humidity, and ambient light level). But those values are not processed yet. By correctly analyzing them, it provides information about the human presence or not in the room. One might need to analyze more in-depth the correlation between the monitoring parameters. This processing could also be done thanks to a deep neural network that would learn from test data to detect presence. One might want to monitor even more parameters. To do so, one starts by processing the data from the microphone. And eventually, add new sensors on the node.

For now, the communication packages follows the LoRa protocol. This means that anyone with a LoRa receptor can intercept the data. One way to avoid it, if it is an issue, is to follow the LoRaWAN protocol. This protocol indeed includes an identification and an encryption phase. However, it requires to use of a LoRaWAN network server, which may become costly for multiple gateways.

Communication is unidirectional, it only uses the downlink. In other words, the node sends to the gateway. All the hardware required for bi-directional communication is already available on the node. And the LoRa protocol handles bi-directional communication as well. Therefore, it only requires a few lines in the code to enable bidirectional communication. This would be interesting for the gateway to be able to send instructions to the node. For instance, acknowledge receipt of the package. It could also send each week, a message indicating when LoRa packages must be sent, taking into account the vacancies. Caution must be made that the node cannot wait for a message at any moment, one way to do this is to have a reception window after each transmission. This allows smarter message scheduling.

Another way to have smarter message scheduling is to send more frequent data when the illuminance is high and therefore the panels harvest more energy. The frequency could also depend on the probability of presence. If a presence is maybe detected, then a new measure is quicker to confirm the information.

And finally, when one is happy with the node PCB design, the network must be deployed. To do so, one must order as many nodes as there are rooms to monitor. Maybe add an optional second gateway for better coverage. Gateways and nodes are then registered in the API, and the dashboard shows the evolution of the monitoring parameters in each room. And for even more impact, the dashboard can be presented at the entry of the Institute thanks to a screen.

Appendices

Appendix A

Energy management

A.1 Energy management chip *e-peas*

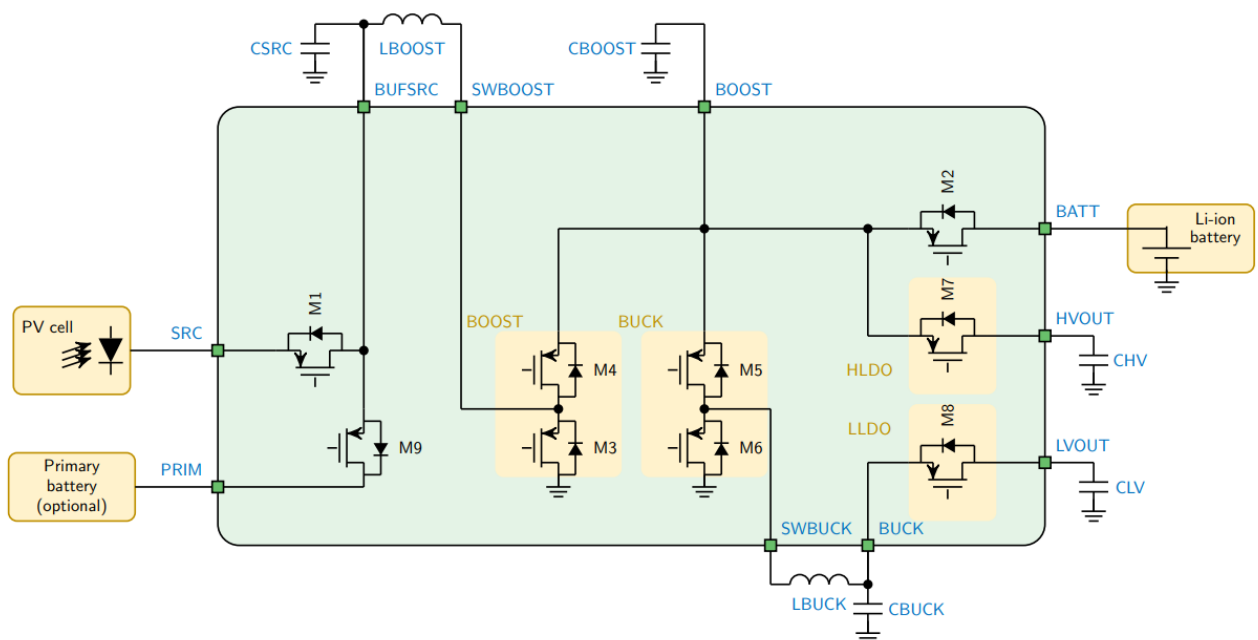


Figure A.1: Simplified schematic view of the *e-peas* chip AEM10941

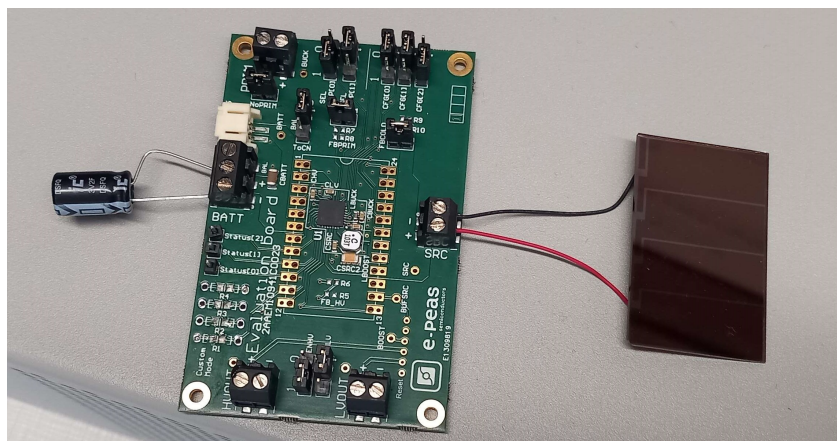


Figure A.2: Solar panel testing setup thanks to the *e-peas* evaluation board.

Appendix B

Smith chart

B.1 Smith chart presentation

This part aims at explaining how to practically match an arbitrary impedance to a fixed target impedance thanks to the Smith chart. This enables the matching values to be found graphically rather than analytically.

First the non-match impedance is reported on the Smith chart. A Smith chart represents the complex value on a polar plot. Two versions of the Smith chart are represented in Figure B.1. The left one highlights the impedance value while the right one highlights the admittance value. The admittance is defined as the inverse of the impedance. Both representations show the same information, and the lines can be overlaid. However, the graph would become difficult to read.

Notice that the center of the plot is the impedance to match: resistance of 50Ω or admittance $0.02 S$. Generally, and unlike this example, the Smith chart has a 1 at its center and represents the inductance divided by its target. This is only a convention that does not change anything when the target impedance is real.

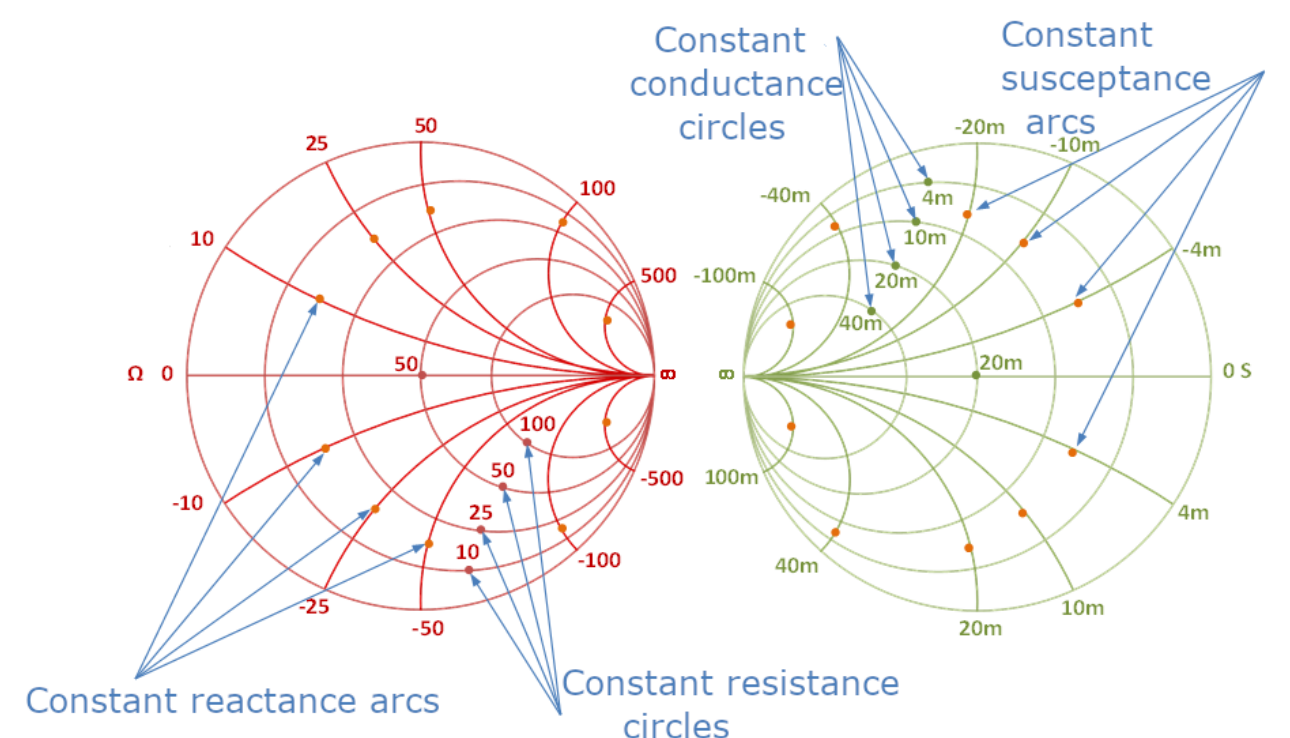


Figure B.1: Smith chart for inductance representation (left) and admittance representation (right).

The Smith chart, therefore, represents a complex impedance value on a circle. The graphs have either constant resistance and reactance circles or constant conductance and susceptance circles. It forms a grid-like pattern. The impedance Z can be decomposed into its real part R and its imaginary part X . Parameter R is called the resistance and X is the reactance. To place the inductance on the chart, one finds the intersection of the resistance constant circle of value R and the reactance constant arc of value X .

$$Z = R + jX$$

One might prefer to use the admittance Y . Similarly, it decomposes into its real part G and its imaginary part B . Parameter G is called the conductance and B is the susceptance. To place the admittance on the chart, one finds the intersection of the conductance constant circle of value G and the susceptance constant arc of value B .

$$Y = \frac{1}{Z}$$

$$Y = G + jB$$

When the two graphs are superposed, the value taken in impedance or in admittance has obviously the same location.

Inductance reactance is above the horizontal mid-line and capacitive reactance is below it.

B.2 Using the Smith chart for impedance matching

When the impedance is placed on the chart if it is not on the center point, it has to be moved toward it. To make the point move, one adds an inductor or a capacitor. On the one hand, if the component is added in series, the point will move following the resistance lines. In fact, those passive components only add a complex value to the initial resistance value. On the other hand, if the component is in parallel, the point will move following the conductance lines. Furthermore, if the component is an inductor the point goes up and if it is a capacitor the point goes down. This behavior is illustrated in Figure B.2. Notice that using resistors for the matching is avoided because it creates losses.

The constant-resistance circles and the constant-conductance circles form an alternative type of grid pattern. Therefore, moving any point to the center can be done using only 2 components. However, not any combination of the 4 behaviors will work. Firstly, it is required that one of the components make the point move on the resistance line and the other on the conductance line. Otherwise, the motion is "1D" and not "2D". Secondly, one component must make the point move upward and the other makes it move downward. This only let 2 combinations:

- L in series and C in parallel
- C in series and L in parallel

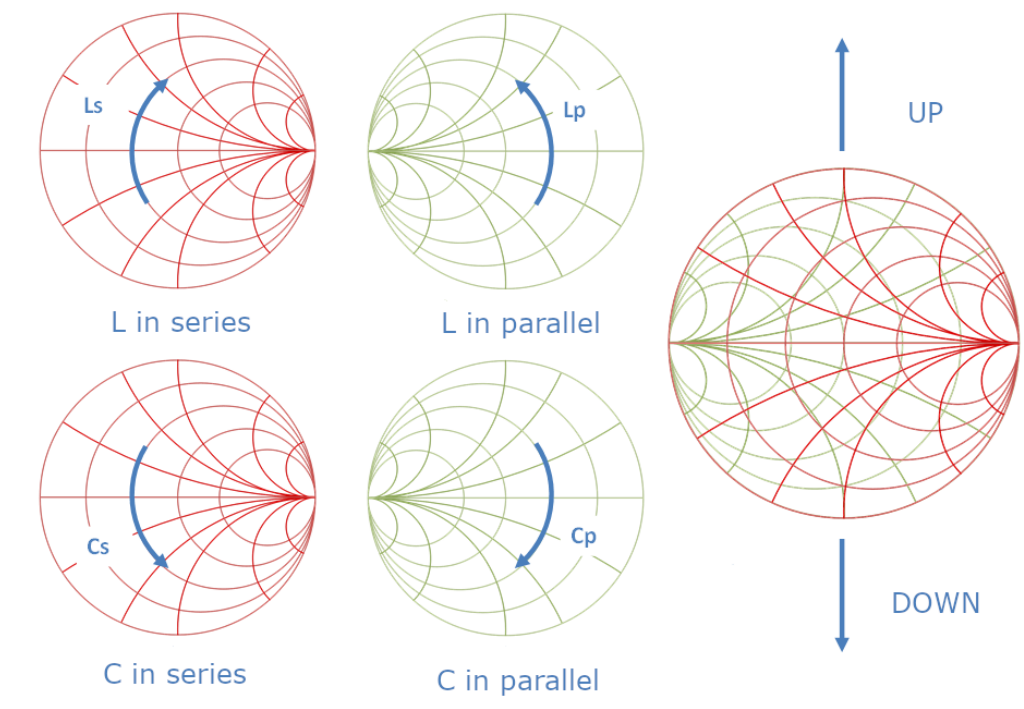


Figure B.2: Representation of the motion of a point on the Smith chart depending on the added component (C or L) in series or in parallel.

An example of impedance matching thanks to the Smith chart is shown in Figure B.3. To ease the process, only the 2 useful circles are shown. One circle passes through Z_a , the impedance of the antenna to match, and the other passes through the matching point at the center. First, the point goes up on the resistance circle thanks to an inductor in series. It reaches the intersection of the 2 circles. Then it goes down thanks to a capacitor in parallel.

This example shows the solution with L in series and C in parallel. However, finding the solution with C in series and L in parallel is trivial. It uses the same 2 circles and instead of going up then down, the point goes down then up. Making a stop at the second intersection of both circles.

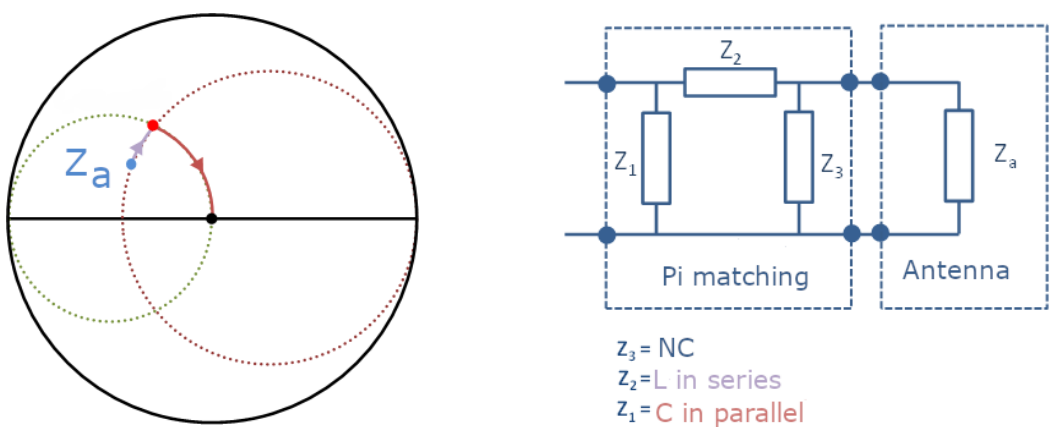


Figure B.3: Example of impedance matching when the point is on the left of the Smith chart.

The values of the components are computed depending on the distance covered on the arc. Caution to the order of those components with regard to the antenna. Z_a is the impedance of the antenna, therefore the closest component to the antenna is the inductor, and then the capacitor. On the matching circuit, Z_3 is not connected.

The matching circuit used is generally a pi-network. It is called this way because of the shape made with the 2 elements in parallel and the element in series. One may ask why using 3 footprints when 2 components are sufficient to do the matching. Actually, those 3 footprints enable putting the element in parallel (C or L) either before or after the other matching element (respectively L or C). In fact, as illustrated in the next example, the order of the components depends on the initial location of the impedance.

In fact, in the second example, illustrated in Figure B.4, the point is on the right of the chart and not on the left. Therefore, for the circles to cross each other, the resistance circle passes through the matching point, and the conductance circle passes through Z_a . It was the reverse way in the previous example. For the impedance to be matched, the point goes down thanks to a capacitor in parallel. Then it goes up on the resistance circle thanks to an inductor in series. On the circuit, the capacitor in parallel is the closest to the antenna, then comes the inductor in series. Z_3 is not connected this time.

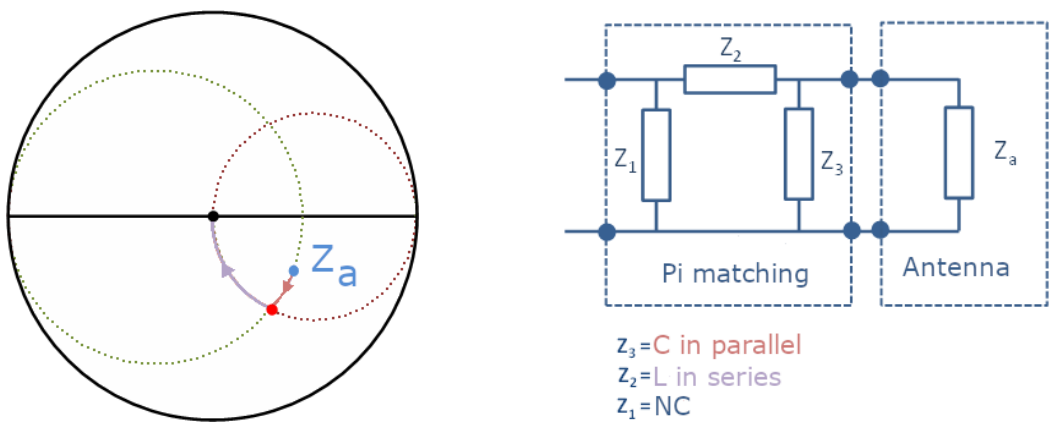


Figure B.4: Example of impedance matching when the point is on the right of the Smith chart.

Smith charts are really powerful tools. They are useful to correctly understand the use of a matching network. However, when matching is needed, multiple impedance matching calculators exist. Those compute almost instantly the values needed in the matching depending on the target impedance and the current impedance.

Appendix C

Antenna design

C.1 Antenna selection







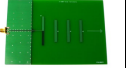


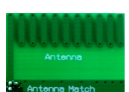


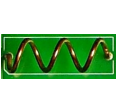



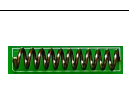
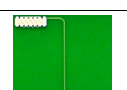

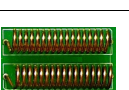
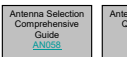

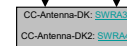

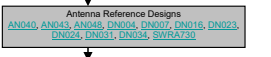

		Antenna Selection Quick Guide						DN035
								
Design / Application Note	SWRU120 *1	SWRA117 *2	SWRA118	SWRA093	SWRA227	SWRA350	SWRA730	
Frequency	2.4 GHz	2.4 GHz	2.4 GHz	2.4 GHz	2.4 GHz (863 MHz - 2.4 GHz)	2.4 GHz	2.4 GHz (433 MHz - 2.4 GHz)	
Typical Efficiency	80 % (EB) 94 % (SA)	68 % (EB)	80 % (EB)	65 % (Zlight2)	94 % (SA)	72 % (SA)	84 % (SA)	
Bandwidth @ VSWR 2:0	280 MHz	101 MHz	100 MHz	150 MHz	354 MHz (SA)	497 MHz	315 MHz	
Dimensions (mm)	26 x 8	15 x 6	46 x 9	45 x 2.5	38 x 25	150 x 100	43 x 25	
								
Design / Application Note	SWRA227 *1	SWRA496 *2	SWRA730	SWRA228	SWRA328	SWRA328	SWRA307	
Frequency	868 / 915 / 920 MHz (863 MHz - 2.4 GHz)	868 / 915 / 920 MHz	868 / 915 / 920 MHz (433 MHz - 2.4 GHz)	868 / 915 / 920 MHz	868 / 915 / 920 MHz	868 / 915 / 920 MHz	868 / 915 / 920 MHz	
Typical Efficiency	98 % (SA)	63 % (EB)	85 % (SA)	80 % (SA)	69 % (EB)	64 % (EB)	48 % (EB)	
Bandwidth @ VSWR 2:0	88 MHz (SA)	37 MHz	96 MHz	40 MHz	62 MHz	56 MHz	56 MHz	
Dimensions (mm)	38 x 25	19 x 12	43 x 25	43 x 20	10 x 28	48 x 8	15 x 29	
						Antenna Support Documentation      		
Design / Application Note	SWRA730 *1	SWRA328 *2	SWRA328	SWRA328 *1	SWRA328 *1			
Frequency	433 MHz (433 MHz - 2.4 GHz)	433 MHz	433 MHz	315 MHz	136 - 240 MHz			
Typical Efficiency	23 % (SA)	26 % (EB)	15 % (EB)	15 % (EB)	7 % (EB)			
Bandwidth @ VSWR 2:0	24 MHz	38 MHz	30 MHz	4 MHz	3 MHz			
Dimensions (mm)	43 x 25	42 x 29	15 x 29	37 x 9	42 x 29			
EB: SmartRF Evaluation Board SA: Stand Alone	*1 First Choice Recommended Antenna		*2 Second Choice Recommended Antenna		SWRA351B	By Richard Wallace		

Figure C.1: Texas Instruments antenna selection quick guide.

C.2 Radiation pattern

- Theta is the rotation around the Y-axis
- Phi is the rotation around the Z-axis

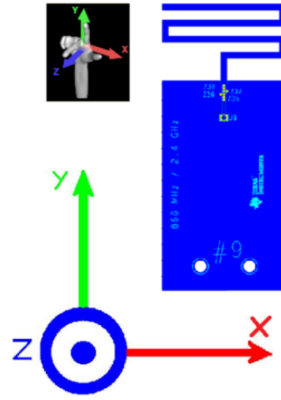


Figure C.2: Axis orientation [9].

Meander monopole antenna

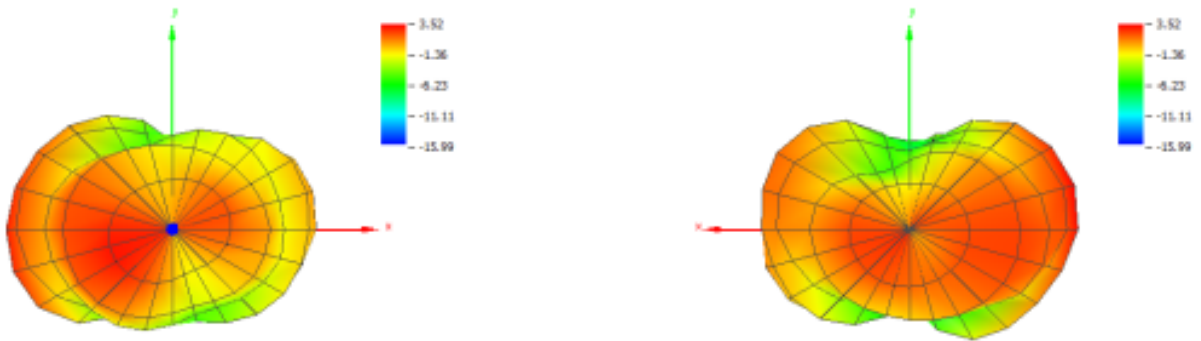


Figure C.3: Meander monopole antenna, radiation patterns.
 Left: $\theta = 0$, $\phi = 0$; Right: $\theta = 180$, $\phi = 0$

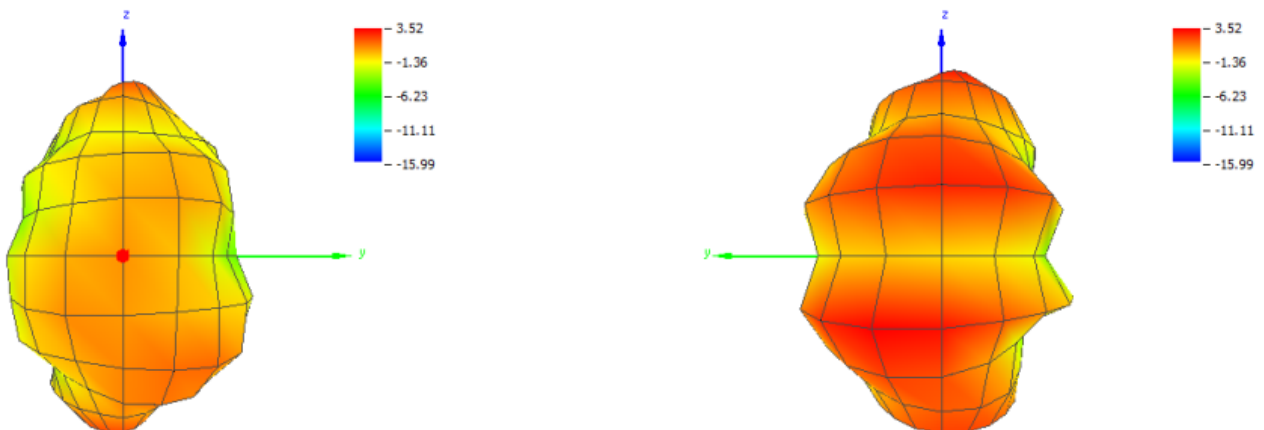


Figure C.4: Meander monopole antenna, radiation patterns.
 Left: $\theta = 90$, $\phi = 0$; Right: $\theta = 90$, $\phi = 180$

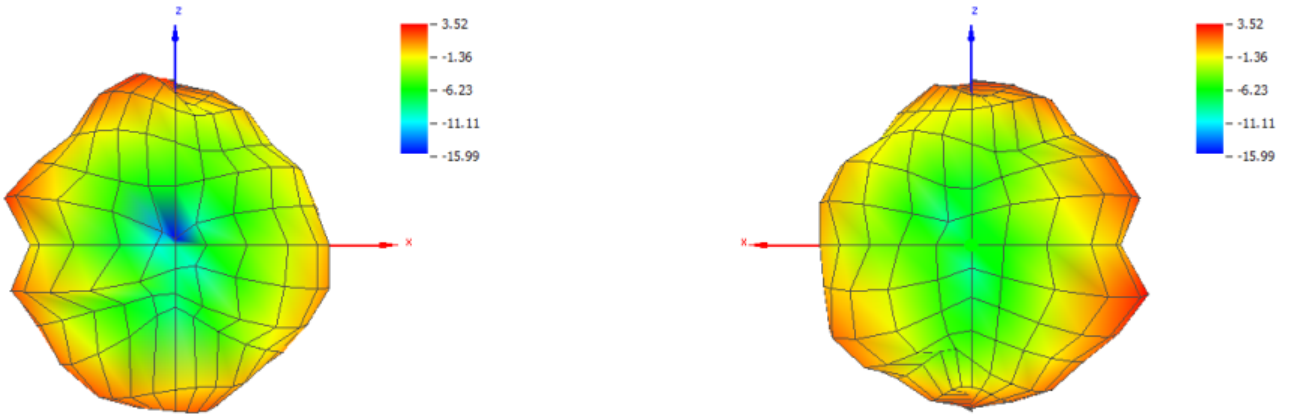


Figure C.5: Meander monopole antenna, radiation patterns.
 Left: $\theta = 90$, $\phi = 270$; Right: $\theta = 90$, $\phi = 90$

Compact helical antenna

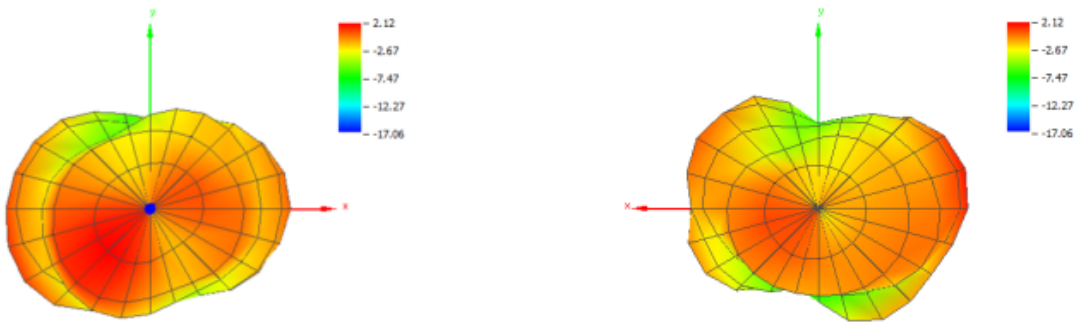


Figure C.6: Compact helical antenna, radiation patterns.
 Left: $\theta = 0$, $\phi = 0$; Right: $\theta = 180$, $\phi = 0$

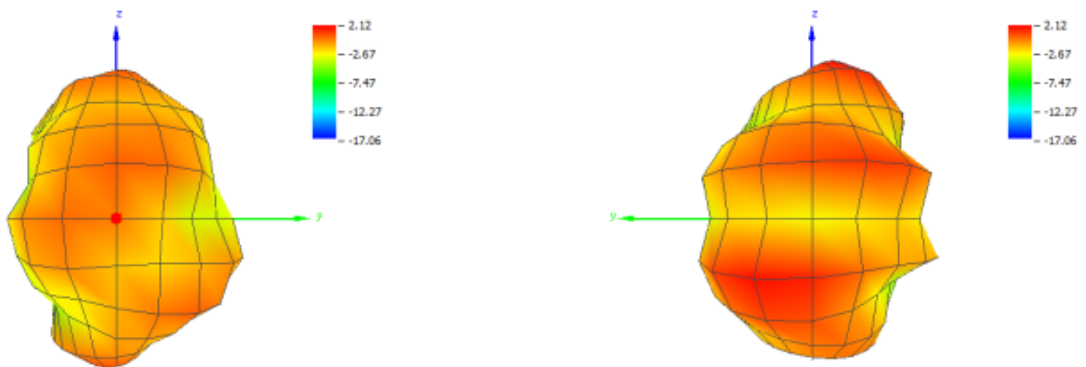


Figure C.7: Compact helical antenna, radiation patterns.
 Left: $\theta = 90$, $\phi = 0$; Right: $\theta = 90$, $\phi = 180$

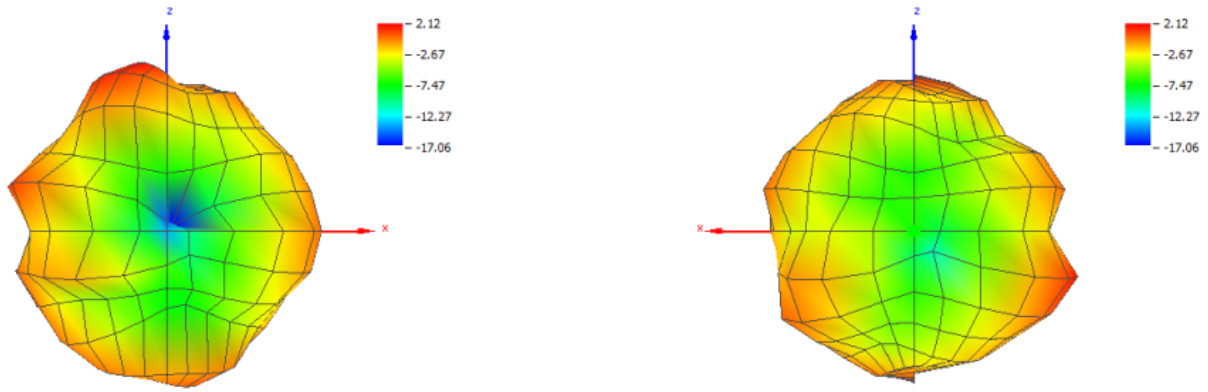


Figure C.8: Compact helical antenna, radiation patterns.
 Left: $\theta = 90$, $\phi = 270$; Right: $\theta = 90$, $\phi = 90$

C.3 Antennas test PCB

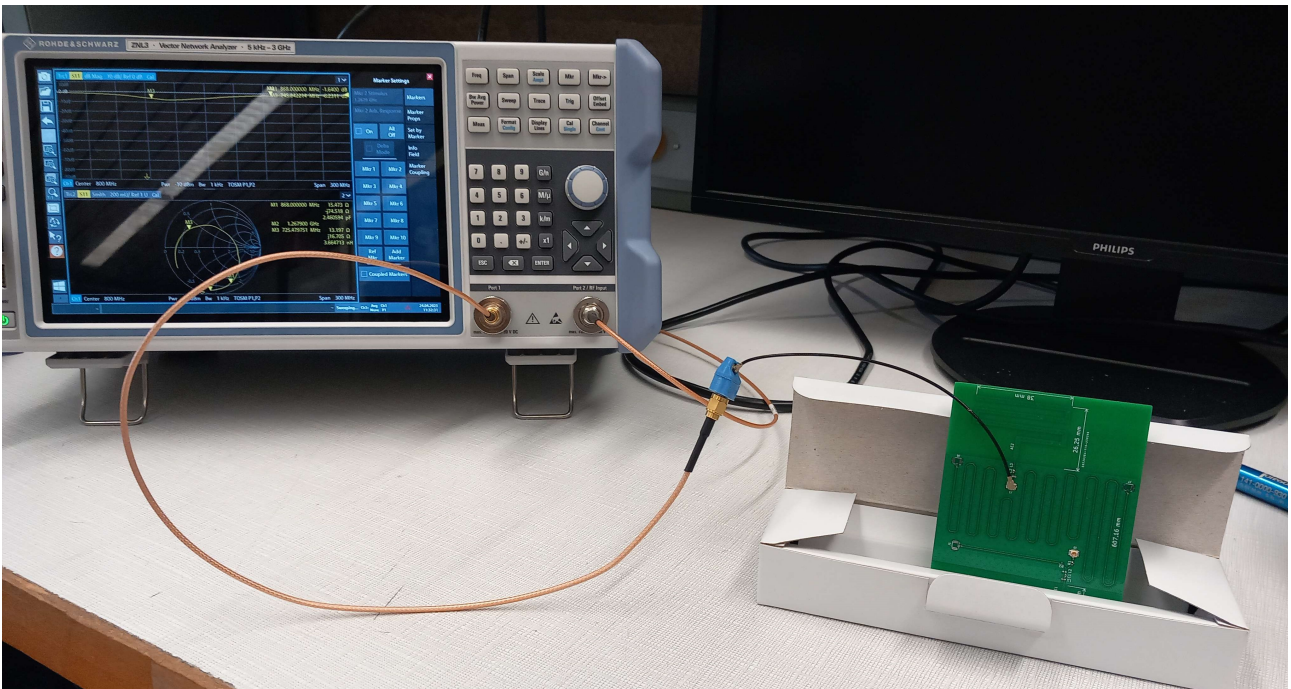


Figure C.9: Antenna test setup thanks to the VNA.

Appendix D

Standalone PCB

D.1 MCU pin assignment

n°	Name	Label	n°	Name	Label	n°	Name	Label
1	PB3	SW0	18	NRST	-	35	PA12	OPT3001_SCL
2	PB4	OPT3001_EN	19	PH3	-	36	PA13	SWDIO
3	PB5	NC	20	RFI_P	-	37	VBAT	-
4	PB6	NC	21	RFI_N	-	38	PC13	NC
5	PB7	HS3003_SDA	22	RFO_LP	-	39	PC14	LSE_OSC_IN
6	PB8	HS3003_SCL	23	RFO_HP	-	40	PC15	LSE_OSC_OUT
7	PA0	NC	24	VR_PA	-	41	VDDA	-
8	PA1	NC	25	VDDPA	-	42	PA14	SWCLK
9	PA2	JLink_Rx	26	OSC_IN	-	43	PA15	HS3003_EN
10	PA3	JLink_Tx	27	OSC_OUT	-	44	VDD	-
11	VDD	-	28	VDDRF	-	45	VFBSMPS	-
12	PA4	status_0	29	VDDRF1V55	-	46	VDDSMPS	-
13	PA5	status_1	30	PB0	BATT_FB_EN	47	VLXSMPS	-
14	PA6	status_2	31	PB2	BATT_FB	48	VSSMPS	-
15	PA7	CTRL_antenna	32	PB12	OPT3001_INT	49	VSS	-
16	PA8	T3902_CLK	33	PA10	T3992_DATA			
17	PA9	T3902_EN	34	PA11	OPT3001_SDA			

Table D.1: STM32WLE5C8 pin assignment

D.2 Standalone PCB bill of materials

Component	Precision	Quantity	Price [€]	Total [€]
PCB		1	10	10
Antenna	868 MHz	1	0	0
Sensors	HS3003	1	4.45	4.45
	T3902	1	1.02	1.02
	OPT3001	1	3.10	3.10
E-peas	AEM10941	1	5.22	5.22
MCU	STM32WLE5C8U	1	8.69	8.69
Crystals	32 kHz	1	0.72	0.72
	32MHz	1	0.55	0.55
Solar panel	Panasonic	1	3.95	3.95
Supercapacitor Capacitor [F]	2	3	1.22	3.66
	0.1 μ	7	0.02	0.14
	47n	1	0.03	0.03
	2.7p	3	0.11	0.33
	68p	1	0.14	0.14
	3.6p	1	0.09	0.09
	5.6p	1	0.14	0.14
	3.3p	1	0.79	0.79
	33p	2	0.13	0.26
	1.3p	1	0.25	0.25
	10 μ	5	0.04	0.20
	22 μ	2	0.17	0.34
	4.7 μ	2	0.08	0.16
	1 μ	1	0.09	0.09
	470n	1	0.09	0.09
	10n	1	0.15	0.15
	10p	2	0.08	0.16
7p	2	0.03	0.06	
Inductor [H]	11n	1	0.16	0.16
	3.3n	1	0.23	0.23
	47n	1	0.20	0.20
	3n	1	0.24	0.24
	8.7n	1	0.21	0.21
	1.8n	1	0.19	0.19
	10 μ	2	0.13	0.26
	15 μ	1	0.11	0.11
	100 μ	1	0.10	0.10
Resistor [Ω]	100	3	0.09	0.27
	10k	6	0.02	0.12
	0	1	0.26	0.26
	100k	2	0.11	0.22
	50	1	0.05	0.05
Connectors	2 pins	4	0.05	0.20
	terminal block	1	1.3	1.3
	ufl	1	0.67	0.67
	swd 20 pins	1	1.3	1.3
	jumpers	3	0.09	0.27
Rf switch		1	0.44	0.44
Button		1	0.20	0.20
TOTAL				51.78

Table D.2: Bill Of Materials (BOM) for a standalone sensing node

Bibliography

- [1] “Semtech website.” <https://www.semtech.com/>. Accessed June 2023.
- [2] Mobilefish, “Lora/lorawan tutorial 12: Modulation types and chirp spread spectrum.” https://www.youtube.com/watch?v=lg0eZWZFKiE&ab_channel=Mobilefish.com. Accessed June 2023.
- [3] Semtech, “Lora module sx1276 datasheet.”
- [4] “Pioneering a new era in the internet of things: Lorawan.” <https://www.iob.watch/lorawan>.
- [5] “Nucleo-wl55jc.” <https://www.st.com/en/evaluation-tools/nucleo-wl55jc.html>. Accessed June 2023.
- [6] STMicroelectronics, “Stm32wl5x datasheet.”
- [7] “Monopole antenna vs dipole antenna-difference between monopole antenna and dipole antenna.” <https://www.rfwireless-world.com/Terminology/monopole-antenna-vs-dipole-antenna.html>. Accessed June 2023.
- [8] “Jlpcb.” <https://jlpcb.com/pcb-impedance-calculator/>. Accessed June 2023.
- [9] T. Instruments, “Application report swra496a.” <https://www.ti.com/lit/an/swra496a/swra496a.pdf?ts=1677569599949>.
- [10] T. Instruments, “Design note dn024: Monopole pcb antenna.” https://www.ti.com/lit/an/swra227e/swra227e.pdf?ts=1677003700601&ref_url=https%253A%252F%252Fwww.google.com%252F.
- [11] T. Instruments, “Design note dn038: Miniature helical pcb antenna.” <https://www.ti.com/lit/an/swra416/swra416.pdf>.
- [12] B. Buurman, J. Kamruzzaman, G. Karmakar, and S. Islam, “Low-power wide-area networks: Design goals, architecture, suitability to use cases and research challenges,” *IEEE Access*, vol. 8, pp. 17179–17220, 2020.
- [13] DigiKey, “Comparaison des technologies sans fil basse consommation.” <https://www.digikey.be/fr/articles/comparing-low-power-wireless-technologies>. Accessed June 2023.
- [14] K. Sandeep and L. Joakim, “Application note an092: Measuring bluetooth low energy power consumption.” https://www.ti.com/lit/an/swra347a/swra347a.pdf?ts=1684436811539&ref_url=https%253A%252F%252Fwww.bing.com%252F. Accessed June 2023.
- [15] “Lora calculator.” <https://lora-developers.semtech.com/build/tools/calculator>. Accessed June 2023.
- [16] “Bluetooth website.” <https://www.bluetooth.com/learn-about-bluetooth/tech-overview/>. Accessed June 2023.
- [17] “Sigfox technical overview.” <https://www.avnet.com/wps/wcm/connect/onsite/03aebfe2-98f7-4c28-be5f-90638c898009/sigfox-technical-overview.pdf?MOD=AJPERES&CVID=magVa.N&CVID=magVa.N&CVID=magVa.N>. Accessed June 2023.

- [18] “Nb-iot vs lte-m : Choisir la meilleure technologie pour vos besoins iot.” https://www.google.com/url?q=https://lab4sys.com/fr/nb-iot-vs-lte-m-choisir-la-meilleure-technologie-pour-vos-besoins-iot/?cn-reloaded%3D1&sa=D&source=docs&ust=1686234606577309&usg=A0vVaw13-B_xfBTbu1jEe8iIRONY. Accessed June 2023.
- [19] “Aem10941 solar energy harvesting.” <https://e-peas.com/product/aem10941/>. Accessed June 2023.
- [20] C. Blair, “Why antenna length is $\lambda/4$.” <https://bestreviews.technology/why-antenna-length-is-lambda-4/>. Accessed June 2023.
- [21] “Breif tutorial on smith charts.” <https://k6jca.blogspot.com/2015/03/a-brief-tutorial-on-smith-charts.html>. Accessed June 2023.
- [22] L. Joris, F. Dupont, P. Laurent, P. Bellier, S. Stoukatch, and J.-M. Redouté, “An autonomous sigfox wireless sensor node for environmental monitoring,” *IEEE Sensors Letters*, vol. 3, no. 7, pp. 01–04, 2019.
- [23] W. Wang, N. Wang, E. Jafer, M. Hayes, B. O’Flynn, and C. O’Mathuna, “Autonomous wireless sensor network based building energy and environment monitoring system design,” in *2010 The 2nd Conference on Environmental Science and Information Application Technology*, vol. 3, pp. 367–372, 2010.
- [24] M. Meli, M. Brütsch, S. Stajic, M. Böbel, D. Lorenz, L. Hegetschweiler, D. Karanassos, and C. Kouzinopoulos, “Low light energy autonomous lorawan node,” in *2020 IEEE 5th International Symposium on Smart and Wireless Systems within the Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS)*, pp. 1–6, 2020.
- [25] A. I. Petrariu, A. Lavric, and E. Coca, “Renewable energy powered lora-based iot multi sensor node,” in *2019 IEEE 25th International Symposium for Design and Technology in Electronic Packaging (SIITME)*, pp. 94–97, 2019.
- [26] M. Rezzouki, S. Ouajih, and G. Ferré, “Monitoring social distancing in queues using infrared array sensor,” *IEEE Sensors Journal*, vol. 22, no. 17, pp. 16764–16771, 2022.
- [27] G. Pasolini, “On the lora chirp spread spectrum modulation: Signal properties and their impact on transmitter and receiver architectures,” *IEEE Transactions on Wireless Communications*, vol. 21, no. 1, pp. 357–369, 2022.
- [28] L. Vangelista, “Frequency shift chirp modulation: The lora modulation,” *IEEE Signal Processing Letters*, vol. 24, no. 12, pp. 1818–1821, 2017.
- [29] “Lora alliance website.” <https://lora-alliance.org/>. Accessed June 2023.
- [30] B. Taoufik, D. Jean-François, C. Jean-Jacques, J. Randa, and A. Guillaume, “Energy consumption model for sensor nodes based on lora and lorawan,” *Sensors*, vol. 18, p. 2104, 2018.
- [31] “Application note an5406: How to build a lora® application with stm32cubewl.” https://www.ti.com/lit/an/swra347a/swra347a.pdf?ts=1684436811539&ref_url=https%253A%252F%252Fwww.bing.com%252F. Accessed June 2023.
- [32] “Stmicroelectronics youtube videos.” <https://www.youtube.com/@stmicroelectronics>. Accessed June 2023.
- [33] “Stmicroelectronics articles.” <https://community.st.com/s/article/lora-communication-within-stm32>. Accessed June 2023.
- [34] S. Santos, “Esp32 lora sensor monitoring with web server (long range communication).” <https://randomnerdtutorials.com/esp32-lora-sensor-web-server/>. Accessed June 2023.

- [35] lewisxhe, “Github page: Lilygo-lora-series schematic t3_v1.6.” https://github.com/Xinyuan-LilyGO/LilyGo-LoRa-Series/blob/master/schematic/T3_V1.6.pdf. Accessed June 2023.
- [36] S. Santos, “Ttgo lora32 sx1276 oled board: Getting started with arduino ide.” <https://randomnerdtutorials.com/ttgo-lora32-sx1276-arduino-ide/>. Accessed June 2023.
- [37] PCBWay, “How to get started with lora lpwan using ttgo lora32.” <https://randomnerdtutorials.com/ttgo-lora32-sx1276-arduino-ide/>. Accessed June 2023.
- [38] P. Dhaker, “Introduction to spi interface.” <https://www.analog.com/en/analog-dialogue/articles/introduction-to-spi-interface.html>. Accessed June 2023.
- [39] Tsla, “Arduino i2c ssd1306 oled screen tutorial.” <https://eecs.blog/arduino-i2c-ssd1306-oled-screen-tutorial/>. Accessed June 2023.
- [40] V. Fragoso, “Environmental sensing using lora and stm32.” <https://www.hackster.io/victorffs/environmental-sensing-using-lora-and-stm32-c0136e>. Accessed June 2023.
- [41] E. Mansour, R. Vishinkin, S. Rihet, W. Saliba, F. Fish, P. Sarfati, and H. Haick, “Measurement of temperature and relative humidity in exhaled breath,” *Sensors and Actuators B: Chemical*, vol. 304, p. 127371, 2020.
- [42] “Carbon dioxide concentration in rooms occupied with people.” https://www.engineeringtoolbox.com/pollution-concentration-rooms-d_692.html. Accessed June 2023.
- [43] H. presto, “Panneau solaire monocristallin ou polycristallin : que choisir ?.” <https://www.habitatpresto.com/mag/electricite/panneaux-photovoltaiques/panneau-solaire-monocristallin-polycristallin>. Accessed June 2023.
- [44] H. presto, “Panneau solaire amorphe, monocristallin ou polycristallin : quelle différence ?.” <https://blog.lumioo.com/2021/02/04/difference-panneau-solaire-amorphe-monocristallin-polycristallin/>. Accessed June 2023.
- [45] SunVival, “Solar cells comparison – amorphous vs monocrystalline vs polycrystalline.” <https://sunvivalguide.com/amorphous-vs-monocrystalline-vs-polycrystalline-solar-panel/>. Accessed June 2023.
- [46] J. Arguelles, “Using monocrystalline vs. amorphous solar panels.” <https://mysmartssurvival.com/using-monocrystalline-vs-amorphous-solar-panels/#>. Accessed June 2023.
- [47] E. Technology, “Series, parallel & series-parallel connection of solar panels.” <https://www.electricaltechnology.org/2020/09/series-parallel-and-series-parallel-connection-of-solar-panels.html>. Accessed June 2023.
- [48] C. S. Kouzinopoulos, D. Tzovaras, P. Bemnowicz, M. Meli, M. Bellanger, M. Kauer, J. De Vos, D. Pasero, M. Schellenberg, and O. Vujicic, “Amanda: An autonomous self-powered miniaturized smart sensing embedded system,” in *2019 IEEE 9th International Conference on Consumer Electronics (ICCE-Berlin)*, pp. 324–329, 2019.
- [49] L. Carosso, L. Mattiauda, and M. Allegretti, “A survey on devices exploiting lora communication,” *Acta Marisiensis. Seria Technologica*, vol. 17, no. 2, pp. 31–35, 2020.
- [50] e. a. Meli, Marcel, “Energy autonomous wireless sensing node working at 5 lux from a 4 cm² solar cell,” *Journal of Low Power Electronics and Applications*, vol. 13, no. 1, p. 12, 2023.

- [51] Arrow, "Supercapacitors: How and when to use." <https://www.arrow.com/en/research-and-events/articles/hands-on-with-supercapacitors>. Accessed June 2023.
- [52] "Humidity." https://en.wikipedia.org/wiki/Humidity#Relative_humidity. Accessed June 2023.
- [53] "A brief tutorial on smith charts." <https://k6jca.blogspot.com/2015/03/a-brief-tutorial-on-smith-charts.html>. Accessed June 2023.
- [54] "Understanding the fundamental principles of vector network analysis." <https://k6jca.blogspot.com/2015/03/a-brief-tutorial-on-smith-charts.html>. Accessed June 2023.
- [55] "Toggle the table of contents smith chart." https://en.wikipedia.org/wiki/Smith_chart#. Accessed June 2023.
- [56] P. Lab, "Kicad controlled impedance traces." https://www.youtube.com/watch?v=0fteCxn5XXA&ab_channel=Phil%E2%80%99sLab. Accessed June 2023.
- [57] Montaro, "Impedance calculator." <https://www.mantaro.com/resources/impedance-calculator.htm>. Accessed June 2023.
- [58] A. theory, "S parameters." <https://antenna-theory.com/definitions/sparameters.php>. Accessed June 2023.
- [59] "Nucleo-wl5jc schematics." https://www.st.com/content/ccc/resource/technical/layouts_and_diagrams/schematic_pack/group1/79/3d/ce/a5/1c/76/40/89/MB1389-WL55JC-highband-D04_schematic/files/MB1389-WL55JC-highband-D04_Schematic.pdf/jcr:content/translations/en.MB1389-WL55JC-highband-D04_Schematic.pdf.
- [60] "Kicad tutorial." https://docs.kicad.org/7.0/fr/getting_started_in_kicad/getting_started_in_kicad.html. Accessed June 2023.