UCLouvain

École polytechnique de Louvain

Design and characterisation of a wireless sensing node for natural-environment monitoring.

Author: **Nicolas Brusselmans**
Supervisors: **David Bol, Jérôme Louveaux**
Readers: **Laurent Francis, Pol Maistriaux, Marco Gonzalez**
Academic year 2022–2023
Master [120] in Electrical Engineering

Acknowledgements

First of all, I would like to thank Prof. David Bol and Prof. Jérôme Louveaux, for their availability and the interest they have shown in my work. The regular meetings and the integration into the ECS group by inviting me to the lunches particularly encouraged me to surpass myself. I also greatly appreciate their constant reflection to give meaning to their reseach projects, beyond performance.

Secondly, I am particularly grateful to Pol Maistriaux and Marco Gonzalez who were ready to help me at any time. They gave me precious practical knowledge that made the fulfilment of my work possible.

During the last difficult days of my writing, I could always count on my roommates, the flourishing garden and the shining sun to relax between two paragraphs. To them : a big thanks.

I would also like to deeply thank my parents who supported me throughout my studies, in addition to financing them, even when the results were not as hoped. Furthermore, I particularly thank my mother for the entire proofreading of this work.

Finally, I lovingly thank Sirga, for her unfailing presence and for having patiently listened during the lenghty discussions about microcontrollers, PCBs and other stuff. The evening walks in the woods to capture data were a great pleasure.

Abstract

The world seems to catch fire every summer, with more and more reports of devastating wildfires and their consequences. Prevention and early detection of forest fires are key components of an efficient firefighting method. Wireless sensing networks are a promising technology in that field. However, experimental prototypes are needed to better learn and understand the specific constraints and limitations inherent to the particular forest environment.

This work proposes such a prototype. After an informed study of the requirements and specifications, the ForestMEv2 device is designed and built. It responds to the need of having a means of characterising and evaluating different performance metrics linked to natural-environment monitoring. Several experiments are conducted in a laboratory as well as in a real forest environment to assess the power consumption, the fire detection ability and the LoRa communication performances of the prototype. Advantage is also taken of its modularity to create an extension board, which focuses on the monitoring of bat population as a bio-indicator of ecosystems.

The results are two functional prototypes that bring valuable information about natural-environment monitoring and that are ready to be used for future projects.

Contents

List of Abbreviations

ADC Analog-to-Digital Converter

AFE Analog Front-End

AI Artificial Intelligence

API Application Programming Interface

BGA Ball Grid Arrays

BOM Bill of materials

BSEC Bosch Sensortec Environmental Cluster

BUI Buildup Index

CFO Carrier Frequency Offset

CPU Central Processing Unit

CSS Chirp Spread Spectrum

D2D Device-to-Device

DAC Digital-to-Analog Converter

DMA Direct Memory Access

DMM Digital MultiMeter

ESD Electrostatic Discharge

EVB Evaluation board

FAT File Allocation Table

FIR Finite impulse response

FWI Fire Weather Index

GPIO General Purpose Input/Output

HAL Hardware Abstraction Layer

IO Input/output

IoT Internet of Things

ISI Initial Spread Index

ISM Industrial, scientific and medical

I²C Inter-Integrated Circuit

LDO Low dropout regulator

LED Light-Emitting Diode

LoRaWAN LoRa Wide Area Network

LPWAN Low-Power Wide Area Network

LVS Low Voltage Signaling

MCU Microcontroller unit

MEMS MicroElectroMechanical System

NVIC Nested Vector Interrupt Control

OpAmp Operational Amplifier

PCB Printed circuit boards

PER Packet Error Rate

PLL Phase-Locked Loop

PMU Power management unit

RAM Random Access Memory

RF Radio frequency

RSSI Received Signal Strength Indication

RTC Real-Time Clock

RX Receiver

SF Spreading Factor

SFO Sampling Frequency Offset

SMA SubMiniature version A

SMD Surface-Mounted Device

SMPS Switched-Mode Power Supply

SNR Signal-to-Noise Ratio

SPI Serial Peripheral Interface

STO Symbol Timing Offset

SWD Serial Wire Debug

TF Transfer Function

TTN The Things Network

TX Transmitter

UART Universal Asynchronous Receiver Transmitter

UI User Interface

USB Universal Serial Bus

VOC Volatile Organic Compounds

WSN Wireless Sensing Network

µSD micro Secure Digital

List of Figures

List of Tables

Introduction

Context

During these recent years, the news has been regularly filled with reports of large wildfires and their devastating consequences. Each summer, the whole world seems to catch fire and this is not quite an exaggeration as several new dramatic records of worst wildfire season have been set these last years. In 2020, almost ten thousand fires have burned over 1.75 million hectares in California, making that year "the largest wildfire season recorded in California's modern history" [1]. At the opposite side of the world, Australia also faced a year earlier an unusual bushfire season, considered a megafire due to the dimension it took and its uncontrollability. Between September 2019 and March 2020, the fires burned a total of 24.3 million hectares, destroying precious natural ecosystems and human infrastructures, as well as provoking the death of at least 34 people [2].

Wildfires are not only a direct threat to human life because people get caught in the fire or later suffer from health problems indirectly linked to the fires, but also an indirect threat in several ways. From the economic point of view, losses are caused by the destruction of houses and other infrastructures, but the total costs are far greater than the direct costs of the fire's destruction, as they must include firefighting costs and wider societal and environmental damage [3]. Furthermore, the environment suffers from a direct loss of ecosystems, with fauna and flora annihilated over large areas, and also from large quantities of greenhouse gases emitted into the atmosphere that contribute to a worsening of the climate change. This in addition results in a positive feedback loop that further creates conditions conducive to the outbreak of fires.

The problem resides in recent trends that the intensity and ocurrence of megafires have increased drastically. To face that problem and reduce its costs, actions on several levels must be considered [3]. The forests and other ecosystems prone to wildfires must be managed so that the risk of a fire breaking out and the seriousness of its consequences are decreased. This includes cutting the total area into smaller parts with the help of firebreaks to limit the fire propagation, or preventively burning some areas to remove fuel and thus decrease the intensity of an uncontrolled wildfire. Another level is the population management. The more scattered around the infrastructures, the more difficult it is to clearly separate them from the flammable ecosystem. Education of people is also an important factor to limit the number of human-started fire and improve the behaviours they adopt in case of a fire event. Finally, because the risk of fire cannot be brought to zero, a fair amount of resources must be deployed in the prevention and detection of fires.

State of the art

The detection of wildfires, and most particularly early detection, is important. Indeed the cost of extinguishing a fire grows exponentially with time. There is a rather well-known saying that one would only need one litre of water to put out a fire that has been burning for a minute, but a thousand litres if it has been burning for 10 minutes. It is thus critical to detect it as early as possible.

Several detection techniques exist, they differ in their advantages and drawbacks [4].

First, there is the simplest form of detection, made by people either on a voluntary basis or engaged by an institution to do so, and then reported to the authorities. This method is neither fast nor efficient, as the fire must already be large to be seen and there must be somebody present at that place.

Then come the techniques that use imaging techniques to recognise a fire by its fumes or flames. Satellites can be used to take images of the Earth for this purpose but suffer from a low

time resolution (it takes between 1.5 and 2 hours to complete a full orbit in Low Earth Orbit [5]) and are very expensive, whereas cameras can record and analyse real-time videos but require to be at the border of the forest for power supply and Internet access. In addition to their already cited disadvantages, using images has a critical drawback : the fire must be very large to be detected, i.e. there must be a big amount of smoke and a direct line of sight to it, and it thus cannot perform early detection.

Other types of cameras use infra-red or LiDAR technologies and detect radiated heat or the light spectral characteristics of smoke to get rid of the constraints of images, e.g. they work in the dark, but still need a big fire and a line of sight.

The last detection technique is the one this thesis focuses on, the Wireless Sensing Network (WSN). The idea is to spread a large amount of independent sensing nodes, that can monitor their environment through one or several sensors, evaluate the risk or presence of fires by means of computation algorithms, then transmit the results to be reported to the authorities. The huge advantage that this technique brings over the others is that it effectively performs *early* detection. Indeed, as it does not depend on any line of sight from the border of the forest or the skies, nor does it need a big amount of smoke or heat to identify a fire, an alarm can be raised in a matter of minutes after the start of a fire. Furthermore, in addition to the detection of an occuring event, WSNs can be used to predict the risk of a fire, by regularly computing the Fire Weather Index (FWI) for example [4].

Contribution

Nowadays, WSN technologies are rapidly evolving because they provide numerous benefits in many different domains. Among these are applications such as military usages, industrial process automation, urban smartification, health assistance or environmental monitoring [6]. While the purpose of the WSN is very versatile, a common feature to most applications is that it relies on nearby human infrastructures to operate. This is because the classical structure of a WSN is composed of a series of end nodes that directly communicate with their base station, which must be directly accessible to bring the Internet and power supply. Wide natural environments create a new challenge because of the long distances between the nodes and their gateway. In that case, a meshed network is preferred for its ability to transmit information hop-by-hop over distances unreachable in a single communication step. Moreover, a forest environment is not like any human-modified environment, such as rural or urban areas. It is thus not yet well-known and characterisation of the behaviour of a WSN in such an environment would bring valuable information.

This master thesis proposes a new prototype for a sensor node of a WSN, based on an already available module which is presented as the use case of [7]. The design choices are based on the analysis of the needs and constraints of such a prototype, specifically for natural-ecosystem monitoring in a forest environment. Once the device is built, it serves as a means of testing and assessing different figures of merit, such as its ability to prevent and detect forest fires or the communication performances achievable in that environment.

Another useful information that this prototype can provide is the environmental impact due to the deployment of an entire WSN, through the computation of a life cycle analysis. This aspect, often overlooked during the design phase, is important to make sure that the proposed solution is relevant and responsible. The environmental costs of the full life cycle of a proposed solutoin must not exceed the benefits brought by its usage, otherwise it is not worth implementing. Such a comparison must be based on quantitative analyses of the positive and negative impacts of both parties. This study [8] proposes a robust methodology for assessing the carbon footprint of WSN's end nodes.

Designing a prototype for experimental purposes is an opportunity to make it versatile and modular. By providing the right interface, the device can easily be extended by adding one or several supplementary functionalities as peripherals, such as other sensors, more computing power and memory, communication links, a graphical interface or else.

As a proof of concept, an extension is developed in this work to extend the natural-environment monitoring capabilities of the prototype.

As already discussed, uncontrolled forest fires lead to ecosystem losses and climate change. This affects not only humans but also all the biodiversity, fauna and flora included. In order to

understand the complex behaviour of biological communities in this kind of situation, it is critical to monitor bio-indicators that show measurable responses to climate change and habitat loss. Bats are identified as such a bio-indicator, because their presence and the size of their population give precious informations about the quality of the underlying ecosystem and biodiversity [9]. Monitoring bats through an automatic acoustic recording method proves to be relevant and effective for extracting useful information about the environment [10]. Concretely, a extension board is proposed as a solution for using the device for that purpose.

Structure

The document is organised in five main chapters, each covering an aspect of this work.

Chapter 1 analyses the already existing module, its different functions and weaknesses, and shows why a new prototype is needed.

In Chapter 2, different possible design directions are compared and choices are expressed from the high-level hardware architecture to the specific components used for the solution.

Then the different prototyping phases are explained in Chapter 3. Those are the construction of the schematic, the layout, the assembly and the functionnality tests.

After that, several experiments are conducted to characterise the power consumption, the detection capabilities and the communication performances of the solution in its target environment. The experimental setups and results are explained in Chapter 4.

Finally, Chapter 5 presents the extension module for bat monitoring, by explaining the design as well as the main performances and limitations of the solution.

Chapter 1

Initial state of the project

The first chapter of this work is focused on the analysis of the already existing prototype, that was conceived for the purposes of another research, presented in [7]. Its differents functionalities are explained, together with background concepts to lay the foundations needed to understand this work.

The module is composed of three main modules, assembled on a breadboard with the help of jumper wires. Figure 1.1 shows the different parts and their wired interconnections : the green rectangle highlights the microcontroller unit (MCU), linked to the sensor circled by the blue rectangle and the radio module in red.

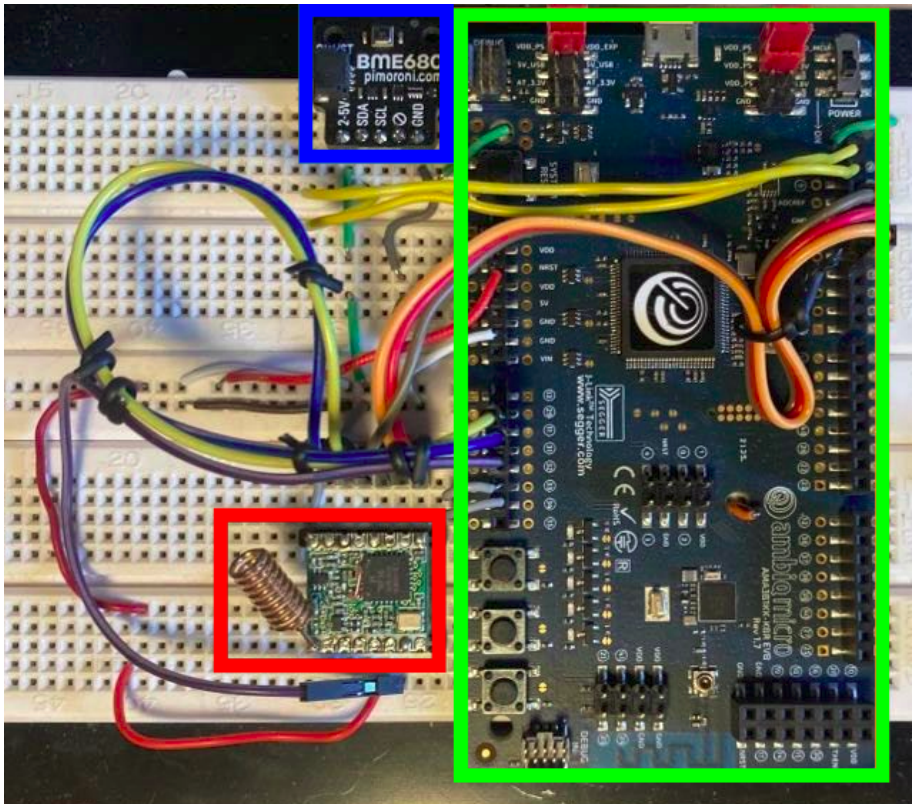


Figure 1.1: The initial module assembled on a breadboard.

1.1 BME680 sensor

The sensor's role is to collect environmental data. It is the first step required in order to evaluate the state of the forest, by transducing a physical quantity into a numerical value that can be manipulated by the system. This sensor, the BME680 from Bosch [11], measures four different quantities : the temperature, the relative humidity, the pressure and the concentration of Volatile

Organic Compounds (VOC). The first three are useful for the *prevention* of forest fires, by regularly computing the FWI, as explained in Subsection 1.2.1. The gas concentration is used to *detect* a burning fire and trigger an alarm if so.

Using a sensor that is sensitive to VOCs has been proven relevant, as a significant amount of thoses gases is emitted when vegetation is burning [12]. The detailed analysis of the sensor and the different transfer functions between physical and electrical quantities are outside the scope of this work. However, a basic knowledge of its operation is required to use it properly, as much for its configuration as for the understanding of its outputs.

The BME680 has a digital interface, i.e. the data is already sampled by an embedded analog-to-digital converter (ADC), and communicates with the system using an Inter-Integrated Circuit (I^2C) data bus. It is a simple serial bus that allows the BME680 to transfer digital data to its host. Through this bus, the host can initialise and configure the sensor as well as retrieve the environmental data captured. The main configuration parameter is the sampling rate of the digital conversion, which can either be a fixed value, e.g. 1Hz, or undefined in which case the sampling is executed whenever asked by the master. Two other parameters can be configured for each of the three first quantities, i.e. temperature, relative humidity and pressure. The first is the oversampling factor, that corresponds to the number of successive acquisitions that are performed then averaged by the sensor. It provides a crude low-pass filter and thus reduce the noise on the measurement at each sample. The second one is the number of taps of the available finite impulse response (FIR) filter. It provides another low-pass filtering effect, but this time using several samples. It should only be used with constant sampling rate, because the filter is based on regularly time-spaced taps, otherwise its action will be meaningless. The BME680 already processes in hardware the digitised quantities to provide an integer representation in physical units. More precisely, the temperature is in hundredths of °C, the relative humidity is in thousandths of % and the pressure in Pa. The range of operation covers entirely the possible values in the target application, with a measurable temperature from -40°C to 85°C, a relative humidity from 0 to 100% and a pressure from 300 to 1100hPa.

The gas sensor however has a more complex operation, for two main reasons. First, because of the physical behaviour used for the sensing. The gas sensor is a thin plate made of an active material that reacts to certain gases, e.g. VOCs in this case. Its resistance varies with the gas concentration with a sensitivity dependant on the temperature of the plate. In order for the sensitivity to be high and stable, the plate needs to be heated to several hundreds of degrees Celsius. The sensing is thus divided into a heating phase, and then the sampling. In addition to the same parameters as for the previous quantities, the gas sensor must be configured with two additionnal values : the target temperature of the plate, and the heating duration. Both have a strong impact on the output resistance that the sensor measures, but the analysis of those dependencies is outside the scope of this work. It should be noted that the objective is not to obtain an accurate value of the gas concentration but rather its variations. To be representative of the reality, a calibration step is required.

The second reason is that the measured quantity, i.e. the resistance, is not simply proportionnal to the gas concentration. Indeed, the transfer function between the gas concentration [VOC] and the output resistance R_{sens} can be modelled as following.

$$R_{sens} = a[\text{VOC}]^m \quad (1.1)$$

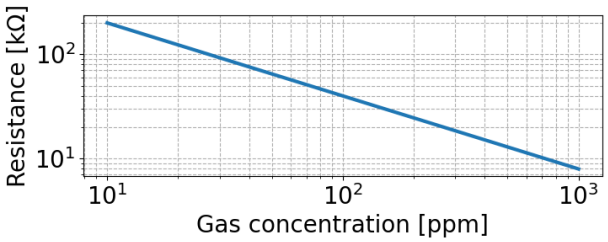


Figure 1.2: Gas sensor transfer function model.

a and m depend on the aforementionned parameters but also on the technical characteristics of the sensor. Furthermore, m is negative and is equal to the slope of the transfer function when plotted in log-log scale (Figure 1.2).

The important behaviour that needs to be highlighted is that the sensor reacts by exhibiting a lower resistance than usual in the presence of gas.

1.2 Apollo3 MCU

The microcontroller is the Apollo3 from Ambiq [13]. It is an ultra-low power MCU, primarily optimised for high energy efficiency in active mode with only 6µA/MHz, that is highly integrated with a ARM Cortex-M4 core, 1MB flash memory and 384kB of RAM. With these high-performance specifications, this MCU is able to manage quite heavy processing tasks while staying low-power and suitable for battery-powered applications.

The MCU is the central part of the system as it is responsible for managing the processing, i.e. running the software that communicates with the peripherals to fetch, process and communicate data. As already mentionned, the microcontroller uses an I^2C bus to get data from the BME680 sensor at a configurable sampling rate. The measured quantities are used as inputs to compute the FWI and detect the present of smoke in the air. Then the results are sent to the radio module using an Serial Peripheral Interface (SPI) bus.

1.2.1 Fire Weather Index

The Fire Weather Index is an index based on meteorological informations that estimates the fire danger under different aspects, such as the risk of ignition or the spreading behaviour. It has been imagined and developped in Canada in the seventies to obtain a day-by-day estimate of the risks and help to prevent unwanted forest fires [14].

The FWI process flow chart is presented in Figure 1.3. It relies on different weather observations that are measured each day at noon to compute fuel moisture codes. With those codes are then computed sub-indices, finally aggregated in the final index.

Fire weather observations Two of the needed observations can be locally captured by the system : the temperature and relative humidity, directly available at the output of the sensor. The others, i.e. wind and rain, should be brought to the system either by an external source through the radio communication channel, or by another sensor. For example, it could be possible to compute the average wind speed based on the output of the pressure sensor [15], even if this is not implemented nor considered in this work.

Fuel moisture codes Those codes represent the moisture of the different types of fuel that compose the forest soil. The dryer the organic matter lying on the floor, the more risk it is of forest fires. The three different codes are linked to three different levels of decomposed organic matters. The Fine Fuel refers to the surface litter, where a fire can start. The Duff Moisture Code represents the moisture in the underneath level, below the surface, and the Drought Code applies to buried flammable matters deep into the soil.

Fire behaviour indices The important and useful indices are the Initial Spread Index (ISI), the Buildup Index (BUI) and the final FWI. The first integrates fine fuel moisture and wind speed to provide an estimate of the ignition risk and spreading behaviour. The BUI, computed from deeper organic matters moisture level, represents the potential heat released by a burning fire, and gives knowledge about the severity of such an event. Those two sub-indices are aggregated into the final Fire Weather Index indicating the total dangerousness and risk of a fire event.

1.3 SX1276 radio

The radio module is the second peripheral, used to receive and transmit data through wireless communication. It is built around the chip SX1276 from Semtech [17]. The MCU initialises, configures and transmits data through a SPI bus. That bus is more complex than I^2C because it uses one more wire to separate the communication from the host to the peripheral from the other data direction, but that allows it to reach a higher clock frequency, a higher datarate and a duplex communication.

1.3.1 LoRa

This chip is specifically designed to communicate through the physical proprietary radio communication technique named LoRa, from “long range”, also owned by Semtech [18].

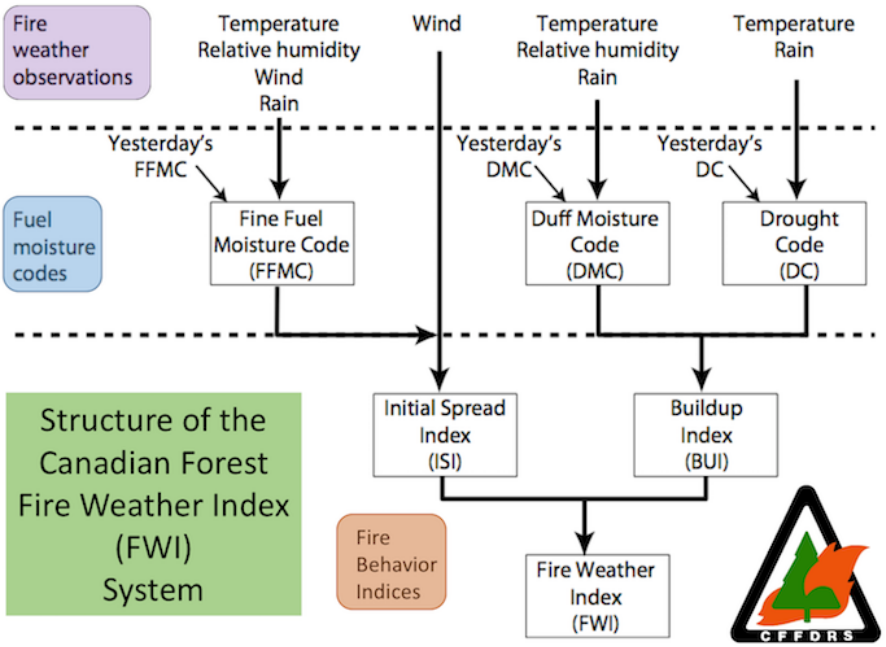


Figure 1.3: Fire Weather Index process flow chart, from [16].

It uses license-free sub-gigahertz radio frequency bands. In Europe, the allowed industrial, scientific and medical (ISM) band used by LoRa is the band between 863 and 870 MHz. The European legislation prohibits the emission at power levels higher than 14dBm [19], even if it is technically achievable by most of the LoRa-compliant chips, including the SX1276.

The physical LoRa modulation is derived from the Chirp Spread Spectrum (CSS) modulation technique. In CSS, each symbol is modulated as linear increase of the frequency cyclically shifted around the central frequency, called a chirp. The bitrate (BR) depends on three important parameters according to Equation 1.2 [20]. The bandwidth (BW) defines the range of frequencies modulated in each chirp, while the Spreading Factor (SF) determines the number of frequency steps in each chirp. The coding rate (CR) defines the fraction of useful bits. The others are redundant and act as a forward error correction code to make LoRa able to endure interference with a low bit error rate.

$$\mathrm{BR} = \mathrm{SF}\ \frac{\mathrm{BW}}{2^{\mathrm{SF}}}\ \mathrm{CR} \tag{1.2}$$

The SF offers a tradeoff between, on the one hand, datarate and consequently sending duration, and on the other hand and the minimum Signal-to-Noise Ratio (SNR) required by the demodulator, as displayed in Table 1.1. LoRa allows to demodulate below the noise floor thanks to that technique. This, in combination with a very high sensitivity, explain the particularly high link budget and thus the capacity of LoRa to communicate over large distances at low power.

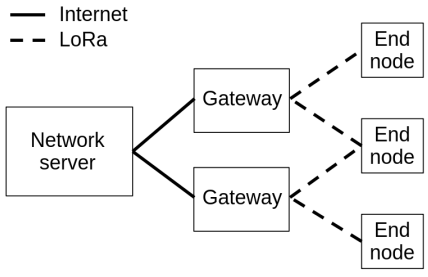


Figure 1.4: LoRaWAN architecture.

On top of the physical layer, there exists a network protocol called LoRa Wide Area Network (LoRaWAN) [21]. It is designed to support a massive integration of Internet of Things (IoT) devices and defines network specifications that the end nodes and gateways must respect to work properly. LoRaWAN makes the bridges between those devices and the Internet by providing a low-power link between them, as shown in Figure 1.4.

Spreading factor	SF7	SF8	SF9	SF10	SF11	SF12
SNR [dB]	-7.5	-10	-12.5	-15	-17.5	-20

Table 1.1: Minimum SNR required for demodulation as a function of the spreading factor, from [17].

1.4 Hardware components

As highlighted in Figure 1.1, the three chips are not directly used alone, but are rather integrated in printed circuit boards (PCB) together with other passive or active components.

Apollo3 EVB The main board is the evaluation board (EVB) of the Apollo3. The most important component is of course the MCU, with its crystal oscillators, decoupling capacitors and other passive components required for its proper operation. Every useful pin connection is available through connectors at the edges of the board.

There is also another microcontroller that is used as a debugger. It is an interface between the main MCU and a host computer. It translates the Universal Serial Bus (USB) communication from the computer to program the MCU or debug it while a software is running. The development environment on the host computer used to program and debug the Apollo3 is Keil µVision, from ARM [22].

The board also embeds components for the management of power supply. The power is provided by the USB connection, either from the host computer or a USB power bank. The voltage is then decreased to 3.3V or 1.8V by a configurable low dropout regulator (LDO) before being delivered to the different components.

Finally, an user interface composed of buttons and a light-emitting diode (LED) provides a way of interacting with the software at runtime.

BME680 EVB The BME680 chip is not alone on its evaluation board. It also comes with an LDO, to regulate an input voltage between 2V and 5V to 1.8V because the sensor operates at that voltage. Furthermore, the two wires of the I^2C bus pass through level shifters so that the host and peripheral can communicate even if their digital voltage is not the same.

RFM95 The radio module is the RFM95 [23]. It is mainly composed of the SX1276 chip with the needed passive components, e.g. decoupling capacitors, plus a radio frequency (RF) chain of components to transmit the output signal of the chip to the external antenna. That chain is a matched-impedance network of components that has several roles, such as filtering the signal or switching between the reception and emission paths.

1.5 Limitations

The presented system is a functional proof of concept, but it suffers from several limitations, that can be summarised in two points.

1.5.1 Power consumption

First, as just explained, the evaluation boards assembled for this system embed not only the required components, but also others that are not stricly necessary. Those extra components may consume a significant amount of energy. That is the case of the LDO and level shifters on the sensor board, which are not needed if the MCU already operates at the right voltage. On the main board, the on-board debugger is really helpful for a fast prototyping but also consumes a lot of energy if it is not disconnected. The radio module does not have unnecessary energy-consuming components, but attention needs to be paid to the fact that it could be too powerful for the European legislation. Indeed the SX1276 features two outputs : a high-power one and a low-power one. While the low-power output can deliver up to 14dBm, which stays in the legal range (see Section 1.3.1), the high-power one goes up to 22dBm. The use of the latter at high power is not authorised in Europe, thus it would be safer if it was not available at all.

The static consumption of the extra components doesn't allow the system to reach a low power budget required by the targeted application. Indeed, the sensor node should either harvest energy in the environment, or operate on batteries for a long time, i.e. months or even years.

1.5.2 Robustness

The second and most important reason why this system needs to be improved is for robustness and practical use. This assembly, made of jumper wires and boards pinned to a breadboard, is quite flimsy. It cannot be easily transported in experimental field, i.e. in a forest environment, without risking to detach the wires, or that a loose part comes off.

Furthermore, the module is rather large. The three boards occupy a surface of about 100cm^2, and the power supply must still be included as it is needed as a secondary part for autonomous operation. Because the sensor nodes must be attached to a tree in a forest to fulfill their role, the smaller they are the better.

Chapter 2

Design choices

In this second chapter, different possible solutions are presented to cope with the limitions discussed previously. They all offer an improved robustness, mainly by proposing an assembly that doesn't require any interconnection jumper wires. To tackle with the extra power consumption, the solutions use the sensor and/or the MCU without their evaluation board, to keep only necessary components as parts of the system.

Six different solutions are considered. They are first described in details, then compared according to different figures of merit, such as the price, the power consumption and the ease of development. The purpose of this in-depth comparison is to objectify the design choices that are made and support the relevance of the chosen solution.

2.1 Competing solutions

Each of the six solutions is explained in the following sections. Figure 2.1 recapitulates them all schematically. To come up with a fair and objective comparison, the scope is strictly limited to the same functionalities. Those can be separated in three functional blocks as presented in the previous chapter : the MCU responsible for the processing, the sensor to acquire data and the radio to transmit them.

2.1.1 EVBs on a breadboard

This is the initial situation. Its structure as well as its limitations are described in details in Chapter 1. It correponds to the baseline of the system from which the different comparisons are made.

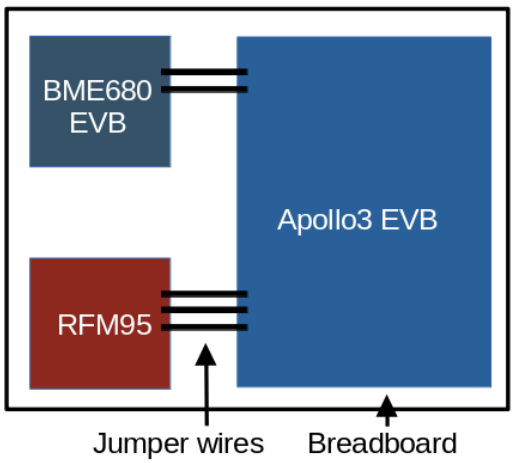
As a short reminder, the main characteristic of this module is that it is composed of three evaluation boards, linked by jumper wires on a breadboard.

2.1.2 Peripherals stacked on Apollo3 EVB

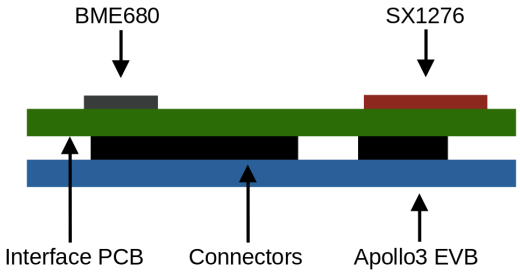
In this solution, the Apollo3 evaluation board is kept, but not the two other modules. Instead, a custom PCB is proposed, that includes the BME680 and the SX1276 chips, together with the other needed components, e.g. decoupling capacitors, pull-up resistors or RF network. It is stacked on top of the Apollo3 EVB by plugging regular 2.54mm pin headers into the dedicated connectors, offering on the one hand the required electrical connections between the two layers, and on the other hand a solid and robust assembly.

The module can still be programmed and debugged through the same software tools and hardware connection (USB) as for the initial situation. It however also still needs a 5V external USB power supply. The drawbacks in terms of power consumption of the peripherals disappear because the unnecessary components are removed, but the ones from the Apollo3 EVB, i.e. mostly the debugger's consumption, are still present.

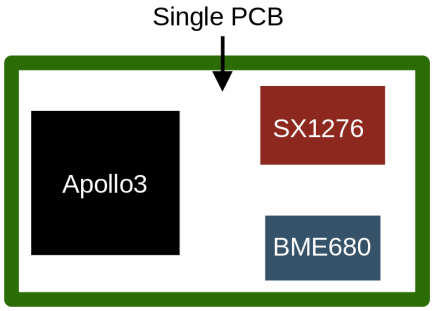
The added custom PCB offers possibilities regarding the integration of new functionnalities. Indeed, it opens the way to adding others sensors or a custom power management unit block for example.



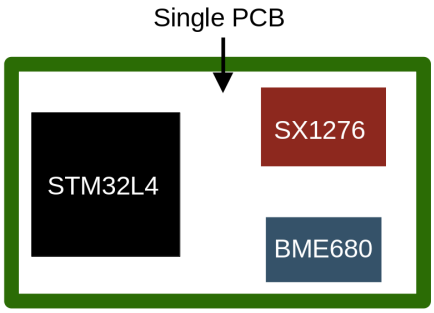
(a) EVBs on a breadboard.



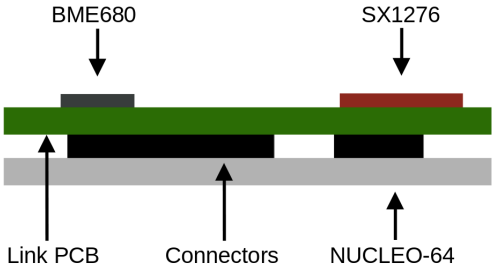
(b) Peripherals stacked on Apollo3 EVB.



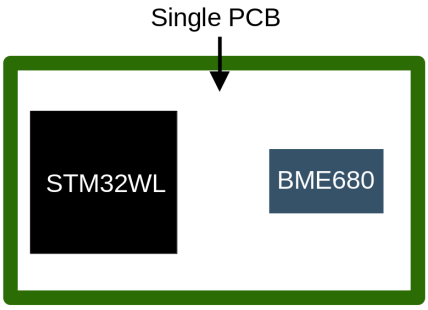
(c) Single custom PCB with Apollo3.



(d) Single custom PCB with STM32L4.



(e) Peripherals stacked on Nucleo board.



(f) Single custom PCB with STM32WL.

Figure 2.1: Schematics of the six competitors.

2.1.3 Single custom PCB with Apollo3

Here, the customisation goes further by proposing the inclusion of the Apollo3 chip onto the custom PCB, resulting in a single board with the three blocks on it.

In addition to the peripheral's chips and their auxiliary components, all the functionnalities that were useful on the Apollo3 EVB need to be added in the new custom PCB. It includes the crystal oscillators, the decoupling capacitors and a complete power management unit (PMU).

For the debugger, two possibilites can be considered : either an on-board debugger, that should be power-gatable because it is not needed during the system operation, or an external debugger, with only its appropriate connector on the board. The external solution is chosen to reduce the design complexity as well as the PCB footprint.

2.1.4 Single custom PCB with STM32L4

The possibility to choose other specific chips for the peripherals, i.e. the BME680 and SX1276, is not considered because no clear limitation about them is identified. However, the Apollo3 MCU suffers from several drawbacks (see Section 2.2) that lead to consider the use of another microcontroller. This solution is thus a single custom PCB with a different MCU and the same peripherals.

The scope of the analysis of potential competitors is limited to the catalogue proposed by STMicroelectronics, a European company that designs and manufactures semiconductors, including microcontrollers. The specific series selected is the STM32L4. The models in this series match the characteristics of the Apollo3, mainly in terms of processing capacity and power consumption. They are 32-bit microcontrollers with a Cortex-M4 core, based on ST's proprietary ultra-low-leakage technology specifically designed for building energy-efficient embedded systems. The different models differ in their flash and RAM memory capacities, as well as in the number of input/output (IO) connections they offer.

2.1.5 Peripherals stacked on Nucleo board

As for the Apollo3, there also exists development boards with STM32 MCUs, called Nucleo boards. They offer the same functionalities as the Apollo3 EVB : essential components for the MCU, a PMU with the USB power source, an on-board debugger, a user interface and standard connectors.

Several Nucleo boards exist that are equipped with a STM32L4 series MCU. All are roughly the same price and only differ with the specific MCU they embed. The Nucleo board model chosen is the Nucleo L452.

In this solution, the Nucleo board is used in the same way as the Apollo3 EVB. A custom PCB with the sensor and the radio modules is stacked on the connectors.

2.1.6 Single custom PCB with STM32WL

ST also proposes an interesting series of wireless microcontrollers that embed an additional radio module in the MCU chip. The STM32WL series provides a Cortex-M4 core with a sub-gigahertz radio supporting different modulation techniques, including LoRa. The radio is based on the Semtech SX126x and is consequently very similar to the SX1276 chip.

There exists two flavours in that series : the STM32WL5x that embeds a main Cortex-M4 plus a secondary Cortex-M0+ used to control the radio, and the STM32WLEx that only includes a single Cortex-M4 core. The latter is chosen to simplify the programming, because some difficulties are expected, particularly to manage the wake-up from sleep modes with the dual-core topology.

With this MCU, the external radio module is not needed, bringing several advantages for the design complexity, power consumption and PCB area. Moreover, ST already provides the radio drives in its intregrated development environment. As for previous solutions, the idea here is to have a single PCB with the MCU and the sensor blocks on it, and an external debugger.

2.2 Comparison

The results of the exhaustive comparison are shown in Table 2.1.

First, the estimated total price of each solution is computed, based on the sum of the main items of the bill of materials (BOM). The source of the prices for each component appears in Appendix A.

Then the current consumption corresponding to each of the tasks that the system can perform is reported from the datasheet of the corresponding component, for a voltage of 3.3V. The tasks are the processing and sleep consumption of the MCU, the radio consumption when in sleep mode, receiver (RX) mode and transmitter (TX) mode, and the sensor consumption when sleeping or acquiring data with a basic configuration [11]. Note that for the radio TX mode, the consumption is compared for a reference emission level of 20dBm because it is a standard level indicated in the datasheets. Even if the system cannot emit at such a high power in Europe (see Section 1.3.1), it can be assumed that this comparison holds at lower transmitted power.

Finally, advantages and drawbacks about the robustness and the ease of development of the different possibilities are noted.

Price The total prices are very similar, except for the two solutions that are based on the Apollo3 EVB. This is because that evaluation board alone costs more than a hundred euros, i.e. around two third of the total price. The reason behind this large difference between the chip's price, only a few euros, and its evaluation board is that the chip is manufactured and sold in large number, which is not the case of the EVB.

The *Others* category regroups all the small auxiliary components such as decoupling capacitors and connectors.

The cheapest solution is the last one, the single custom PCB with the STM32WL, with a cost of around 75€. The merged MCU and radio in a single chip is around ten euros cheaper than two separate components.

Consumption The Apollo3 is optimised for ultra low consumption in active mode. Its 6µA/MHz current outperforms by far the one of the STM32 MCUs, with 36 and even 72µA/MHz for the STM32WL. However, the sleep consumption is better for the latter, by a factor around three. Because the system is sleeping most of the time, the STM32 MCUs are less energy-consuming in average.

The radio module is the SX1276 in all the cases except the last one where it is included in the STM32WL. This last case consumes less in sleep, TX and RX mode because the energy efficiency of the MCU chip is higher. In sleep mode, the value for the MCU and the radio for the last case is the same because it is the same chip. To compute the total consumption for a certain use case, they should thus not be summed.

The sensor's consumption is the same for all cases because they all use the BME680. For the first case, because the EVB of the sensor is used, some additional power is dissipated by its LDO and level shifters.

In total for a use case where the system is awake for a short fraction of the time, senses and uses the radio regularly, the most advatageous case is again the last one, because it combines a decrease of the MCU consumption in average, plus a more efficient radio in all three modes of operations.

Robustness This qualitative criterion reflects the solidity of the assembly proposed. From the poor robustness of the initial situation, two cases are highlighted : either a medium one by stacking two different boards, or a strong and compact solution by having only one module. The three solutions that propose a single custom PCB are equivalently winners on this point.

Ease of development Several positive or negative remarks about the development of the proposed solution are explained here.

First, it must be noted that the single PCB with the Apollo3 MCU must be eliminated because it is not technically feasible. Indeed, the package of the MCU chip is of the ball grid array (BGA) type, as shown in Figure 2.2. The pins of the chip are located under it and they must be electrically isolated and connected to separate pads. The available electronics workshop is not equipped with the required tools to properly solder that kind of packages.

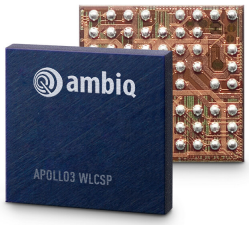


Figure 2.2: BGA package of the Apollo3, from [24].

Secondly, there is only little documentation available about the Apollo3 EVB. A crude reference schematic of the board can be downloaded from Ambiq's website, but does not come with any other explanation. It is thus really hard to understand the purposes and capabilities of the different components present on the evaluation board without an in-depth study. In contrast, ST offers lots of manuals and examples to use their products, which makes the job of utilising or designing with their product quite easy and safe.

But choosing a new MCU comes with a cost in terms of development time. Most of the software that runs on the MCU must be rewritten as it depends on its specific hardware architecture. The computational parts, e.g. the FWI algorithm, can use the same code integrated and compiled in the new project, but the platform-specific parts, e.g. the initialisation of the memories and clocks or the communication through SPI and I^2C, must be entirely translated. Fortunately, ST comes with a development suite that is user-friendly and makes prototyping fast by proposing a lot of reference resources.

Final choice The choice is for the last proposed solution, a single custom PCB with the STM32WL MCU. It solves the robustness and consumption problems of the initial module while being the cheapest solution. It however requires a complete design and the translation of the codebase.

	EVBs on a breadboard		**Single custom PCB with Apollo3**		**Peripherals stacked on Apollo3 EVB**		**Single custom PCB with STM32L4**		**Peripherals stacked on Nucleo board**		**Single custom PCB with STM32WL**	
BOM [€]	Apollo3 EVB	108.07	Apollo3	3.53	Apollo3 EVB	108.07	STM32L4	10.54	Nucleo L452	13.72	STM32WLE5	10.87
	BME680 EVB	22.21	PCB 10x5cm	25	PCB 10x5cm	25	PCB 10x5cm	25	PCB 10x5cm	25	PCB 10x5cm	25
	RFM95	9.95	SX1276	9.31	SX1276	9.31	SX1276	9.31	SX1276	9.31	BME680	13.96
	Jumper wires	3	BME680	13.96	BME680	13.96	BME680	13.96	BME680	13.96	Others	25
	Breadboard	8.3	Others	25	Others	25	Others	25	Others	25		
Total price [€]		**151.53**		**76.8**		**181.34**		**83.81**		**86.99**		**74.83**
Consumption												
MCU [µA/MHz]		6		6		6		36		36		72
MCU sleep [µA]		1		1		1		0.375		0.375		0.36
Radio sleep [µA]		0.2		0.2		0.2		0.2		0.2		0.36
Radio TX [mA]		120		120		120		120		120		87
Radio RX [mA]		10.8		10.8		10.8		10.8		10.8		4.82
Sensor [mA]		12		12		12		12		12		12
Sensor sleep [µA]		0.15		0.15		0.15		0.15		0.15		0.15
Robustness of the assembly		Poor		Strong		Medium		Strong		Medium		Strong
Ease of development			⊖ MCU package not solderable		⊖ Bad documentation from Ambiq ⊕ Existing codebase		⊕ Good documentation from ST ⊖ Need for codebase translation		⊕ Good documentation from ST ⊖ Need for codebase translation		⊕ Good documentation from ST ⊖ Need for codebase translation	

Table 2.1: Comparison of the six considered solutions.

Chapter 3

Design of the prototype

The prototyping of the chosen solution is divided in four steps. The first one is the schematic, whose role is to provide a netlist linking all the required elements. Then, once a specific component and its footprint are assigned to each element of the circuit, the layout step indicates their physical disposition on the PCB and the connections between them. The software suite used in this project for those two design steps is the free and open-source software KiCAD. After the PCB is manufactured, the last two steps are the assembly of the modules and the test phase, during which a software is developed to verify that all the hardware components can be properly used.

In Figure 3.1 is presented a high-level block schematic of the prototype. The central block is the MCU with the radio included, together with its auxiliary components, e.g. decoupling capacitors, crystal oscillators. The sensor is another block and it communicates with the MCU. The RF output of the radio module passes through an RF network before going out to the antenna. Other pins of the MCU are linked to connectors for debugging, probing or external usage. Finally, the PMU is responsible for providing a stable power supply to the system from the batteries.

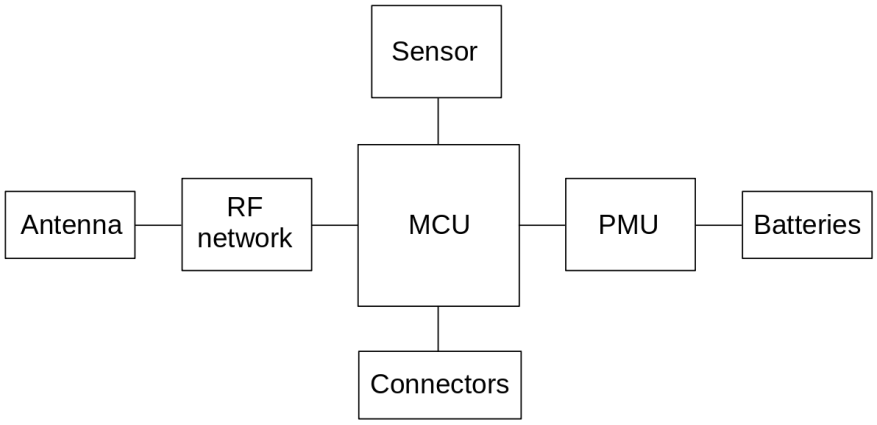


Figure 3.1: High-level block schematic of the prototype.

3.1 Schematic

The schematic of the different blocks are successively presented hereunder and Appendix B contains the complete one. Appendix C also presents the complete BOM for the design.

3.1.1 Power management unit

The role of the PMU is to provide a voltage source to the system that is stable with respect to the state of charge of the batteries, to the current drawn by the system and to other variations like the temperature. This stability is assured by an LDO, which converts the voltage of the batteries to a fixed, lower, predefined voltage. The chosen operating voltage for the system is 1.8V. All the different active components, i.e. the MCU, the sensor and the RF switch (see Section 3.1.6), can

operate either at 1.8V or 3.3V. For power consumption reasons, the lower one is chosen. Moreover, it requires only two batteries instead of three.

The LDO is the TPS7A03 [25]. It features the required stability, for an output current of maximum 200mA, and an ultra-low quiescent current of 200nA, which is particularly important for this application as the system is mostly sleeping. It is smaller than the sleep current of the MCU, 360nA in standby mode or 1.07µA in stop mode [26]. The LDO can be enabled or disabled by driving the pin EN high or low respectively, but in this design it is hardwired to VDD and is thus always on.

The battery pack is composed of two AAA batteries connected in series to reach a nominal voltage of 3V and it is connected to J3. J8 allows to connect another power source, e.g. a DC power supply for measurement experimentations. The choice of the power source is done by the mechanical slider switch S1.

The datasheet of the LDO specifies that 1-µF capacitors are needed at its input and output to stabilise the voltages.

J4 are pin headers that provide a way of supplying the system bypassing the LDO, for experimental purposes for example. In normal operation, the output of the LDO is connected to VDD by a jumper.

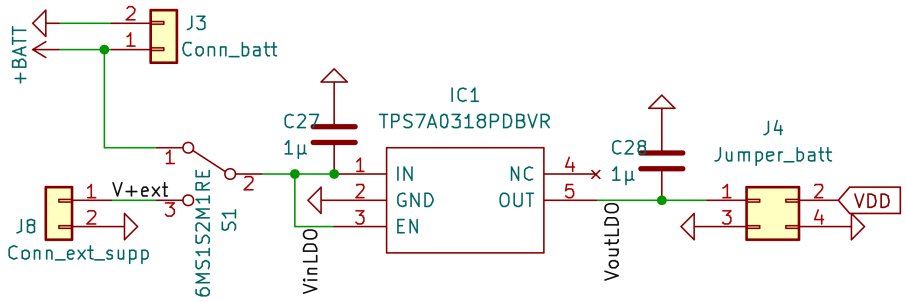


Figure 3.2: Power management unit.

3.1.2 Microcontroller unit

The microcontroller has many pins that can be grouped in several categories depending on their functionnality.

General Purpose Input/Output (GPIO) The GPIO pins, highlighted in blue in Figure 3.3, can be configured to provide many different functions, such as SPI or I^2C communication busses, ADC, digital-to-analog converter (DAC), external interruption triggers or simple configurable digital output pin that can supply or sink up to 20mA of current [26].

Some are used to interact with different blocks of the system. Their role is explained in the section related to that specific block.

Power pins The pins circled in red are either ground pins or supply pins. While the ground is the same everywhere, different supplies exist for different voltage levels or to separate the noise from one part to another.

VDDA is the analog VDD and it is separated from VDD by a filter to eliminate the switching noise, i.e. current spikes at each digital clock cycle, and thus have a clean analog voltage reference.

Four pins are used to control a switched-mode power supply (SMPS), which is basically a buck converter. Unlike an LDO that reduces the voltage by dissipating the power associated with that voltage drop, i.e. the current is the same at the input and the output, an SMPS converts voltage with a high power efficiency. Its output voltage is used as the reference voltage of the radio module inside the chip through an external connection.

The last power pin is VBAT. Its main operation mode is when it is connected to a security battery, for example a coin cell, such that some important features can run even in absence of voltage at VDD, e.g. the real-time clock (RTC) and some backup registers. It can also be directly connected to the main battery positive voltage to give a means of monitoring it and its state of

charge, by feeding it to an ADC after a division by a factor three [26]. This design doesn't use any of these possibilities and VBAT is simply connected to VDD.

RF pins The green squares highlight the pins related to the radio module. Those are the inputs and output, as well as the ones that deliver the RF power. It should be noted that the high-power output is not used for the reasons explained in Section 1.3.1.

Others The other pins are special ones that fulfil a specific role. Four of them are connected to crystal oscillators that oscillate at a reference frequency, one at 32.768kHz for the digital clocks and the RTC, the other at 32MHz for the phase-locked loop (PLL) control system of the radio module. The TCXO pin can be connected to a Temperature Compensated X (Crystal) Oscillator to provide an even more stable reference clock.

BOOT0 is a pin that indicates the bootloader of the MCU where to find the instructions to execute after a reset. If it is tied high, it can initialise some peripheral communication links like SPI to be programmed. In the opposite case, it assumes the flash memory contains the program and starts running it. Because this design uses an external Serial Wire Debug (SWD) debugger, it should always be connected to ground through a 10-kΩ resistor by a jumper.

The last pin is NRST. It is the active-low signal used to reset the MCU to program it or restart the execution of the program.

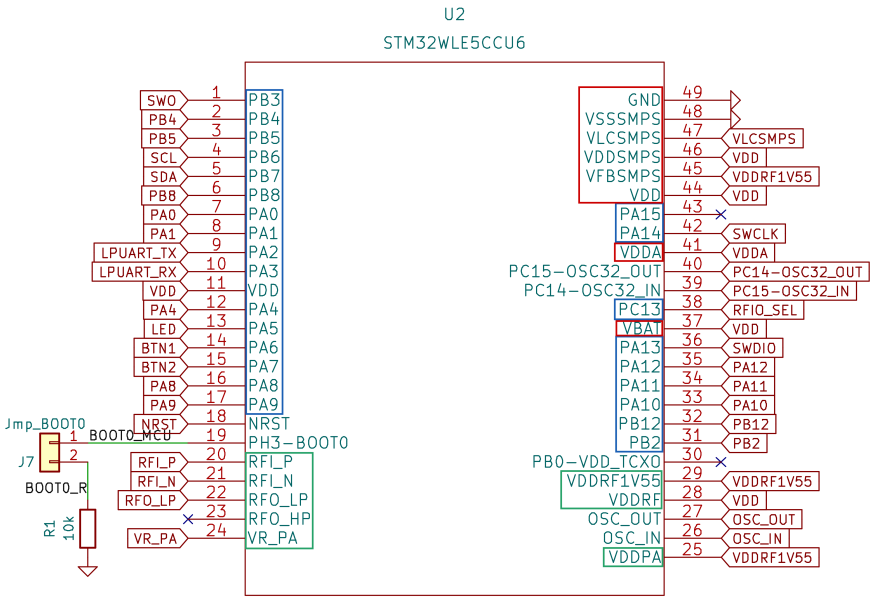


Figure 3.3: Microcontroller unit.

The MCU chip needs several external passive components for its proper operation, as displayed in Figure 3.4.

Decoupling capacitors filter the power supply noise and provide an energy reserve to cope with fast-changing AC currents consumed by the MCU. The rule of thumb is to place one 100-nF capacitor near each VDD pin, plus one big 4.7-µF per supply group. The big one stores more energy, while the smaller have a higher quality factor to deliver quickly the needed current, in order for the voltage to stay stable. The radio module and the SMPS use additionnal capacitors for their operation.

VDDA and VDD are separated by the ferrite bead L1, that acts like an inductor specifically designed to filter switching noise.

The two crystal oscillators are also displayed, with two capacitors needed for the low-frequency one.

All those components and their value are selected following a reference schematic from ST for the STM32WL MCU [27].

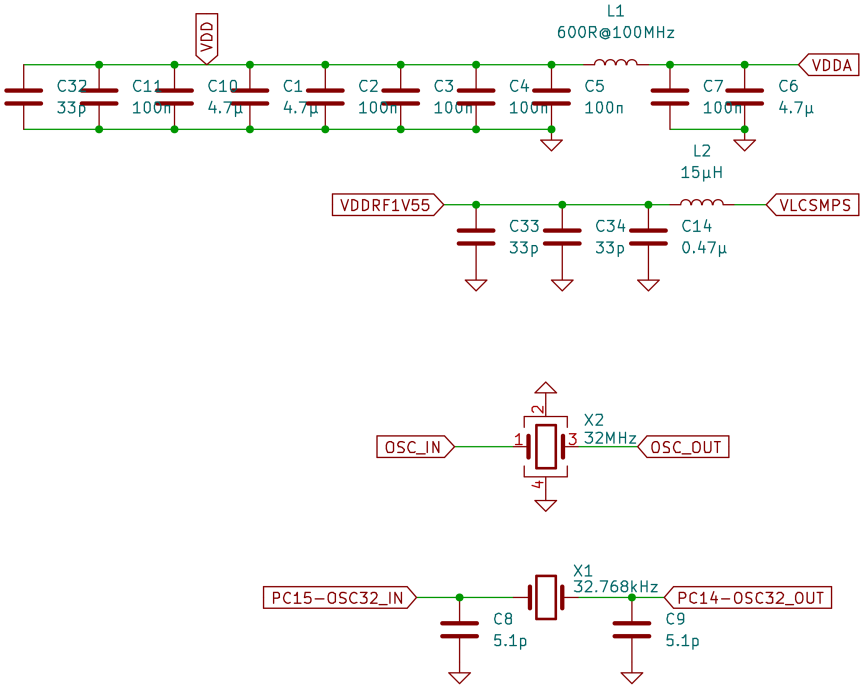


Figure 3.4: Decoupling capacitors and crystal oscillators.

3.1.3 Debugger

The external debugger is the STLINK-V3MINIE [28]. It is a tiny debugger from ST compatible with all STM32 MCUs. As shown in Figure 3.5, it makes the bridge between the host computer and the target application by communicating as a serial USB device with the computer, and through a STDC14 cable with the MCU.

The STDC14 connector has 14 pins and is able to program and debug the microcontroller through SWD, but also redirect an Universal Asynchronous Receiver Transmitter (UART) communication bus to the host. The MCU can thus print informations on the host's console. It uses GNDDetect to sense the presence of a target board, VDD to adapt the digital voltage of the communication busses and NRST to reset and program the MCU.

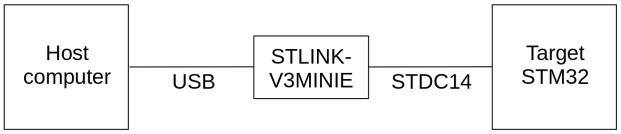


Figure 3.5: Debugging with the STLINK-V3MINIE.

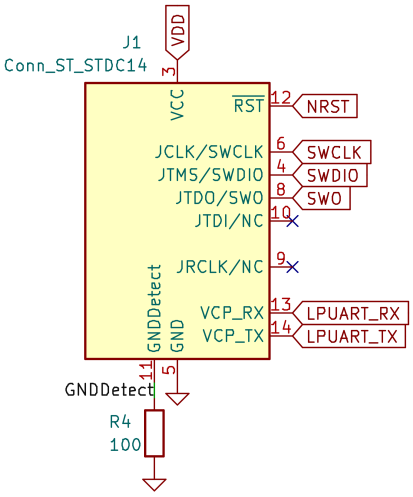


Figure 3.6: Debugger connector.

3.1.4 User interface

The user interface (UI) provides a way of interacting with the software at runtime.

Two user buttons trigger external interruptions on falling-edge when they are pressed. The voltage is normally tied high by an internal pull-up resistor in the MCU, and brought to zero by pressing the button. A diode is present for electrostatic discharge (ESD) protection because users touch the device at that location. The RC filter prevents glitches to happen, that could trigger multiple interruptions at a single button press.

The NRST button is wired to the reset line to provide a hardware reset feature.

Finally, an LED is connected to a GPIO pin through a 330-Ω current-limiting resistor. The LED nominal ratings are 1.6V and 5mA. With that resistor value, the current is 5mA at 3.3V and 0.6mA at 1.8V, which stays below the capacity of the GPIO pin of 20mA. At 1.8V, the current is way below the nominal one but it is enough to make the LED shine.

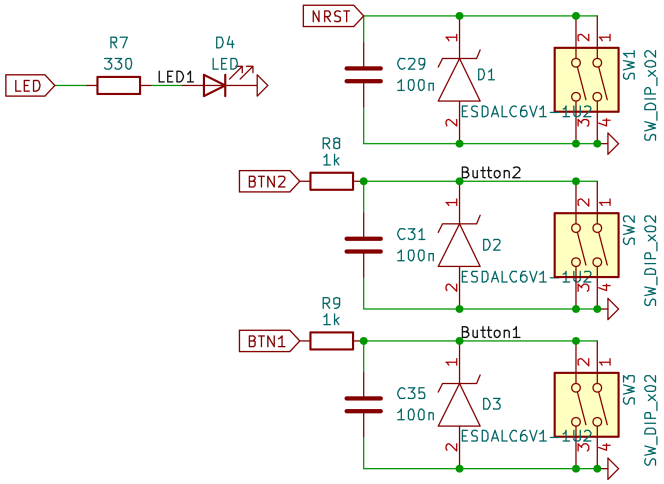


Figure 3.7: User interface.

3.1.5 Sensor

The BME680 can communicate either by an I²C bus, the one used in this design, or a 3- or 4-wire SPI bus. Pins 2 to 5 are connected differently for each case. For I²C, CSB is tied high, SDO low and pins 3 and 4 are the bus. They both need a pull-up resistor to operate properly.

The value of the components, i.e. pull-ups and decoupling capacitors, are suggested in the reference datasheet of the sensor [11].

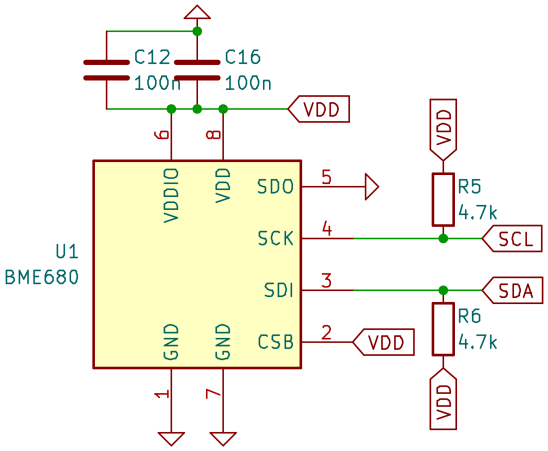


Figure 3.8: Sensor.

3.1.6 Radio frequency network

The RF network links the RF input and output of the MCU to the antenna and provides different functions to the RF signal. This schematic is reproduced from the reference from ST [27], with the value specifically chosen to be optimised for a 868-MHz carrier frequency at a maximum output power of 14dBm. No in-depth analytical study of the network's behaviour is conducted, but the role of each block is identified.

In the middle of the network, U3 is an RF switch used to connect the antenna path either to the output or the input path of the MCU. The supply line through a resistor and with capacitors is indicated in the reference schematic, as is the digital control line.

The RF input path passes through a balun, from BALanced-UNbalanced, whose role is to link the asymmetrical line to the symmetrical RF input composed of a positive and a negative RFI pin.

RFO_LP outputs the RF signal while VR_PA provides its power. It goes through a band-pass filter to remove noise outside the band of interest.

Finally, another filter is present before reaching the SubMiniature version A (SMA) connector J2 where the antenna is plugged.

Several components have a null value. Those are placeholders not necessary for the frequency and power used, but they can accomodate a useful component in case one wants to change these parameters. Concretely, 0-H inductors are short-circuited and 0-F capacitors are left open.

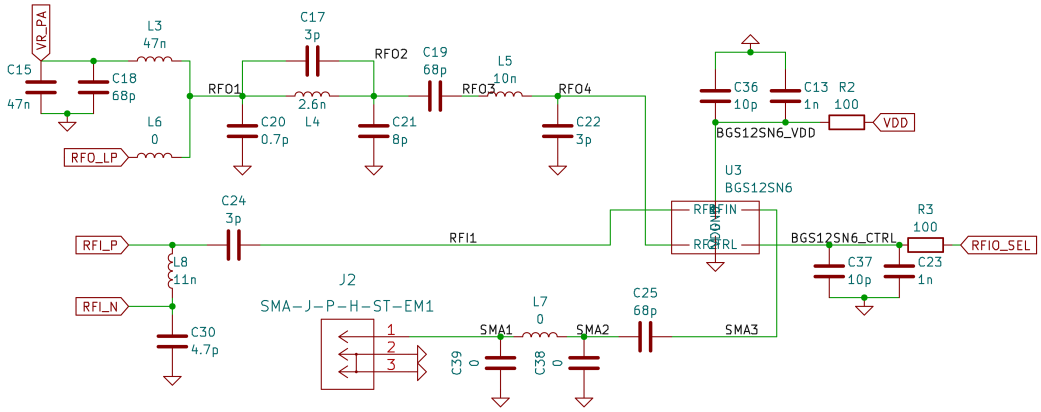


Figure 3.9: Radio frequency network.

3.1.7 Connectors and probes

The current application only uses a fraction of the available GPIO pins. The unused ones are wired to two 8-pin connectors, together with VDD and two ground pins, such that the design is modular and can easily be extended with new functionnalities. An example of such a extension is presented later in this work. It uses several GPIO pins to provide an additionnal SPI bus, an ADC and control GPIOs.

The probes are male pin headers that can be used for debugging the digital communication busses. In case one or the other fails, the waveforms of the messy signal can be retrieved there for investigation.

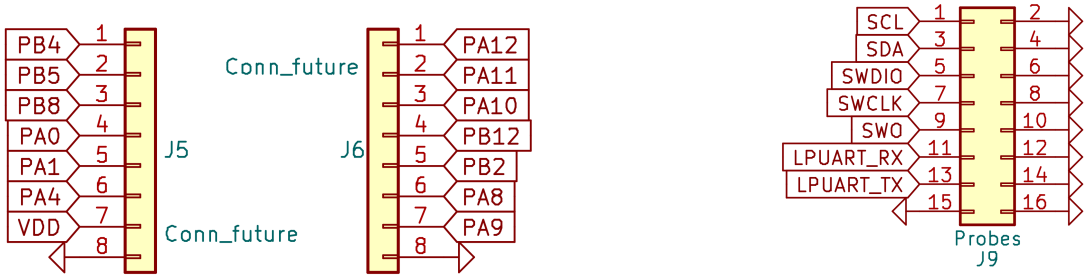


Figure 3.10: Connectors and probes.

3.2 Layout

During the layout phase, the footprints of the components are placed next to each other on the 2D plane that will become the PCB. Then copper traces are drawn to connect the different nets together. The geometry is important because the physical properties of the copper traces have an impact on the electrical quantities they transport.

The general disposition of the different functional blocks is displayed in Figure 3.11. The antenna is placed at the edge of the device to ease a possible future integration in a casing. It is also in the same plane as the PCB because in the application, the system must be attached to a tree vertically, with the antenna pointing upwards. The sensor is also placed near the antenna at the edge, so that it can easily be exposed outside the casing.

The connectors are each placed on one side of the PCB, to provide a good mechanical robustness when an extension board is stacked up. Three fixation holes are drilled through the module to provide a way of fixing it to a tree or elsewhere.

Several layout considerations and choices are explained in the following subsections.

A complete layout view can be found in Appendix D.

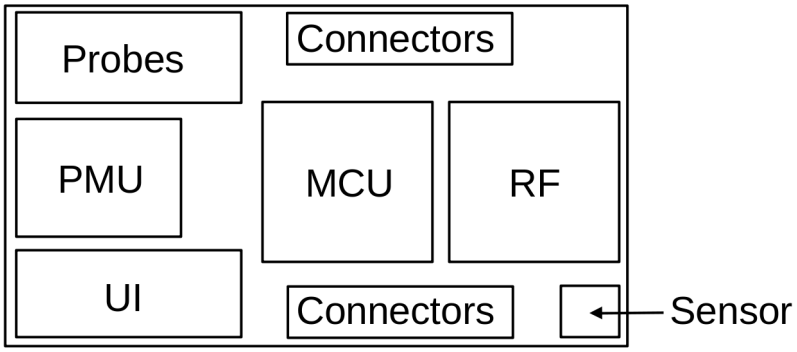


Figure 3.11: Disposition of the functional blocks on the PCB.

3.2.1 Physical stackup

The PCB is a stackup of several layers of different materials as presented in Figure 3.12. The thickest layer is the core material or substrate that assures the mechanical robustness of the PCB. On each of its side is printed a layer of copper protected by a solder mask, that covers everything except the pads where the components must be soldered. A legend is printed on the top side to indicate any kind of information.

The thickness of each layer is the standard one proposed by EuroCircuit, so that the manufacturing is as fast and as cheap as possible. For such a simple design with neither much components nor important current, two thin layers of copper are enough.

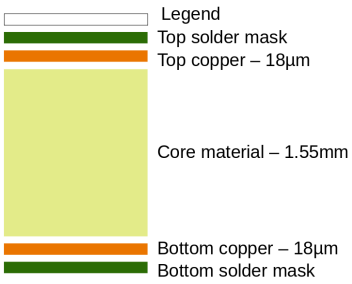


Figure 3.12: Cross-section of the physical stackup.

Bottom copper The bottom copper layer is mostly filled by a ground plane. It is useful to provide a low-impedance return path for all the currents that are consumed by the system. Moreover, it harmonises the capacitance between each of the top copper traces to have as little mismatch effects as possible from the layout geometry.

The ground plane is interrupted here and there to allow signal tracks to cross. The electrical connections between the bottom and top copper layers are done by means of vias, i.e. plated holes that are conductive.

Top copper The top copper layer is where the pads of the components and the interconnection tracks are drawn.

3.2.2 Footprints

Most of the components used are surface-mounted devices (SMD), which means that they are soldered on pads on one side of the PCB. In this design, all the SMD components are placed on the top side.

Others, like the connectors, are through-hole components. They go through the PCB and are soldered in a plated hole.

Many dipole SMD components have standard footprints, like capacitors, inductors, resistors and diodes. Small-size components are chosen to obtain a compact layout, but the footprints are a little larger than normal to ease the soldering process. It also allows to easily change a component in case one wants to update some part of the system, because the pads can be reached by a small soldering iron tip. In Figure 3.13 are displayed top views of the normal-sized footprint and the used one, the larger, with the component placed on it. The grey zone is the pad. The standard size of these components is 0402 (1mm $\times$ 0.5mm), except the ferrite bead and the few large capacitors and inductors that are bigger.

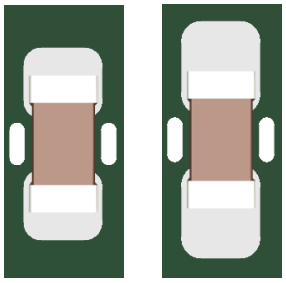


Figure 3.13: Normal and larger dipole footprints.

3.2.3 Trace widths

Three different trace widths are used in the design, depending on the type of signal they carry.

For regular nets that are likely to carry a reasonable amount of current, e.g. GPIOs, a medium trace width of 0.25mm is used.

Thinner tracks of 0.15mm are dedicated to digital busses that must switch fast from high to low. The small total area of the trace means that the total capacitance is low, and thus that the switching power consumption is also low.

Finally, 1-mm traces are drawn for supply lines to reduce their resistance, because a fair amount of current must flow in those.

3.2.4 Decoupling capacitors

The decoupling capacitors must be placed as close as possible to their respective chip. Indeed, each copper trace exhibits a certain impedance that results in voltage drop when current is flowing. To provide a fast energy source with a high quality factor, the trace length between the capacitor and the chip must be minimised.

3.2.5 RF network

Matched impedance

When a high-frequency signal goes through a PCB trace, the non-idealities of the copper tracks cannot be ignored because they can be very destructive. Indeed, the trace must be considered as a transmission line. The main parameter to control is the characteristic impedance of the line, defined as $Z_0 = \sqrt{\frac{L}{C}}$, with L the inductance and C the capacitance per unit length of the line. Z_0 must be the same for all the different blocks, i.e. the RF path of the MCU, the RF network, the SMA connector and the antenna. All have $Z_0 = 50\Omega$, thus the RF network must match this value.

Different techniques exist to control the characteristic impedance of a PCB line. The two common ones are the use of a microstrip line or a coplanar wave guide (Figure 3.14). KiCAD provides a calculator tool to compute Z_0 as a function of the physical dimensions and the substrate parameters, i.e. the dielectric constant, the dissipation factor and the resistivity, or the other way around.

Microstrip line Because the substrate is thick, the capacitance between the top and bottom layer is small. To reach $Z_0 = 50\Omega$, the required width for the signal copper trace is around 3mm, which is not feasible in practice because the components are too small.

What is usually done to cope with this situation is to use a 4-layer PCB. In that case, two layers of copper are separated by a thin substrate at each side of the main substrate. It allows to

(a) Microstrip line.

(b) Coplanar wave guide.

Figure 3.14: Two common techniques to control a line impedance.

increase the inter-layer capacitance and have a smaller required trace width. However, the price of a 4-layer PCB is roughly two times the price of a regular 2-layer one.

Coplanar wave guide In this case, the capacitance depends on the space between the signal trace and the coplanar ground planes. For a standard price, EuroCircuit's constraint on the minimum clearance between traces is 0.15mm. For $Z_0 = 50\Omega$, this gives a trace width of 0.75mm which is totally acceptable. However, coplanar wave guide is well suitable for long traces with uninterrupted ground planes. In this case, it is not possible because there are a lot of components in the way of the line.

In reality, such techniques may not be necessary in this design. Indeed, an unmatched characteristic impedance only plays a destructive role when the length of the transmission line is at least a significant fraction of the wavelength of the signal it carries. At 868MHz, the wavelength is about 35cm. The length of the line is the distance between the MCU and the SMA connector, which is only about 2.5cm, not even a tenth of the wavelength.

For that reason, none of the explained techniques is implemented. Instead, the width of the RF trace is 0.6mm, to match the width of the pads of the components that compose the network.

RF shield

The RF network is circled by a large ground trace drawn on the top copper layer and regularly connected to the bottom plane by vias. The role of this RF shield is to electromagnetically isolate the RF network from the rest of the system, to prevent it to be polluted by capacitive or inductive coupling with, for example, the switching noise of the digital blocks.

3.3 Assembly

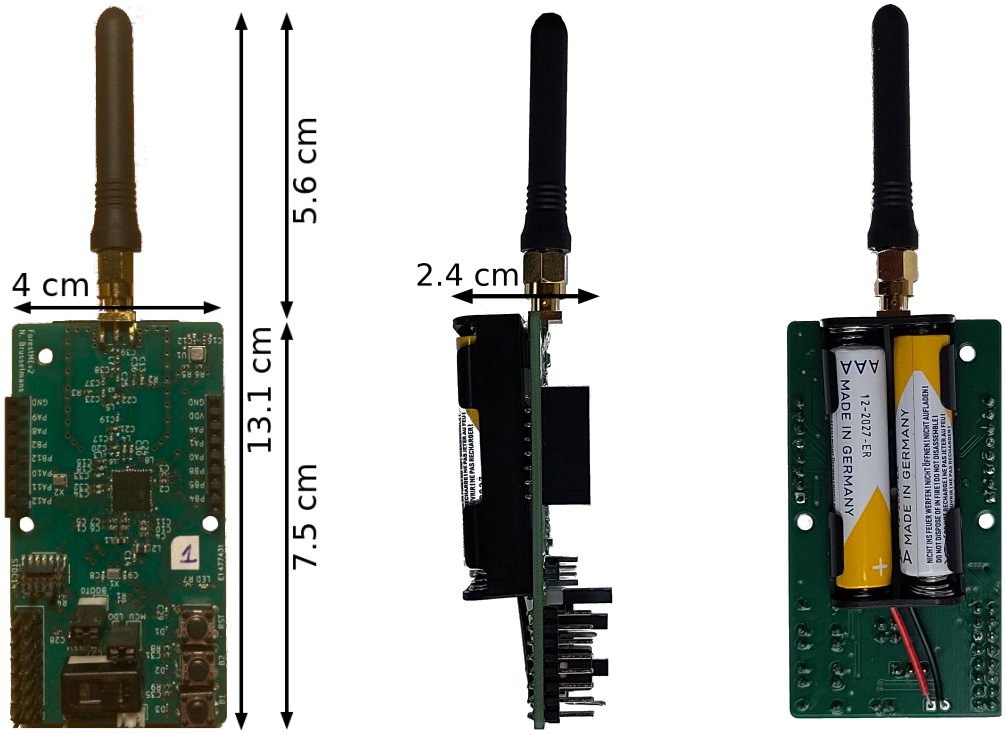
Once the PCBs are manufactured, the assembly consists of soldering each of the components at its place. First, soldering paste is applied on all the SMD pads and the SMD components are placed by hand at the right location, with the help of a microscope. Then, the paste is melted in an oven following a precise temperature curve. Finally, the through-hole components are soldered by hand with an iron.

This project requires six modules to conduct different experimentations (see Chapter 4). All six are fully functional, after being assembled this way.

Figure 3.15 presents the top, bottom and side views of the assembled modules, numbered from 1 to 6 and named ForestMEv2 after the so-called first prototype presented in Chapter 1.

The total dimensions of the module are $7.5 \times 4 \times 2.4$ cm, plus the antenna with a length of 5.6cm for a total length of 13.1cm.

The battery pack is glued with adhesive to the back of the PCB and wired to the connectors with flexible cables.



(a) Top view. (b) Side view. (c) Bottom view.

Figure 3.15: Views of one of the assembled modules.

3.4 Test software

The objective of the test software is to build all the necessary code to be able to access and use through software the different hardware functionalities. No concrete application is built, only the useful hardware abstraction function prototypes.

To program the STM32 MCU, ST provides an all-in-one development environment named STM32CubeIDE. It includes all the required tools to program the target microcontroller.

Initialisation code generator STM32CubeMX is one of the tools of STM32CubeIDE. It provides a graphical interface for the configuration of the MCU. A first panel allows to build the clock tree from the different available clock sources. On the second one, each of the MCU pins can be configured to manage the desired function, e.g. a GPIO or an I^2C bus. All the fixed configuration parameters are accessible from this panel.

In addition to this graphical interface, the tools embeds a code generator that autonomously builds the initialisation code for all the configured elements. It also includes a set of hardware abstraction layers (HALs), that hides the complexity of specific register operations and provides simple function prototypes to use hardware components.

Compiler The environment embeds the compiler needed to build the binary program for the MCU. The toolchain used is a flavour of GCC for STM32 MCUs.

Programmer and debugger The last tools used in this work are the programmer and debugger. They are directly linked to the STLINK debugger to provide a way of flashing the target directly from the development environment.

The following sections describe the software architecture and parameters for each of the different hardware components.

3.4.1 Clocks

Three clock sources are used for different purposes, each with its advantages and drawbacks.

Low-frequency crystal oscillator The 32.768-kHz external crystal oscillator generates the clock for the RTC. That specific frequency is historically chosen to be easily divided to 1Hz. Indeed, because $2^{15} = 32768$, a simple 15-bit counter overflows every second.

The generation of that clock consumes a low power because it is at a low frequency, and the use of a crystal generates an accurate and stable clock with very few jitter. This is mandatory for the use of an RTC, because a small frequency deviation or period jitter accumulates over time and this would result in an offset in the actual time measured by the RTC.

High-frequency crystal oscillator However, the use of a low-frequency oscillator is not suited for the high-frequency clocks required for the digital operation or the generation of the RF carrier. It would need a big multiplication factor that would be a problem for power consumption, stability and jitter performances.

Instead, the other crystal oscillator is used to generate a 32-MHz clock reference for the radio module. It consumes more power due to its increased frequency, but it can be disabled when the radio module is not used.

High-frequency internal oscillator The third clock source is an internal RC oscillator, whose frequency can be configured between 1 and 48MHz. This type of oscillator generates a high-frequency clock at low power, but is neither accurate nor jitter-free. It is used to clock the main digital operations of the MCU that don't require any of these two properties, as well as all the internal and external communication busses.

3.4.2 User interface

As shown in the design schematic, three GPIO pins are used for the user interface.

PA5 is configured in output mode, initially tied low. It only requires to bring it high to light on the LED connected to it, by using the provided HAL function.

PA6 and PA7 are wired to the two user buttons and configured as external interruption lines, with their internal pull-up resistor enabled as explained in Section 3.1.4. For the interruptions to work as expected, they must be enabled in the Nested Vector Interrupt Control (NVIC). The role of this block is to listen continuously to the interruption lines, so that it can interrupt the Central Processing Unit (CPU) or wake it up if it is sleeping in case of a button press or another monitored event.

3.4.3 Sensor

The way of interacting with the sensor can be represented as a stackup of software and device layers, as presented in Figure 3.16. The description of the different layers in the following sections follows a bottom-up approach.

BME680 sensor The lowest layer is the sensor device that needs to be accessed by software for its configuration and for data retrieval. It is physically linked to the MCU by the I²C bus.

Hardware Abstraction Layer HAL functions exist to manage the I²C bus. They provide simple interfaces for reading and writing from and to registers of the BME680. They are automatically generated by STM32CubeMX when configuring the I²C bus.

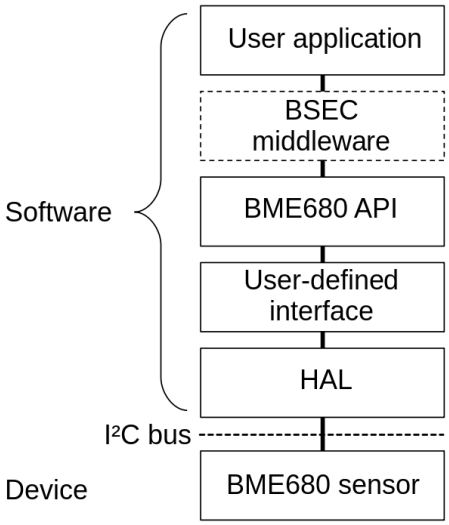


Figure 3.16: Sensor software stack.

User-defined interface Because the BME680 provides several modes of communication (see Section 3.1.5) and the chosen type of bus can be wired to more than one pin location on the MCU, a specific interface must be written depending on the actual hardware.

Concretely, the read and write accesses from the upper layer are redirected to the specific HAL function responsible for making use of the right bus. In this case, the first I^2C bus is used and located on pins PB6 and PB7.

BME680 API Bosch provides an official Application Programming Interface (API) for its sensor. It is composed of a dozen functions to configure the mode of operation, set the different parameters for the four sensed quantities (see Section 1.1), retrieve the data and perform other utility tasks.

As does the HAL for the MCU, this API hides the complexity of the sensor and provides simple and easily understandable functions to manage the peripheral.

BSEC middleware On top of the API, Bosch also offers a middleware named Bosch Sensortec Environmental Cluster (BSEC). It is a high-level signal processing software for the BME680 that can be used to produce aggregated information from the acquired raw data. Its possible usages regroup many applications, such as air quality monitoring or weather forecasting.

Unlike the API whose source code is fully provided, the BSEC software is distributed as a compiled file, together with its interface header.

The software developed in this work only targets experimentations on the prototype that doesn't need such a high-level processing. It is thus not included in the software. However, a future real-world application could benefit from its integration to efficiently prevent and detect forest fires.

User application The user application is the highest layer on the stack. It is the control software that uses all the lower layers to build a specific application. It can either use the functions provided by BSEC, or in this case directly the API to interact with the sensor.

3.4.4 Radio

The software stack for the radio module presents the same type of architecture as the one for the sensor, as can be seen in Figure 3.17.

Radio module The lowest layer is the radio device. This time, there is no external communication bus because it is included in the MCU chip. Its registers are accessible through an internal SPI bus.

User-defined interface This interface makes the link between the operation mode configured in software and the custom hardware. Indeed, the radio driver must control the RF switch, an external component on the PCB, through the GPIO pin named RFIO_SEL (see Figure 3.9). Concretely, the user-defined interface is composed of an initialisation function of the GPIO pin and a control function that sets it high or low to switch between the RX and TX paths.

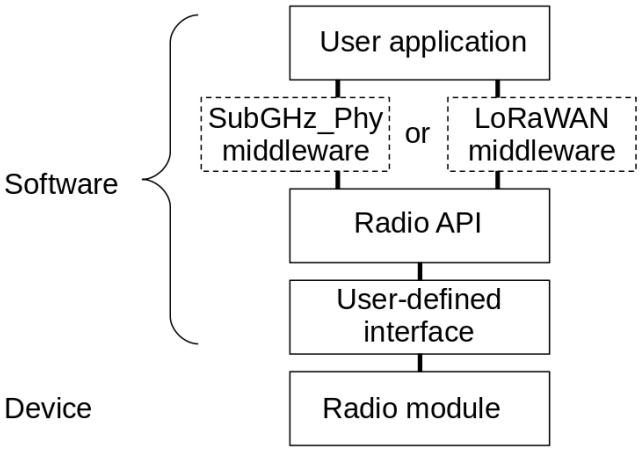


Figure 3.17: Radio software stack.

Radio API The API for the radio is provided by its designer, SemTech. It is already included and generated by STM32CubeMX when the radio is enabled and provides necessary low-level functions to manage the peripheral.

Middlewares Several middlewares are available to use the radio driver in different ways. Two of them are used and explained herebelow.

SubGHz_Phy The SubGHz_Phy middleware is the simplest one. The user has directly access to the physical layer to send and receive LoRa packets, with funtions like `Radio.Send(bytes)` and `Radio.Rx()`. The first configures the radio in TX mode then transmits a LoRa packet containing the `bytes` data, while the latter puts it in RX mode. When the radio finishes to send or receives a packet, it notifies the CPU through an interruption.

LoRaWAN This second middleware is much more complex as it implements the LoRaWAN protocol. The device acts as an end node on the network.

Compliance of the device with the protocol is demonstrated using an existing, freely accessible LoRaWAN network, named The Things Network (TTN) [29]. Many LoRaWAN gateways are deployed around the world, belonging either to TTN or to private users. Anyone can set up a gateway and add it the TTN network, provided that is it connected to the Internet. After registering to TTN's free services and encoding a private application, a message captured by any gateway that matches the private application key is redirected to TTN's console on their website.

The module works perfectly with that protocol, as several dummy packets have been successfully transmitted from and to the end device using one of the gateways located in Louvain-la-Neuve.

User application The application developed in this work targets two types of uses for the radio, regarding the experiments. The ones are tests concerning the performance and characteristics of the LoRa communication, and the others use the radio for data transmission. Both use the simple SubGHz_Phy middleware to have control over the physical layer for the first, and to transmit packets from one device to another for the latter. In that case, one module acts as a gateway to receive the packets from the others, acting as end nodes.

3.4.5 Default parameters

In the test application that is developed for the characterisations, the configuration parameters of the radio and the sensor are chosen to be typical values of a real application.

Sensor The effect of the different sensor's parameters are already described in Section 1.1. Table 3.1 shows their default value. The FIR filter is not used. The target temperature is in the middle of the possible range indicated in the datasheet. It also states that this temperature is usually reached after 20 to 30 milliseconds of heating, so its duration is set a little longer than that.

Tap number of FIR	Oversampling factor	Target temperature	Heating duration
0	8	300°C	50ms

Table 3.1: Default parameters of the sensor.

Radio By default, the radio emits packets at maximum power, i.e. 14dBm, and in SF7. The packet length is 16 bytes, structured in five fields that contain a 2-byte sender identification number (ID), together with four fields of the size of the sensor's data, i.e. temperature (T), relative humidity (H), pressure (P) and gas resistance (G), as presented in Table 3.2. The temperature is coded on two bytes while the other types of data are on four bytes.

Length	2	2	4	4	4
Field	ID	T	P	H	G

Table 3.2: The different fields inside a packet.

Chapter 4

Characterisation

Several experiments are conducted in order to characterise the module.

First, a premiliminary test is performed to assess the influence that the use of the radio module has on the sensor's behaviour.

The second characterisation aims to evaluate the power consumption of the module to perform different tasks.

Then, the sensor's ability to detect forest fires is assessed, by exposing it to burning fires.

Finally, the communication performances of the radio are evaluated, in a controlled laboratory environment as well as in a real forest environment.

4.1 Influence of the radio on the sensor

The BME680 sensor, in blue in Figure 4.1, is located right next to the RF path, highlighted in red. This proximity could be the cause of many non-idealities regarding the behaviour of the sensor. For example, the digital I^2C communication might be disrupted by capacitive or inductive coupling, that could potentially either prevent a successfull communication or flip some of the bits and thus corrupt the transmitted data. Another undesirable influence could be that the use of the radio disturbs the physical sensing operation, such that the measured quantities are biased.

The objective of this experiment is to detect such an influence, if any exists, and quantify its effect in order to be able to counteract it by a preventive calibration of the sensor or a post-processing on the acquired data.

4.1.1 Setup

While the sensor continuously acquires data, the mode of operation of the radio is changed to assess its influence for all the possible situations.

The sensor is configured with the default parameters mentionned in Section 3.4.5. Moreover, an acquisition of each of the four quantities is performed every second.

The total experiment lasts for 50 minutes. During the first 20 minutes, the radio stays idle to capture the baseline data acquired by the sensor. Then, it is set in TX mode for 10 minutes, during which dummy packets are sent continuously, i.e. whenever the transmission of a packet is finished, another is direcly sent. After that there are 10 more minutes with the radio in idle mode

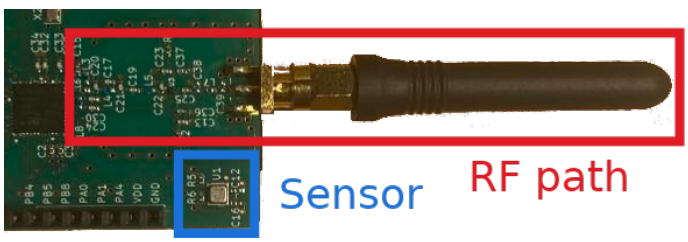


Figure 4.1: Proximity of the sensor to the RF path.

to recover from previous influences and restore the baseline. Finally, the radio is set in RX mode for the last 10 minutes.

The only radio parameters that plays a role is the output power level. It is configured to the maximum power of 14dBm to observe the result in the worst case. When sending packets continuously, the spreading factor and the size of the packets have no influence, because the RF signal that can disturb the sensor have the same properties in all cases.

CPU temperature In addition to the sensor's data, another temperature measurement is performed at the same rate in the MCU. Indeed, it is equipped with an internal temperature sensor linked to an ADC. Unlike the BME680 that directly outputs a physical quantity in °C, the output of the ADC is a 12-bit binary number.

The datasheet of the MCU [26] specifies two important parameters to compute the actual temperature based on the binary output, displayed in Table 4.1 : the ADC's output voltage at 30°C, V_{30}, and the average slope or sensitivity of the sensor, Avg_Slope.

Neither of the two is precise, as reflected by the possible range of values. However, because the sole purpose of that measurement is to be linked to the temperature of the sensor by its general behaviour, only the trend is required and not accurate values.

With these parameters, the CPU temperature T_{CPU} can be computed from the binary number b by Equation 4.1, with $N = 12$ the number of bits of the ADC and VDD = 1.8V the supply voltage.

V_{30}	0.76 ± 0.02 V
Avg_Slope	2.5 ± 0.2 mV/°C

Table 4.1: CPU temperature sensor parameters.

$$T_{CPU} = 30°C - \frac{V_{30} - b\frac{VDD}{2^N}}{\text{Avg\_Slope}} \tag{4.1}$$

4.1.2 Results

Figure 4.2 shows the four sensed quantities during the whole experiment. The periods during which the radio is active are highlighted, in orange for the TX mode and in green for the RX mode. The sensor's data is plotted in blue in all four graphs, while T_{CPU} is in orange.

The first identified possible non-idealities, i.e. disturbances on the I^2C bus, is not present, as all the transmission were successfully executed. There however exists an influence on the sensed quantities, as detailed herebelow.

Temperature When looking only at the temperature measurement of the sensor, a roughly linear increase can be observed in the TX zone, and another one less pronounced in the RX one. At this stage, two hypotheses can be put forward concerning the cause of this phenomenon.

The first one is that part of the RF power emitted from the antenna creates induced currents in the metallic structure of the sensor and heats it up. The heat progressively stored linearly increases the temperature, at least up to the point where the heat dissipation in the air equals the one generated by the currents. If this hypothesis is true, no such temperature evolution would be observed in the MCU, because it is located farther away from the antenna and packaged in a plastic enclosure.

The second hypothesis is that the heat originates from the MCU and is conductively transferred to the sensor through the core material and the copper traces of the PCB. As will be shown in Section 4.2, the MCU consumes power when using the radio in TX and RX modes. All of that power, minus the actually emitted power for the TX mode, is dissipated as heat in the MCU, thus its temperature increases.

The plot of the CPU temperature validates the second hypothesis. Indeed, a strong increase of its temperature can be observed when the radio is transmitting, and a smaller one in RX mode. The linear increase of the temperature of the sensor can be explained by the physical behaviour of the heat conduction, because the materials of the PCB have a certain thermal resistivity and time constant.

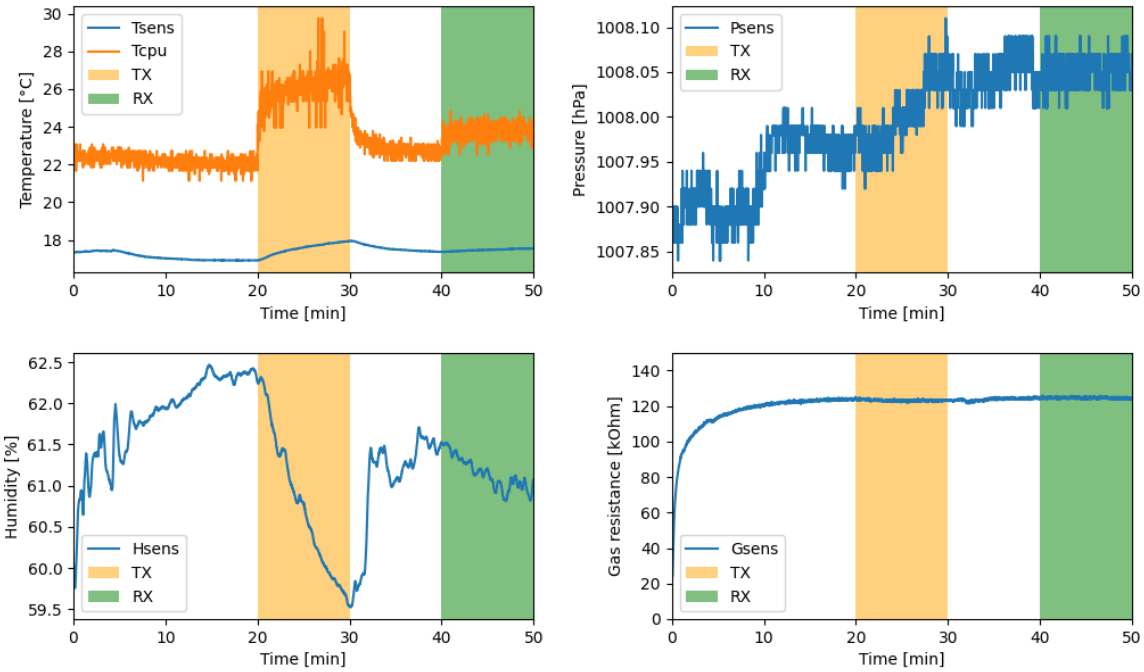


Figure 4.2: Influence of the radio on the sensed quantities.

Relative humidity The relative humidity is direcly linked to the temperature. Indeed, the hotter the air, the more humidity it can contain. For the same absolute value of humidity, the relative humidity is thus lower when the temperature increases.

That behaviour can clearly be observed in the graph, with a decrease of the relative humidity when the radio is used, following the linear behaviour of the temperature.

Pressure No correlation can be identified concerning the sensed pressure and the use of the radio.

Gas concentration Similarly, the gas concentration stays stable, no matter if the radio is used or not.

However, the gas measurement needs around 10 minutes to reach a stable state when starting its operation. It is because the target temperature of the heated plate of the gas sensor takes some time to be stable. The time taken to reach the stable operation, as well as the baseline resistance value are expected to strongly vary with the gas sensing parameters, i.e. the target temperature, the heating duration and the sampling rate.

4.1.3 Discussion

Because the use the radio has an influence on the temperature and the relative humidity measured by the sensor, caution must be exercised in a real application. No radio operation must be performed during a reasonable amount of time, e.g. 3 minutes, before acquiring data. Otherwise, wrong measurements could result in a computed FWI that does not reflect the reality.

To mitigate this constraint, it should be noted that the radio is not likely to be used as intensely as in this experiment, but rather for very short periods of time, i.e. seconds, once in a while. Moreover, in a real application, the order of operations is usually first the data acquisition and then the transmission.

4.2 Power consumption

The objective of this experiment is to characterise the power consumption of the device in every possible scenarios, to assess its total power budget. In the end, it allows to correctly size either the battery pack for a target life span or an energy harvesting unit to provide enough power to the system.

For each of the functional tasks that can be performed, e.g. send a packet or acquire data, the energy required to execute it is measured.

4.2.1 Setup

The experimental setup is composed of a power supply, the 2410-C SourceMeter from Keithley, wired to the device and put in series with a Digital MultiMeter (DMM), the DMM 6500, also from Keithley. The power supply outputs a stable voltage, and the DMM measures the current consumed by the device under test.

The voltage is supplied after the LDO, to provide an accurate measurement of the system of interest only. To compute the total current consumption in an application that uses the LDO, like the current device with a battery pack, its quiescent current of 200nA should be taken into account.

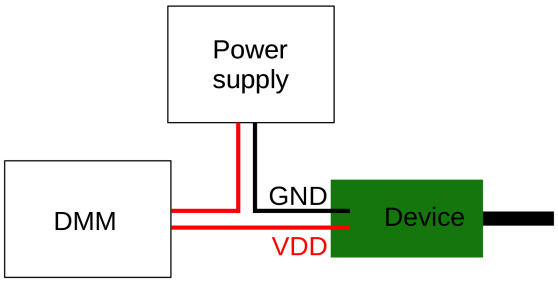


Figure 4.3: Experimental setup for the power consumption measurements.

Because the system can operate either at 1.8V or at 3.3V, both cases are measured and compared.

In order to obtain the energy per task based on the current consumption, it is multiplied by the supply voltage to get the power, then multiplied again by the time taken to perform the task.

4.2.2 Results

Processing

Supply voltage [V]	Processing [mW]	Low-power modes [mW]
1.8V	2.25	0.73
3.3V	5.15	1.77

Table 4.2: Power consumption during processing tasks.

The main source of power consumption for processing tasks is the CPU. Its frequency is fixed at 4MHz for this experiment, but the power is direcly proportional to the frequency. The supply voltage goes not only to the MCU, but also to the peripheral components and the BME680 sensor which is in sleep mode for those measurements.

In addition to the active mode, three low-power modes are considered. When using them, parts of the MCU are stopped to save power, such as the main clock. In STOP0 mode, the Random Access Memory (RAM) stays on such that when the system wakes up, data stored in it is still present. In Standby mode, the RAM is off to save even more power, but then no volatile data can be retained. The deepest low-power mode is in Shutdown, when only the RTC and three specific GPIO pins are still alive and can wake up the system.

In theory, the deeper the low-power mode, the more power it saves. However in this case, they all give the same result, presented in Table 4.2, which is orders of magnitude above the values given in the datasheet [26]. It means that something else than the MCU is pumping current, such as pull-up resistors. It could be the internal pull-up resistors of the GPIOs or the one that are connected to the I^2C bus, but that hypothesis is not verified.

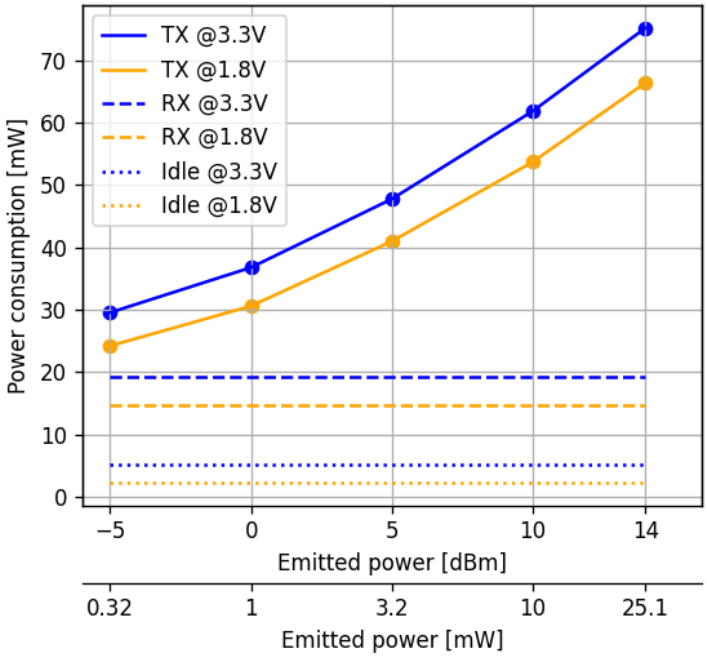


Figure 4.4: Power consumption of the radio for different tasks.

Supply voltage		1.8V	3.3V
TX	-5 dBm	24.2	29.5
	0 dBm	30.6	36.8
	5 dBm	41.0	47.8
	10 dBm	53.7	61.8
	14 dBm	66.4	75.2
RX		14.6	19.2
Idle		2.2	5.1

(a) Values from Figure 4.4.

Spreading factor	Time [ms]	Energy at 14dBm [mJ]	
		1.8V	3.3V
SF7	52	3.45	3.91
SF8	93	6.18	6.99
SF9	165	10.96	12.41
SF10	330	21.91	24.82
SF11	660	43.82	49.63
SF12	1319	87.58	99.19

(b) Time and energy consumption to send a 16-byte packet at a power of 14dBm.

Table 4.3: Power consumption to perform radio tasks.

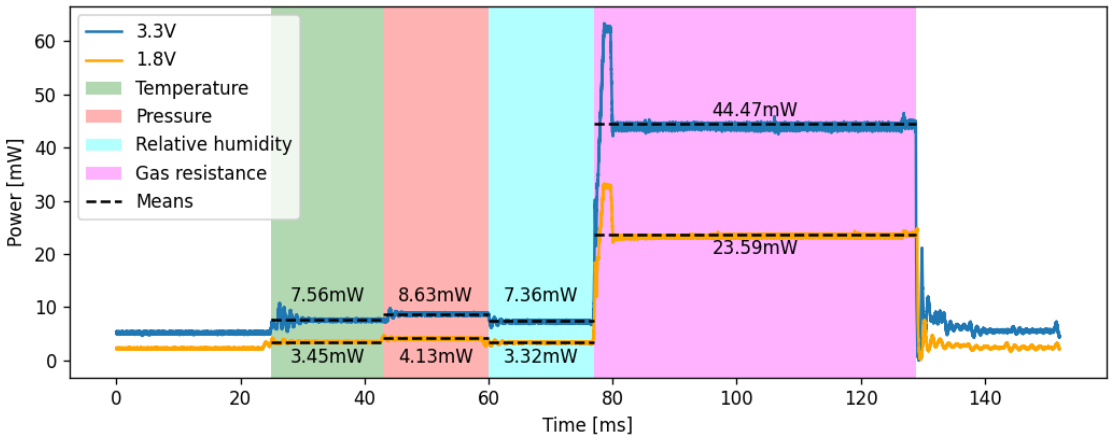


Figure 4.5: Power consumption of the sensor.

Sensed quantity	Time [ms]	Power [mW]		Energy [mJ]	
		1.8V	3.3V	1.8V	3.3V
T	17.5	3.45	7.56	0.06	0.13
P	17.5	4.13	8.63	0.07	0.15
H	17.5	3.32	7.36	0.06	0.13
G	55	23.59	44.47	1.30	2.45
Total for TPHG	107.5	13.86	26.60	1.49	2.86

Table 4.4: Energy consumption per sensed quantity.

Radio

Figure 4.4 and Table 4.3a show the power consumption of the device to perform tasks linked to the radio module. The idle mode represents the base consumption of the MCU, when the CPU is in active mode. On top of that, the consumptions in RX and TX modes for different emitted power levels are added.

The consumption in RX may appear low, but as the energy consumed grows linearly with time, it rapidly becomes a fair amount of energy. Indeed, when implementing the LoRaWAN protocol for instance, RX slots are typically defined with a fixed duration to allow the end node to receive data from the gateway. In TX mode, there is a base consumption greater than the actual emitted power that grows with the power level. Indeed, when emitting less than 1mW, the radio consumes around 25mW and 30mW for 1.8V and 3.3V respectively.

In Table 4.3b are presented the duration and energy consumption in addition to the power required to send a 16-byte packet for different spreading factors. The sending duration doubles at each SF step, although at small SF we can see the effect of a small overhead. Anyway, it still takes 25 times more energy to send the same packet in SF12 than in SF7, instead of $2^5 = 32$ times more if it doubled each time.

Sensor

Figure 4.5 and Table 4.4 show the power and energy consumption of the sensor when it acquires all four types of data.

With an oversampling factor of 8, it takes 17.5ms to perform the successive acquisitions for each of the three first quantities. The relative humidity measurement consumes a little more than the two others. The gas concentration measurement begins with the heating phase during 50ms, and then a single ADC acquisition. The peak at the beginning of the heating phase corresponds to the actual heating of the sensor's plate, then the target temperature is reached and is kept stable with a lower current.

It can be noticed that the BME680 embeds an LDO to regulate its supply voltage. Indeed, the current is the same for both voltages, and thus the ratio of the power consumptions is equal to $\frac{3.3}{1.8} \approx 1.85$.

4.2.3 Discussion

For all the tasks, the lower voltage supply, i.e. 1.8V, consumes less power than the higher one. While this result is obvious for the sensor that uses an LDO, the MCU than manages the radio and the processing unit operates through an SMPS which is more efficient at low voltage because it is optimised for 1.8V.

This characterisation presents the current state of the system. No particular effort is made to optimise and reduce the power consumption at this point. However, it would be necessary to do it for implementing a real application where life span or power budget is critical.

4.3 Detection

The objective of this characterisation is twofold.

First, it assesses the system's ability to effectively detect forest fires through the measured quantities of the sensor, particularly the concentration in VOCs. Knowing the quantitative effects of a fire on the data acquired is critical to implement a mechanism for triggering alerts based on, for example, a threshold value or a behavioural pattern recognition.

Secondly, an important parameter for the deployment of a WSN is the required density of nodes. It is directly linked to the area coverage of each node and thus to the maximum distance of detection of a fire.

The experiments were conducted outdoors, by detecting burning fires in an environment as close as possible to a real forest environment.

The first one revealed several shortcomings in its setup, which were taken into account in the second one to obtain more conclusive results.

4.3.1 Detection during a fire prevention exercise

We had the opportunity to join a fire prevention exercise taking place at FACT Training, Limal [30]. While a group of ten people participated to the exercise, we could set up the sensors and acquire data throughout the training session.

Setup

The layout of the courtyard where the exercises take place is presented in Figure 4.6. Three containers, arranged in a U shape, surround a big fireplace. Meteorological measurements specify that a wind of around 20km/h blows in a direction slightly off the opening of the containers.

With such a wind, the smokes clearly move in a specific direction and don't spread out everywhere. The sensors are thus set up in that direction from the fire. Three poles each supporting two ForestMEv2 devices, at respective heights of 1.75m and 3.5m are placed at three different distances from the fireplace, i.e. 3, 25 and 65 meters.

The six sensors acquire data with the default parameters (see Section 3.4.5) once every three seconds, then send them in a LoRa packet. They are synchronised by the receiver at start-up such that a TX time slot is allocated to each. Because the RTCs are accurate, no significant deviation of the sampling frequency is observed in the timescale of interest and thus there are no collision in the transmission channel.

The initial prototype, presented in Chapter 1, is used as the receiver. It is always in RX mode and dumps the content of any received packet through UART to a computer connected by USB.

Two of the performed exercises are interesting for this experiment. The first one is an exercise that takes place inside one of the containers. Smoke is accumulated inside and released in very large quantities once in a while by opening the container's door. The second is an extinguishing exercise on the central fireplace, where a fire is lit and then put out with an extinguisher. The fuel is a liquid vegetal oil that burns cleanly without producing soot. Because it is vegetal, it is assumed to reject VOCs that can be measured by the sensor [12]. Figure 4.7 shows photographs of the two scenes with one of the poles visible.

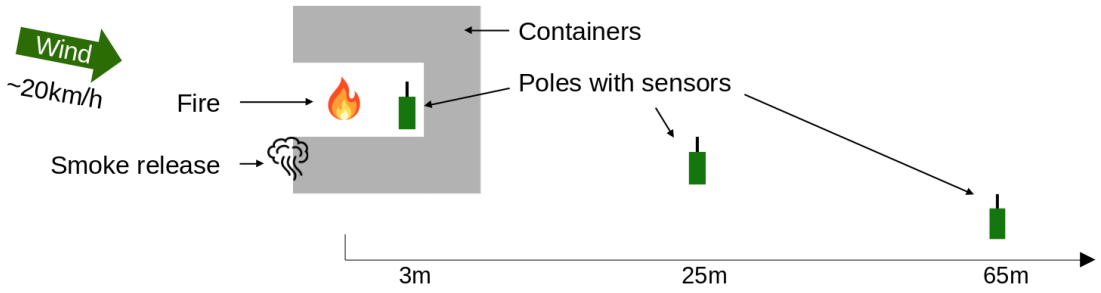


Figure 4.6: Setup for fire detection during the fire prevention exercise.

(a) Smoke released by opening the door.

(b) Flaming fire lit up.

Figure 4.7: The two exercises of interest. The arrows point to the closest pole with sensors.

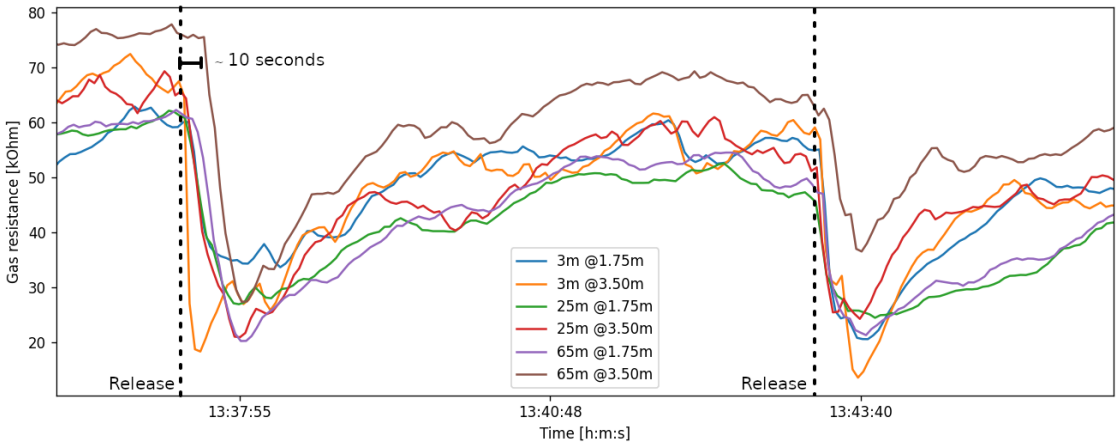


Figure 4.8: Sensed gas resistance during two smoke releases.

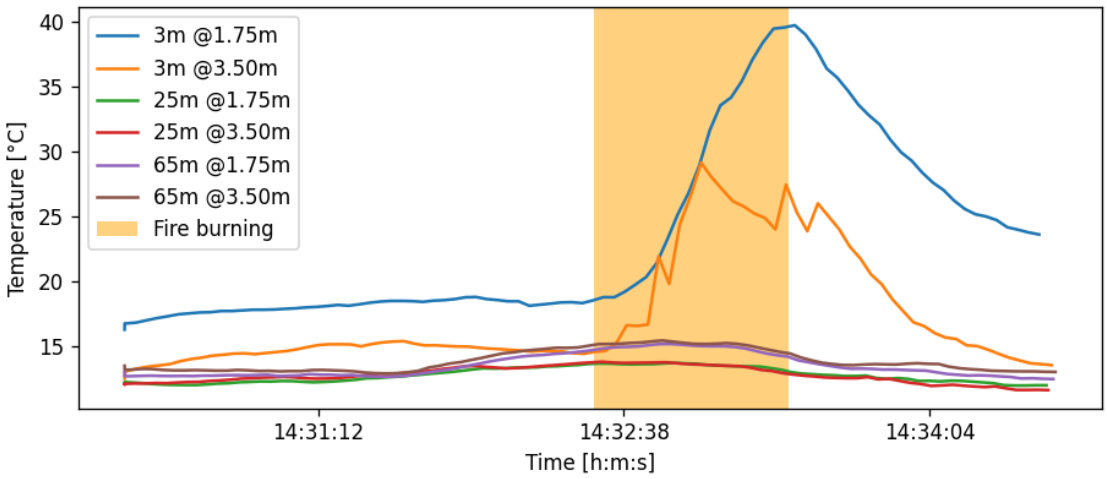


Figure 4.9: Detection of a fire in the central fireplace, rapidly extinguished.

Results

Smoke release Figure 4.8 presents the gas resistance measured by the sensor during 10 minutes. A first release of smoke happens around 13:37 and a second one around 13:43. As a reminder from Section 1.1, the gas concentration is inversely linked to the resistance, i.e. the smaller the resistance, the higher is the concentration in VOCs.

After each of the two events, the smoke moves fast due to the high wind speed and quickly reaches all six sensors. The reaction is immediate with a decrease of at least 50% in the gas resistance compared to its initial value. Because the large quantity of smoke is released at once, it acts as a single smoke packet and then nothing. The resistance thus gradually ramps back up to the pre-event value.

The farthest sensors detect the smoke some time after the closest one due to the time taken by the smoke to propagate. It roughly corresponds to the speed of wind, with approximately 10 seconds taken to cover 62 meters for a speed of about 6.2m/s $\approx$ 22km/h.

Flaming fire At 14:32:30, the liquid oil in the central fire place is ignited. It burns very hot with large flames, then is put out within a minute by using an extinguisher. As it can be seen in Figure 4.9, the two sensors that are on the pole three meters away from the fireplace measure a quick increase in temperature of at least 15°C.

This phenomenon is due to the radiated heat from the flames. It is only perceptible in direct view of the flames and at a short distance. Indeed, the sensors located on poles at 25 and 65 meters don't measure any significant change of temperature. A variation of a few degrees is most probably due to a passing ray of sunlight that temporarly heats the sensor.

The lower sensor on the closest pole (blue curve) measures a baseline temperature higher than all the others because it is located at a height lower than the containers. Because of this, it is protected inside the U-shape and is heated by the nearby container in which fires are burning for other exercises. This protection from the winds also allows it to reach a higher temperature when the flaming fire is lit up.

Discussion

The presented results allow to confirm two detection features of the system. First, smoke is rapidly and unambiguously detected whenever it reaches the sensor, even at 65 meters from the source. Secondly, when the fire is located very close and in line of sight, the sensor measures the radiated heat as a rise of temperature.

No information about the detection of a fire can be extracted from the pressure and relative humidity measurements. Furthermore, the two heights at which sensors were set up don't lead to a conclusive difference in terms of measured quantities.

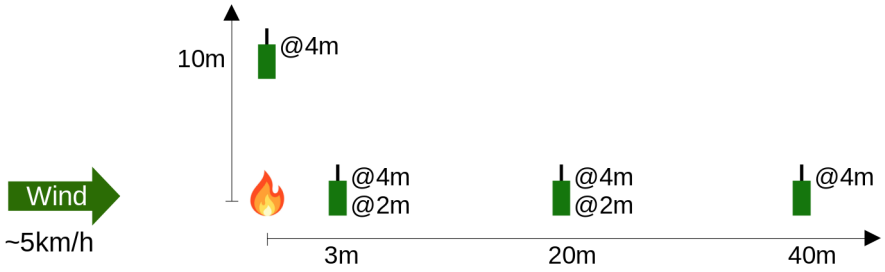


Figure 4.10: Setup for fire detection in a forest environment.

(a) Sensor attached to a tree.

(b) Smoke reaching the sensors.

Figure 4.11: The experimental setup in a forest environment. The arrows point to the closest sensors from the fire.

The detection characteristics of the system could be better analysed with a setup that is closer to a forest fire starting up. Three main shortcomings are identified in this experiment.

- The containers form a screen that disturb the smoke propagation and block the radiated heat from reaching the sensors behind them.
- Because exercises are performed one direcly after the other and some fires, such as the one in the container for the first exercise, never fully stop, there is always smoke in the air spread out a bit everywhere. It is thus difficult to identify a baseline gas resistance that represented a normal forest situation.
- The fires are ignited and extinguised rapidly. It doesn't allow to observe the full evolution of the measured quantities during one single long fire event.

4.3.2 Detection of a forest fire

To cope with the shortcomings of the first experiment, another one is performed with an improved setup. It takes place in a private wood which corresponds well to a real forest environment.

Setup

As for the previous experiment, six sensors are set up at different locations and the initial prototype is used as the receiver. They are attached to trees at distances of 3, 20 and 40 meters from the fireplace in the wind's direction. At 3m and 20m, one sensor is attached at a height of 2m and another one at 4m. Only one sensor is set up at a distance of 40m and a height of 4m. The last

sensor isn't located in the wind's direction, but at a 90° angle from it at a distance of 10m and a height of 4m.

Figures 4.10 and 4.11 show respectively the setup schematically and photographs of it.

Results

In the tables and graphs presenting the results, the sensors are named according to their location, e.g. "d3 h2" is the sensor set up at a distance of 3m and a height of 2m.

The gas resistance measured during all the experiment is shown in Figure 4.12. The other sensed quantities are not presented because they show no interest for smoke detection, as highlighted by the previous experiment. The complete temperature, relative humidity and pressure data are exhaustively presented in Appendix E.

Two phases can be identified on the graph. During the first 40 minutes, the initial state of the forest is sensed to capture the baseline resistance value of the gas sensor. Then, the fire is started and data is acquired for 60 more minutes. It ends before the fire is completely extinguished but it burns lower and lower for the last 30 minutes of acquired data.

Several minutes before the start of the fire and for the three first minutes after its ignition, the receiver crashed and no data could be captured. This inconvenience is however not critical for the study of the impact of the fire on the sensed data.

In the baseline situation, the gas resistance measured by the six sensors is stable around 100kΩ, with little variation ($\sigma < 3\text{k}\Omega$). The non-zero standard deviation is mostly explained by environmental noise, as illustrated by the curves of the two sensors located at 20m that are subject to the same fluctuations due to the movements of the air they both sense. Device-to-Device (D2D) variability is the source of the difference in the means of the different curves. Indeed, the manufacturing process makes the gas sensors different from each other, with a sensitivity and a base resistance, i.e. the parameters from Equation 1.1, that are not the same.

After the fire is lit up, its smoke propagates towards the sensors that detect it. The sensed resistance drop instanteneously when smoke reaches it, and climbs back up when there is no more smoke, like it was previously seen in the other experiment.

The five sensors located in the direction of the wind from the fireplace react strongly. The mean resistance drops by a few tens of kΩs, with minimum values reaching down to only 4kΩ. The closer the sensor is to the fire, the lower the minimum resistance. Indeed, the smoke is denser at the beginning and disperses as it spreads. The standard deviation is bigger than in the baseline situation because the resistance drops then climbs back up each time a cloud of smoke passes through the sensor.

A difference between the two sensors attached to the tree located at 20m can be identified. Because the smokes rises up as it travels, the top sensor (4m) reaches a lower minimum resistance, has a lower mean and a higher standard deviation than the lower one (2m).

The sixth sensor, set up at a 90° angle from the wind's direction, does not sense anything, except at the very beginning when a bit of smoke reaches it. At that moment, the wind doesn't blow steadily but rather shifts in multiple directions now and then. Its mean and standard deviation stays close the to values of the baseline situation.

Discussion

This experiment shows that the gas sensor embedded in the BME680 is well adapted for detecting the smoke from a forest fire. As soon as smoke reaches the sensor, the resistance strongly decreases by at least one order of magnitude, making a threshold-based detection algorithm easily implementable.

However, it suffers from a major drawback. The smoke needs to touch the device to be detected, so it doesn't work if the sensor is not located in the wind's direction from the fireplace, or if there is no wind at all and the smoke goes directly upwards.

The distances involved in this experiment are not large enough to reach the maximum distance at which smoke can be detected, because of the lack of space in the private wood where it took place. At 40m, the sensor still clearly reacts, not much less than the one located closer to the fire. It can be expected that if the smoke goes on spreading into the forest, it would still be detectable from far away, i.e. several hundreds of meters. But that assumptions cannot be verified with the acquired data, it would need further investigation.

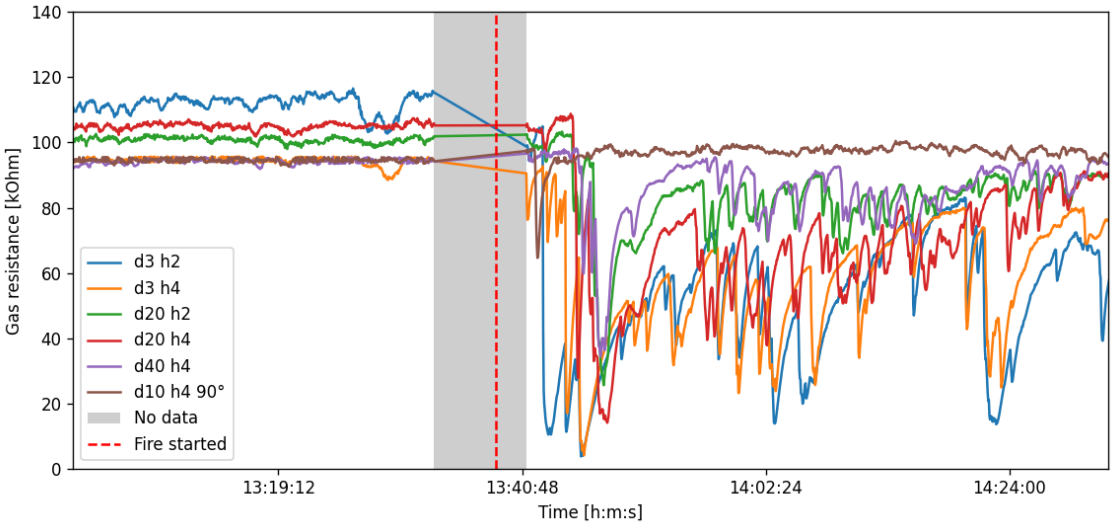


Figure 4.12: Detection of a fire in a forest environment.

Sensor location		d3 h2	d3 h4	d20 h2	d20 h4	d40 h4	d10 h4 90°
Baseline	Mean	112.0	94.4	100.8	105.0	94.2	94.6
	σ	2.8	1.3	0.9	1.0	0.7	0.6
	Min.	102.7	88.5	98.0	101.8	91.8	92.6
Fire burning	Mean	51.9	58.2	82.6	69.8	85.8	96.9
	σ	20.1	17.6	11.1	18.4	9.9	3.1
	Min.	3.7	4.0	25.6	14.0	34.4	64.6

Table 4.5: Mean, standard deviation (σ) and minimum (Min.) values of sensed resistance in kΩ before and after starting a fire.

An important factor revealed by this experiment is that the complex spreading behaviour of the smoke plays a critical role in the detection. If it doesn't stay near the ground, the detection is not possible. However, the smoke emitted in the early stages of a fire is very dense because it is charged in moisture from the fuel as it dries. It thus tends to stay low and easily reaches the sensors. The physics behind the propagation of smoke is complex to model and this subject is out of the scope of this work.

Another point to mention about the smoke detection is that the time required to detect a starting fire directly depends on the speed of propagation. If the smoke spreads slowly, it could take a long time before reaching the closest sensor.

Finally, when a fire is burning, the gas resistance is not always low, as it depends on the presence of smoke in the specific place where the sensor is located. In this experiment, the sensing rate is way higher than the frequency of the resistance fluctations, such that any small variation is clearly detected. In a real application, it would be useful to sample at a sufficiently high rate to be sure not to miss a low peak of the resistance value. The period of one peak is approximately one minute, so the sampling period should be lower than 30 seconds to detect it automatically, according to the Nyquist–Shannon theorem. But sampling at a high rate comes with a great cost in power consumption. A good compromise could be found by adapting the sampling frequency as a function of the FWI, i.e. when the risk of fire is high, the sampling rate should be high.

4.4 Communication

The last series of experiences focuses on the characterisation of the communication performances of the device.

First, laboratory experiments are conducted to extract in a controlled environment the accuracy of the emitted power and the Packet Error Rate (PER) as a function of the SNR of the received signal.

Then, the performances are assessed in a forest environment to evaluate the maximum distance at which two devices can communicate with each other, by analysing the evolution of the SNR and the Received Signal Strength Indication (RSSI) with distance.

4.4.1 Emitted power

The objective here is to verify that the functional blocks involved in the radio communication, i.e. the MCU's radio module and the RF chain, operate as expected to transmit the right amount of power to the antenna.

One possible cause of disturbance that could cause losses in the transmission is the unmatched impedance of the RF network. As explained in Section 3.2.5, the copper traces don't have the same 50Ω characteristic impedance as the one of the MCU and the antenna.

Setup

LoRa packets are sent from the device with the default parameters (see Section 3.4.5), apart from the power level that is sweeped from −5dBm to 14dBm. Instead of the antenna, the RF chain's output is connected to a spectrum analyser, the MS2713E from Anritsu. This instrument measures the incoming RF power for a certain range of frequencies. Its main configuration settings are displayed in Table 4.6. The central frequency set in the spectrum analyser is also the central frequency of the LoRa configuration.

Central frequency	Span	Number of points	Resolution
866MHz	180MHz	550	327.3kHz

Table 4.6: Spectrum analyser frequency settings.

Results

As shown in Figure 4.13, the device emits a very accurate power level. This accuracy tends to decrease when the power level decreases, but still remains close to the configured value. At 14dBm, the error is only 0.06dBm while at −5dBm it reaches 0.3dBm.

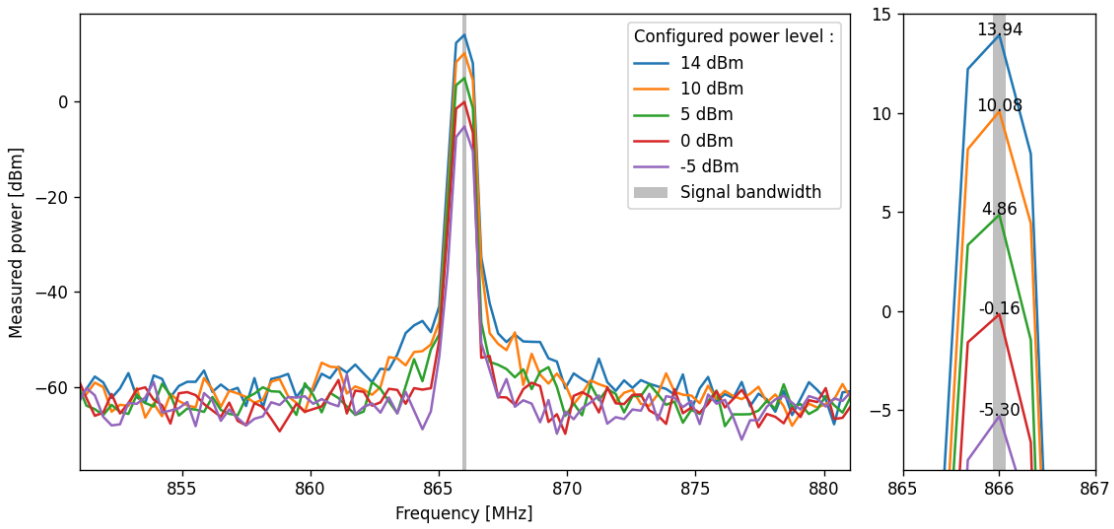


Figure 4.13: Actual emitted power as a function of the configured power level in software. The graph on the right is a zoom on the peak values.

The actual emitted power tends to be a little lower than the configured one. A possible explanation relies on the fact that the RF chain implemented on the PCB is optimised for a power of 14dBm, so when the emitted power is lower than that, the RF chain is less effective.

An explanation is required to understand why the peak value on the graph actually corresponds to the signal power. The total emitted power is the integral of the power on the bandwidth of interest. Because the signal bandwidth is smaller than the frequency resolution, the central point at 866MHz captures all of it and thus indicates the total emitted power of the device.

In conclusion, this experiment shows that the software configuration of the power level is effective because it closely corresponds to the actual emitted power, which means that there is no significant disturbance or loss coming from the RF chain or other functional block involved in the radio transmission.

4.4.2 Packet Error Rate

The packet error rate reflects the quality of communication as a key indicator to consider a communication link as being reliable. The objective of this experiment is to characterise the tradeoff existing between the spreading factor and the PER. According to LoRa theory, increasing the SF by 1 decreases the minimum required SNR by 2.5dB to correctly demodulate a packet (see Section 1.3.1).

That property is assessed in a controlled laboratory environment, to have the best control possible on the SNR such that it is stable. Indeed, in a real environment, the common way to attenuate the signal is to increase the distance between the devices, but the channel's attenuation is often neither constant nor accurately known in that case.

Setup

In order to reach a sufficiently low SNR such that it is at the limit of the demodulation capabilities of the radio module, the signal needs to be attenuated. To this end, the communication channel, which is by default composed of the antennas and the air in between, is replaced by an SMA cable that provides an assumed lossless transmission channel. Four 30dB RF attenuators, for a total of 120dB, are connected between the transmitter and the receiver devices.

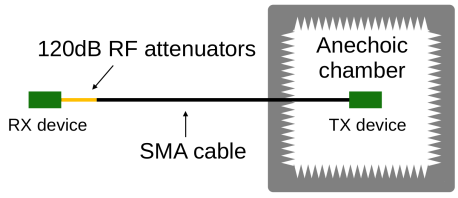


Figure 4.14: Setup for the PER measurements.

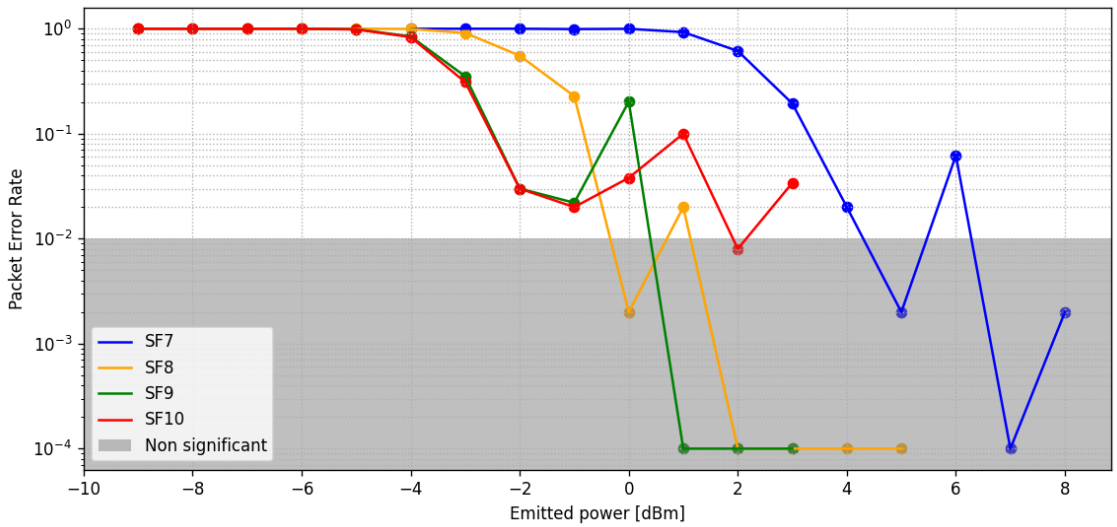


Figure 4.15: Computed PER as a function of the emitted power.

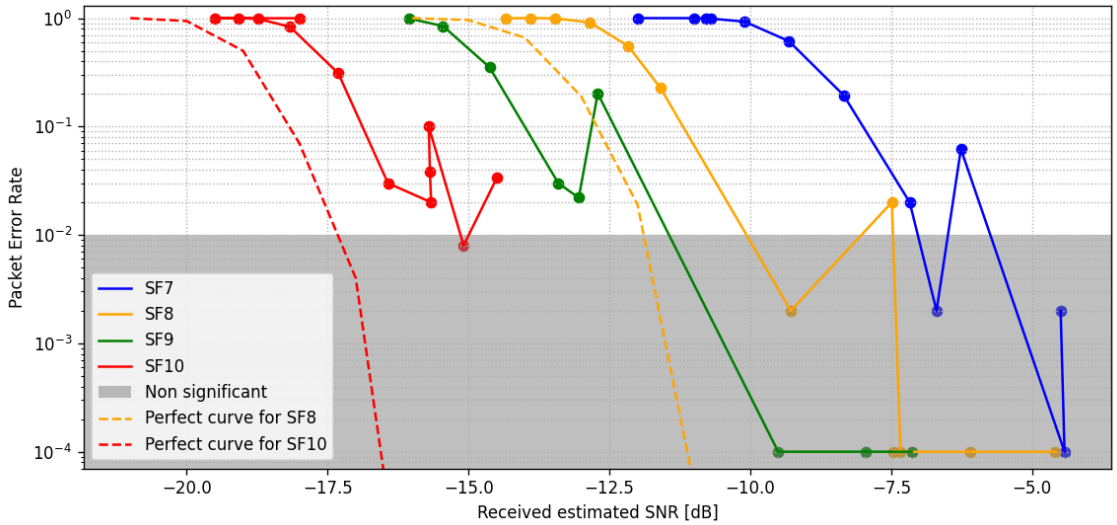


Figure 4.16: Computed PER as a function of the estimated SNR. The theoretical curves are extracted from [31].

The devices are isolated from each other by placing one inside an anechoic chamber and the other one outside. Otherwise, the communication is still successful even by completely attenuating the wired channel, because the RF signal from the transmitter is emitted from the PCB tracks and the SMA connector. The high sensitivity demodulation capabilities of LoRa makes the reception possible at very low RF power. When the antennas are not plugged in the connectors, the RSSI is about -60dBm at a distance of one meter, which is way above the RX sensitivity of the radio that can go down to -148dBm [26].

While sweeping the power level of the transmitter by 1-dB steps, 500 packets are emitted with the default parameters (see Section 3.4.5) for the power levels in an interval that leads to a received SNR and RSSI around the demodulation capabilities of the receiver. Because the time taken to send a packet approximately doubles each time the spreading factor is incremented, it takes a significant amount of time to perform the experiment at high SF. It is thus only done for SF7 to SF10.

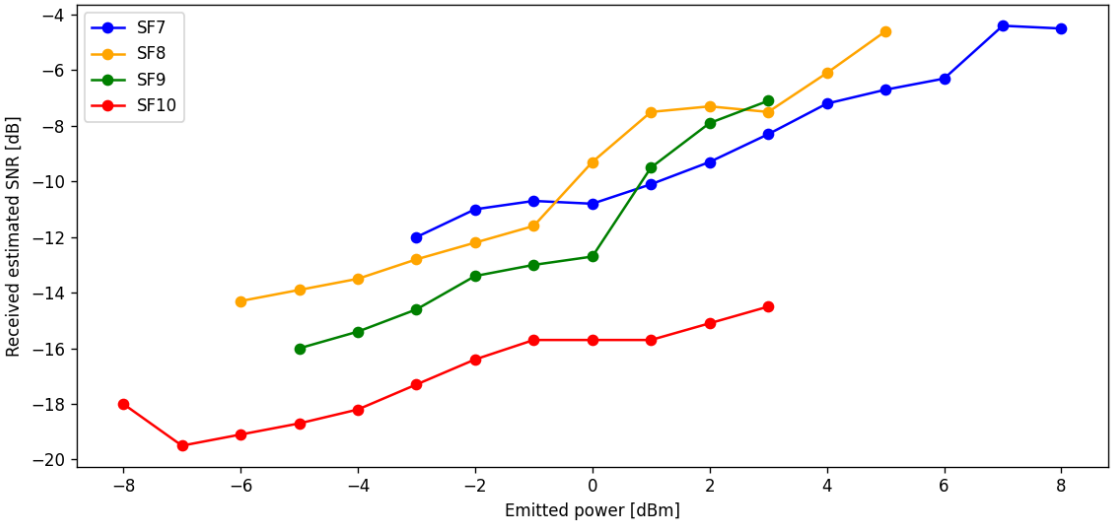


Figure 4.17: Variation of the SNR at fixed power level for different SF.

Results

Figure 4.15 shows the computed PER over 500 packets for each SF and at the measured TX power levels. At the receiver, that power is attenuated by 120dB, in addition to the losses in the RF chains of both devices. Because there are only 500 packets to compute the PER, it is significant down to 10^{-2}. Below that value, too few trials are done to obtain a statistically robust estimation. The power levels that lead to a PER of zero are brought back to 10^{-4} so they appear on the plot as data points.

The SF8 curve shows an improvement in the communication performances compared to SF7. The emitted power can be 4dBm lower in SF8 before experiencing the same drop in PER. Some data points are out of the trend because of shortcomings in the setup. At some point, something unidentified goes wrong and the communication is strongly degraded during several minutes, maybe because of interferences but it is not further investigated. The sudden PER increase at 6dBm for SF7 and 1dBm for SF8 are due to that phenomenon.

However, SF9 doesn't show the same improvement compared to SF8, and SF10 is even as bad as SF9 with no improvement at all. The higher power data points for those two curves where the PER is between 10^{-1} and 10^{-2} can be explained by the same phenomenon that degrades some points of the two first curves. But in this case, because the experiment takes much more time because of the high SF, the probability of having a setup failure for some minutes during the 500-packets run is higher, thus perfect PER never happen.

Instead of plotting the PER as a function of the TX power, it can be done with the SNR as shown in Figure 4.16. The SNR is computed in hardware inside the radio chip for each received packet. The average SNR is computed on each group of 500 packets sent at a given SF and TX power. In this graph, the expected curves appear, with an increase of approximately 2.5dB of the link budget for each SF step. The curves are however around 2dB away from the theoretical one. That can be explained by the fact that the latter assumes perfect synchronisation between the devices, which is not the case. Indeed, three different offsets can degrade the RX performance of the demodulator. The Carrier Frequency Offset (CFO) and the Symbol Timing Offset (STO) are usually well compensated, but the Sampling Frequency Offset (SFO) is a lot more complex to correct and leads to synchronisation issues [31].

The two previous graphs show that the SNR measured by the receiver, for a fixed TX power, depends on the SF. The relation between those quantities is presented in Figure 4.17. When the SF is higher, the measured SNR is degraded for the same TX power compared to lower SNR, at least for SF8 and higher.

A decrease of the SNR can be explained either by a decrease in signal power, or an increase in noise power. In the previous characterisation, it was demonstrated that the transmitter outputs an accurate power level. The noise is supposed to be constant, because the noise sources, i.e. the

(a) Density of the forest.

(b) A ForestMEv2 device attached to a tree.

Figure 4.18: Woods of Villers-la-Ville in April.

channel, the RF chain and the RF path inside the radio, don't depend on the SF because it has no influence on the properties of the signal.

A possible explanation is thus that this dependency comes from synchronisation issues. The effects of the SFO could be perceived as a decrease of the signal power. The higher the SF, the longer is the duration of a packet and thus the offset in frequency makes the jitter accumulate over time up to greater values that would explain a loss in SNR.

Discussion

Although the PER/SNR behaviour of the receiver is as expected, the dependency of the SNR on the SF results in few or no improvement at all when increasing the SF.

This means that in an actual application, trading datarate for link budget would cost a lot in power consumption but give almost nothing in return.

4.4.3 Performances in a forest environment

LoRa is known for allowing communication over large distances with a low power budget. It is stated that the maximum range can reach up to 2 to 5km in urban areas and even up to 15km in rural areas with a line of sight [18]. However, a forest environment is not like those type of areas. The trees form a dense shield that prevents the RF signal from travelling that far.

In this experiment, the communication between two ForestMEv2 devices is characterised to assess the actual distance at which a packet can be transmitted in a real forest environment. Moreover, the path loss that the RF signal undergo is modelled by measuring the evolution of the SNR and the RSSI with distance.

These are essential parameters to be quantified in order to know what is the minimum radio configuration, in terms of SF and TX power, required to communicate over a given distance. On a larger scale, it allows to model the total consumption and limitations of an entire WSN given its node density.

Setup

The experiment has been conducted in the woods of Villers-la-Ville, in April. The leaves have already appeared on the trees and other plants at that moment of the year, but the forest is not yet in its densest state, as shown in Figure 4.18. It can be considered as an average density that leads to an average attenuation of the signal, while in winter the woods are less dense and in summer more.

Figure 4.19 presents the experimental setup used for this characterisation. A path as straight and as flat as possible is selected, along which five out of the six devices are set up 50m apart.

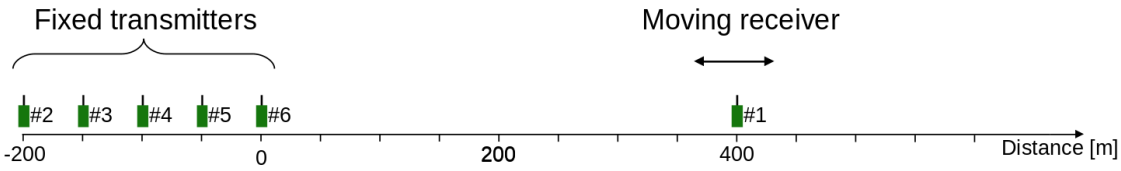


Figure 4.19: Experimental setup of the communication measurements in a forest environment.

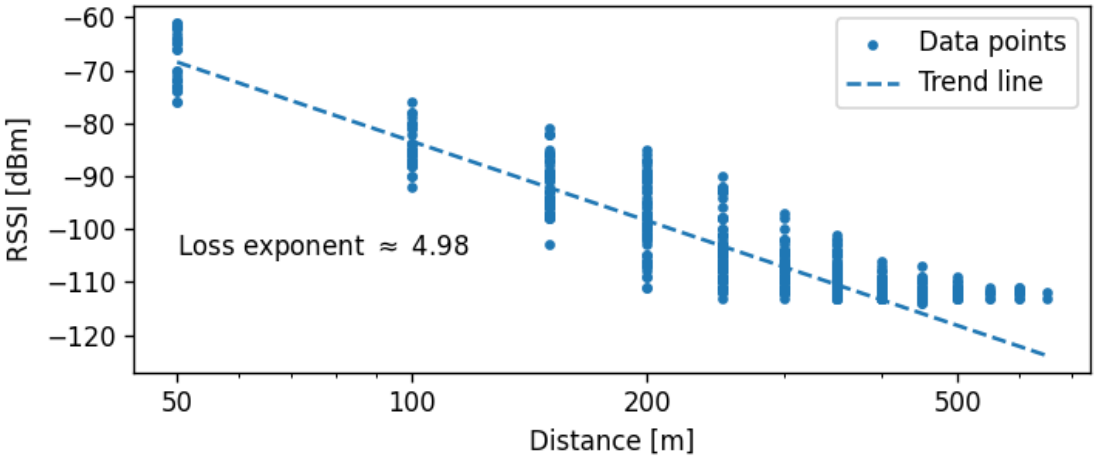


Figure 4.20: RSSI as a function of the distance.

They are attached to trees at a distance of 20m from the path, all on the same side, such that the path has little influence on the propagation, i.e. it does not act as a wave guide.

The sixth device is used as the receiver. It is moved in 50m steps and receives packets during 10 minutes at each location.

Each transmitter sends a packet every 12 seconds and loops through the six spreading factors, with the other parameters being the default ones (see Section 3.4.5). The transmitters are synchronised and assigned a time slot for the transmission, so there are no collision. Twelve seconds are needed to have enough margin on the time slot to accomodate the transmission of a packet in SF12 that takes around 1.5 second. The receiver thus potentially captures data from all five transmitters on all six SF once every 72 seconds.

The RSSI and SNR values are computed in hardware inside the radio chip for every demodulated packet. Details about the way those values are computed are unknown. The output values have poor resolution of only 0.25dB for the SNR and 1dB for the RSSI.

Results

RSSI Figure 4.20 shows the evolution of the RSSI with distance, in log-log scale. Each point represents a correctly demodulated packet, from all six senders at all SF combined.

The curve follows the typical path-loss model, where the loss increases with distance and thus the RSSI decreases according to Equation 4.2, where d is the distance and n the loss exponent.

$$\text{RSSI [dBm]} \propto -10\, n\, \log_{10} d \tag{4.2}$$

In the free-space path loss model, where the only attenuation is due to the spreading of the signal, $n = 2$. While it can be lower than that value, by using wave guiding for example, the loss exponent is often greater than that, meaning that other attenuation sources exist. In the case of a forest, the vegetation and ground attenuate the signal, such that the computed loss exponent is a little less than 5. It corresponds to what was expected, falling in the range of values of an existing study that measured loss exponents between 3.83 and 5.24 in a similar forest environment [32].

The trend line is computed only for the distances that lead to RSSI values far from the sensitivity limit. Indeed, from 250m and more, the PER is non-zero and only the part of the RSSI distribution

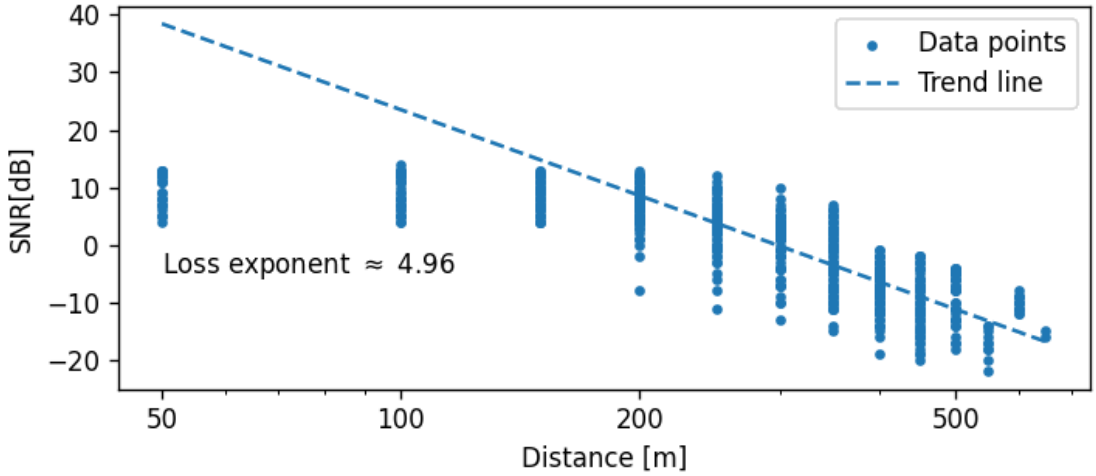


Figure 4.21: SNR as a function of the distance.

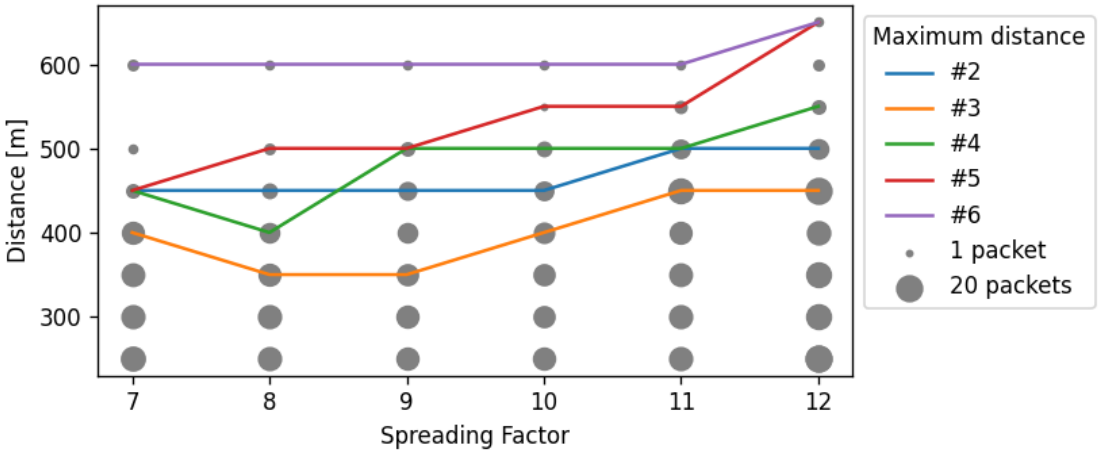


Figure 4.22: Maximum distance at different SF for a correct packet transmission.

that falls above the sensitivity limit is correctly demodulated.

The measured limit for the sensitivity is −115dBm. It is higher than the values stated in the datasheet that can reach down to −138dBm at SF12 [26]. Two hypotheses can explain this phenomenon.

- The external components present on the PCB, i.e. the RF network and the antenna, generate noise that decreases the SNR below the required one before the signal power can reach the sensitivity limit.
- The values expressed in the datasheet are measured with experimental conditions different from the ones used in this characterisation. The values of the datasheet are optimal and may not be systematically reachable in an application.

SNR The evolution of the SNR with distance, presented in Figure 4.21, follows the same trend as the RSSI, which is expected as the noise has a constant power and the signal power is the RSSI. The computed loss exponent is thus very close to the previous one.

The computed path loss model doesn't apply to the SNR at low distance. It can be seen on the graph that the SNR saturates between 5 and 13dB even when all the packets are correctly received. Again, two explanations can be put forward.

- It can simply come from the hardware computation of the SNR which is not intended to work correctly at high SNR. Indeed, LoRa usually works at low SNR to save power, and it is not useful to accurately compute it when it is high enough for a perfect demodulation. However, no such information was found in the datasheet, though it would probably appear if it was the case.
- Another cause can be that an automatic controller adjusts a variable gain amplifier to limit the signal power when it is high enough for perfect demodulation. The signal is kept low while the noise generated later in the RF demodulation chain limits the SNR to that value. When the gain control reaches its maximum value, it cannot go further and the signal power drops according to the RSSI.

Maximum distance In Figure 4.22, the grey balls represent the number of packets received from any transmitter, at each distance as a function of the spreading factor. This size of the ball being proportionnal to the number of packets received, it gives an idea of the PER but it is not rigorous because the number of packets sent per location is not constant and not monitored. Indeed, the receiver stayed around 10 minutes at each place, but that duration was not exactly respected because the purpose of the experiment was not to extract the PER. The number of packets received decreases with distance, as does the actual PER because the SNR is more and more degraded.

In addition to the balls, the lines represent, for each individual emitter, the maximum distance at which at least one transmission is successful. Three analyses can be done on those results.

First, there is a strong variability between the different channels. The emitters are set up 50m apart from each other, so the communication channel they face is different for each of them. Even in an environment that was chosen for its homogeneous density of vegetation, a difference of nearly 50% of maximum distance is observed between emitters #3 and #6.

Secondly, a single channel itself is not constant, and the path loss model doesn't take into account the granularity of the location of obstacles. Emitter #6 reflects this behaviour. Indeed, for SF7 to SF9, not a single packet is received at 550m but well 50m farther. It means that the specific location where the receiver was set up at 550m from the sixth emitter had a particularly bad channel due to, for example, a big tree right in the way or topographic reasons like height difference behind a mound or else.

The last point concerns the tradeoff between maximum distance and spreading factor. As mentionned in Section 1.3.1, increasing the SF by one reduces the minimum required SNR for a correct demodulation by around 2.5dB. Such a gain in the link budget is supposed to allow a nearly doubled distance at every SF step when the free-space path loss model is applicable, i.e. with a loss exponent of 2. In this case with $n \approx 5$, Equation 4.2 can be used to computed the ratio of distance gained with a 2.5dB-increase in the link budget, to find that it only increases the distance by 12.2% at each SF step. While that theoretical gain is already not much, the results are even worse than that with gains between 1.6% and 7.6% by SF step for the different transmitters.

This could partly be explained by the results of the previous experiment, where it is shown that the SNR is degraded when the SF is increased, for the same signal power, thus compensating the gain in link budget.

Discussion

This experiment demonstrates interesting results that lead to useful things to remember when implementing a WSN in a forest environment.

First, due to the dense vegetation of a forest, the attenuation is very strong with a high loss exponent. The simple path loss model might not be the most suitable to model a forest environment. A probabilistic approach of the distribution of the tree's location may be a solution.

Secondly, this strong attenuation has a large impact on the tradeoff that LoRa allows between the spreading factor and the communication range. The cost of increasing the SF is that the datarate is divided by two at each step. For an application that sends fixed-size messages, it rather means that the emitting time and thus the energy consumption is doubled. But that cost only gives a poor increase in reachable distance, making it not worth it in a low-power application.

Chapter 5

Extension for bat monitoring

The ForestMEv2 prototype is first and foremost designed for experimental purposes. To this end, versatility and modularity are key features to open the way to many different studies, characterisations or proofs of concept. In order to demonstrate the ability of the created prototype to be modular, this last chapter proposes an extension board which plugs into the main device to add extra hardware components and functionalities.

The chosen application, i.e. acoustic monitoring of bats as a bio-indicator, capitalises on the acquired knowledge about forest environment. Most particularly, the environmental monitoring and data communication techniques and performances studied in the last chapter form the basic architecture on top of which the new application is built.

The chapter is divided in three sections. First, high-level design choices are expressed to justify the relevance of the chosen solution, with the technical specifications required for the acoustic monitoring, as well as the identification of the main challenges and bottlenecks.

Then, the prototype in itself is explained, as for the main device, by detailing the schematic, layout and software used.

Finally, the current applicative performances are stated, together with possible improvement areas.

5.1 Design choices

5.1.1 High-level architecture

The process of automatic acoustic recording of bat populations is divided in two steps. First, data is acquired on the field by a microphone and digitised into numerical values. Then it is classified, commonly by a machine learning algorithm, that can tell whether or not a bat is present, its species or other type of information.

Such classification algorithms already exist or are currently developed [33] [34] [35]. This work focuses thus on the data acquisition, useful in large amount as a first step to train those kind of algorithm, then to actually monitor the bats.

Two common techniques are used for that kind of application. They are described in the following paragraphs with their respective advantages and drawbacks.

Edge AI

The Edge Artificial Intelligence (AI) refers to a set of techniques used to acquire and transmit data from an IoT network. The main feature is that each IoT node performs signal processing operations directly on the acquired data. It then only transmits aggregated information instead of the whole raw data. The benefits of this operation is twofold : on the one hand, it shifts the computing load from the gateway or network server to the end nodes, and on the other hand it drastically reduces the amount of data that needs to be transmitted, thus largely relaxing the datarate constraint on the network.

In the target application, it means that the classification algorithm must be executed on the microcontroller, and only the answer of the algorithm must be transmitted through LoRa. Unfortunately, this idea is not possible to implement because of the tremendous amount of resources

this type of AI requires. The amount of available RAM is the most important limiting factor, as the algorithm developed in [33] needs several gigabytes of RAM while microcontrollers like the STM32WL usually provide up to several hundreds of kilobytes. The complex and irregular nature of acoustic signals, coupled with the very high sampling frequency required to capture ultrasounds, justify that requirement.

LPWAN

The second technique is named Low-Power Wide Area Network (LPWAN) and proposes the opposite. Instead of having the computing constraint on the end node, it relies on an efficient network, able to transmit data at very low energy. LoRa is a good competitor as it is a low power physical technique that allows to communicate over large distances, but it suffers from a low datarate.

Transmitting the complete raw data often represents a too great load for such a network. In this application, it is clearly impossible to transmit such an amount of data because LoRa has a maximum datarate of around 25kbit/s (see Equation 1.2).

An intermediate solution is to combine the best of both Edge AI and LPWAN by extracting main features of the raw data and only transmit them through the network.

Chosen solution

Because neither of the susmentionned techniques are suited for the bat monitoring application, another solution is chosen. Instead of transmitting either the raw data or the extracted features, the data is written on an external non-volatile memory located on the end node. It removes both the computing and the network constraints. Additionally, the raw data is available for other uses than identification, such as studying the social communication through acoustic signals between bats [36]. This type of study can certainly not be conducted with only the results of a classification algorithm, and perhaps also not with only signal features.

The realisation of the prototype requires to add two main functionalities on the extension board. First, an analog front-end (AFE) to capture the acoustic signal. Secondly, a digital interface to an external flash memory. A common type of memory is a micro Secure Digital (µSD) card, which is easily accomodated in this kind of project thanks to its small size and standard interface.

There already exists an open-source project that proposes those functionalities for a bat monitoring application [37]. This work is greatly inspired by that project and reuses several of its components and schematics. The TeensyRecorder is the current most developed design, which is an ultrasound recorder for bat monitoring that is based on a Teensy MCU card. It fits in a box with a size of approximately $25 \times 15 \times 7$ cm and is equipped with two lithium batteries that ensure several hours of recording. With that amount of energy, the TeensyRecorder can be set up and acquire data for one or two nights.

This work proposes the same functionalities but with several major improvements.

- A reduced size, with the extension board stacked on top of the main module, the device holds in the palm of a hand.
- A reduced power consumption, allowing the device to record data for a longer duration.
- The functionalities already present on the ForestMEv2, i.e. the ability to communicate through LoRa and the environmental sensor. The transmission of packets can be used for instance to send summaries of the acquired sounds, like the total duration recorded or the storage capacity used. The environmental data, for its part, can be used together with bat population monitoring to highlight specific behaviours or other correlations.

5.1.2 Specifications

The following specifications are chosen for the proposed solution.

Bats emit sounds with a frequency ranging from 9kHz to 200kHz [38]. According to the Nyquist–Shannon theorem, the sample rate must at least 400kHz to correctly detect the higher frequencies.

Because the ADC provides a 12-bit resolution, continuously acquiring data corresponds to a datarate of 4.8Mbit/s or 600kB/s. This is the rate at which data must be written on the µSD card.

The target life span is in the order of magnitude of weeks. The device should be able to record data for several tens of nights.

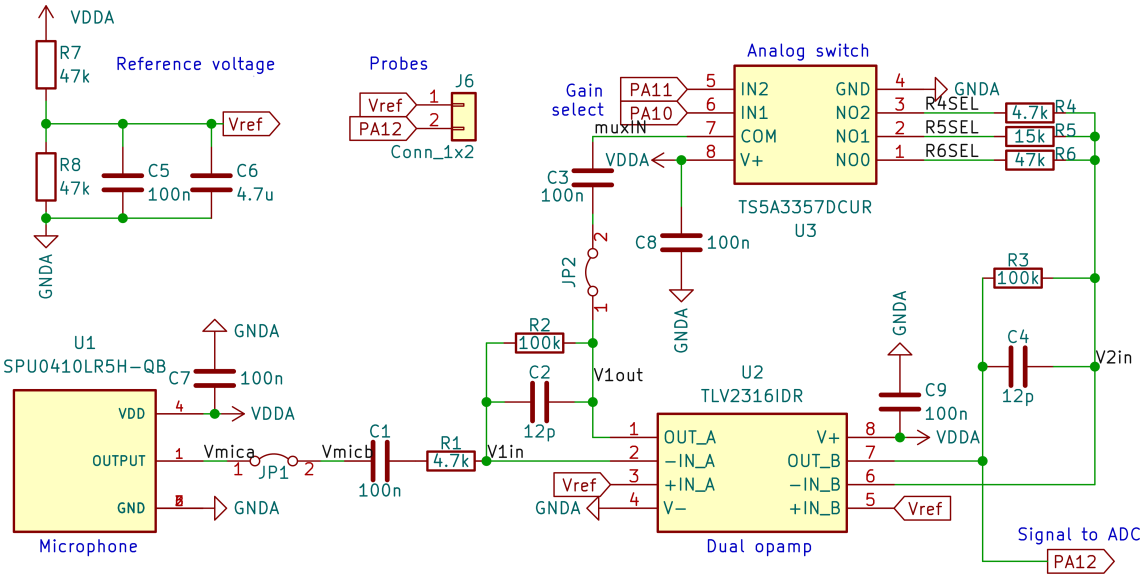


Figure 5.1: Analog front-end.

5.1.3 Preliminary bottleneck analysis

The following preliminary study roughly analyses, by using orders of magnitudes, the different parts of the system to identify the bottlenecks, if any exists.

Bats are mostly around in the evening and the beginning of the night. If the sound is recorded only when it is useful, i.e. when the signal's energy is high at high frequency, it can be assumed that each night is worth around two hours of recording. At 600kB/s, it represents around 4.3GB per night. A 64-GB µSD card has thus the capacity to record data during two weeks, which falls in the specifications. It should be noted that this capacity can easily be extended because there exist much larger µSD cards nowadays.

A µSD card from SanDisk [39] consumes around 100mA when used for writing operations. This current is far greater and is destined to be consumed for a much longer time than the one for any of the other tasks. As a simple estimation, it can thus be assumed that for the whole device operation, the consumption is equal to 100mAh per hour of recording. An AAA battery has a capacity of around 1000mAh, which represents only five nights of recording.

The battery is thus the bottleneck and its capacity should be increased to match the memory life span. In such a case, the device runs out of battery at the same time as it runs out of storage capacity, such that both are replaced together.

5.2 Prototype

The extension board is created following the same design phases as the ForestMEv2 device. The following sections detail the components and electrical circuits used, aswell as the layout and the software architecture.

Complete views of the schematic, the layout and the BOM are available in Appendix F.

5.2.1 Analog front-end

The AFE fulfils multiple roles. The microphone is the sensor, responsible for transducing acoustic signal into an electrical voltage. It then passes through an amplifier, that provides voltage gain and filters out the frequencies oustside the bandwidth of interest. The signal is finally directed to one of the ADC channels of the MCU.

Schematic

The architecture of the AFE is mostly inspired by [37], with only small adjustments.

Bandwidth	IO range	VDD_{min}	I_q	Noise at 1kHz
10MHz	Rail-to-rail	1.8V	400µA/ch	$12nV/\sqrt{Hz}$

Table 5.1: Main specifications of the TLV2316 OpAmp.

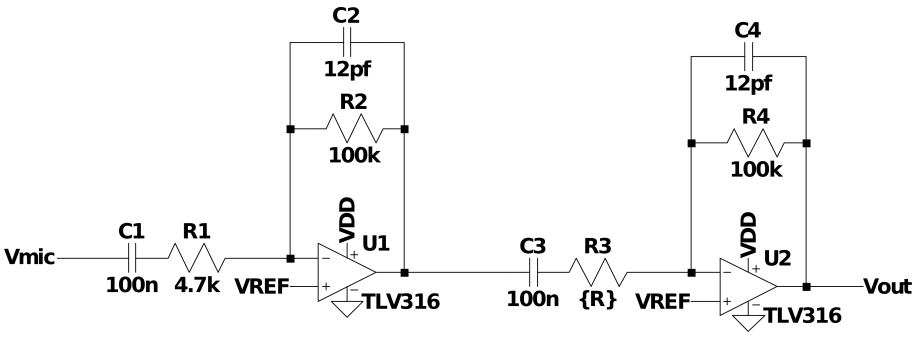


Figure 5.2: Dual OpAmp amplifying and filtering chain of the AFE.

The microphone is the SPU0410LR5H-QB from Knowles [40]. It is a microelectromechanical system (MEMS) low-noise microphone with a particularly stable transfer function (TF) at high frequency, i.e. above 20kHz, as measured by [37], even if Knowles doesn't primarly provide it for ultrasound applications.

The amplification and filtering functions are performed by a dual operational amplifier (OpAmp), together with the resistors and capacitors associated. More details about the transfer function of that chain are provided in the following section. The chosen OpAmp is the TLV2316 [41]. Its main specifications are displayed in Table 5.1. The high bandwidth provides enough gain even at a signal frequency of 200kHz. Rail-to-rail operation is mandatory when working with low-voltage, otherwise the swing is strongly impacted. This OpAmp can work either at 1.8V or at 3.3V, to stay compatible with the main device whatever the supply voltage. Finally, it is a low-noise and low-power OpAmp. Its current consumption is in the same order of magnitude as the one of the MCU, i.e. around 1mA.

The third active block of this schematic is an analog switch, the TS5A3357 [42]. Depending on the 2-wire digital control signal applied on IN1 and IN2, COM is connected to either NO0, NO1, NO2 or nothing. It allows to dynamically, i.e. at runtime by software, adjust the gain of the amplifier chain, by choosing through which of the three resistors the signal goes.

The reference voltage Vref is the half of VDDA. It is generated by a voltage divider and stabilised by two capacitors, that operate in the same way as decoupling capacitors (see Section 3.1.2). Thanks to high resistance values, it only draws 140µA of current at 3.3V.

Finally, jumpers and probes are placed to be able to access and retrieve important signals for debugging or experimental purposes.

Transfer function simulation

The amplifying and filtering chain is composed of two almost identical stages. Each of them applies a band-pass filter and gain to the signal. The high-pass characteristic is mandatory to decouple any DC signal between the different stages and the low-pass one is required as an anti-aliasing filter. The only difference between the two stages resides in the possibility to tune the gain of the second one. The values of the chosen components are shown in Figure 5.2, together with their interconnections.

With these values, the three possible total gains are 33dB, 43dB or 53dB. This choice allows to keep the signal inside the allowed range between 0V and VDD, while maximising the swing. It is expected that the MEMS microphone outputs a voltage of several millivolts, up to 20mV. For a microphone voltage of 10mV, it respectively gives an output signal of 0.45V, 1.4V or 4.5V for the different gains. Of course, in the latter case, the signal would be trimmed at 3.3V.

The capacitance values are chosen such that the high cut-off frequency is at 133kHz. The highest frequencies are a little attenuated but this is necessary to provide the required anti-aliasing feature, for a sampling rate right at the Nyquist frequency. The low cut-off frequency slightly depends on the chosen gain, because the resistor responsible for providing gain also acts on it. It

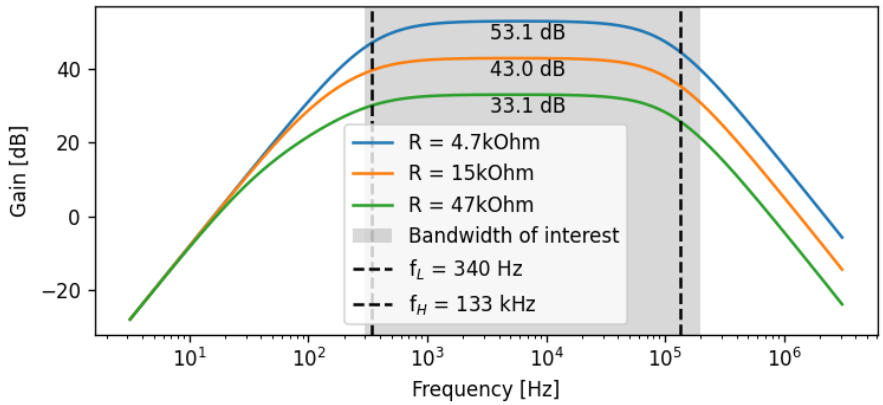


Figure 5.3: Simulated transfer function of the filtering and amplifying chain.

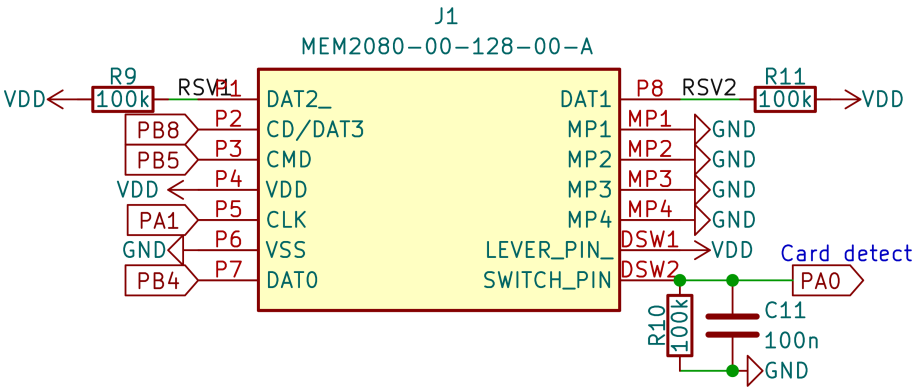


Figure 5.4: Digital storage.

varies around 340Hz, thus the audible sounds are mostly captured.

The total transfer function is simulated and displayed in Figure 5.3 for the three possible gains.

5.2.2 Digital storage

All µSD cards share the same standard hardware interface. The high-performance SD bus is an SPI-like communication bus with four parallel data channels and a command channel, in addition to the clock. It is however also compatible with the simpler SPI interface, which is the means of communication used by most microcontrollers. In this case, because only one data channel is used in each direction, i.e. master to slave or slave to master, the maximum reachable datarate is expected to be divided by at least four.

Figure 5.4 presents the schematic of the µSD connector used on the extension board. To communicate in SPI mode, some pins have another function than the default one. SPI uses four wires : Chip Select (DAT3) that notifies the slave that a communication is addressed to it, MOSI (CMD) and MISO (DAT0) which are respectively the master-to-slave and slave-to-master data channels. The two unused data channels DAT1 and DAT2 must be tied high by pull-up resistors for power consumption reasons, otherwise a floating or low voltage could bring them in a passing state.

When a card is inserted in the connector, a mechanical lever is pressed and connects the LEVER and SWITCH pin. That functionality is used as a card detection feature by being connected to a GPIO of the MCU. The resistor and capacitor prevent glitches from happening.

Finally the MP pins are the mechanical connections of the metallic structure of the connector to the PCB.

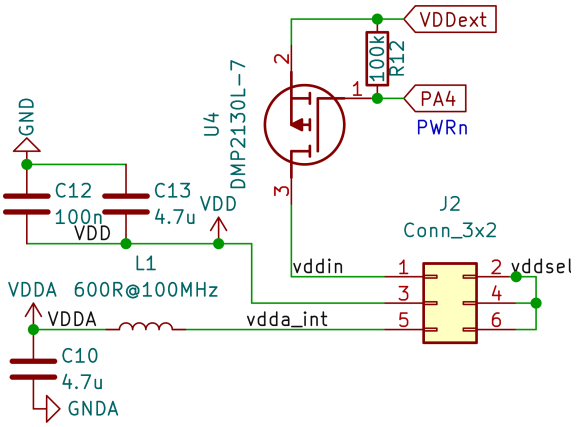


Figure 5.5: Power management unit.

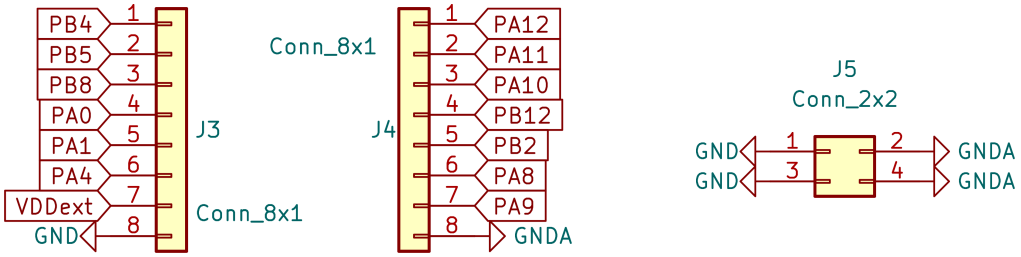


Figure 5.6: External connectors and ground pins.

5.2.3 PMU

The simple power management unit, displayed in Figure 5.5, allows to completely cut off the supply of the extension board. This if useful because the analog part of the board consumes a lot of static current when supplied, mostly the dual OpAmp with its total quiescent current of 0.8mA. It uses a PMOS transistor [43] normally blocked by a pull-up resistor on its gate. When it is activated, the pull-up consumes only 33µA which is negligeable compared to the active components consumption. That specific transistor is chosen for its very low equivalent resistance in passing mode, which is of the same order of magnitude as the one of the copper traces, around 100mΩ.

This feature is critical for a monitoring application, because the detection must only be operated for several hours a day. Outside these time slots, a large amount of energy can be saved by cutting the supply.

In addition to this power switch and as for the main device, a ferrite bead is used to separate VDD from VDDA and attenuate the switching noise. Decoupling capacitors for the active components are also present on this schematic.

J2 normally connects all its pins with external jumpers, which can be removed for experimental purposes.

5.2.4 Connectors

The two 8-pin connectors are the same as the ones on the main device, such that they can be plugged into it, with male pins downwards. Female headers are also present upwards to be able to accomodate another potential extension board on top of it.

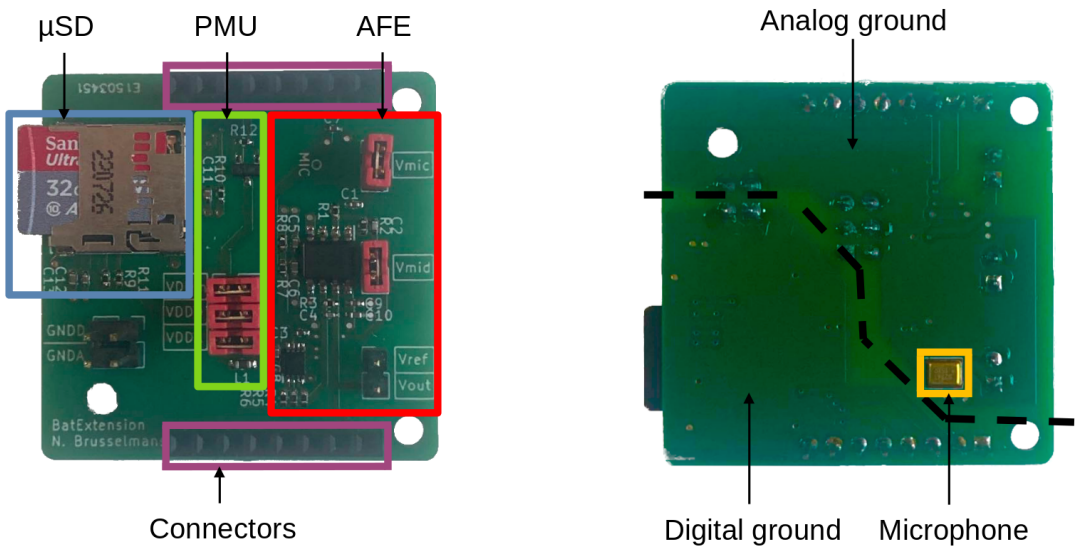
Four ground pins are provided for debugging and experimental purposes.

5.2.5 Layout and assembly

The layout and assembly phases are conducted in the same way as for the ForestMEv2 devices. Two copies of this board, named BatExtension, are produced.

The different parts composing the PCB are highlighted in Figure 5.7.

The digital and analog supplies are separated not only by having different VDDs, but also different ground planes and ground pins. Indeed, each of them is connected to the main PCB by



(a) Top view.

(b) Bottom view.

Figure 5.7: Views of the assembled extension board.

one of the two available ground pins, located at opposite sides of the device.

The microphone is soldered underneath the extension board, with a hole through the PCB pointing outwards. This is because the air vent of the microphone is located below the component.

5.2.6 Supply voltage adaptation

The SD Association states that a new protocol exists for µSD card to operate at 1.8V, named Low Voltage Signaling (LVS) [44]. However, no such LVS card exists yet on the market. The choice is thus for a regular card that is supplied and communicates at 3.3V.

The entire prototype being compatible with that voltage level, only two changes must be made to adapt it. The 2-AAA battery pack is replaced by a larger one that holds three of them. Its nominal voltage of 4.5V is then decreased and stabilised by a 3.3-V LDO. Both components are replaced on two ForestMEv2 devices, #1 and #2, to properly accomodate the BatExtension board.

While this change induces an increase of power consumption, as characterised in 4.2, it also allows more swing for the analog signals.

5.2.7 Software

In this section the configuration of the pins and blocks of the MCU involved in each of the three functions are explained, together with the software architecture used to access the hardware functionalities.

PMU

PA4 is used as a digital IO normally tied low to block the PMOS.

AFE

For the AFE, PA10 and PA11 are also used as digital IO pins to select the gain of the amplification chain.

PA12 is linked to the ADC, which is triggered by a timer at the sampling frequency. The acquired samples are automatically transferred in the RAM by the Direct Memory Access (DMA) without requiring any operation from the CPU and using the double buffering technique. The allocated memory block is divided into two buffers of equal size. Once the first one is fully filled with samples, it is processed by the CPU while the second one is filled. This technique allows to continuously acquire data from the ADC and process it in real time. The STM32WL provides

64kB of RAM. 48kB are used for the two buffers, each storing 12000 samples of 2 bytes each. This gives two blocks each containing a total of 30ms of recording at 400kHz.

µSD

The card detection line is wired to PA0, which is configured as a digital input. Its digital voltage can be read through software at any time. It can also be configured as an external interruption line to interrupt the CPU when a card is inserted or removed.

The sofware to use the µSD card is more complex and can be represented with the same architecture as the one for the sensor and the radio of the main device, as shown in Figure 5.8.

µSD card It is the device that is accessed by the SPI bus through software, in order to read or write data.

HAL These functions provide simple interfaces to simplify the use of the SPI bus. They are called by the upper layers to communicate with the µSD card.

User-defined interface As it is done for the sensor and the radio, this layer makes the bridge between standard functions and the custom hardware, by specifying which SPI bus to use on which pins.

Driver The driver manages the low-level commands for the communication protocol with SD. The source code and its integration in the STM32 project are provided by [45].

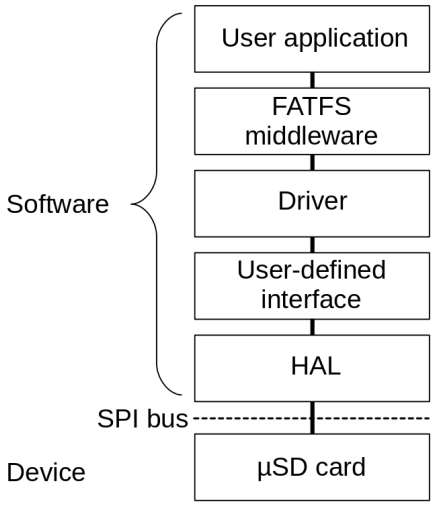


Figure 5.8: µSD software stack.

FATFS middleware External flash drives like µSD cards are rarely used to store raw data. Instead, the classical way to work is by using a standardised file system, such that it can be read by any computer. The FATFS middleware provides structures to manage the simple File Allocation Table (FAT) file system. The memory is thus organised in folders and files, and raw data is written inside named files.

This middleware is integrated in STM32CubeMX and its code is generated and included in the projet by the tool, but requires the lower software layers already explained to operate with a of custom hardware.

User application In the current user application, a file is created and written to each time the AFE is enabled, to record all the acoustic signal captured.

5.3 Performances

In order to maximise the reachable SPI clock frequency, as well as the rate of processing operations, the MCU main clock frequency is increased to its maximum value, 48MHz.

The main performances in terms of power consumption and ability to capture data are assessed and compared to the initial specifications of the extension board.

Then, different ideas of future improvements are stated.

	Baseline	AFE	µSD write
Total power consumption [mW]	20.62	29.11	195.36
Block power consumption [mW]	20.62	8.48	166.25

Table 5.2: Power consumption of the extension board, for different modes of operation.

5.3.1 Power consumption

The power consumption measurements are conducted using the same experimental setup as the one from Section 4.2.

This time, because the MCU clock is higher, the base consumption, i.e. with the MCU running but the extension board powered off by its PMU, reaches a value of 20.62mW. This baseline value can be substracted from subsequent measurements to isolate and evaluate the consumption of the extension board only.

Two cases are distinguished. The first one is when the extension board is powered on. In that case, the µSD card is in sleep mode and its consumption is negligeable compared to the one of the AFE. In the second situation, data is written on µSD. The power consumption values are displayed in Table 5.2. Substractions are used to compute the consumption of a each individual block.

As expected, the writing operation is the one that consumes the most power. However, its current consumption is lower than the one indicated in the datasheet, with only 50.34mA compared to 100mA [39]. Two explanations can justify this result. First, the datasheet is rather old and doesn't exactly correspond to the µSD card model used. New technologies certainly help at improving the energy efficiency of flash memories. Secondly, as discussed in the following section, the writing datarate is much lower than the maximum one, meaning that there is probably room for a faster writing operation that would lead to higher power consumption. But the power is not necessarily proportional to the datarate, as the µSD card operates by erasing and writing entire blocks of memory at a time.

5.3.2 Data capture

Even if all the different functional blocks work as expected, i.e. the AFE correctly transduces acoustic waves which are digitised and written on the µSD card, the most important specification could not be reached. Indeed, the current maximum sampling frequency is 10kHz instead of 400kHz. That allows to capture a signal up to a frequency of 5kHz which doesn't even include the low-frequency content of the acoustic signals of bats.

The bottleneck is the writing operation on the µSD card. Indeed, even if the SPI clock of 24MHz theoretically allows to transfer up to 3GB of data per second, the actual writing speed cannot exceed 20kB/s, which is far below the 600kB/s required.

5.3.3 Future improvements

Three directions of future improvements can be identified at the end of this chapter.

- The most important shortcoming to overcome is obviously the µSD card writing speed. It can be improved by working at two different levels.

  The easiest solution is to use a more performant µSD card. Indeed, they are classified according to their speed, and the one used in this work belongs to the C10 category, which is supposed to be able to write at 10MB/s. Even if there seems to be a large amount of margin in the performances reachable with this specific card, perhaps the use of a more advanced one can help at writing faster.

  The other layer at which work is needed is the driver software, which is currently taken as is from [45]. While it seems to perform a decent job at writing and reading the card, no detailed investigation is performed about the way it works. It could certainly be optimised and enhanced to improve the writing speed.

- The experimental characterisation of the AFE would bring valuable information needed to assess the quality of the recordings that this device is able to capture. It would include the measurement of the transfer function of the microphone up to high frequencies, together with the one of the amplifying and filtering chain.

- Finally, the software developed for this work is basic and only provides experimental features. A real application would need a much more complex software, with processing features like signal energy threshold detection or power consumption optimisations.

Conclusion

This master thesis proposes a new prototype of a WSN end node, primarly designed for experimental purposes. The knowledge generated by the different characterisations and experiments conducted give a better understanding of the constraints and limitations of such a network for natural-environment monitoring, and most particularly in a forest environment. Through this, it adresses the current challenges that would benefit from a more extensive monitoring, such as the protection of an environment which is endangered by uncontrolled forest fires.

First, the preliminary study of the needs assesses the different constraints linked to the three needed functionalities of the prototype : sensing, processing and communicating. Then, an informed choice is made for the best suited solution. It is a single custom PCB that uses the STM32WL chip which includes both the MCU and a modified SX1276 radio.

After an in-depth thinking of the components, the electrical circuit as well as the layout, the designed solution is a mechanically robust and compact prototype with dimensions of $7.5 \times 4 \times 2.4$ cm and a price of 75€ per piece. The development of an adapted software shows that every hardware feature works as expected on the first try and that the six ForestMEv2 devices are suited to conduct experiments and characterisations.

The first of them shows that an intensive use of the radio before sensing is not recommended. Indeed, the current consumed by the radio slowly heats up the sensor and can shift the measured temperature by 1°C and the relative humidity by 3%.

Power consumption measurements indicate that when operating the device at low supply voltage, i.e. 1.8V, it consumes less power than at 3.3V. In that case, the energy required to send 16 bytes of data varies between 3.45mJ and 87.58mJ for SF7 to SF12 respectively. To capture the four quantities with the sensor, i.e. the temperature, the relative humdity, the pressure and the gas resistance, it requires 1.49mJ.

The ForestMEv2 is perfectly suited for the detection of fires in a real forest environment. Indeed, the gas resistance quickly and unambiguously reacts in the presence of smoke by decreasing to at least 50% of its initial value. However, it requires to be in direct contact of the smoke. It thus needs to be set up in the wind's direction and at the right height. The temperature also reacts by increasing from 5 to 10°C when the sensor is at short distance and in line of sight of flames.

The last set of experiments assesses the communication performances in a laboratory as well as in a real forest environment. It shows that because of the link between SNR and SF and the high RF attenuation of that particular environment, trading datarate, thus energy, for link budget, thus range, is not worth it. Indeed, for each doubling of the energy required to send a packet, the range benefit is at most +7.6%.

The extension board designed for bat monitoring validates the modularity and versatility features of the prototype, by adding an analog sensing chain and an external storage system to the initial device. However, the low writing speed of 20kB/s makes it unusable without further improvement.

Finally, the prototype designed in this work already brings some new and valuable information, but it also opens a lot of possibilities.

This project could be followed by the development of a fully deployable prototype, with a robust casing to face all types of weather, an applicative software and optimisations on the power consumption level. It could also embed an energy harvesting system to extend its life span even further.

Another way of improvements is to further work on the characterisation of the existing features, by analysing the impacts of the different parameters of the sensor, or the underlying effects behind the path loss in a forest and the relation between SF and SNR.

The modularity of the prototype could be exploited by improving the existing extension board, or even create a new purpose and make another extension.

That only represents some of the possibilities that this work brings on the table and that could serve as a basis for future works.

References

[1] Wikipedia contributors, *2020 California wildfires — Wikipedia, The Free Encyclopedia*, [Online; accessed 7-April-2023], 2023. [Online]. Available: https://en.wikipedia.org/wiki/2020_California_wildfires.

[2] Wikipedia contributors, *2019–20 Australian bushfire season — Wikipedia, The Free Encyclopedia*, [Online; accessed 7-April-2023], 2023. [Online]. Available: https://en.wikipedia.org/wiki/2019%E2%80%9320_Australian_bushfire_season.

[3] R. Bailey and J. Yeo, "The Burning Issue : Managing wildfire risk," *MarshMcLennan*, 2019.

[4] A. Ahmad A. A.author = Ahmad A. A. Alkhatib, "A Review on Forest Fire Detection Techniques," *International Journal of Distributed Sensor Networks*, vol. 2014, no. 597368, p. 12, Mar. 2014. DOI: http://dx.doi.org/10.1155/2014/597368.

[5] T. G. Roberts. "Popular Orbits 101." (2017), [Online]. Available: https://aerospace.csis.org/aerospace101/earth-orbit-101/.

[6] D. Kandris, C. Nakas, D. Vomvas, and G. Koulouras, "Applications of Wireless Sensor Networks: An Up-to-Date Survey," *Applied System Innovation*, vol. 3, no. 1, 2020, ISSN: 2571-5577. DOI: 10.3390/asi3010014. [Online]. Available: https://www.mdpi.com/2571-5577/3/1/14.

[7] P. Maistriaux, T. Pirson, M. Schramme, J. Louveaux, and D. Bol, "Modeling the Carbon Footprint of Battery-Powered IoT Sensor Nodes for Environmental-Monitoring Applications," *IoT '22: Proceedings of the 12th International Conference on the Internet of Things*, pp. 9–16, Nov. 2022. DOI: https://doi.org/10.1145/3567445.3567448.

[8] T. Pirson and D. Bol, "Assessing the embodied carbon footprint of iot edge devices with a bottom-up life-cycle approach," *Journal of Cleaner Production*, vol. 322, p. 128 966, 2021, ISSN: 0959-6526. DOI: https://doi.org/10.1016/j.jclepro.2021.128966. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0959652621031577.

[9] G. Jones, D. Jacobs, T. Kunz, M. Willig, and P. Racey, "Carpe noctem: the importance of bats as bioindicators.," *Endang Species Res*, vol. 8, pp. 93–115, 2009. DOI: https://doi.org/10.3354/esr00182.

[10] P. Stahlschmidt and C. A. Brühl, "Bats as bioindicators – the need of a standardized method for acoustic bat activity surveys," *Methods in Ecology and Evolution*, vol. 3, no. 3, pp. 503–508, 2012. DOI: https://doi.org/10.1111/j.2041-210X.2012.00188.x. eprint: https://besjournals.onlinelibrary.wiley.com/doi/pdf/10.1111/j.2041-210X.2012.00188.x. [Online]. Available: https://besjournals.onlinelibrary.wiley.com/doi/abs/10.1111/j.2041-210X.2012.00188.x.

[11] *Low power gas, pressure, temperature and humidity sensor*, BME680, Rev. 1.0, Bosch, 2017.

[12] P. Ciccioli, M. Centritto, and F. Loreto, "Biogenic volatile organic compound emissions from vegetation fires," *Plant, Cell & Environment*, vol. 37, no. 8, pp. 1810–1825, 2014. DOI: https://doi.org/10.1111/pce.12336.

[13] *Ultra-low Power Apollo MCU Family*, Apollo3 Blue MCU, Rev. 1.0.2, Ambiq, Feb 2022.

[14] C. E. Van Wagner, *Development and structure of the Canadian Forest Fire Weather Index System.* O. Canadian Forestry Service Headquarters, Ed. Forestry Technical Report, 1987.

[15] A. K. Ramanathan, L. M. Headings, and M. J. Dapino, "Flexible piezo-capacitive pressure sensor for measurement of wind speed and direction," in *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2022*, D. Zonta, B. Glisic, and Z. Su, Eds., International Society for Optics and Photonics, vol. 12046, SPIE, 2022, p. 1 204 607. DOI: 10.1117/12.2614329. [Online]. Available: https://doi.org/10.1117/12.2614329.

[16] NWCG. "Fire Weather Index (FWI) System." (2021), [Online]. Available: https://www.nwcg.gov/publications/pms437/cffdrs/fire-weather-index-system.

[17] *137 MHz to 1020 MHz Low Power Long Range Transceiver*, SX1276/77/78/79, Rev. 7, Semtech Corporation, May 2020.

[18] Wikipedia contributors, *LoRa — Wikipedia, The Free Encyclopedia*, [Online; accessed 14-Mai-2023], 2023. [Online]. Available: https://en.wikipedia.org/wiki/LoRa.

[19] LoRa. "Rules and regulations." (2018), [Online]. Available: https://lora.readthedocs.io/en/latest/#rules-and-regulations.

[20] A. Augustin, J. Yi, T. H. Clausen, and W. Townsley, "A Study of LoRa: Long Range & Low Power Networks for the Internet of Things," *Sensors*, vol. 16, p. 1466, Oct. 2016. DOI: 10.3390/s16091466.

[21] LoRa Alliance. "LoRaWAN." (2023), [Online]. Available: https://lora-alliance.org/.

[22] ARM. "Keil µVision." [Accessed 15-Mai-2023]. (), [Online]. Available: https://www.keil.com/.

[23] *Low Power Long Range Transceiver Module*, RFM95/96/97/98(W), V1.0, HopeRF Electronic.

[24] Ambiq. "Apollo3." (2023), [Online]. Available: https://ambiq.com/apollo3/.

[25] *Nanopower IQ, 200-nA, 200-mA, Low-Dropout Voltage Regulator With Fast Transient Response*, TPS7A03, Texas Instruments, September 2022.

[26] *Multiprotocol LPWAN 32-bit Arm® Cortex®-M4 MCUs, LoRa®, (G)FSK, (G)MSK, BPSK, up to 256KB flash, 64KB SRAM*, STM32WLE5xx STM32WLE4xx, DS13105 Rev 11, STMicroelectronics, October 2022.

[27] *STM32WL UFQFPN48 Reference board*, Based on MB1720, Rev. A, STMicroelectronics, 21-Dec-20.

[28] *STLINK-V3MINIE debugger/programmer tiny probe for STM32 microcontrollers*, DB4541, Rev 2, STMicroelectronics, December 2021.

[29] The Things Industries. "The Things Network." (2023), [Online]. Available: https://www.thethingsnetwork.org/.

[30] FACT Training (Protection Unit). "Prévention incendie." (2023), [Online]. Available: https://training.protectionunit.com/fr/formations/prevention-incendie.

[31] M. Xhonneux, O. Afisiadis, D. Bol, and J. Louveaux, "A Low-Complexity LoRa Synchronization Algorithm Robust to Sampling Time Offsets.," in *IEEE Internet of Things Journal*, 2021, p. 14. DOI: 10.1109/JIOT.2021.3101002. [Online]. Available: http://hdl.handle.net/2078.1/251518.

[32] T. Rama Rao, D. Balachander, and N. Tiwari, "UHF short-range pathloss measurements in forest & plantation environments for wireless sensor networks," in *2012 IEEE International Conference on Communication Systems (ICCS)*, 2012, pp. 194–198. DOI: 10.1109/ICCS.2012.6406137.

[33] L. Dierckx, M. Beauvois, and S. Nijssen, "Detection and multi-label classification of bats," in *Advances in Intelligent Data Analysis XX*, T. Bouadi, E. Fromont, and E. Hüllermeier, Eds., Cham: Springer International Publishing, 2022, pp. 53–65, ISBN: 978-3-031-01333-1.

[34] M. Beauvois and L. Dierckx, "Automated detection of bat species in Belgium.," Prom. : O. Bonaventure, S. Nijssen, 2021. [Online]. Available: http://hdl.handle.net/2078.1/thesis:30594.

[35] M. Franco and C. Lipani, "Automated monitoring of bat species in Belgium.," Prom. : O. Bonaventure, S. Nijssen, 2020. [Online]. Available: http://hdl.handle.net/2078.1/thesis:25168.

[36] G. Chaverri, L. Ancillotto, and D. Russo, "Social communication in bats," *Biological Reviews*, vol. 93, no. 4, pp. 1938–1954, 2018. DOI: https://doi.org/10.1111/brv.12427. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/brv.12427. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/brv.12427.

[37] J.-D. Vrignault. "Pibatrecorder – teensyrecorders." (2018), [Online]. Available: https://framagit.org/PiBatRecorderPojects/TeensyRecorders.

[38] Maryland Department of Natural Resources. "Bat Echolocation." (2023), [Online]. Available: https://dnr.maryland.gov/wildlife/Pages/plants_wildlife/bats/batelocu.aspx.

[39] *SanDisk microSD, microSDHC and microSDXC*, 80-36-03335, Rev. 2.6, SanDisk, July 2012.

[40] *Zero-Height SiSonic Microphone*, SPU0410LR5H-QB, Rev. H, Knowles, 2013.

[41] *10-MHz, Rail-to-Rail Input/Output, Low-Voltage, 1.8-V CMOS Operational Amplifiers*, TLVx316, SBOS752A, Texas Instruments, 2016.

[42] *Single 5-Ohm SP3T Analog Switch 5-V/3.3-V 3:1 Multiplexer/Demultiplexer*, TS5A3357, SCDS177B, Texas Instruments, 2018.

[43] *P-Channel Enhancement Mode MOSFET – DMP2130L*, DS31346, Rev. 8-2, Diodes Incorporated, 2019.

[44] SD Association. "Low Voltage Signaling." (2023), [Online]. Available: https://www.sdcard.org/developers/sd-standard-overview/low-voltage-signaling/.

[45] H. Pearce (kiwih). "Tutorial: An SD card over SPI using STM32CubeIDE and FatFS." (2020), [Online]. Available: https://01001000.xyz/2020-08-09-Tutorial-STM32CubeIDE-SD-card/.

Appendix A

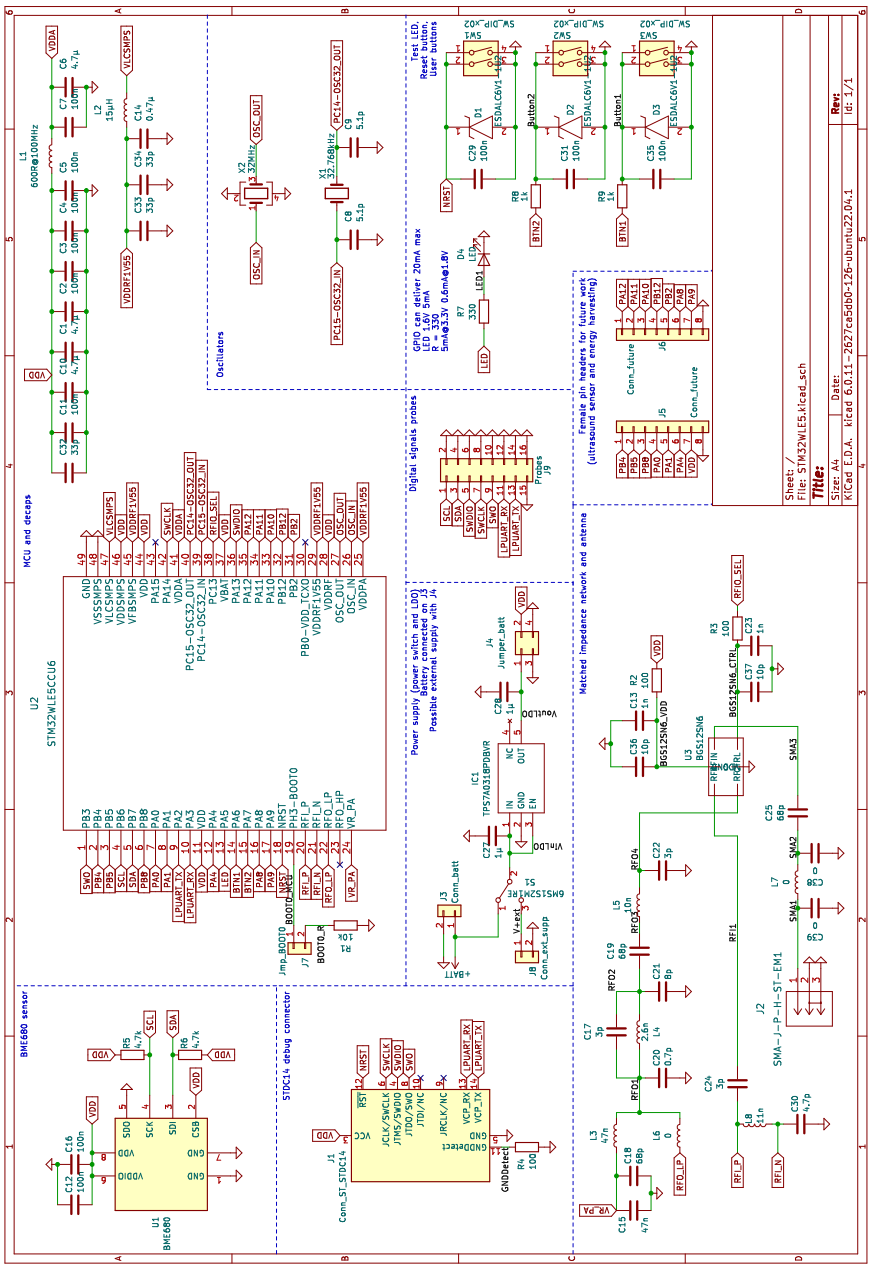
Price sources of the components of the BOMs

In order to compare the price of the six different possible solutions, the BOMs are computed by summing the price of the main components. All the remaining parts are aggregated in the category *Others*, which is evaluated at 25€ *a posteriori*, i.e. it is the actual cost of those components for the developed solution.

Component	**Price [€]**	**Link**
BME680 EVB	22.21	https://www.digikey.at/en/products/detail/pimoroni-ltd/PIM357/9356663
BME680	13.96	https://www.digikey.com/en/products/detail/bosch-sensortec/BME680/7401317
RFM95	9.95	https://www.elektor.fr/seeed-studio-rfm95-ultra-long-lora-transceiver-module-eu868
SX1276	9.31	https://www.mouser.be/ProductDetail/Semtech/SX1276IMLTRT?qs=rBWM4%252BvDhIceYQoeMKd%2FQQ%3D%3D
Apollo3	3.53	https://www.ambiq.top/en/apollo3-blue-soc-96-mhz-384kb-bga
Apollo3 EVB	108.07	https://www.ambiq.top/en/apollo3-blue-soc-eval-board
PCB 10x5cm	20	https://www.eurocircuits.com/, for 5 pieces.
STM32L4	10.54	https://www.st.com/en/microcontrollers-microprocessors/stm32l4-series.html#
Breadboard	8.3	https://shop.mchobby.be/fr/fils-prototypage/90-breadboard-grande-taille-3232100000902.html
Nucleo L452	13.72	https://estore.st.com/en/products/evaluation-tools/product-evaluation-tools/mcu-mpu-eval-tools/stm32-mcu-mpu-eval-tools/stm32-nucleo-boards/nucleo-l452re.html
STM32WLE5	10.87	https://estore.st.com/en/catalogsearch/result/?q=STM32WLE5CC

Appendix B

Complete design schematic



Appendix C

Bill of materials

Reference	Value	Component	Manufacturer
C1, C6, C10,	4.7µ	GRM155R60J475ME47D	Murata
C2, C3, C4, C5, C7, C11, C12, C16, C29, C31, C35,	100n	GRM155R61H104KE19D	Murata
C8, C9,	5.1p	GJM1555C1H5R1WB01D	Murata
C13, C23,	1n	GRM1555C1H102JA01D	Murata
C14,	0.47µ	C1005X5R1E474K050BE	TDK
C15,	47n	GRM155C71H473KE19D	Murata
C17, C22, C24,	3p	GJM1555C1H3R0WB01D	Murata
C18, C19, C25,	68p	GRM1555C1E680JA01D	Murata
C20,	0.7p	GJM1555C1HR70WB01D	Murata
C21,	8p	GJM1555C1H8R0WB01D	Murata
C27, C28,	1µ	GRM155R61E105KA12D	Murata
C30,	4.7p	GJM1555C1H4R7WB01D	Murata
C32, C33, C34,	33p	04025A330JAT2A	KYOCERA AVX
C36, C37,	10p	04025A100JAT2A	KYOCERA AVX
D1, D2, D3,	ESDALC6V1-1U2	ESDALC6V1-1U2	STMICROELECTRONICS
D4,	LED	5988A10107F	DIALIGHT
IC1,	TPS7A0318PDBVR	TPS7A0318PDBVR	Texas Instruments
J1,	Conn_ST_STDC14	FTSH-107-01-L-DV-K-A	SAMTEC
J2,	Conn_Coaxial	SMA-J-P-H-ST-EM1	SAMTEC
J4,	Jumper_batt	90131-0122	MOLEX
J5, J6,	Conn_future	2212S-08SG-85	MULTICOMP PRO
J7,	Jmp_BOOT0	22-03-2021	MOLEX
J8,	Conn_ext_supp	22-03-2021	MOLEX
J9,	Probes	2213S-16G	MULTICOMP PRO
L1,	600R@100MHz	BLM18AG601SN1D	Murata
L2,	15µH	LQM21DN150M70L	Murata
L3,	47n	LQW15AN47NG00D	Murata
L4,	2.6n	LQW15AN2N6B80D	Murata
L5,	10n	LQW15AN10NG80D	Murata
L6, L7,	0	AC0402FR-070RL	YAGEO
L8,	11n	LQW15AN11NG80D	Murata
R1,	10k	RC0402FR-0710KL	YAGEO
R2, R3, R4,	100	MCWR04X1000FTL	MULTICOMP PRO
R5, R6,	4.7k	MCMR04X4701FTL	MULTICOMP PRO
R7,	470	MCWR04X4700FTL	MULTICOMP PRO
R8, R9,	1k	MCWR04X1001FTL	MULTICOMP PRO

S1,	6MS1S2M1RE	6MS1S2M1RE	MULTICOMP PRO
SW1, SW2, SW3,	SW_Push	B3F-1020	OMRON
U1,	BME680	BME680	Bosch Sensortec
U2,	STM32WLE5CCU6	STM32WLE5CCU6	STMICROELECTRONICS
U3,	BGS12SN6	BGS12SN6E6327XTSA1	INFINEON
X1,	32.768kHz	NX2012SA-32.768K-STD-MUB-1	NDK
X2,	32MHz	NX2016SA-32M-EXS00A-CS06465	NDK
	Antenna	TI.08.C.0111	Taoglas
	Battery holder	BH2AAAW	Hammond Manufacturing
	Debugger	STLINK-V3MINIE	STMICROELECTRONICS

Appendix D

Complete view of the layout

All the manufactured layers are displayed in Figure D.1. The bottom copper is in blue and the top copper in red.

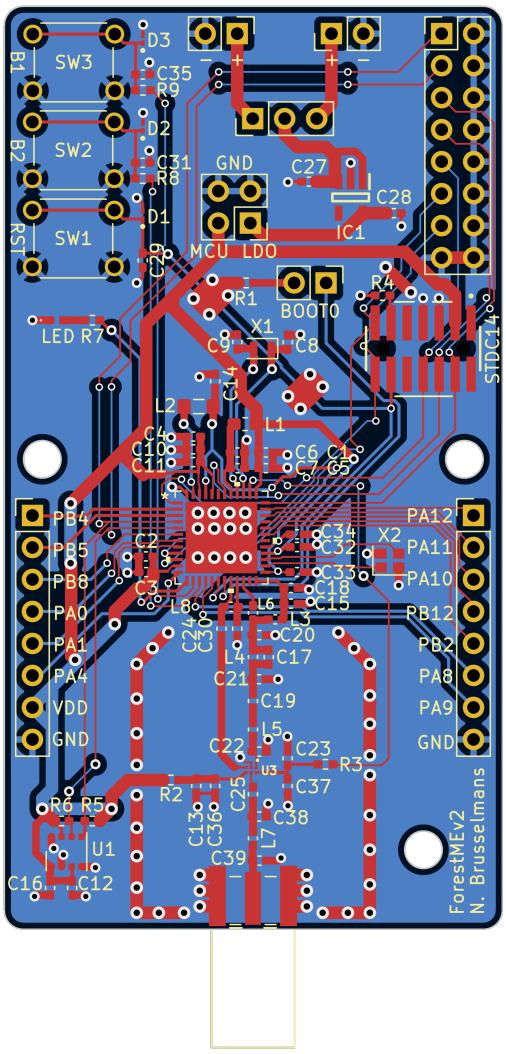


Figure D.1: Complete view of the layout.

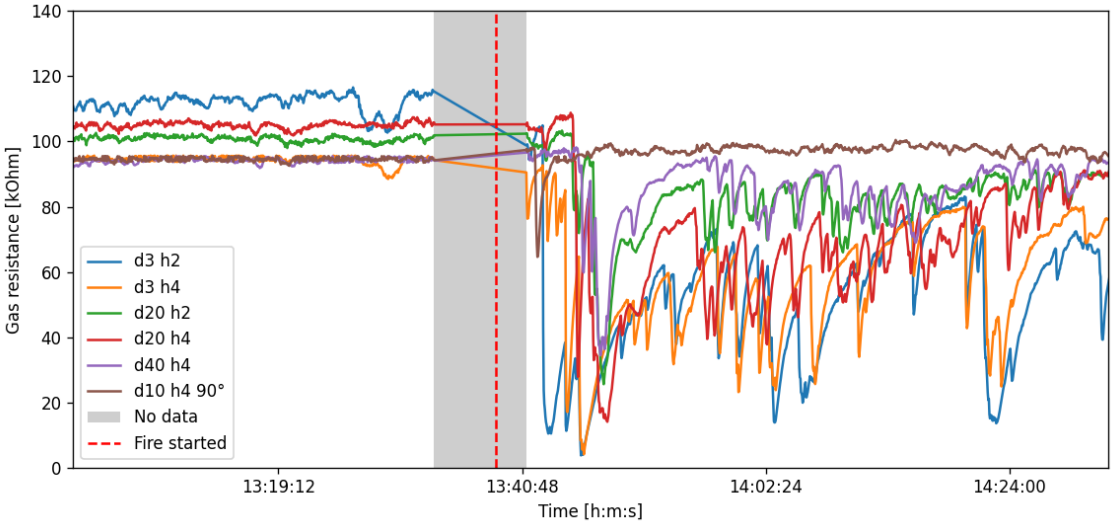
Appendix E

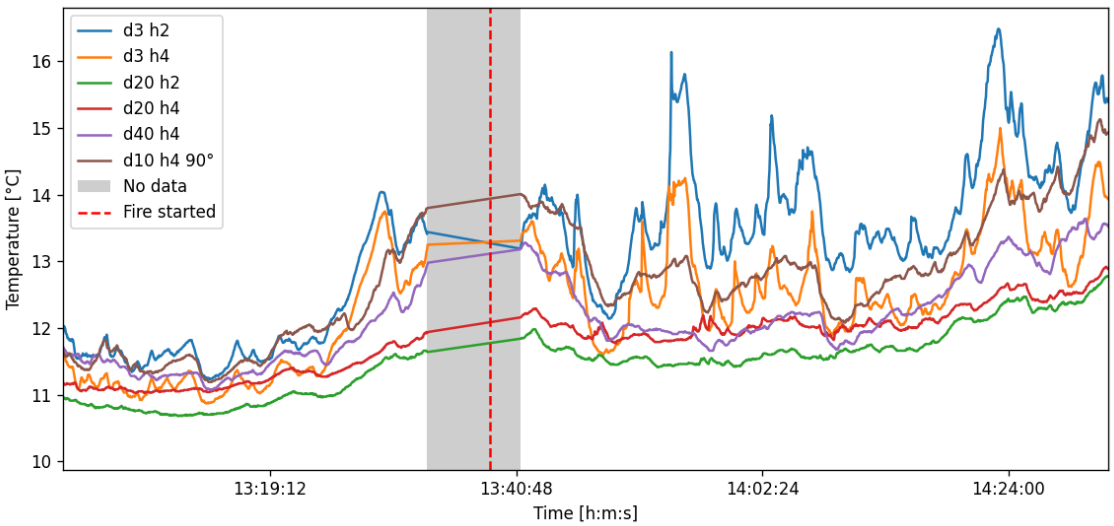
Complete captured data during a fire in a forest environment

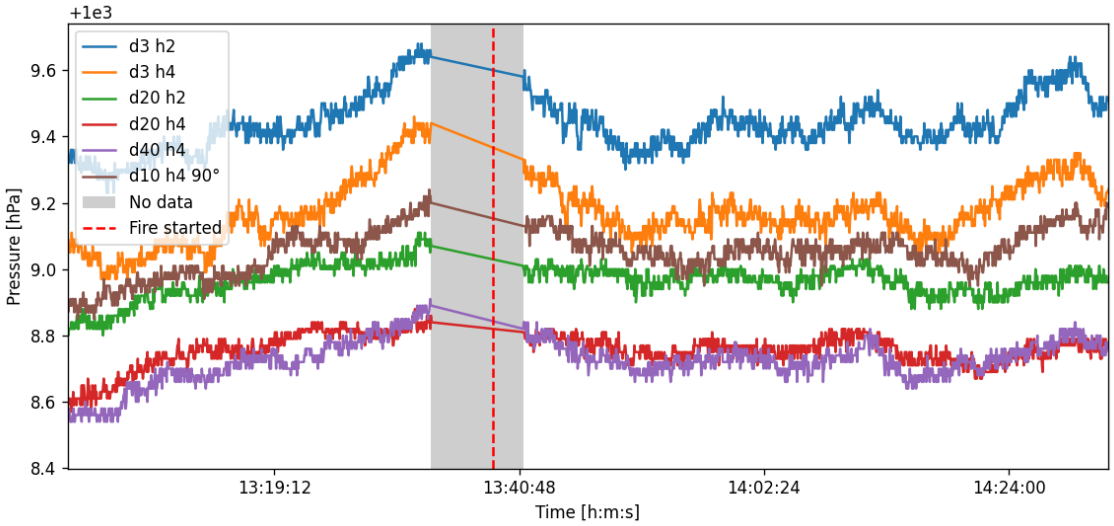
The graph of the gas resistance is described and analysed in details in Section 4.3.2.

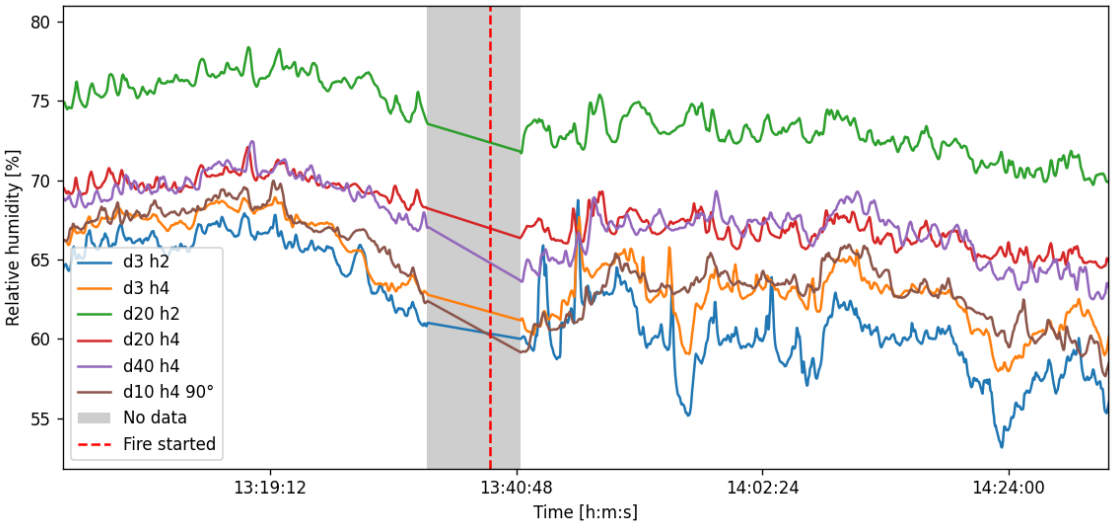
In Section 4.3.1, it is explained that the pressure and relative humidity don't give any useful information for the detection of forest fire. Those quantities are used to compute the FWI.

The temperature can give an information about the proximity of a fire by capturing the radiated heat, as also explained in Section 4.3.1. On this graph, the two sensors set up close to the fireplace measure rises of temperature when the flames are large or when hot smoke reaches them.




d3 h2
d3 h4
d20 h2
d20 h4
d40 h4
d10 h4 90°
No data
Fire started
Temperature [°C]
16
15
14
13
12
11
10
13:19:12
13:40:48
14:02:24
14:24:00
Time [h:m:s]


+1e3
d3 h2
d3 h4
d20 h2
d20 h4
d40 h4
d10 h4 90°
No data
Fire started
Pressure [hPa]
9.6
9.4
9.2
9.0
8.8
8.6
8.4
13:19:12
13:40:48
14:02:24
14:24:00
Time [h:m:s]


d3 h2
d3 h4
d20 h2
d20 h4
d40 h4
d10 h4 90°
No data
Fire started
Relative humidity [%]
80
75
70
65
60
55
13:19:12
13:40:48
14:02:24
14:24:00
Time [h:m:s]

Appendix F

Extension for bat monitoring : schematic, layout and BOM

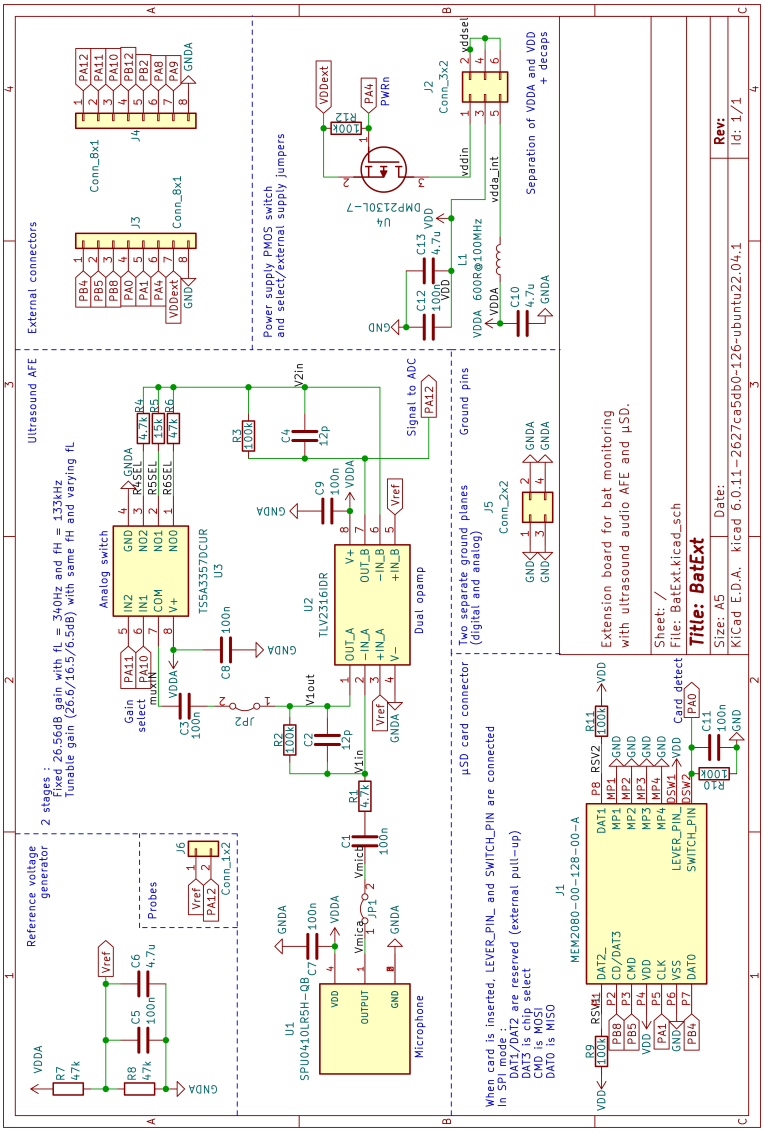


Figure F.1: Complete schematic of the extension board.

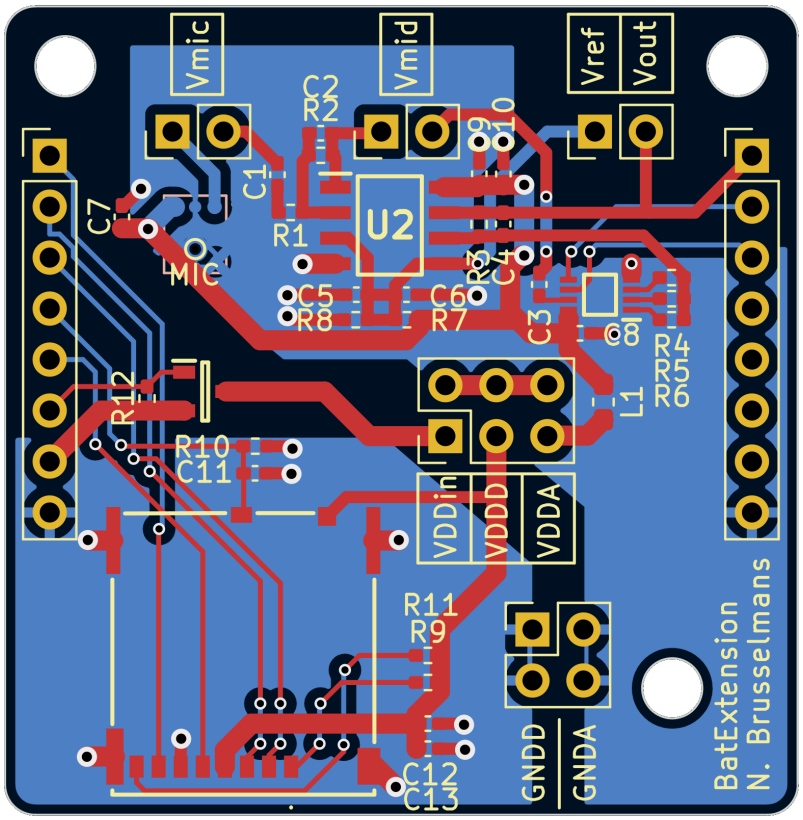


Figure F.2: Layout of the extension board.

Reference	Value	Mouser reference
C1, C3, C5, C7, C8, C9, C11, C12	100n	581-04025C104KAT4A
C2, C4	12p	581-04025A120F4T2A
C6, C10, C13	4.7u	581-0402YD475MAT2A
J1	MEM2080-00-128-00-A	640-MEM2080-0012800A
J2	Conn_3x2	649-1012938190601BLF
J3, J4	Conn_8x1	474-PRT-09279
J5	Conn_2x2	649-1012938190403BLF
J6, JP1, JP2	Conn_1x2	649-1012937890201BLF
L1	600R@100MHz	BLM18AG601SN1D
R1, R4	4.7k	MCMR04X4701FTL
R2, R3, R9, R10, R11, R12	100k	603-RC0402FR-07100KL
R5	15k	279-CRGCQ0402F15K
R6, R7, R8	47k	603-RC0402FR-0747KL
U1	SPU0410LR5H-QB	721-SPU0410LR5H-QB
U2	TLV2316IDR	595-TLV2316IDR
U3	TS5A3357DCUR	595-TS5A3357DCUR
U4	DMP2130L-7	621-DMP2130L-7
	Batt_Holder_3xAAA	546-BH3AAAW
	LDO_3.3V	595-TPS7A0333PDBVR
	Jumper	710-60900213521

Table F.1: Bill of material of the extension board. The components are referenced according to their Mouser part number.

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain
Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | **www.uclouvain.be/epl**