POLITECNICO DI TORINO

MASTER's Degree in ELECTRONIC ENGINEERING

MASTER's Degree Thesis

Wireless Measuring System for Freeze Dryers and Harsh Environments

Supervisor

Prof. Simone CORBELLINI

Candidate

Andrea MACCARIO

December 2023

In the 19^{th} *century Thomas Seebeck observed that when two dissimilar metals were joined at their ends and there was a temperature difference between the two junctions, an electric current would flow. This important discovery brought to the development of thermocouples, rugged temperature sensors, extensively used these days in challenging environments.*

Summary

Temperature measurement is nowadays fundamental for scientific research and in modern industry. It is crucial to be able to measure temperature in different environments, including the most challenging ones. As an example, industrial lyophilization requires reliable temperature monitoring under harsh conditions. Lyophilization, also known as freeze drying is the process that is used to remove water from biological and/or pharmaceutical products in order to improve their shelf-life while preserving as much as possible the original characteristics. The process takes place inside an isolated stainless steel container at low pressure and low temperature in three different phases called freezing, primary drying and secondary drying. During the entire process the temperature of the product, but also the temperature of the shelves, has to be accurately monitored in order to have satisfactory outcome at the process conclusion. There are different factors that contribute to make the lyophilization process challenging for a measuring system. The primary of them emerges from the extremely low temperatures experienced during the freezing phase, reaching as low as $-50\,^{\circ}\mathrm{C}$. In addition every component introduced inside the freeze dryer must undergo sterilization, in order to prevent the introduction of unwanted microorganisms. Furthermore, the requirement of vacuum for the correct physical transformations makes difficult to introduce cables into the chamber. Thus, the sensing equipment must rely on battery power and necessitates a long-range radio capable of transmitting data from within the freeze dryer to a central receiver positioned potentially tens of meters away. Another harsh environment where the deployment of a measuring system requires particular care is the agriculture domain. In this field extensive monitoring of temperature and humidity is very important therefore the sensing equipment has to be sufficiently rugged to withstand bad weather conditions, high level of humidity and a significant dust exposure. There are a lot of different applications in this sector that require a measurement system implementing wireless data transmission. A first example is represented by greenhouses where temperature and humidity, are the most important parameters that have to be maintained under control to successfully grow crops. Another example is related to the temperature monitoring of the soil which plays a fundamental role in agricultural practices. In

this context the thesis work focuses on developing an affordable, accurate and low power consumption wireless device on a compact printed circuit board (PCB) for temperature monitoring inside freeze dryers but designed to extend its applications to other harsh environments and physical quantities. This is done by creating a system, with a modular architecture where most of the components can be reused for different applications and different types of analog sensors. In particular the system is battery powered with long range wireless data transmission and a data logger functionality implemented by a non volatile flash memory, increasing the reliability of the system in the case of failure during the radio propagation. It performs temperature measurements by means of six thermocouples due to their robustness, small uncertainty and ease to fit inside vials. It incorporates a GPS, even though it is not exploited in the case of the lyophilization, it becomes fundamental when the device is used for environmental monitoring in large areas where it is required to know its position. Moreover, the PCB is designed to be resin coated to withstand sterilization and harsh environments, however after the resin solidification, the board components are no longer accessible therefore reset functionality, battery charging and microcontroller's software update have to be considered properly. Furthermore, given that the wireless device operates on battery, it is fundamental to minimize current consumption through the implementation of a low-power approach by means of both hardware and software techniques. The thesis starts illustrating all the constrains that are typically found inside freeze dryers which define the main characteristics of the measuring system. Subsequently it moves to the definition of the architecture that is used motivating the choices made in terms of microcontroller and various components. Following the hardware description, the thesis delves into the software analysis, covering the firmware running on the microcontroller and the description of the two applications developed to support the user interaction with the device. The first one for managing the configuration phase while the second one for receiving and visualizing the measurements. The final part is reserved to the metrological characterization and experimental verification performed to validate the device's functionalities. The tests have shown both a satisfactory accuracy from all the six thermocouples and a remarkably low current consumption, less than $4\,\mu$A during the sleep state. Considering a Lithium battery with a capacity of 400 mAh, the device is expected to operate for a few years in low power mode.

Table of Contents

List of Tables

List of Figures

Acronyms

PCB
Printed Circuit Board

GPS
Global Position Satellite

BLE
Bluetooth Low Energy

SPI
Serial Peripheral Interface

I2C
Inter-Integrated Circuit

IC
Integrated Circuit

CMOS
Complementary Metal Oxide Semiconductor

BJT
Bipolar Junction Transistor

MOSFET
Metal Oxide Field Effect Transistor

PTAT
Proportional To Absolute Temperature

RTD

Resistive Temperature Detector

PTC

Positive Temperature Coefficient

NTC

Negative temperature Coefficient

PLC

Programmable Logic Controller

PGA

Programmable Gain Amplifier

PCM

Pulse Code Modulation

SAR

Successive Approximation Register

GPIO

General Purpose Input Output

ASCII

American Standard Code for Information Interchange

IDE

Integrated Development Environment

GUI

Graphical User Interface

TUE

Total Unadjasted Error

SMD

Surface Mount Device

Chapter 1

Temperature monitoring in lyophilization process

The goal of this thesis is to design, build, and characterize a wireless sensor containing multiple temperature sensing probes. The device has to be capable of real-time temperature measurements during lyophilization processes, but its architecture has to be sufficiently modular to allow the extension of its applications to other challenging environments and physical quantities. This chapter provides a brief overview of the lyophilization process as well as its associated challenges, then it discusses the proposed solution.

1.1 Overview on the lyophilization process

Lyophilization is the process that exploits sublimation, that is the direct transition of a substance from solid to gas, in order to extract up to 99 % of water from biological and/or pharmaceutical product (e.g. proteins, peptides, vaccines, etc.). The primary objective of lyophilization is to increase the shelf-life of a particular product, especially when it needs to be stored or transported for a long time. Among the dehydration methods, it is the one that provides better results in terms of product integrity mainly due to the low temperature involved. The process takes place inside an isolated stainless steel chamber equipped with a vacuum system that lowers the pressure during the drying process to few pascal. The chamber has shelves that are loaded with products and a specific fluid that flows inside them to transfer the heat required for the sublimation. Residual water vapour is then trapped using refrigerated plates called process condenser. An example of a freeze drying machine is reported is Figure 1.1. The entire process is characterized by three different phases:

- Freezing
- Primary drying
- Secondary drying

During the first phase the temperature is slowly decreased so that all the water crystallizes as ice. Low temperature is mandatory since the sublimation can take place only if the water is below its triple point that is the temperature for which the three phases of the substance coexist. Figure 1.2 reports an example of phase diagram that clarifies the concept of triple point. An accurate temperature monitoring is fundamental during this phase in fact, it is not possible to lower the temperature too fast otherwise the integrity and properties of the product are not maintained. In the following step called primary drying, the pressure is reduced to a value of 5 to 10 pascal while the temperature of the shelves is increased to allow the sublimation to start. Following the initial phase of gradually reaching the desired shelf temperature, the heat provided by the shelves and the heat removed due to sublimation reach an equilibrium, bringing the system to a stable state. At this point based on the shelf temperature, pressure, and sublimation rate, the product will acquire a specific product temperature. This temperature becomes crucial for the successful implementation of the freeze-drying recipe. When these parameters are not controlled correctly, the product can shrink or collapse and of course this condition must be avoided during the normal operation of the freeze dryer.

Figure 1.1: Industrial freeze dryer

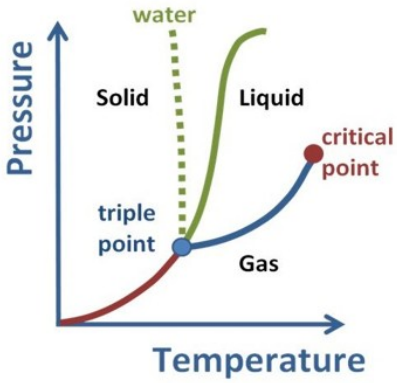


Figure 1.2: Water phase diagram

At the end of this phase up to 95 % of the initial water is removed. During the last phase the temperature of the shelves is further increased even to values above 0 °C to remove unfrozen water molecules mainly by desorption and diffusion. In

fact an higher temperature is necessary to break the remaining bonds between the water molecules and the frozen material. At the end of the entire process, which can last up to some days (depending on the recipe), the water content inside the product is in the range of 1-4 %. Consequently the product is ready to be stored and it can be re-hydrated as soon as it is needed.

1.2 Measurement system challenges

The freeze dryer environment defines some constrains that have to be accurately considered during the design phase of the measurement system. Below they are briefly discussed:

- **Vacuum chamber**: one initial challenge is represented by the sealed container necessary to sustain the vacuum environment. In this scenario conventional wire connections are not feasible and even if it would be possible to use air-tight connectors, they are too expensive and difficult to set up. As a result two key features of the system can be defined: data communication necessitates a wireless approach based on radio signals while the energy source must be localized to the sensor with a battery powered approach being the best solution. There is also another possibility for powering the system that uses the same technology involved in Radio Frequency IDentification (RFID) that is called *inductive coupling*. With this technology the energy is transferred by means of an oscillating magnetic field generated by an alternating current flowing inside the transmitter coil. At the receiver coil, the oscillating magnetic field induces an alternating voltage that has to be rectified to generate a DC signal capable of powering all the parts of the system. The solution is quite interesting since it could be used to avoid the use of the battery but it has some significant drawbacks that can not be neglected. The operating range is an example, since for having an optimal power transfer the two coils has to be very close to each other (typically in the range of few centimeters) and with the two axis aligned. This is a limiting factor when the measurement system has to be placed in an industrial freeze dryer that can also be some meters long.

- **Steel walls and presence of water**: when it comes to real-time data transmission, care must be taken during the selection of the radio transceiver since the freeze dryer's stainless steel walls have the potential to shield radio signals and a considerable amount of water could interfere with the propagation of radio waves. For this reason it would be convenient to have a non volatile memory that can save all the measurements for post processing purposes or to retrieve all the data because some of them could get corrupted or lost during the radio propagation.

- **Low temperature and pressure**: a further consideration pertains to the extremely low temperature and pressure conditions to which the system is exposed. As a result, careful component selection is necessary, taking these factors into account.

- **Process duration**: due to the extended duration of the lyophilization process, which can span over more days, care must be taken regarding the system's battery life duration. As a consequence a low power approach is required to reduce the average current consumption as much as possible.

- **Sterilization**: prior to placement within the freeze dryer, all the equipment has to be properly sterilized. There are different techniques that can be adopted for this purpose including the sterilization by heat using low pressure and sub-atmospheric steam, gaseous sterilization by means of ethylene oxide and irradiation. In order to apply one of these methods, the PCB has to undergo resin coating in such a way that no damage occurs to the components. In addition the resin provides a good protection to the PCB making it more robust and water resistant.

1.3 Commercial solutions

Among the different commercial solutions, the one which is closer in terms of characteristics required for temperature mapping inside freeze dryers is called TEMPRIS which stands for Temperature Remote Interrogation System. It is a battery-free solution that incorporates an antenna, a centering piece, the sensor itself and the probe tip as it can be seen in Figure 1.3. The system can be inserted inside a vial without wires that exit from it due to the small dimension giving the possibility of a fully automated sensor loading inside the vials. Since no battery is present, power to the sensor is derived from the activation of a passive transponder by an amplitude-modulated signal within the 2.4 - GHz ISM band generated by an interrogating unit, with evaluation of the back-scatter response. The signal is then demodulated at the transponder with an envelop detector circuit which, in turn, triggers a resonance circuit based on quartz technology. The stimulated resonator will then oscillate at a frequency that depends on the temperature detected. Thus the TEMPRIS system exploits the temperature dependence of the oscillating frequency of a resonator as a temperature sensing element. The sensor provided good results and it is considered to become a valuable tool for cycle development [1]. Despite its positive aspects, the system can still be subjected to some problems, in particular, being a battery free architecture is has to rely on a transmitter that provides the excitation signal for the quartz oscillator. The solution works well when the transmitter and the receiver are sufficiently close but as soon as the

distance increases the power drops dramatically. This factor could prevent the use of the system on large industrial freeze dryers. Furthermore the fact that the TEMPRIS is battery free solution, prevents the possibility to save the readings on a memory, as a consequence in the case of a transmission fail, data are lost. Another consideration is related to the dimension of the sensing device; it is well established that vials with probes exhibit a variation in terms of nucleation and freezing behavior of the solution, with respect to the vials without the probe. This effect typically leads to the formation of fewer and larger ice crystal that reduce the product resistance shortening the drying time. As a consequence, due to the TEMPRIS sensor's notably larger size compared to a standard thermocouple, the potentially negative impact of this effect might be more pronounced.

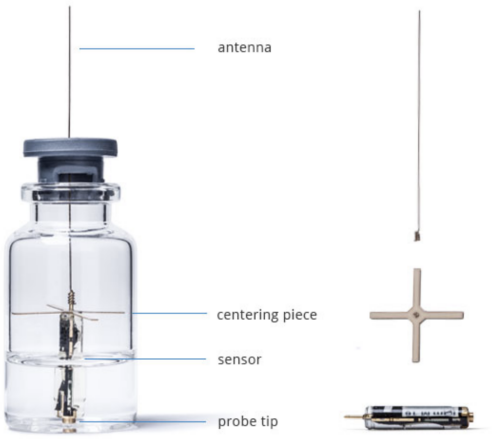


Figure 1.3: TEMPRIS sensor

1.4 Proposed solution

Given the absence of existing commercial solutions with the specified characteristics, the proposed design aims at addressing all the points needed to create a precise, reliable and robust sensor for temperature monitoring inside freeze dryers and difficult environments. Starting from an existing architecture, the project aims to improve it. This involves considering the possibility to resin the entire PCB to make it resistant to harsh environments, adding more temperature sensors, using a new microcontroller, featuring a short range radio for easy configuration of the device, integrating a radio module capable of long-distance transmission, and developing new software for the data visualization, system configuration and data acquisition. Figure 1.4 shows a block diagram of the system

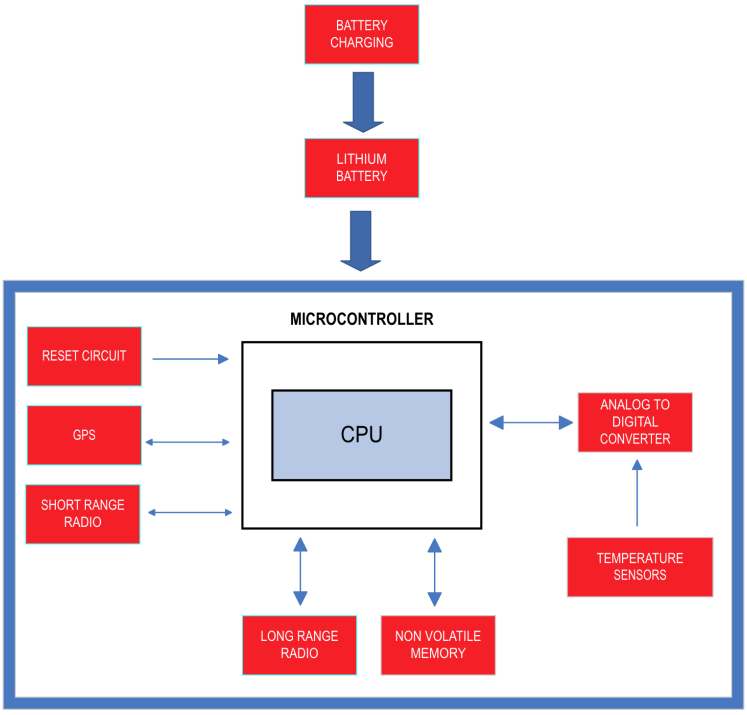


Figure 1.4: System block diagram based on the requirements

The main components are the following:

- **Microcontroller**: it is the brain of the entire system and is used to manage all the tasks, from the data acquisition, manipulation and transmission.

- **Short range radio**: it is the radio used for the configuration of the device

- **Long range radio**: it is the radio that handles the data transmission toward the receiver used for the data collection.
- **Flash memory**: it is used as a data logger to store all the measurements that are contemporary transmitted by the long range radio.
- **GPS**: the GPS module is used to have information about the position of the sensor. This is not intended for the lyophilization process but for other applications where the device is far away from the receiver, it becomes very useful.
- **Battery**: it is used to power the entire system. A Lithium-ion battery is the best choice due to its compact size, good capacity and elevated number of charging cycles.
- **Battery charging**: it is used to recharge the battery when its voltage goes below a certain threshold.
- **Reset circuit**: the reset circuit is the part of the device that manages the reset of the microcontroller in case the radio transmission or the inter-module communication develop some problem. Due to the resin coating that is necessary to protect the PCB from humidity and allow it to withstand sterilization, the reset circuit can not employ a simple button but it has to rely on a different solution.
- **Temperature sensor and conditioning circuit**: it is the device that actually senses the temperature. A conditioning circuit is required to correctly acquire the sensor's signal.

In addition to the measurement system itself, a receiver that can allow the visualization of data in real time on a PC has to be design as well as the software that allows to perform the configuration of the device.

Chapter 2

Hardware design and component selection

The design of a wireless device that is capable to perform temperature measurements in different and difficult environments, requires the selection of the most suited components available to tackle the task maintaining the balance between effectiveness and affordability. In addition as for all the systems that have to acquire physical quantities through some sensors, the conditioning circuit plays an important role since it allows to adapt the electrical characteristic of the sensor's output to the ADC input. This chapter illustrates more in depth the proposed architecture, the design choices for the different part of the measuring system as well as the components that are selected. The final sections report the complete schematic and the PCB layout, which are created using the freely available software, Kicad.

2.1 Architecture's block diagram

Figure 2.1 shows a more detailed block diagram with respect to the one presented in the previous chapter putting in evidence the component choices. In particular it shows that the temperature sensors are six thermocouples which are split between two analog to digital converters, three for each one, it highlights the presence of the cold junction temperature sensor used to measure the temperature of the reference junction of the thermocouples in order to perform the compensation procedure required to retrieve the correct temperature value from the sensors and it put in evidence the utilization of Bluetooth Low Energy as short range radio technology for easy device configuration from portable devices. The block diagram depicted in Figure 2.1 reveals a notable distinction between certain blocks – some are highlighted in yellow, while others are in light blue. This differentiation

serves to emphasize components dedicated to creating a wireless device specific for temperature measurements (yellow) that are the thermocouples and the temperature sensor used for the cold junction compensation procedure, as opposed to components serving as a foundation for constructing a separate wireless device specialized in monitoring various other physical parameters like humidity, light, gases, and more (light blue). In this case the components are the analog to digital converter, the GPS, the LoRa module and the flash memory. As a result, the thesis primarily concentrates on the design of a wireless system for temperature monitoring. However, thanks to its high modularity, this architecture can be repurposed for other applications, rendering it highly versatile.

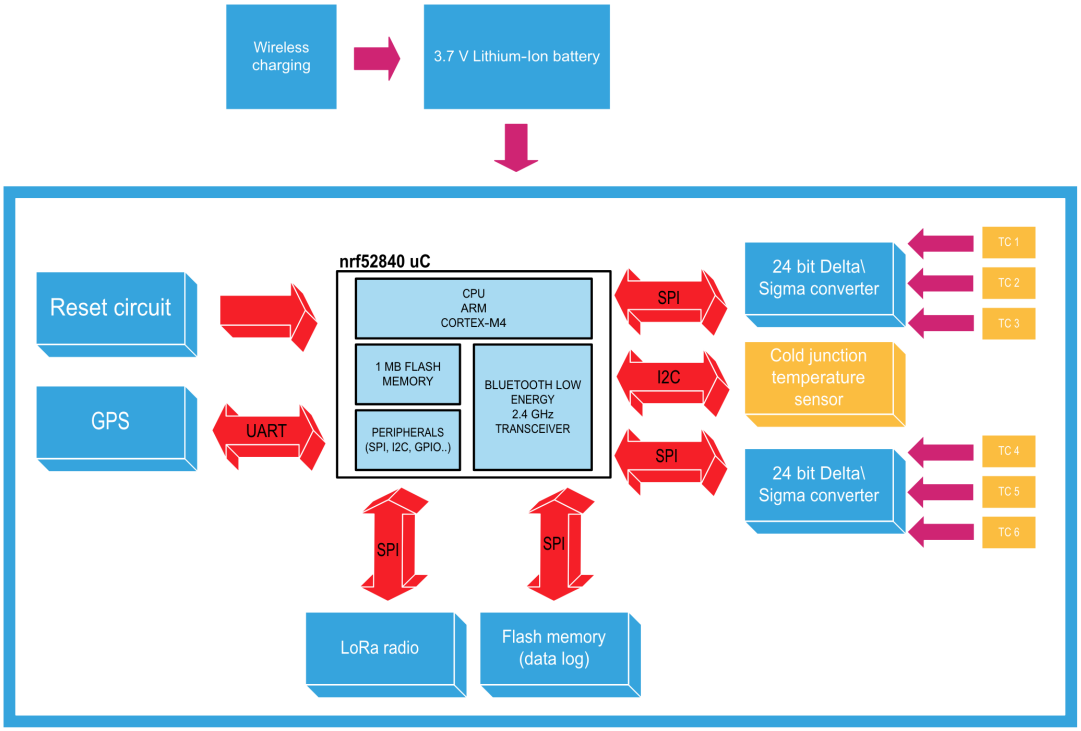


Figure 2.1: Complete system block diagram

2.2 Components overview

This section describes the main components and circuits that have been used for the realization of the system.

2.2.1 Analog to digital converter

The signal generated by each thermocouple has to be acquired by an ADC for successive manipulations. The one that is selected is the AD7799 from Analog Devices that is a low power, low noise, complete analog front ends for high precision measurement applications. Its core contains a low noise 24 bit Σ-Δ ADC with three differential inputs that are used to acquire the voltage from three thermocouples. Given the presence of a total of six thermocouples, two units are necessary. On-chip features include a low-side power switch, reference detect, programmable digital output pins, burnout currents used to identify an open thermocouple, and an internal clock oscillator. The output data rate from the part is software-programmable and can be varied from 4.17 Hz to 470 Hz. The average current consumption while it is operating is 300 μA but it becomes less than 1 μA in power down mode. The ADC is equipped with a set of configuration, status and data registers which can be accessed via SPI communication. Figure 2.2 reports the block diagram of the Σ-Δ ADC used [2].

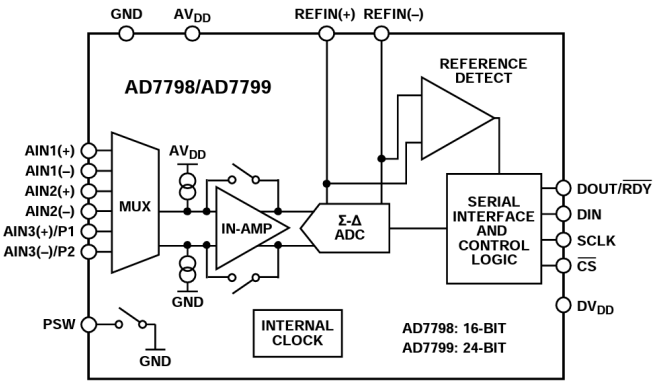


Figure 2.2: AD7799 block diagram

The choice for this particular type of ADC is related to their very high resolution and low noise that is fundamental when acquiring small signals like the ones generated by thermocouples. Furthermore, it has already been used for other similar projects therefore it is well tested. The module is also equipped with an internal PGA characterized by different gains eliminating the necessity for another external amplifier on the PCB. Another important aspect is related to the presence

of the power switch functionality. The power switch is a programmable switch toward ground. Given the low power approach that has been adopted during the design, the presence of this switch proves highly advantageous since it allows to disconnect the ground pin of other modules like the 1.25 V reference and the temperature sensor used for the cold junction compensation in such a way that the current consumption is reduced to the minimum when the system is not taking measurements. In terms of footprint, the part comes in a 16-lead TSSOP package so the space occupied on the PCB is minimal.

Sigma-Delta ADC background theory

Given the presence of the Σ-Δ ADC in the measuring system, it is important to briefly look at the theory behind it to understand how it works. The core of the converter is the Σ-Δ modulator whose output is a sequence of zeros and ones called pulse code modulated signal or PCM. The density of zeros and ones depends on the input signal being converted; the closer the signal to the reference voltage, the higher the number of ones and vice versa. Figure 2.3 reports the block diagram of the converter where it is possible to see all the components [3].

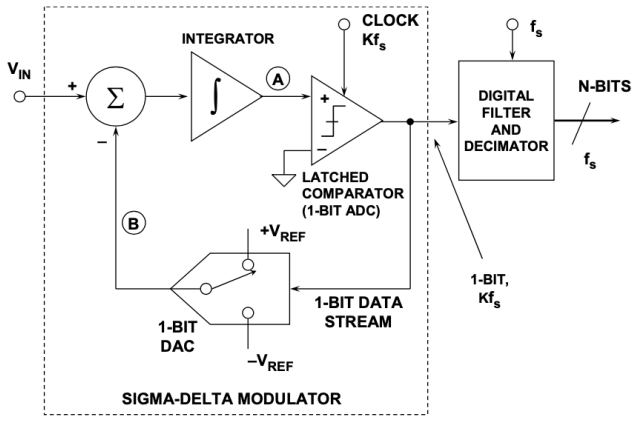


Figure 2.3: Sigma-Delta ADC block diagram

The ADC is based on an adder, an integrator, a 1-bit ADC (comparator), a 1-bit DAC and a digital part composed by a filter and a decimator. The structure of the converter is a typical feedback control loop where the error, that is the difference between the analog input V_{IN} and the digital output converted back to analog, needs to be minimized. With a positive error, the integrator output increases up to a point where the voltage at its output is larger the threshold of the comparator triggering a one at the comparator output. Now the error becomes negative so the integrator ramps in the opposite direction. As a consequence the

closed loop system tries to maintain the error minimized by periodically changing the output of the comparator between one and zero. Averaging a certain number of samples allows to obtain the digital representation of the analog input signal with the number of bits that depends on the number of taps of the digital filter. As already anticipated, Σ-Δ ADCs are characterized by a very high resolution and the reason behind this is related to the concepts of oversampling and noise shaping. Every analog to digital converter is characterized by the so called quantization noise that is generated every time that a continuous signal is mapped into a discrete set of values, as a result there will be some different analog values associated with the same digital code. The quantization error can be defined as

$$\epsilon_q = \frac{1}{2} LSB \tag{2.1}$$

where LSB is the resolution of the ADC and is given by its input range divided by 2^N. Considering the variance of the quantization noise it is possible to define the quantization noise power as follow, where $\rho(\epsilon_q)$ is the probability density function of the quantization error supposed to be uniform

$$P_{\epsilon_q} = \int_{-\frac{LSB}{2}}^{\frac{LSB}{2}} \epsilon_q^2 \cdot \rho(\epsilon_q)\, d\epsilon_q = \frac{LSB^2}{12} \tag{2.2}$$

the noise power is uniformly distributed between DC and the Nyquist frequency which is $f_s/2$ where f_s is the sampling frequency. If a sampling frequency K times higher is considered, then the quantization noise power remains the same but it becomes distributed over a wider frequency range contributing to increase the resolution. Oversampling is the first technique used by the Delta-Sigma modulator to achieve high resolution; generating samples at the output of the modulator with a rate that is much higher than the actual output data rate. The second big advantage is the so called noise shaping. Considering the frequency domain block diagram reported in Figure 2.4, it is possible to write the transfer function of the output with respect to the input and the output with respect to the quantization noise to get the following result

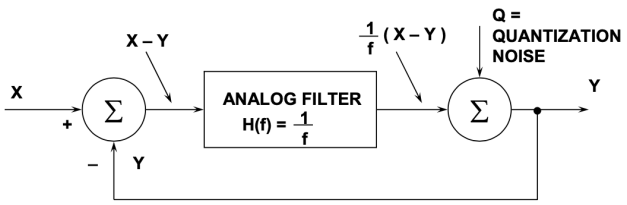


Figure 2.4: Sigma-Delta ADC block diagram in the frequency domain

$$Y = \frac{X}{1+f} + Q\frac{f}{1+f} \tag{2.3}$$

where Y is the output, X is the input and Q is the quantization noise. From this equations it can be noted that the integrator (or analog filter) has a low pass filtering effect on the input while it has an high pass filtering effect on the quantization noise form which the name noise shaping. The result is that the noise now lies on the portion of bandwidth above the pass band of the digital output filter contributing to a further reduction of the noise. Figure 2.5 reports the representation of the noise shaping

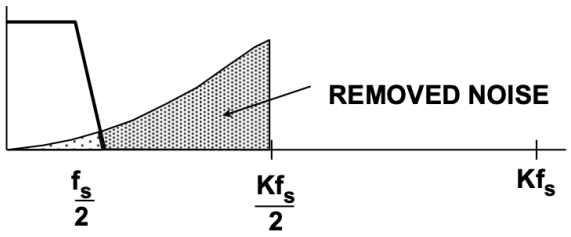


Figure 2.5: Noise shaping

eventually the high number of samples at the output of the modulator is reduced by the decimator to produce a lower output data rate.

2.2.2 Microcontroller

The microcontroller selected is the nRF52840 from Nordic® Semiconductor which is the most powerful of the nRF52 series. The reason behind the choice of this particular product is due to the presence of a Bluetooth Low Energy (BLE) transceiver that is needed for the configuration phase of the device; this feature contributes to the overall reduction of the PCB dimension since no external Bluetooth module is required. The choice of BLE as short range radio communication is related to the fact that it is supported by almost all portable devices like smartphones. The nRF52840 is fully multiprotocol capable and in addition to the protocol support for BLE, it also supports Bluetooth mesh, Thread, Zigbee, 802.15.4, ANT and 2.4 GHz proprietary stacks but for this project they are not used. The nRF52840 is built around the 32-bit ARM® Cortex™-M4 CPU with floating point unit running at 64 MHz. The module features numerous analog and digital peripherals, among them there are 5 timers configurable both in capture and compare mode, 4 SPI master, 2 I^2C compatible two-wire master/slave, 2 UART, a SAR ADC and more. To allow the interaction of the microcontroller with other components 48 GPIOs are presents. An important consideration is related to the fact that the peripherals can be mapped to any I/O pin drastically increasing the flexibility of system. In terms of power supply, voltages from 1.7 V to 5.5 V are supported. Due to the battery power nature of the sensor developed in this project, accurate power management is essential, in particular low power modes where the nRF52840 can reach current consumption as low as 1.5 μA [4]. The integration of the microcontroller on the PCB has been done exploiting a small and compact module from Ebyte called E73-2G4M08S1C that already incorporates a ceramic antenna for the BLE communication, the 32 MHz quartz oscillator and some bypass capacitors in correspondence of the power supply pin. In addition it leads out most of the I/O pins of the nRF52840 for multilateral development including the serial debug interface used to flash the code. Figure 2.6 reports a snapshot of the module.

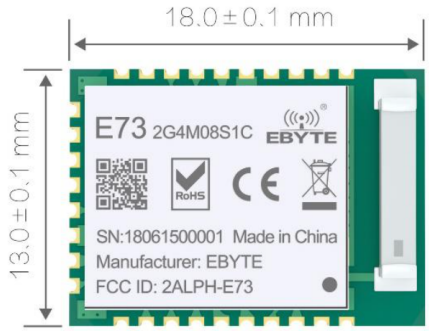


Figure 2.6: nRF52840 based module

2.2.3 LoRa Transceiver

The wireless data transfer between the device and a central receiver is managed exploiting the LoRa communication. LoRa stands for Long Range and it is typically used in all the applications requiring low power consumption, long transmission range and low bit rate. It is based on a unique way of encoding information that is called chirp spread spectrum modulation. This particular modulation scheme uses the chirp as the basis for each modulated symbol that needs to be transmitted where a chirp is simply a sinusoidal signal whose frequency increases or decreases linearly in time. As a consequence for each symbol the chirp can be seen as a sort of carrier while the information is represented by the different frequency shifts present at the beginning of each symbol [5]. The constellation defining the available symbols is generated by a parameter called spreading factor (SF); in particular, the number of symbols is in the range $s = \{0, 1, 2, \ldots, 2^{SF} - 1\}$ where SF can be from 6 to 12. The spreading factor is used to define two values: the number of raw bits which can be encoded by that symbol and the number of chips that each symbol have namely 2^{SF}. Other important parameters that define, along with SF, the final data rate are the bandwidth (BW) of the channel and the coding rate (CR). Different options are available for the bandwidth starting from 7.8 kHz up to 500 kHz, with higher values leading to higher data rates. Coding rate instead is used to increase the reliability and the robustness of the link by means of cyclic error coding to perform forward error detection and correction. The cyclic coding is expressed as the ratio of useful information bits to the total bits transmitted and possible values that can be configured are 4/5, 4/6, 4/7, 4/8. The data rate expressed in bit per second can be computed putting together all these information by the following formula:

$$R_b = \frac{BW}{2^{SF}} \cdot SF \cdot CR \tag{2.4}$$

where $BW/2^{SF}$ represents the symbol rate, SF is the number of bits per symbol and CR is the cyclic error coding fraction.
The chip incorporating the LoRa modulation selected is the Semtech SX1276 working at 868 MHz. The integrated circuits embeds all the required analog circuitry to perform both transmission with a LoRa modulator and reception with a LoRa demodulator, as well as the required configuration, status and data register accessible through an SPI interface. With the default register values the spreading factor is set to 7, the bandwidth is set to 125 kHz and the coding rate is 4/5 meaning that every 4 bits of information, one redundant bit is added. These parameters correspond to an effective output data rate of 5.4 kbps which is a good compromise that allows to have a sufficiently low time on air which, in turns, means less current consumption but still an outstanding receiver sensitivity (-121 dBm theoretically). Figure 2.7 reports a snapshot of the module used. The use of the sophisticated

Chirp Spread Spectrum (CSS) modulation of LoRa, with respect of a more common Amplitude Shift Keying (ASK) or Frequency Shift Keying (FSK) provides better results in noisy environments, moreover it significantly increases the range that can be achieved between the transmitter and the receiver.

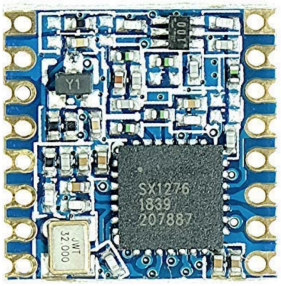


Figure 2.7: LoRa module based on Semtech SX1276 IC

2.2.4 Flash memory

The flash memory is used as a backup to retrieve data that could get lost during the real time wireless transmission. The model used for this application is a Winbond W25Q32FV with 4 MB of storage. The device is ideal to store every type of data including voice, text and others. It operates on a single 2.7 V to 3.6 V power supply with a current consumption of 4 mA when it is active and 1 μA when put in power down mode. To exchange data with a central microcontroller the module supports the standard dual SPI interface as well as the quad SPI. The memory's internal structure is organized into distinct levels, comprising blocks, sectors, and pages. At the most fundamental level, there are pages, each with a capacity of 256 bytes. These pages are grouped into sectors, with each sector accommodating 16 pages. In turn, 16 sectors collectively form a block, and this memory module contains a total of 64 blocks. Consequently, there are a total of 16,384 programmable pages. The page program instruction is used to perform a write operation to previously erased (0xFF) memory locations from a minimum of 1 byte to a maximum of 256 byte at a time. To erase the memory there are different options: it is possible to erase groups of 16 pages corresponding to 4 kB, groups of 128 corresponding to 32 kB, groups of 256 corresponding to 64 kB or the entire memory with the chip erase command. The memory supports also special features like the hardware protection of data. Lastly, in terms of physical packaging, this component is available in an 8-pin SOIC package, ensuring a compact footprint [6].

2.2.5 Reference junction temperature sensor

The sensor used to acquire the temperature of the reference junction of the thermocouples is the ADT7410 from Analog Devices. The ADT7410 is a precise digital temperature sensor with an uncertainty of $\pm 0.5\,°C$, enclosed in a compact SOIC package. This advanced sensor incorporates a temperature reference based on a band gap, coupled with a 13-bit ADC, enabling it to precisely measure and digitize temperatures with a resolution of 0.0625 $°C$, when the ADC operates at 13-bit. However, it also offers the flexibility to enhance the resolution further, allowing users to configure it to 16-bit mode, achieving a temperature resolution of 0.0078 °C by writing to the appropriate internal configuration register. The sensor can be powered with voltages ranging form 2.7 V to 5.5 V with an average supply current of 210 μA that reduces to 2 μA in shutdown mode. The ADT7410 is rated for operation in the temperature range from -55 °C to 150 °C. The communication of the part with a central microcontroller is enabled by the I^2C interface supporting four different addresses that are configurable using the pins A0 and A1. Using the internal registers is is possible to select between two different operating modes: continuous conversion which is the default one where the sensor continuously acquire a new temperature value and stores it in the temperature value register, or one shot mode where only one sample is acquired and then it goes into shutdown mode [7]. The functional block diagram of the ADT7410 is reported in Figure 2.8

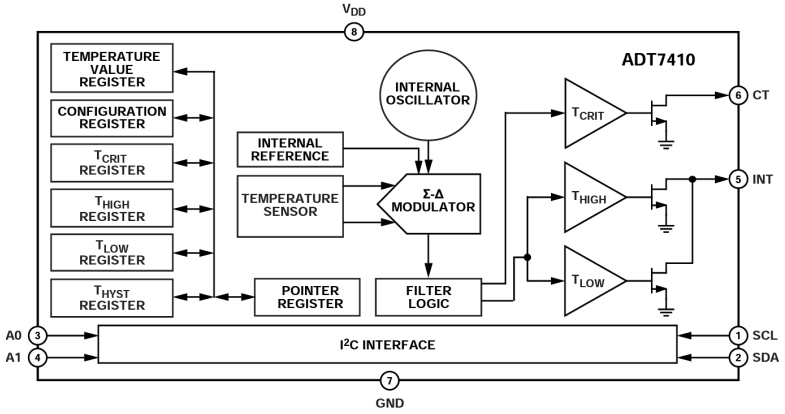


Figure 2.8: Block diagram of the ADT7410

2.2.6 Linear charge management controller

Given the possibility to recharge the battery that powers the system, it is necessary to include in the design a linear charge management controller that implements the correct procedure to safely and correctly recharge a Lithium-ion 3.7 V battery. The MCP73831 - 2ACI/0T from Microchip Technology Inc., which is the module selected for this project, employs a constant-current/constant-voltage charge algorithm with selectable preconditioning and charge termination. During the constant voltage phase the output is set to 4.2 V which is the maximum voltage a Li-Ion battery can reach when it is fully charged. Instead the value of the current during the constant current phase can be selected by placing a resistor between the PROGRAM pin and the V_{ss} pin. This current is maintained until the voltage at V_{BAT} pin reaches the regulation voltage of 4.2 V. The value of the current can be computed by exploiting the following equation

$$I_{REG} = \frac{1000V}{R_{PROGRAM}} \tag{2.5}$$

thus the lower the resistor, the higher the current that in turns requires less time to completely recharge the battery. Of course having an higher charge current means the capacity of the battery will decrease faster over time. As a consequence the proper choice of this current value is fundamental; the preferred fast charge current for Lithium-Ion cells is at the 1C rate, with an absolute maximum current at the 2C rate. Charging at this rate provides the shortest charge cycle times without degradation to the battery pack performance or life [8]. For this application a 6.8 $k\Omega$ resistor has been selected in order to have a constant current of around 150 mA. This is because it is not required to have a fast charging.
To monitor the state of the charge, a specific pin is used to provide status information. This pin exhibits a three-state behavior, transitioning into a high-impedance state when the battery is unconnected or the module is in shutdown mode. Subsequently, it shifts to a low state while the charge algorithm is actively running, eventually returning to a high state upon successful charge completion. Connecting an LED from V_{DD} to that pin allows to visually see when the charge is complete. In terms on footprint, the module is provided with an SOT-23-5 pin package.

2.2.7 GPS module

The GPS module has been included in the design in order to get the real time position of the device. This is not particularly important inside freeze dryers but it becomes very useful for example if the device is part of a sensor network that works in large open areas. The model that has been selected is the GOKU GM10 Nano V3 GPS which is based on the M10050 GNSS chip from ublox. It supports concurrent reception of four GNSS (GPS, GLONASS, Galileo, and BeiDou) with output update rate of 1 Hz or 10 Hz. The module integrates 4 pins: two of them are for the power supply (between 3.3 V and 5 V) while the other two are for the UART serial communication thus the RX and TX pins. The data can be received with two different interface protocols that are NMEA and proprietary UBLOX.

Protocol description

The description is focused on the NMEA 0183 protocol since it is the most widely adopted by the GPS modules available on the market. The protocol encompasses a large number of navigation data that are divided into different sentences (more than 50). Each sentence carries different data but the structure is the same for all. Below it is reported the generic string that is received through the serial interface

```
$IDMSG,D1,D2,...DN*CS<CR><LF>
```

The first thing that can be noted is the fact that the data are transmitted in ASCII string and not binary data like in the UBLOX protocol, as a consequence more bits are used to encode the information since every ASCII character is represented by 8 bits. Looking at the structure of the message reported above, it is possible to identify the protocol structure rules

- The $ sign at the beginning is used to indicate the start of a sentence
- ID after the dollar represents the talker identifier which are two letters that describe the source of information, for example GP indicates GPS, GL indicates GLONASS, GA indicates Galileo and BD indicates Beidou.
- The other three characters MSG represent the sentence identifier which are three letters which describe the content of the sentence in terms of data that can be found. Common sentence identifiers are GGA that is GPS fixed data, GLL that is Geographic Position for latitude and longitude or RMC for Recommended Minimum Specific GPS/Transit Data.
- After the first 6 characters there is a comma that defines the starting point of the data fields. Each data is separated by a comma.

- After the last data DN there is an asterisk which acts as checksum delimiter
- The checksum field contains two ASCII characters which indicate the hexadecimal value of the checksum
- The carriage return <CR> and the line feed <LF> combination terminate the sentence

Among all the possible sentences that are generated by the GPS module, the GLL is sufficient to get position in terms of latitude and longitude as well as the current time. If more information are required, it is possible to exploit the RMC sentence that contains also the altitude, the speed and the date as well as other data. The GLL sentence is

```
$GPGLL,llll.lll,a,yyyyy.yyy,b,hhmm.dddd,A,i,*hh<CR><LF>
```

while the RMC sentence is

```
$GPRMC,hhmm.dddd,A,llll.ll,a,yyyyy.yy,b,c.c,d.d,eeeeeee,f.f,g,i*hh<CR><LF>
```

2.2.8 Temperature sensors

Due to their robustness, size and sufficiently low uncertainty, the temperature sensors exploited in this project are the thermocouples. In fact they can be easily inserted inside pharmaceutical vials. They are based on the physical phenomena called Seebeck effect which consist in the generation of a voltage across the length of a metal that is subjected to a temperature difference between its two ends. When two dissimilar metals are joined together at one end, a thermocouple is created. The side where there is the joint is the measuring junction while the other end is the reference junction. Based on the metals used, different types of thermocouples are realized and they are characterized by different letters. For example the T type is made of copper and constantan. In order to retrieve the absolute temperature when the reference junction is kept at a temperature different from 0 °C, it is necessary to perform the cold junction compensation procedure that requires the use of an additional sensor to measure the temperature of the reference junction. Subsequently, by exploiting the polynomial equations providing a conversion from temperature to voltage and vice versa, it is possible to retrieve the final temperature of the measuring junction. More information about how thermocouples work, their uncertainty classification and cold junction compensation are reported in Appendix A

2.2.9 Wireless charging

In order to withstand sterilization the device needs to undergo resin coating, for this reason it is not possible to rely on USB-C connectors for charging purposes since the whole PCB is going to be covered by a resin layer. The wireless charging is a valid alternative that does not require physical connection between the charger and the board. Thanks to the physical phenomena of magnetic induction the transmitter coil generates an alternating magnetic field that induces a voltage to the receiver coil which is used to power the integrated circuit that is used to manage the charging phase of the Lithium-Ion battery. The module used is reported in Figure 2.9 during a validation test where two boards are visible: the one connected to the power supply is the transmitter while the longer module is the receiver that is going to be soldered on the PCB.

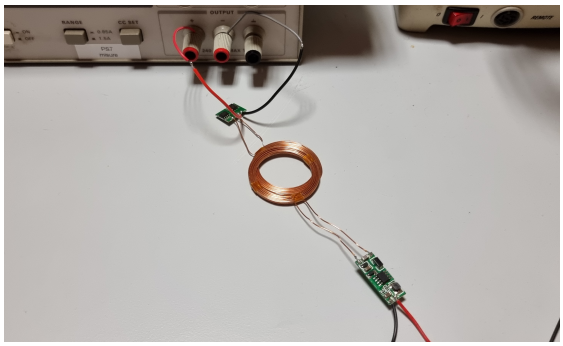


Figure 2.9: Wireless charging circuit

2.3 Signal acquisition and ADC configuration

Due to the presence of different types of thermocouples, the first thing to do is the selection of the most appropriate one. In this case T type are selected since they are very stable thermocouples and often used in extremely low temperature applications such as cryogenics or ultra low freezers. They are formed by copper and Constantan that is a copper–nickel alloy. Having the copper as one lead, when the thermocouple is soldered on the PCB, only one additional junction is going to be formed since the copper to copper connection is transparent. They are typically constituted by a brown sheath for the positive conductor that is the copper and a white sheath for the negative one which is the Constantan. The main reasons behind the choice of thermocouples as temperature sensors are the following:

- small footprint
- low cost
- high durability in harsh environments

moreover, being an active sensor it does not require external excitation like RTDs or thermistors, meaning that there are no self-overheating problems that can corrupt the measurements. The only drawback to deal with is related to the non linear response of the sensor, but this is not a big problem since the computations to obtain the temperature from the voltage or vice versa are performed exploiting the polynomial expressions on an hardware that has a floating point unit, capable of speeding up the computations quite a bit.
The measuring system is formed by six T type thermocouples in such a way that is possible to monitor the temperature of different points. As already mentioned, this is mandatory inside freeze dryers where the temperature of some vials as well as the temperature of the shelves has to be monitored. The thermoelectric voltage generated is very small, in fact inside the range from -60 °C to 60 °C the voltage varies from -2.153 mV to 2.468 mV. High precision ADCs are required to convert the small voltage for the subsequent computations.

2.3.1 Thermocouple's DC bias

To correctly acquire the thermocouple's voltage, proper biasing of the sensor is necessary to set the common-mode voltage of the thermocouple within the specified common-mode voltage range of the programmable gain amplifier (PGA) inside the ADC module. This is important because in the case of non biased thermocouple, its voltage is purely differential and its common mode voltage can violate the common mode input range of the PGA resulting in additional offset error that will

impair the measurements. In the case of the AD7799, when the PGA is active, the common mode voltage must be greater than or equal to 0.5 V. There are different types of biasing but the most common are the biasing resistors or the voltage source. The first type is reported in Figure 2.10

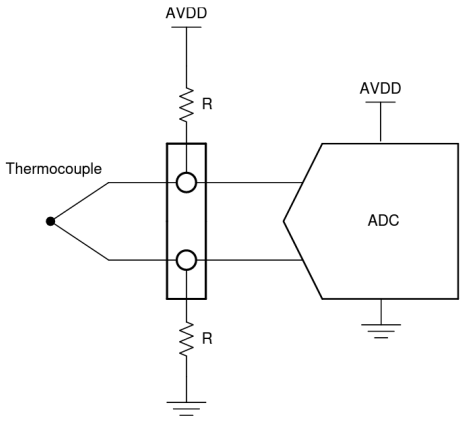


Figure 2.10: Theromocuple's biasing resistor

the value of the resistor is typically chosen in the range 500 kΩ to 1 MΩ in such a way that the biasing current is sufficiently low to avoid the introduction of additional voltage generated by the interaction of this current with the resistive leads of the thermocouple. By choosing two equal resistors, the common mode voltage is set to mid-supply and this is the choice that is typically made. To compute this voltage, it is possible to exploit the Thevenin's equivalent circuit of the thermocouple as reported in Figure 2.11 where AIN0 and AIN1 are the two input pins of the ADC, V_{TC} and R_{TC} are Thevenin's equivalent voltage and resistance respectively while R_{B1} and R_{B2} are the two biasing resistors.

The voltage at the AIN0 pin can be computed as

$$V_{AIN0} = V_{TC} + (R_{B2} + R_{TC})\frac{V_{DD} - V_{TC}}{R_{TC} + R_{B1} + R_{B2}} \tag{2.6}$$

while the voltage at the AIN1 pin has the expression

$$V_{AIN1} = R_{B2}\frac{V_{DD} - V_{TC}}{R_{TC} + R_{B1} + R_{B2}} \tag{2.7}$$

exploiting the definition of common mode voltage and considering the two resistors R_{B1} and R_{B2} with the same value leads to the final result

$$V_{CM} = \frac{V_{AIN0} + V_{AIN1}}{2} = \frac{V_{DD} - V_{TC} + V_{TC}}{2} = \frac{V_{DD}}{2} \tag{2.8}$$

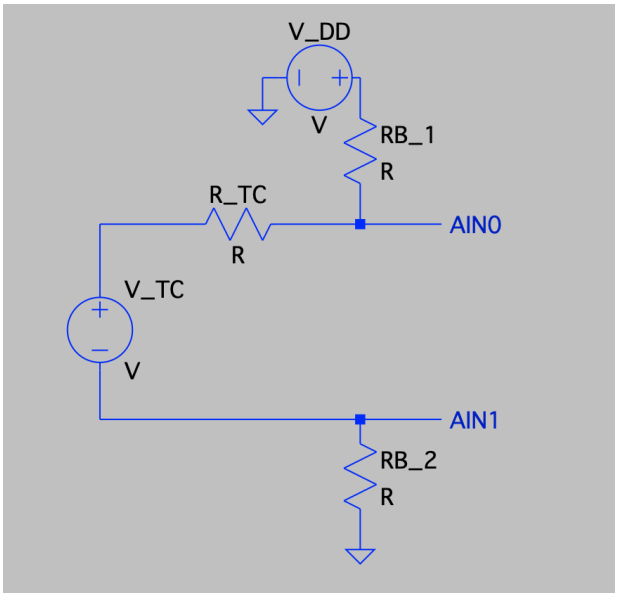


Figure 2.11: Thevenin's equivalent circuit

with this type of biasing the operating point of the sensor is set to mid supply. The presence of the two resistors can also be exploited to identify a burned out thermocouple. In fact when the thermocouple works normally, it generates the thermoelectric voltage that is read by the ADC, but when it opens due to a fault the two resistors pull apart the inputs leading to a reading that is either full scale or a value that is still much higher than the nominal voltage generated by the thermocouple.
The other approach that can be used to set the common mode voltage of the sensor within the common mode range of the PGA exploits a voltage source attached to the negative lead of the thermocouple instead of the biasing resistors. Figure 2.12 reports the schematic the difference with the previous approach is that now there is no more the bias current that flows inside the leads of the thermocouple. This is an advantage because the solution eliminates the additional voltage generated by that current. As in the previous case it is possible to use the Thevenin equivalent circuit for the thermcouple in order to compute the common mode voltage. Figure 2.13 reports the circuit exploited. Now the computation is very simple since

$$V_{AIN0} = V_{BIAS} + V_{TC} \tag{2.9}$$

$$V_{AIN1} = V_{BIAS} \tag{2.10}$$

$$V_{CM} = \frac{V_{AIN0} + V_{AIN1}}{2} = \frac{2V_{BIAS} + V_{TC}}{2} \tag{2.11}$$

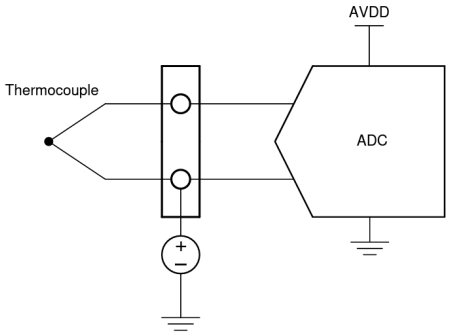


Figure 2.12: Thermocouple's biasing voltage

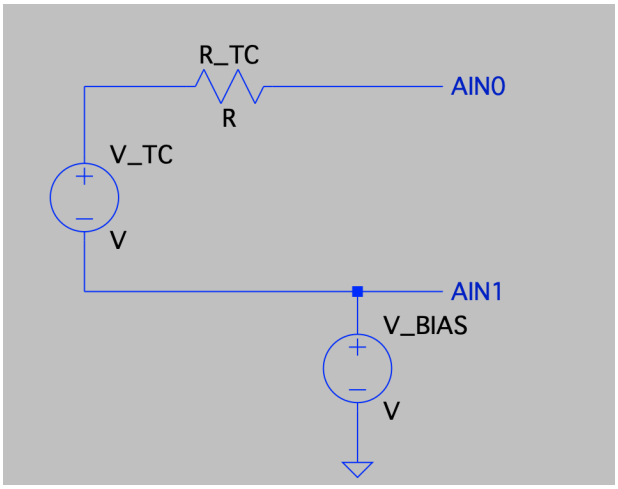


Figure 2.13: Thevenin's equivalent circuit

with this type of polarization, the common mode voltage depends on the thermoelectric voltage generated by the thermocouple. Considering the range from -60 °C to 60 °C, $V_{TC_{min}}$ = -2.153 mV while $V_{TC_{max}}$ = 2.468 mV. As a result the common mode voltage becomes

$$\frac{2V_{BIAS} + V_{TC_{min}}}{2} \leq V_{CM} \leq \frac{2V_{BIAS} + V_{TC_{MAX}}}{2} \tag{2.12}$$

This solution is the one adopted for the measuring system. In particular the bias voltage is retrieved from the same voltage that provides the reference for the ADC.

The integrated circuit that has been selected is the REF3312 which is a low-power, precision, low-dropout voltage reference from Texas Instruments, with an output voltage of 1.25 V and an output current of 5 mA maximum. The supply current is also pretty low with a typical value of 3.9 μA. With this choice the common mode voltage of the thermocouple falls in the range

$$1.249\,\mathrm{V} \leq \mathrm{V_{CM}} \leq 1.251\,\mathrm{V} \tag{2.13}$$

considering that the supply voltage will be set to 3.3 V this is not exactly at mid supply but still within the limits of the common mode range of the PGA. This solution has the advantage of eliminating the biasing resistor that would contribute to an additional voltage errors but care must be taken in the identification of a burned out thermocouple. Indeed without those resistors it is not possible to identify an open thermocouple but it is necessary to rely on an additional feature that is available on the ADC that are the Burnout Currents. The AD7799 is equipped with burnout currents that can be used to verify that an external transducer is still operational before attempting to take measurements on that channel. Once those currents are enabled by writing to the configuration register, they flow to the external circuit. If the thermocouple is still closed, given its small resistance, no significant voltage is dropped by the 100 nA current but when it accidentally opens, the resistance seen by the input pins becomes very high and the voltage measured by the ADC becomes full scale allowing for the identification of the open transducer. An additional RC low pass filter is present at each thermocouple output to smooth out high frequency noise.

2.3.2 ADC parameters configuration

In order to acquire the thermoelectric voltage generated by the thermocouple, the AD7799 needs to be correctly configured. The first thing to do is to identify the range of operation for the thermocouple that in this case is considered from -60 °C to 60 °C, corresponding to a voltage range of -2.153 mV to 2.468 mV. The next step is to calculate the maximum gain allowed without over-ranging the PGA based on the maximum thermocouple voltage and reference voltage. The instrumentation amplifier inside the ADC has different gain options starting from 1 up to 128 in steps of power of 2. Considering the gain setting of 128 and a voltage reference of 1.25 V, the differential input voltage range is defined as

$$\text{Differential input voltage range} = \pm\frac{V_{REF}}{GAIN} = \pm\frac{1.25\,V}{128} = \pm 9.76\,\mathrm{mV} \tag{2.14}$$

this range is enough since -2.153 mV and 2.468 mV are well below the limits. An important consideration is related to the absolute input voltage range that can be

applied to the individual analog input pins of the ADC. The AD7799 datasheet states that with a gain greater that 4, this voltage has to be greater than GND + 300 mV and lower that AVDD - 1.1 V = 2.2 V otherwise, linearity and noise performance degrade. In this case, the negative input AIN(-) is tied up to the 1.25 V reference voltage, as a consequence the positive input AIN(+), whose voltage is referenced to the AIN(-) varies from 1.25 V - 2.153 mV = 1.248 V to 1.25 V + 2.468 mV = 1.252 V. With these values both the voltages at AIN(-) and AIN(+) are within the declared limits.
After passing through the instrumentation amplifier, the thermocouple voltage is amplified by a factor of 128 generating the following values

$$V_{TC_{min}} \cdot \text{GAIN} = -2.153\,\text{mV} \cdot 128 = -0.275\,\text{V} \tag{2.15}$$

$$V_{TC_{max}} \cdot \text{GAIN} = 2.468\,\text{mV} \cdot 128 = 0.316\,\text{V} \tag{2.16}$$

From this range it is possible to compute the percentage of ADC code used

$$\text{ADC codes used} = \frac{128 * [2.47\,\text{mV} - (-2.15\,\text{mV})]}{1.25\,\text{V} - (-1.25\,\text{V})} \cdot 100 = 23.65\,\% \tag{2.17}$$

Considering that the ADC is 24 bit, the number of codes in the measurement range is 0.2365 $\cdot 2^{24} = 3968550$ which corresponds to a resolution of

$$\text{Temperature resolution} = \frac{60\,^\circ\text{C} - (-60\,^\circ\text{C})}{3968550} = 0.03 \times 10^{-3} \frac{^\circ\text{C}}{\text{LSB}} \tag{2.18}$$

The AD7799 offers both unipolar and bipolar configuration. For this application the bipolar has been selected in such a way that both positive and negative voltages with respect to the pin AIN(-) can be measured. When the ADC is configured for bipolar operation, the output code is offset binary, with a negative full-scale voltage resulting in a code of 000...000, a zero differential input voltage resulting in a code of 100...000, and a positive full-scale input voltage resulting in a code of 111...111. The output code for any analog input voltage can be represented as

$$Code = 2^{N-1} \times \left[\frac{AIN \times GAIN}{VREF} + 1\right] \tag{2.19}$$

Exploiting the previous formula, for the maximum and minimum codes are

$$Code\ @\ T_{max} = 2^{23} \times \left[\frac{2.47\,\text{mV} \times 128}{1.25\,\text{V}} + 1\right] = 10510322 \tag{2.20}$$

$$Code\ @\ T_{min} = 2^{23} \times \left[\frac{-2.15\,\text{mV} \times 128}{1.25\,\text{V}} + 1 \right] = 6541773 \tag{2.21}$$

The AD7799 offers two different types of conversion mode: Continuous-Conversion Mode or Single-Conversion Mode. In the first the ADC continuously performs conversions and places the result in the data register with the rate defined inside the mode register. In the second mode instead the ADC powers up and performs a single conversion, then after 1 ms necessary to power up and settle the oscillator plus a conversion time of $2/f_{ADC}$ where f_{ADC} is the update rate of the converter, the result is put inside the data register and the module enters sleep mode again. For this application the Continuous-Conversion Mode is the one selected since it is used to perform the acquisition of a certain number of samples to eventually perform an average. The update rate of 50 Hz allows to have the best noise rejection for this reason it is the value used. As soon as the samples are acquired, the ADC is put in power down mode. Table 2.1 summarize the parameters used to configure the AD7799

ADC configuration parameters	
Reference voltage	1.25 V
Gain	128
Coding	Bipolar
Conversion mode	Continuous
Update rate	50 Hz

Table 2.1: AD7799 configuration parameters

2.4 Battery voltage measurement circuit

The system is powered by a Lithium ion battery with a nominal voltage of 3.7 V. Even though the low power approach allows to use the sensor for a long time, it is still necessary to have the possibility to measure the voltage of the battery in such a way that it can be recharged as soon as its charge level drops below a certain threshold. For this task the microcontroller's onboard successive approximation register (SAR) ADC is exploited. Its characteristics are software configurable such as the number of bits (8/10/12), the reference voltage for each channel, the gain of the internal amplifier or the acquisition time based on the maximum source resistance. Due to the fact that the microcontroller is powered by a 3.3 V low-dropout voltage regulator, the analog pin accepts only voltages up to V_{DD} that is 3.3 V but the battery when fully charged reaches 4.2 V while it drops to 2.7 V when it is completely empty, thus it is necessary to shift the voltage levels by means of a voltage divider. The resistors' absolute values have been selected to minimize the leakage current during the measurement while the capacitor prevents fluctuations in the voltage level during the acquisition. Figure 2.14 reports the portion of schematic responsible for the battery voltage measurement.

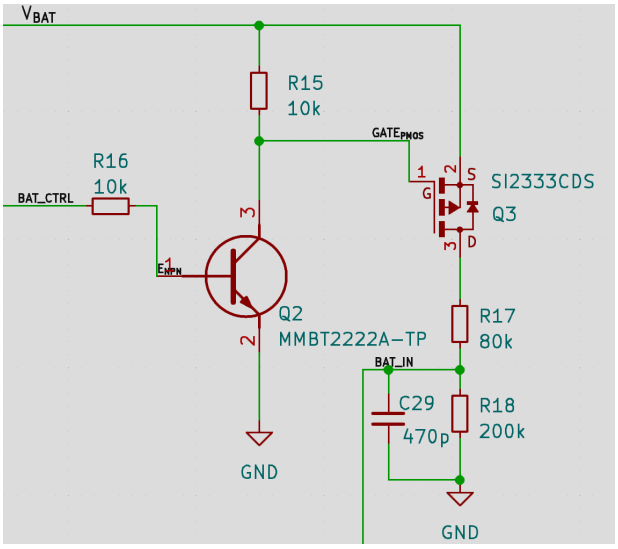


Figure 2.14: Battery voltage measurement circuit

With $R_{17} = 80\,\text{k}\Omega$ and $R_{18} = 200\,\text{k}\Omega$ the maximum voltage of 4.2 V is converted in 3 V while the minimum voltage of 2.7 V becomes 1.93 V. The leakage current, computed as 4.2 V / 280 kΩ, is equal to 15 μA. The node BAT_IN is then fed to the analog input pin of the microcontroller. From the Figure it is possible to see that there are other components in addition to the resistors of the voltage

divider. These are necessary to actually connect the battery voltage to the ADC. The NPN transistor in the common emitter configuration is used as a switch to set the voltage at the gate of Q3. The values of the resistors R_{15} and R_{16} are selected to guarantee that the BJT is in the saturation region when the microcontroller generates the BAT_CTRL command. In fact when the signal BAT_CTRL goes high, the NPN transistor Q2 turns on connecting the gate of the PMOS Q3 to ground, as a consequence the threshold voltage of Q3 becomes negative turning on Q3. Now the voltage V_{BAT} is applied to the voltage divider in such a way that the measurement can be performed. The R_{ON} resistance of Q3 can be neglected since it is more that an order of magnitude lower than R_{17}.
The nRF52840 on board SAR ADC has been configured with the following parameters

- Single ended input
- Reference voltage equal to VDD/4
- Gain equal to 1/4
- Bit resolution equal to 10

With these parameters the ADC input range becomes V_{DD} and the voltage resolution or LSB becomes $V_{DD}/2^{10} = 3.2\,\text{mV}$, which is sufficient to perform a battery voltage measurement. To further reduce the noise, up to 20 samples are acquired before performing an average that will represent the final result.

2.5 Reset circuit

There are situations where the wireless or the inter-module communication could get stuck and the measuring system needs to be reset so that is can start again without any trouble. For this reason the sensor has been equipped with a particular type of reset circuitry that works even if the entire PCB has to be resin due to the fact that it has to work inside freeze dryers where everything has to be sterilized, or in harsh environment to protect the PCB. As a consequence a simple button is not sufficient since as soon as the resin solidifies, no more access to the PCB is available. In order to reset the microcontroller an high to low transition on a particular pin has to be performed. The solution that has been adopted relies on the DRV5032 in SOT-23 package that is a magnetic switch from Texas Instruments. Figure 2.15 reports the internal block diagram of the device.

The IC incorporates an Hall effect sensor that is used to evaluate the intensity of the external magnetic field applied, analog signal conditioning, and a low-frequency oscillator that enables ultra-low average power consumption. When the intensity

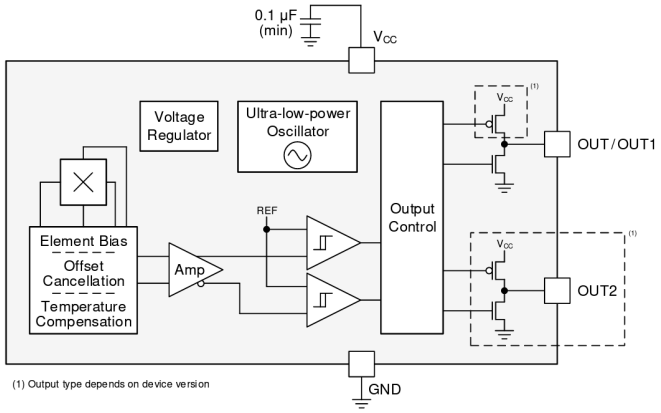


Figure 2.15: DRV5032 block diagram

of the magnetic flux exceeds a certain threshold (called B_{OP}) that depends on the particular type of DRV5032, the devices outputs a low voltage. The output remains low until the flux density decreases below the the B_{RP} threshold, then the output goes high due to the push pull output configuration. Figure 2.16 reports the magnetic response where the thresholds are defined. There are also different options in terms of the polar expansion that has to be provided to the device to make it switch; in this case the omnipolar option has been selected so that it is only sufficient to put the magnet sufficiently close to the device without worrying about north or south pole [9].

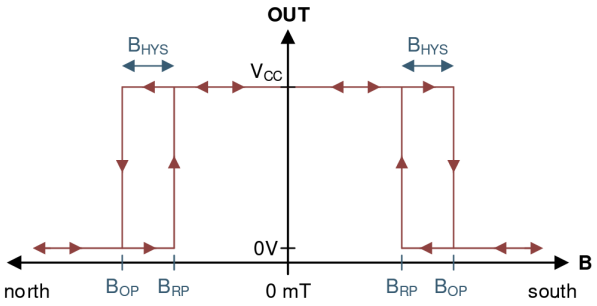


Figure 2.16: DRV5032 omnipolar magnetic response

The complete reset circuit schematic for the microcontroller reset is reported in Figure 2.17. In addition to the Hall effect switch, other components are for its correct operation.

From the scheme it is possible to highlight that the V_{DD} pin of the integrated circuit I^2C is connected to the V_{CRG} label that represents the supply voltage coming from the wireless charging module. As a consequence the DRV5032 is able to work

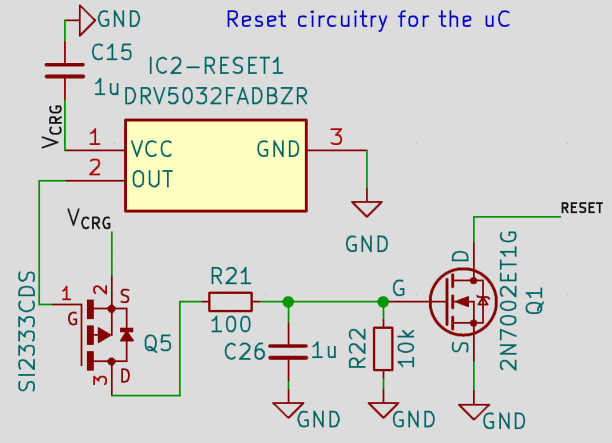


Figure 2.17: Reset circuit schematic

only when the system is charging so that the V_{CRG} voltage is available. This choice is made to enhance the circuit's reliability, preventing accidental reset activation in the presence of strong magnetic fields during normal operation. Thus a reset of the device can only occur during the charging state. Regarding the other components, the PMOS Q5 and the NMOS Q1 as well as the resistor R_{22} are necessary to have the correct voltage on the reset pin of the microcontroller in the different cases: when $V_{CRG} = 0$ then OUT is also 0 and since the source of Q5 is 0 due to its connection with V_{CRG} then V_{GS} is not sufficiently negative to put Q5 into conduction, for this reason R_{22} is used as pull down resistor to ensure that the NMOS Q1 remains off as well, correctly preventing the connection of the reset pin to ground. As soon as V_{CRG} becomes different form 0, if the magnet is not present then OUT is equal to V_{CRG} and again Q5 does not conduct as in the previous situation. When the magnet is brought closer to the Hall effect switch, OUT goes to 0 but now the source of Q5 has a voltage different from 0 causing it to conduct. This, in turn, generates a V_{GS} voltage greater than the threshold for Q1, allowing Q1 to conduct, connecting the reset pin to ground. The resistor R_{21} and the capacitor C_{26} are used to form a low pass filter necessary to filter out possible spikes that could be present during the turn on time of the DRV5032 IC.

2.6 Low power hardware considerations

Due to the fact the device is battery powered, it is important to reduce its current consumption to the minimum when it is not acquiring new samples or it is not transmitting new data. This is achieved by exploiting transistors used as switches. For example the GPS module is the one that draws the higher amount of current, as a consequence after the acquisition of the device's position, its ground connection is cut by driving to an high logic level the NMOS connected to its ground pin. The same technique is adopted for the other components like the LoRa radio, the memory or the digital thermometer. The choice to cut the ground connection instead of the positive supply has been dictated by the presence of an already existing switch toward ground inside the two ADCs called power switch. As a consequence to exploit them and maintain the coherence, all the switches are toward ground. An important note is that every time the ground connection is cut, all the other pins of the module need to be set to an high logic level to have the same electrical potential on the remaining pins.

2.7 Circuit schematic

The complete schematic of the wireless sensor is reported in Figure 2.18. To highlight the different blocks of the circuit described in the previous sections, there are dashed lines boxes around them. Starting form the bottom of the schematic, it is possible to identify the six thermocouples with the corresponding filters that are connected to the two ADCs (3 thermocouples each) differential inputs. On the left side there are the integrated circuit responsible for the management of the battery charge and, above it, the voltage regulation section that provides both the supply voltage for all the modules and a reference voltage used for the thermocouple bias and the ADCs reference. On the right side there is the digital thermometer used to measure the temperature of the reference junction of the thermocouples to perform the cold junction compensation. In the center it is present the brain of the device, namely the microcontroller. Moving toward the top of the schematic it is possible to see the flash memory, the GPS module and the LoRa radio as well as two other sections that are the battery voltage measurement circuit and the reset circuit. The first one is responsible for the acquisition of the battery's voltage used to estimate its remaining charge, the second one instead is used to perform the reset of the module.

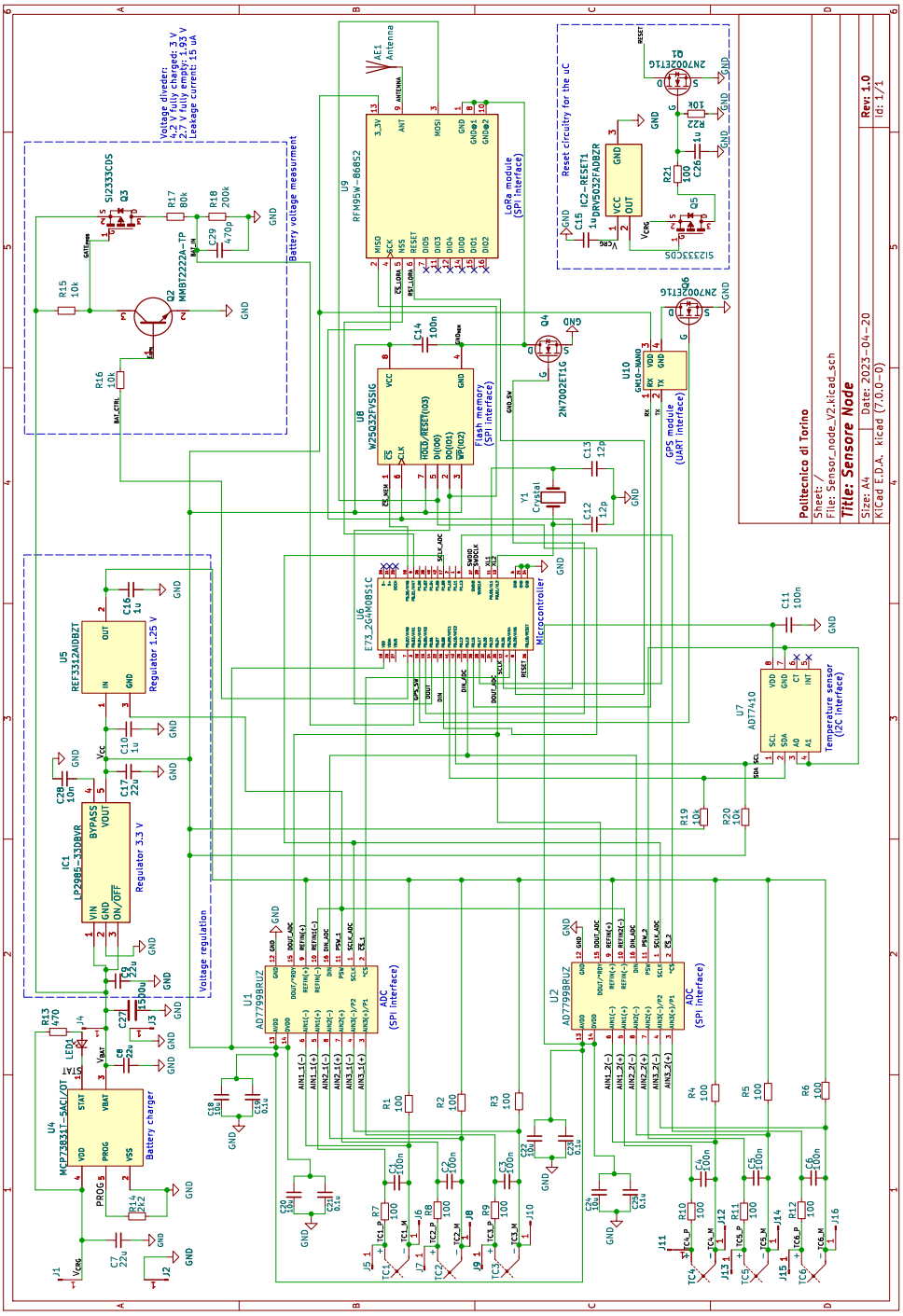
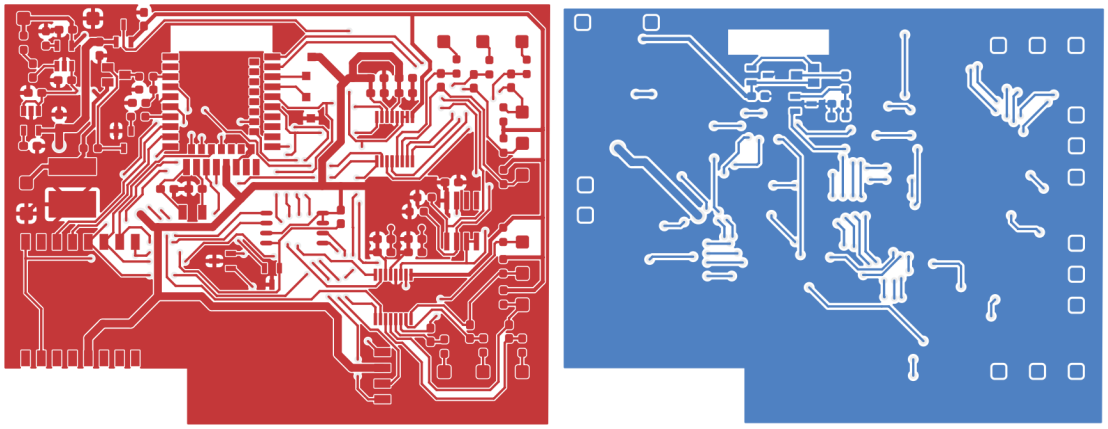


Figure 2.18: Schematic of the device

2.8 PCB layout

The PCB layout was created using the freely licensed software KiCad. The board comprises two layers, with the majority of components residing on the front copper layer, while only the reset circuit is placed on the bottom layer. Both surfaces have been filled with large copper areas connected to the negative terminal of the battery to establish ground planes. The advantages of ground planes are numerous, among them there are the easier layout since all the ground pins of the components can be routed without additional traces and the significant lower impedance of the plane with respect to a trace that contributes to reduce the the disturbance created by the switching current drawn by the digital circuits. Most connections are routed on the front copper layer, while the bottom layer is dedicated to resolving complex intersections where traces intersect. In such cases, vias are employed to transition between the two layers.
The trace width differs depending on the type of connection. Signal lines utilize a width of either 0.3 mm or 0.5 mm, chosen based on the connection's length and available space around the modules. In the case of the positive supply from the regulator, a 1 mm width is employed to further reduce trace resistance, ensuring uniform voltage supply to all integrated circuits. Ground connections, on the other hand, are either directly on the front copper fill or on the bottom one, especially in areas where it's not feasible to place the copper fill on the front layer due to the manufacturing capabilities limitation.
An important consideration is related to the connection of the reference junction of the thermocouples on the PCB. In order to avoid the creation of additional voltage along the copper traces that connect the thermocouple's reference junction to the ADC, it is necessary to have the two thermocouples' leads on an isothermal block to have the same temperature. This block is a connector made by a large metal mass with a certain thermal capacitance. Due to the limited dimensions of the board, having six connectors for the six thermocouple is not feasible, as a consequence the solution adopted for this project is the maximization of the copper fill around the junctions of the PCB exploiting the two ground planes on the top and on the bottom layer. As a consequence the thermocouples' leads are soldered between the copper fill of the two planes . Another consideration is related to the position of the digital thermometer used to acquire the temperature of the reference junction. In particular is has to be placed sufficiently close to the junction to have a reliable measurement. In this project, since T type thermocouples are used, there is only one junction created that is between Copper and Constantan. Figure 2.19 is reports the front and the bottom Copper layer in two separate images while Figure 2.20 reports all the layer superimposed including the Silkscreen, the Soldermask and the paste. The black areas are keepout area in correspondence of the two antennas. Figure 2.21 and Figure 2.22 report the 3D render of the 2 sides of the PCB



(a) PCB layout front copper layer **(b)** PCB layout bottom copper layer

Figure 2.19

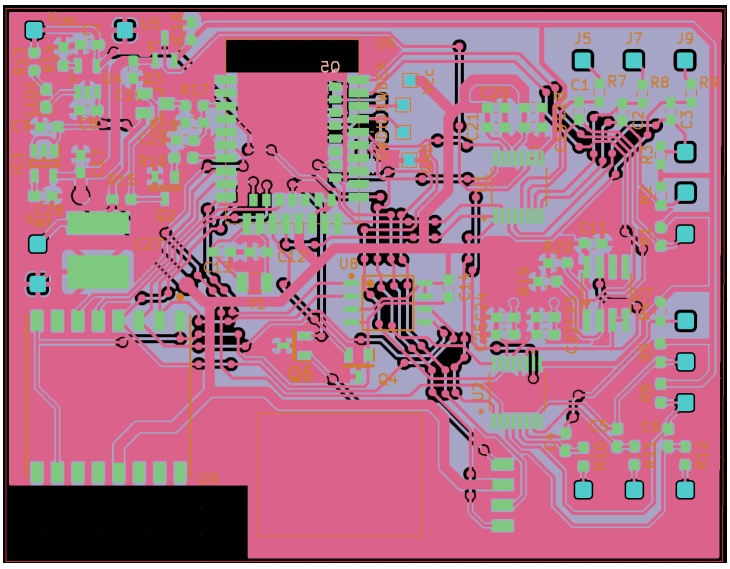


Figure 2.20: PCB layout with all the layers superimposed

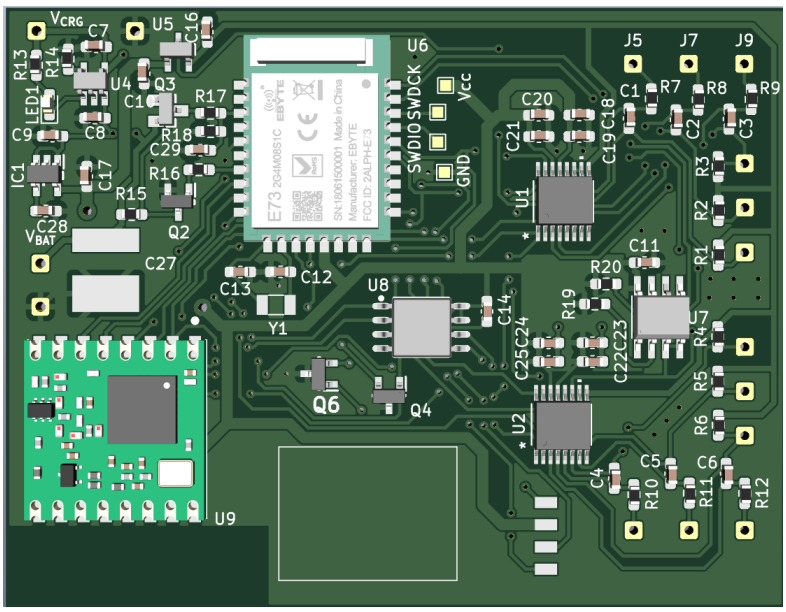


Figure 2.21: 3D render of the front side of the PCB

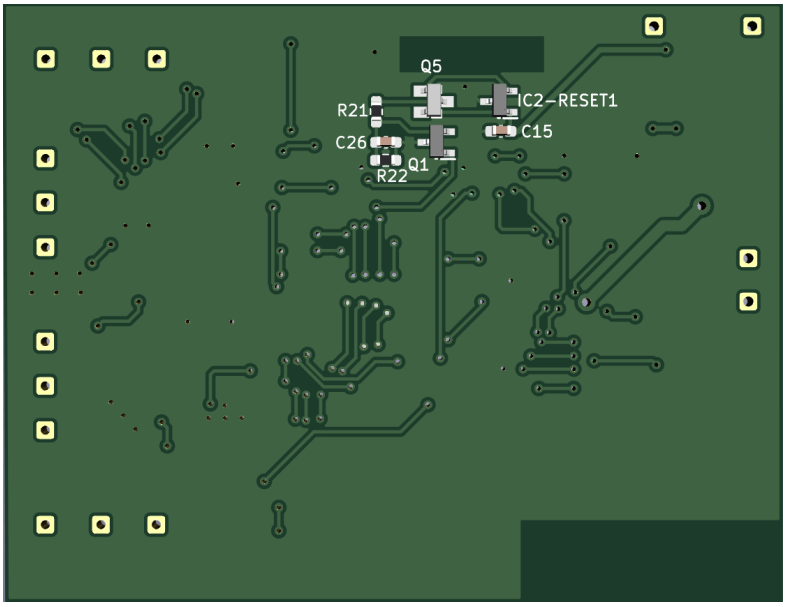


Figure 2.22: 3D render of the back side of the PCB

Chapter 3

Firmware design

This chapter describes the functionalities of the firmware that runs on the microcontroller as well as the software developed for interfacing with the sensor that encompasses the configuration phase and data visualization.

3.1 Development environment

The first thing to do before starting the programming phase is the setup of the development environment which represents the set of tools required to write, flash and debug the code that will be stored inside the microcontroller's memory. The main components that are typically part of this chain are:

- A text editor or and Integrated Development Environment (IDE) that is used to write the code and if possible debug it
- A compiler to translate the high level code to the machine code
- A breakout board hosting the desired microcontroller that can be used to test the application during the prototyping phase
- A programmer needed to flash the code on the microcontroller

The IDE has a big advantage with respect to the classic text editor namely it allows to use the debug functionality. In the case of the Nordic devices used for this application, a powerful IDE called SEGGER Embedded Studio is available with a free license. In particular it is an integrated development environment with project management tools, editor and debugger supporting Arm Cortex devices. Full debug support including Real Time Terminal (RTT) output is also available. It is important to note that the license is only valid for Nordic Semiconductor target devices.

During the prototyping phase it is not convenient to work with the chip hosting the microcontroller that will be mounted on the PCB due to the difficulties in accessing its pins. For this reason the best choice is to use a breakout board with pin connectors that facilitate the connection of external wires. This project involved the use of two distinct boards: the nRF51 and the nRF52840 MDK USB Dongle. The first, shown in Figure 3.1, is the official development kit board from Nordic, featuring the nRF51422 chip and an Interface MCU for programming on-chip or external modules by means of the serial wire debug interface. This board had been previously employed in other projects and was repurposed for this project as a programming tool due to the absence of the correct nRF52 chip. This board also enable the possibility to have debug functionality thanks to the SEGGER J-Link OB Debugger. The second board, depicted in Figure 3.2, is a custom-made board housing the nRF52840 chip, which served as the primary development platform for the wireless sensor. By soldering header pins onto the nRF52840 MDK USB Dongle, the microcontroller's GPIOs become readily accessible for interfacing with other components

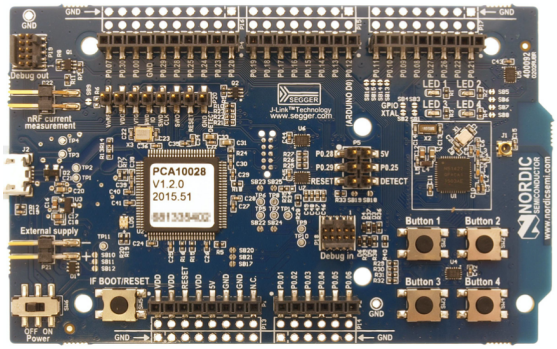


Figure 3.1: nRF51 DK

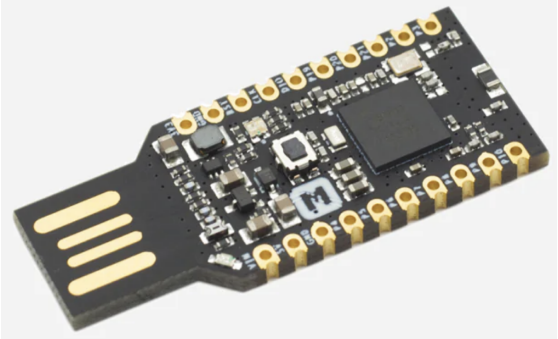


Figure 3.2: nRF52840 MDK USB Dongle

To program nRF51 or nRF52 devices mounted on external boards it is necessary to connect 4 cables from the nRF51 DK to the custom board. Figure 3.3 shows how to perform the connection. In particular the development kit is equipped with a Debug out connector called P20. From this connector four pins are brought to the custom board:

- EXT VTG which connects to the supply pin of the custom board VDD and it is used to detect an external chip and tell the interface MCU to program the external chip instead of the on-board nRF51.
- EXT SWDIO which connects to SWDIO on the custom board.
- EXT SWDCLK which connects to SWDCLK on the custom board.
- GND DETECT which connects to GND pin of the custom board.

To enable the device programming, the PCB is equipped with four pads enabling the possibility to flash new programs to the microcontroller during the prototyping phase. The nRF51 DK is also equipped with a micro-USB connector that allows the direct connection of the board to a PC.

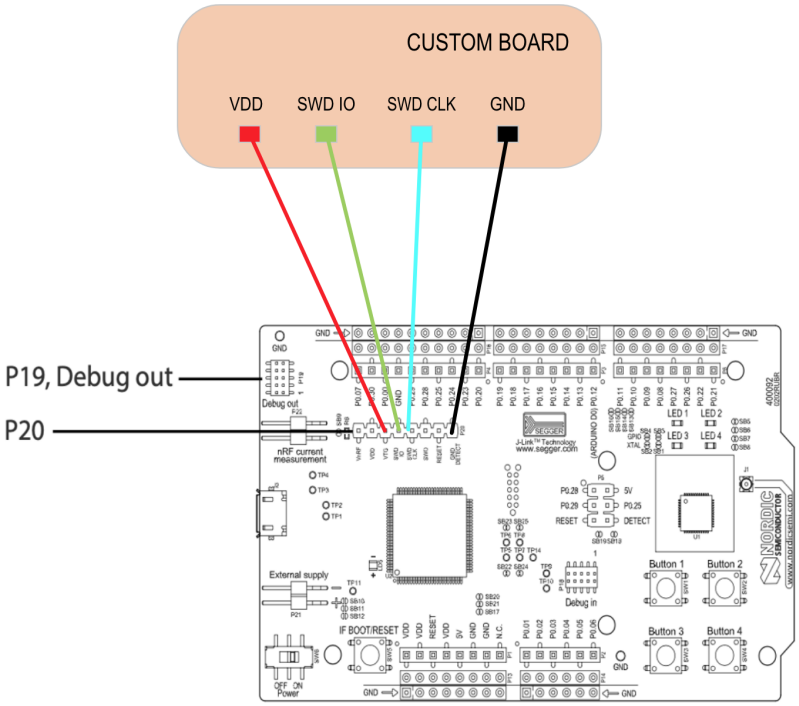


Figure 3.3: External board programming

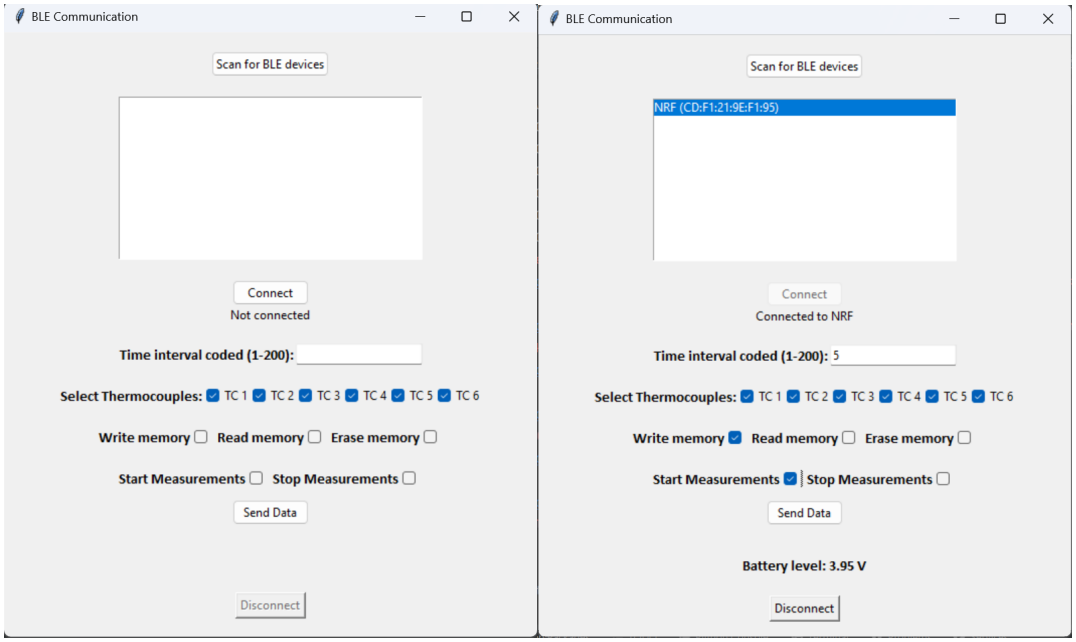
3.2 Configuration and control software

The configuration and control software of the device is fundamental to set the parameters for the acquisition and to decide when to start or stop the measuring phase. Within this project, these tasks are executed through a dedicated Graphical User Interface (GUI) developed in Python and capable of Bluetooth Low Energy support. In particular the application has the following features:

- Bluetooth scanning to find the device and connect to it when needed. Due to the fact that a lot of different Bluetooth devices can be within the range of the PC performing the scanning operation, a filtering action based on the name of the device is performed so that the listbox is populated only with its name and address.
- Visualization of the battery voltage every time that a new connection with the device is established
- Individual selection of each one of the 6 thermocouples to have the maximum flexibility
- Selection of the time interval between two consecutive measurements with a specific coding that guarantees a large range of possible values
- Possibility to control the beginning and the end of the measurement phase
- Possibility to save the acquired data on an on-chip non volatile flash memory, retrieve them at a later time reading the memory and erase it when starting a new acquisition.

The first problem to be solved is the correct management of the BLE communication between the application and the sensor. In this case a specific Python library called Bleak is used. Bleak is an acronym for Bluetooth Low Energy platform Agnostic Klient and it supports reading, writing and getting notifications from GATT servers, as well as functions for discovering a BLE device and connecting to it. It works with asynchronous or non-blocking functions thus it requires the use of the asyncio Python's library and an event loop that schedules their executions. More information about BLE communication and how data are represented can be found in Appendix B.
The GUI is realized with Tkinter which is a standard Python library specifically developed for the creation of graphical user interfaces. In particular it provides a wide variety of built-in widgets (buttons, labels, entry fields, etc) that are the basic building blocks used for the creation of GUIs. An extremely important aspect when working with graphic interfaces is the responsiveness namely the window

needs to be updated continuously in such a way that every action performed by the user is reflected on it. In the case of Tkinter, there is a loop that waits for an event to occur to update the window and runs forever until the window is closed. This is a problem when it is used in conjunction with the asyncio library because the functions used by the Bluetooth library Bleak are executed within another event loop that manages the execution of asynchronous tasks. The solution adopted in order to correctly manage both the loops is to create an infinite loop that calls the Tkinter update() method at regular intervals maintaining the GUI responsive while executing the other asynchronous tasks concurrently. The update() method is used to manually update the graphical user interface, allowing pending events in the event queue and redraw operations to be processed immediately. Figure 3.4 reports the GUI in the case of not connected and connected device after the scanning process.



(a) GUI with device not connected **(b)** GUI with device connected

Figure 3.4

3.2.1 Code organization

The code is organized with two classes: the first one is called App and represents the running application while the other is called Window and it represents the actual window hosting the GUI. The App class has a method called execute that is a coroutine. A coroutine is a special function in Python that can be paused and resumed, allowing for the execution of asynchronous operations. The execute coroutine defines an instance of the class Window and then it runs the coroutine show() from the latter class to continuously update the GUI exploiting the update() method as described before.
The Window class instead is used to define all the GUI widgets used to interact with the sensor and the event handlers in response to user actions. In particular it includes the following components

1. Initialization of the main window
2. Buttons used for scanning BLE devices nearby, connecting to the selected device and sending data after configuration
3. A listbox to display the BLE device's name and address found during the scanning
4. An input field for setting a time interval between two consecutive measurements
5. Checkboxes to select thermocouples' number, for configuring memory and for measurement settings
6. Methods for handling user interactions and BLE device communication

For all the handling methods involving the usage of routines from Bleak such as the BleakScanner.discovery() or the the BleakClient.connect(), due to the fact that they are asynchronous, it is necessary to call the create_task() that is a method from asyncio used to create and schedule the execution of coroutines. Once the task is created, it runs concurrently with other tasks on the event loop. For example the scan button is generate with the following code

```
1 self.scan_button = ttk.Button(self.root, text="Scan for BLE
    devices", command=lambda: self.loop.create_task(self.
    scan_and_populate_listbox()))
2 self.scan_button.pack()
```

Listing 3.1: Scan button

and the associated coroutine that acts as event handler, scheduled with the create_task() method is

```
async def scan_and_populate_listbox(self):
        devices = await BleakScanner.discover(timeout=5)
        self.listbox.delete(0, tk.END)
        for device in devices:
            if str(device.name) == "NRF":
                self.listbox.insert(tk.END, f"{device.name} ({
    device.address})")
```

Listing 3.2: Coroutine scheduled after pressing the scan button

Another consideration is related on how the GATT client (which is the PC that runs the Python application) retrieves the information about the battery voltage from the GATT server (which is the wireless sensor). In order to reduce the power consumption to the least, only one voltage acquisition is performed, at every new connection with the device. In particular, the notify operation is exploited; as soon as the connection is established, the start notify command is issued and the sensor will push the new battery voltage acquired, to the client. To do so, the custom characteristic created on the server needs to have the notify property activated. In the following it is reported the coroutine used to perform the connection and the battery voltage acquisition by means of the notify operation.

```
async def connect_selected_device(self):

        selected_index = self.listbox.curselection()
        if selected_index:
            selected_device = self.listbox.get(selected_index)
            print(selected_device)
            device_name, device_address = selected_device.split("
    ", 1)
            device_address = device_address[1:-1]  # Remove
    parentheses
            self.name = copy.deepcopy(device_name)
            self.client = BleakClient(device_address)

            await self.client.connect()
            if self.client is not None:
                if self.client.is_connected:
                    self.terminate_button.config(state=tk.ACTIVE)
                    self.device_label.config(text=f"Connected to {
    device_name}")
                    self.connect_button.config(state=tk.DISABLED)
                    await self.client.start_notify(CHAR_UUID, self
    .handle_rx)
                    await asyncio.sleep(5.0)
                    await self.client.stop_notify(CHAR_UUID)
```

Listing 3.3: Coroutine of connection

3.2.2 Data packet format

After the user inputs all the commands, the application combines them in a sequence of bits with a specific order in such a way that the code running on the measuring system's microcontroller can extract each piece of information to do the required actions. In particular each action needs to have a corresponding bit or bit sequence that encode the action performed by the user that can be for example the star of the measuring phase or the reading operation from the memory. The bytes that are sent to the device via Bluetooth Low Energy are always seven and they represent the following information

- **Time interval between two consecutive measurements**: it is represented by a number between 1 and 200 where each number encodes a particular time interval. To allow the selection of a reasonable amount of different intervals, the encoding uses a non constant step that changes based on the amount of seconds represented. For example the number between 1 and 29 represent the seconds with step of 1 then from 30 to 44 the step changes to 2 therefore 30 represents 30 seconds but 31 represents 32 seconds and the step further increases for the number from 45 to 62 where it becomes 5 seconds. In this way with only 8 bit it is possible to encode a large number of intervals spanning from 1 seconds to 24 hours. The complete encoding solution is reported in Table 3.1 where the first column reports the number that is inserted in the input field of the GUI, the second column is the step, the third one reports the formula required to retrieve the time interval based on the input encoded number and the last column reports the time that it is possible to represent with the given input numbers

Numbers (n)	Step	Equivalent in Seconds	Time Represented
1 ... 29	1 s	n	1 s... 29 s
30 ... 44	2 s	(n - 30)·2 + 30	30 s... 58 s
45 ... 62	5 s	(n - 45)·5 + 60	1 min... 2 min 25 s
63 ... 77	10 s	(n - 63)·10 + 150	2 min 30 s ... 4 min 50 s
78 ... 92	20 s	(n - 78)·20 + 300	5 min ... 9 min 40 s
93 ... 102	30 s	(n - 93)·30 + 600	10 min ... 14 min 30 s
103 ... 117	60 s	(n - 103)·60 + 900	15 min ... 29 min
118 ... 141	300 s	(n - 118)·300 + 1800	30 min ... 2 h 25 min
142 ... 171	900 s	(n - 142)·900 + 9000	2 h 30 min ... 9 h 45 min
172 ... 200	1800 s	(n - 172)·1800 + 36000	10 h ... 24 h

Table 3.1: Time interval encoding

Since the representation of numbers up to 200 requires 8 bit, the first byte

composing the data packet sent to the wireless sensor is the encoded time interval.

- **Number of thermocouples selected**: the system is equipped with 6 thermocouples that can be activated separately by flagging the corresponding checkboxes on the GUI, therefore it is necessary to know which of them has to perform the measurement and which not. For this reason 6 bits are used to encode this information: when the thermocouple is selected the corresponding bit is set to 1 otherwise it is set to 0. As a consequence the second byte that is sent represents the thermocouples selected.

- **Write, read or erase the memory**: for this purpose 2 bits are exploited in order to encode four possible situations based on the checkbox that is flagged by the user. The couple 00 is used to encode the write operation, 01 for the read operation, 10 for no operation with the memory and 11 for erasing the memory.

- **Start or stop the measuring phase**: the start and stop operations are encoded with one single bit: set to 1 for a star operation and set to 0 for a stop operation. This bit is put together with the other 2 bits related to the memory operations. In this way these 3 bits are joined together to form the third byte of the sequence

- **The current timestamp in Unix epoch format**: every measurement that is performed by the device is going to be associated with the date and time at which it occurs. For this reason it is necessary to provide a starting point to the microcontroller so that it can associate the correct date and time to each measurement and keep it updated with one of its internal timers. This starting point is provided by the Unix epoch timestamp that is the number of seconds that have elapsed since January 1, 1970 at midnight UTC time minus the leap seconds. This number is retrieved exploiting the time library of Python. Every time that the user sends a new command to the device, the current timestamp is sent together with the other data. The microcontroller will simultaneously start a timer increasing every one second to keep this number updated. The Unix timestamp is represented with 4 bytes and it occupies the last 4 bytes of the packet sent to the device.

The function send_data() is used in the Python code to retrieve the user's inputs to the GUI widgets and to format the data packet that is sent to the wireless sensor. The specific function used to actually send data to the specific GATT characteristic is write_gatt_char() from Bleak. It is a write with response operation (sometimes called a Request) where the GATT client will write the data then wait for a response from the remote device.

3.3 Microcontroller firmware

The firmware running on the microcontroller is the software that manages all the operations of the sensor from the Bluetooth Low Energy communication to the acquisition of the thermocouples' voltages and the transmission of all the required information to the central receiver. The code is written in C language and it exploits the Nordic Software Development Kit (SDK) to communicate with the whole set of peripherals required. The SDK is a set of libraries and drivers developed by Nordic to make easier the interaction with the microcontroller, in fact it is possible to exploit high level functions instead of directly writing to the microcontroller's registers, to speed up the development flow. Furthermore to correctly manage the data exchange during the configuration phase based on the Bluetooth Low Energy technology, Nordic provides a precompiled and linked binary software called SoftDevice, implementing the protocol stack of BLE therefore high level functions can be used to configure every aspect from the physical layer to the GATT and GAP protocol.

3.3.1 Drivers for external modules

The device is formed by the microcontroller and a set of modules that are the LoRa for the long range data transmission, the analog to digital converters, the reference junction thermometer, the non volatile flash memory and the GPS. These modules integrate data, configuration and control registers that are accessed via different type of serial interfaces like UART, SPI and I^2C to allow the interaction with the modules. In order to communicate with them, a set of drivers has been developed. Each component has its own C and h file to maintain the separation with the rest of the code so that they can be re used even for other projects. The GPS is the only component that does not require its library because the sentences that are sent to the microcontroller through UART interface, are programmed by means of a software that is provided by the manufacturer of the GPS.

AD7799 driver

The AD7799 is equipped with nine registers as it can be seen in Figure 3.5, accessed via SPI communication which allows to retrieve data from the module and configure all the parameters for the acquisition. All communication to the part must start with a write operation to the communication register. The data written to the communication register determines whether the next operation is a read or write operation, and to which register this operation takes place. After the read or write operation is complete, the interface returns to its default state, where it expects a write operation to the communication register.

RS2	RS1	RS0	Register	Register Size
0	0	0	Communication register during a write operation	8 bits
0	0	0	Status register during a read operation	8 bits
0	0	1	Mode register	16 bits
0	1	0	Configuration register	16 bits
0	1	1	Data register	16 bits (AD7798)/24 bits (AD7799)
1	0	0	ID register	8 bits
1	0	1	IO register	8 bits
1	1	0	Offset register	16 bits (AD7798)/24 bits (AD7799)
1	1	1	Full-scale register	16 bits (AD7798)/24 bits (AD7799)

Figure 3.5: AD7799 registers

The driver is formed by a set of high level functions that simplify the interaction with the module. For example to select a specific gain of the PGA, the setGain() with the required argument can be used. All these functions exploit other 2 base functions that are required to write and read data to and from the module. In particular they are called writeRegister() and readRegister(). They implement the actual data transfer from and toward the ADC by means of the high level driver provided by Nordic to interact with the SPI peripheral. Due to the fact that the registers have different lengths, an argument is provided as input to specify whether the write or read operation is going to be performed on an 8, 16 or 24 bits register. The two functions are reported below

```
1  static uint32_t readRegister(uint8_t n_bytes)
2  {
3      uint8_t tx_buffer[n_bytes];
4      uint8_t rx_buffer[n_bytes];
5      uint8_t spi_write_length = 0;
6      uint8_t spi_read_length = n_bytes;
7
8      spi_xfer_done = false;
9      nrf_drv_spi_transfer(&spi_instance, tx_buffer,
       spi_write_length, rx_buffer, spi_read_length);
10     while(!spi_xfer_done){};
11     if(n_bytes == 1)
12         return (uint32_t)rx_buffer[0];
13     else if(n_bytes == 2)
14         return ((uint32_t)rx_buffer[0] << 8) |
       ((uint32_t)rx_buffer[1]);
15     else if(n_bytes == 3)
16         return ((uint32_t)rx_buffer[0] << 16) |
       ((uint32_t)rx_buffer[1] << 8) | ((uint32_t)rx_buffer[2]);
17 }
```

Listing 3.4: Function used to read data from the AD7799

```
static bool writeRegister(uint16_t value, uint8_t n_bytes)
{
    ret_code_t err_code;
    uint8_t tx_buffer[n_bytes];
    uint8_t rx_buffer[n_bytes];
    uint8_t spi_write_length = n_bytes;
    uint8_t spi_read_length = 0;

    spi_xfer_done = false;

    if (n_bytes == 1) // 8 bits transfer
    {
        tx_buffer[0] = (uint8_t)value;
        nrf_drv_spi_transfer(&spi_instance, tx_buffer,
   spi_write_length, rx_buffer, spi_read_length);
        while(!spi_xfer_done);
    }
    else if(n_bytes == 2) // 16 bits transfer
    {
        tx_buffer[0] = (uint8_t)(value >> 8);
        tx_buffer[1] = (uint8_t)value;
        nrf_drv_spi_transfer(&spi_instance, tx_buffer,
   spi_write_length, rx_buffer, spi_read_length);
        while(!spi_xfer_done){};
    }

    if(err_code == NRF_SUCCESS)
      return true;
    else
      return false;
}
```

Listing 3.5: Function used to write data to the AD7799

LoRa driver

The LoRa module exploits the same serial interface of the AD7799 therefore the strategy used to communicate with it is very similar. There are some differences though, in particular the selection of the register where the read or write operation is going to be performed is done based on an address. If a write operation is required then the MSB of the register's address is set to 1 whereas for a read operation is set to 0. After the first address byte, the second byte will be either the data that needs to be written or the data provided by the module. The available

registers are more than 30 so they are not reported here. The two base functions for reading and writing are listed below

```
static uint8_t readRegister(uint8_t address)
{
    spi_write_length = 1;
    spi_read_length = 2;
    spi_tx_buffer[0] = address & 0x7F;
    spi_xfer_done = false;

    nrf_gpio_pin_clear(SPI_SS);

    APP_ERROR_CHECK(nrf_drv_spi_transfer(&spi_instance,
   spi_tx_buffer, spi_write_length, spi_rx_buffer,
   spi_read_length));
    while(!spi_xfer_done);

    nrf_gpio_pin_set(SPI_SS);
    return spi_rx_buffer[1];
}
```

Listing 3.6: Function used to read data from the LoRa module

```
static void writeRegister(uint8_t address, uint8_t value)
{
    spi_write_length = 2;
    spi_read_length = 0;
    spi_tx_buffer[0] = address | 0x80;
    spi_tx_buffer[1] = value;
    spi_xfer_done = false;

    nrf_gpio_pin_clear(SPI_SS);

    APP_ERROR_CHECK(nrf_drv_spi_transfer(&spi_instance,
   spi_tx_buffer, spi_write_length, spi_rx_buffer,
   spi_read_length));
    while(!spi_xfer_done);

    nrf_gpio_pin_set(SPI_SS); // release the device
}
```

Listing 3.7: Function used to write data to the LoRa module

ADT7410 driver

The ADT7410 is the reference junction temperature sensor and it is equipped with 14 registers as reported in Figure 3.6 but only some of them are actually exploited. The communication is based on the I^2C protocol therefore the module has its own address. To increase the flexibility, there are two pins that can be set to either one or zero to make up to 4 different addresses. In this case they are set to zero. As for the AD7799 and the LoRa, the strategy that permits to read and write data from and to the module remains the same except for the presence of the different serial interface. The address pointer register is always the first register written to during a write to the ADT7410. It must be set to the address of the register to which the write or read transaction is intended. As already mentioned not all the registers are used in this application, for example the T_{high}, T_{low} and T_{crit} are not required since they allow to set specific temperature thresholds. The important ones instead are the status register to verify whether a new temperature value is available, the configuration register to set the acquisition mode (continuous or single) and the temperature value register which contains the current temperature acquired. Exploiting the functions writeRegister() and readRegister() reported below it is possible to set the specific bits required to interact with the module. These functions have the same name as the ones used for the other two modules described before, but they are different because they exploit the I^2C interface instead of the SPI.

Register Address	Description	Power-On Default
0x00	Temperature value most significant byte	0x00
0x01	Temperature value least significant byte	0x00
0x02	Status	0x00
0x03	Configuration	0x00
0x04	T_{HIGH} setpoint most significant byte	0x20 (64°C)
0x05	T_{HIGH} setpoint least significant byte	0x00 (64°C)
0x06	T_{LOW} setpoint most significant byte	0x05 (10°C)
0x07	T_{LOW} setpoint least significant byte	0x00 (10°C)
0x08	T_{CRIT} setpoint most significant byte	0x49 (147°C)
0x09	T_{CRIT} setpoint least significant byte	0x80 (147°C)
0x0A	T_{HYST} setpoint	0x05 (5°C)
0x0B	ID	0xCX
0x0C	Reserved	0xXX
0x0D	Reserved	0xXX
0x2E	Reserved	0xXX
0x2F	Software reset	0xXX

Figure 3.6: ADT7410 registers

The two functions used to exchange data with the ADT7410 are the following

```
static bool writeRegister(uint8_t reg_address, uint8_t
    value)
{
     ret_code_t err_code;
     uint8_t i2c_tx_buffer[2];
     i2c_tx_buffer[0] = reg_address;
     i2c_tx_buffer[1] = value;
     i2c_xfer_done = false;
     err_code = nrf_drv_twi_tx(&i2c_instance,
    ADT7410_ADDRESS, i2c_tx_buffer, sizeof(i2c_tx_buffer),
    false);
     while(!i2c_xfer_done){};
     if(err_code == NRF_SUCCESS)
       return true;
     else
       return false;
}
```

Listing 3.8: Function used to write data to the ADT7410 module

```
static bool readRegister(uint8_t reg_address, uint8_t *
    result, uint8_t length)
{
     ret_code_t err_code;
     i2c_xfer_done = false;
     err_code = nrf_drv_twi_tx(&i2c_instance,
    ADT7410_ADDRESS, &reg_address, 1, true);
     while(!i2c_xfer_done){};
     i2c_xfer_done = false;
     err_code = nrf_drv_twi_rx(&i2c_instance,
    ADT7410_ADDRESS, result, length);
     while(!i2c_xfer_done){};
     if(err_code == NRF_SUCCESS)
      return true;
     else
       return false;
}
```

Listing 3.9: Function used to read data from the ADT7410 module

Non volatile flash memory driver

The communication with the flash memory produced by Winbond, is performed by means of an already existing library that had been developed for previous projects. The only changes that have been made to the code are related to the command used to set or clear the MOSI, MISO, CS and SCK pins.

3.3.2 Bluetooth Low Energy configuration

In this project the BLE is used during the configuration phase of the device due to the fact that it allows to transmit data with a much lower current consumption with respect to the LoRa. In addition the configuration of the device is typically performed close the device itself therefore the long range transmission of LoRa is not required in that specific case. To enable BLE communication it is necessary to create a custom service with a custom value characteristic to receive data from the Python application and then based on those data perform the required actions. The operations required to create the custom service and the custom value characteristic are

1. **Creation of a custom base UUID** → The UUID (Universally Unique Identifier) for Bluetooth Low Energy (BLE) is a 128-bit value typically represented as a 32-character hexadecimal string. BLE uses UUIDs to uniquely identify services, characteristics, and descriptors. In this case it is obtained from an online UUID generator.

2. **Implementation of a custom service** → The process involves including essential header files, defining macros for Custom Service instances, creating structures for initialization and status information, and implementing the initialization function which initializes the Custom Service structure, sets the connection handle and adds the Custom Service UUID to the BLE stack's table using the function the SoftDevice sd_ble_gatts_service_add().

3. **Service initialization and UUID advertising** → This step is required to put the 128-bit UUID in the advertisement packet so that other BLE devices can see the Custom Service of this device. In addition the custom service is initialized in the main.c file inside the function used for initializing services that will be used by the application. Moreover adding a vendor-specific UUID to the BLE stack requires the incrementation of the RAM space reserved for the SoftDevice.

4. **Implementation of the custom value characteristic** → After the creation of the service, it is necessary to add a characteristic. This is done creating a function named custom_value_char_add() that is used to define the attribute

value of the characteristic. Several metadata variables are declared within the function, including characteristic metadata which are used to specify the properties of the characteristic (read, write and notify), Client Characteristic Configuration Descriptor metadata and others. Within this function, it is also defined the dimension in bytes of the characteristic value attribute. Eventually, the characteristic is added using the function provided by the SoftDevice called sd_ble_gatts_characteristic_add().

5. **Event handling from SoftDevice** → In this step, event handling for the SoftDevice is added to manage the Custom Service's connectivity states. A new function named ble_cus_on_ble_evt() is created and registered as an observer to ensure it receives SoftDevice events. Based on the type of event received (connection, disconnection or write) a specific action will be executed. For example in correspondence of a write event occurring when new configuration data are sent to the device, a function is called to extract the data received and perform the corresponding actions.

6. **Notification of the custom value characteristic** → This step is used to enable bidirectional communication between the device and the connected device. This is required to send the battery voltage value measured by the sensor to the PC. For this purpose a new function called ble_cus_custom_value_update() is introduced. The function updates the GATT table with the new custom value received as argument then, if a valid connection is present, it notifies the connected device by utilizing the sd_ble_gatts_hvx() function provided by the SoftDevice. Furthermore, to enable or disable notifications, the CCCD (Client Characteristic Configuration Descriptor) is introduced. In this case the function called in response to a write event needs to be modified to handle the situation where the PC writes to the descriptor to enable or disable notifications.

Every time that new data arrive to the device, the function called to handle the write event extract from each byte the required information, and save them into a structure. Based on these values, the code performs the required actions.

3.3.3 Measurement phase

When the device receives the command to start the measuring phase, the ADCs and the reference junction thermometer are activated to get the required data that will be sent to the central receiver. Due to the stringent current consumption requirements, the ADCs start every set of measurements from the sleep state therefore they need to be activated before acquiring new samples. The first step to do is to initialize the SPI communication and then reset their internal registers to the default values. This is necessary because they could remain in low power mode for long periods of time therefore some internal registers' configurations can change leading to wrong acquisitions. The ADCs are then put in idle mode where the ADC filter and modulator are held in a reset state, although the modulator clocks are still provided. Following the start-up, it is possible to configure the ADCs' internal registers. This is done after the selection of the correct ADC instance, indeed there are two of them managing 3 thermocouples each therefore, based on the thermocouples that are selected, the configuration bits need to be sent to either ADC 1 or ADC 2, after driving low the corresponding chip select pin. When the correct AD converter is selected, it is possible to choose the right channel by writing to the configuration register, then the coding type namely bipolar by writing to the same register, the gain and eventually the update rate of the filter, in this case 50 Hz, by writing to the mode register. These operations are repeated within a for loop, cycling through each bit of a variable. Each bit represents the state of the i^{th} thermocouple, taking the value of 1 if the thermocouple is selected and 0 otherwise. For every active channel, a sequence of 13 samples is acquired. The first five are discarded to eliminate a cross-talk effect that has been registered during some tests, while the other 8 samples are averaged to reduce the random noise superimposed to the thermocouple's voltage. After the acquisitions, the ADCs are put again in low power mode where their current consumption is lower than 2 μA. To correctly estimate the temperature at which the measuring junction of the thermocouple is, it is also necessary to acquire the temperature of the reference junction. As for the two AD converters, also the ADT7410 goes in power down during the time interval between consecutive measurements. Thus the first thing to do is the initialization of the I^2C interface to be able to sent data to the module. After that, there is only one configuration to do namely set the result temperature on 16 bits which provides the highest resolution. Acquiring each new temperature takes 240 ms, thus requiring 1.5 seconds to average at least six values. Due to this time constraint, it is impractical to perform a sequential operation where ADT7410 values are acquired first and then, upon completion, the ADCs are activated. A more efficient approach is to initiate the ADT7410 start command, acquire the voltage of thermocouple i, and subsequently retrieve the ADT7410 temperature value. Figure 3.7 reports a flow chart of what has been described so far.

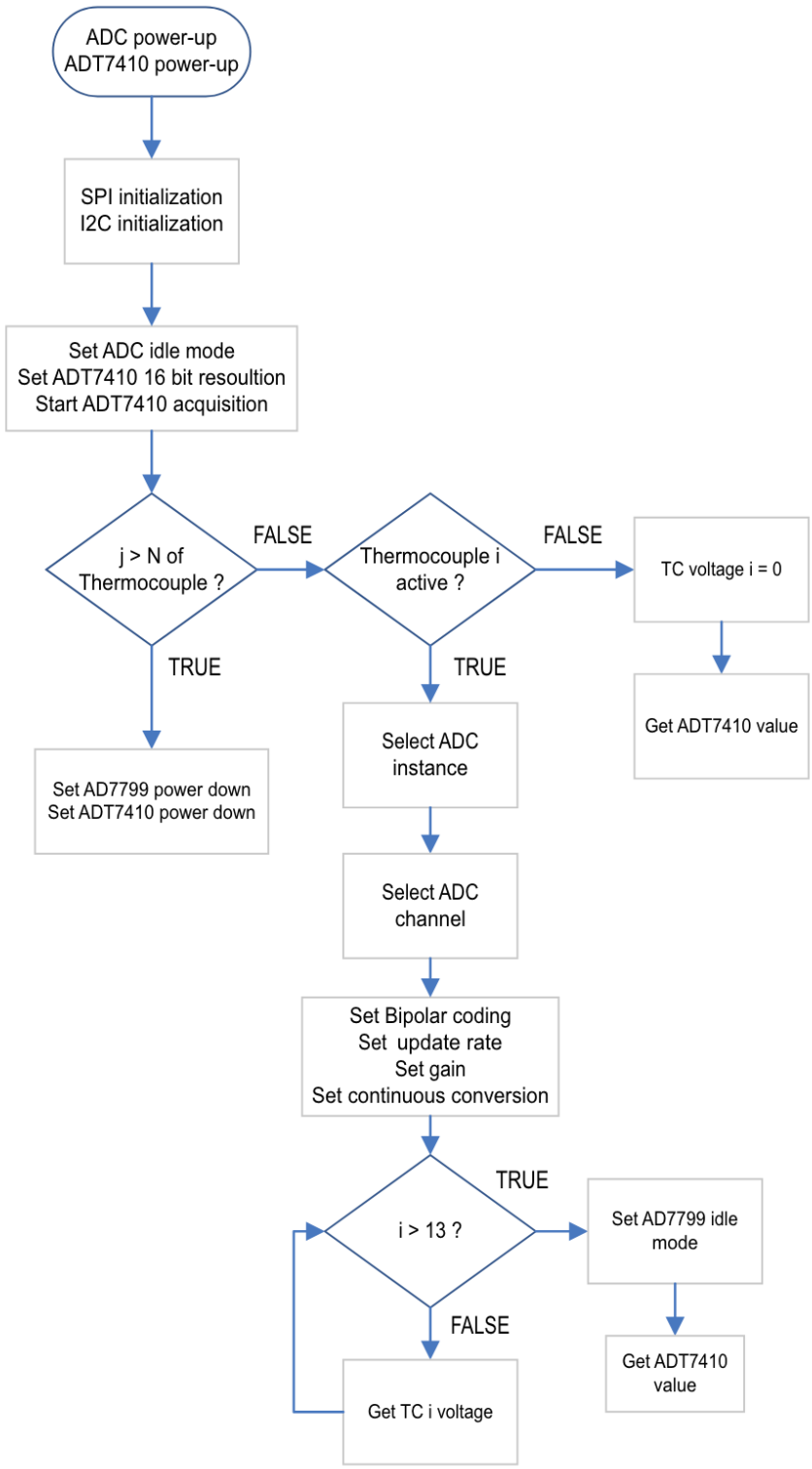


Figure 3.7: Measuring phase flow chart

3.3.4 Wireless data transmission

After the completion of the measuring phase, the ADC readings, the reference junction temperature and the timestamp associated with each measurement can be sent. The LoRa radio during the sleep state is disconnected from the supply by cutting its ground connection with an NMOS acting as a switch, therefore it needs to be re-activated. The SPI peripheral needs also to be initialized since during the sleep state it is uninitialized to eliminate a possible source of current consumption. The data transmission procedure starts with the computation of the number of thermocouples that are active based on the raw data of the ADCs. In fact for inactive sensors, the corresponding code is 0 which is a value that can not be possible for an active thermocouple. This number allows to define the size of the packet that will be sent in terms of bytes. The size is not always the same but is depends on the number of active thermocouples. The first byte of the sequence is indeed the number of active thermocouples since it is necessary to extract correctly all the other data at the receiver side. The next bytes are the raw data coming from the ADCs where each sample is 24 bit wide thus it is necessary to decompose it in 3 chunks of 8 bits each in such a way that they can be stored correctly in a uint8_t variable array. This field can vary between 3 (one thermocouple active) and 18 (six thermocouples active) bytes. After that there are the raw data coming from the reference junction temperature sensor. In this case 16 bit are decomposed in two chunks of 8 bits each. The last information to be sent is the timestamp where again a decomposition is required to have four separate 8 bit values. The LoRa radio is capable of transmitting from 1 to 256 bytes for each data transfer. In this case is more than enough since the maximum number of bytes that are going to be sent is 25 when all the thermocouples are active. The minimum is 10 bytes for the case of only one active sensor. Table 3.2 reports an example of packet in the case of one thermocouple active.

N of TC active
TC 1, byte 1
TC 1, byte 2
TC 1, byte 3
ADT7410, byte 1
ADT7410, byte 2
Timestamp, byte 1
Timestamp, byte 2
Timestamp, byte 3
Timestamp, byte 4

Table 3.2: Packet example in the case of one thermocouple active

3.3.5 Data memorization

If the memorization bit is set to one, all the data sent by the radio are also going to be stored in the non volatile flash memory. In the perspective of reducing as much as possible the current consumption, the memory has its ground pin connected to a switch controlled by the microcontroller. In particular it is the same NMOS used for the LoRa module for this reason when the radio is active, the memory becomes active as well. The communication with the part is done via SPI. The serial interface is shared with the LoRa exploiting two different chip select pin. As already anticipated in the chapter related to the hardware description the memory has a capacity of 4 MB and it is organized into 16384 programmable pages of 256 bytes each. The write command allows to write from 1 to 256 bytes for each operation. If the remaining bytes of a certain page are less than the bytes of the packet that is going to be stored, a new write command needs to be issued. Thus the code, before every write operation, needs to check whether the packet size is greater than the remaining space in the current page, then if the space is sufficient, a single write operation is performed otherwise the packet is split in such a way that the remaining space in the current page is filled with the first part of the packet, the other part is then written to the next page. The packet format used is the same that is transmitted namely the first byte contains the number of active thermocouples, the next bytes contain the ADC codes of the thermocouples' voltages, after that there are two bytes for the ADT7410 raw temperature while the last 4 bytes represents the timestamp in Unix format. This byte sequence though, is used only during the prototyping phase. In the final implementation the memory will be filled with temperature values already converted (not raw data) and a compressed timestamp based on the same encoding used to decide the measurement frequency.

3.3.6 Low power software considerations

For the hardware part the current consumption minimization is obtained by connecting the ground pin of the different modules to an NMOS transistor used as a switch in order to cut their supply. This approach allows to eliminate also the small current that would be present if the module had been put in its power down mode. Regarding the software part, other aspects need to be discussed. The first one is related to the timers that are used; in particular there are two possible solutions, the first one uses the timer peripheral that relies on the 16 MHz quartz oscillator the other one instead uses the RTC which relies on the 32 kHz low frequency oscillator. The latter is much more efficient indeed it can work with only 0.1 μA compared to more than 1 mA of the timer peripheral therefore the choice is for the RTC. The other consideration is related to the peripheral's use, in particular the SPI, the I^2C and the UART. When they are active, their current consumption is

of the order of 1 mA therefore it is fundamental to disable them as soon as the communication is finished. In addition tuning advertising parameters is essential in fact the longer the advertising interval, the less current will the device consume during advertising. The only drawback is that the time required by a central device to connect to the sensor increases. For this reason in the first 60 seconds after a disconnection the advertising is fast namely there is a new advertising packet every 0.3 s, then for the rest of the time the device enters the slow advertising state where the period between packets increases up to 10 s.

3.4 Receiver firmware and data visualization

The reception of data sent by the LoRa module mounted on the PCB is performed by another LoRa module working as a receiver that is controlled by an Arduino Nano. The same function can be done by any other microcontroller. When a new packet is detected, an integrity verification of the bytes received is performed before further processing. This is done exploiting a mechanism called CRC which involves adding a certain number of redundant bits to a data stream before transmission. The receiver calculates the CRC based on the received data and compares this newly calculated CRC with the CRC appended to the end of the packet. If the two coincides, the packet can be processed otherwise it is discharged. The processing consists in the reconstruction of the different ADC codes corresponding to the thermocouples' voltages, the reconstruction of the reference junction temperature data and the reconstruction of the timestamp. This is due to the splitting operation performed at the transmitter side where data are sent as sequences of 8 bit. The received quantities are still raw data therefore they are converted into actual voltages (for thermocouples) and temperature (for the ADT7410) before being serially sent. After the conversion, data are ready to be sent to the Python application for real time visualization. The communication between Python and Arduino is managed by the serial interface thus the Arduino code writes the packets on the serial interface and the Python application retrieves them exploiting the serial library. Once data are available, to get an actual temperature from the thermocouples' readings it is necessary to perform the compensation procedure. This is done in three steps:

1. Conversion of the reference junction temperature measured by the ADT7410 into an equivalent thermoelectric voltage

2. Summation of the equivalent reference junction temperature voltage to the thermocouple voltage

3. Conversion of the total voltage to the final temperature

Given the possibility to measure both positive and negative temperatures, the coefficients of the polynomial equations used for the conversions are different depending on the temperature and voltage sign. More details about thermocouples can be found in Appendix A. After the computation of the temperature values of the different thermocouples it is possible to visualize all the results. In particular the Python script implements three main functionalities: the first one is the visualization of temperature values in the text format on the PC terminal, the second one is the graphical visualization of them with separate plots for each thermocouple while the third functionality is related to the data storing in a comma-separated values (csv) file for post processing purposes. Figure 3.8 reports the text window while Figure 3.9 reports the graphical representation of the incoming data regarding thermocouple number 1.

```
09:21:17 -> TC 1: 20.082 °C, TC 2: 20.06 °C, TC 3: 20.018 °C, TC 4: 20.057 °C, TC 5: 20.092 °C, TC 6: 20.025 °C, ADT7410: 20.141 °C
09:21:21 -> TC 1: 20.082 °C, TC 2: 20.073 °C, TC 3: 20.035 °C, TC 4: 20.06 °C, TC 5: 20.112 °C, TC 6: 20.018 °C, ADT7410: 20.148 °C
09:21:25 -> TC 1: 20.073 °C, TC 2: 20.073 °C, TC 3: 20.03 °C, TC 4: 20.065 °C, TC 5: 20.107 °C, TC 6: 20.025 °C, ADT7410: 20.148 °C
09:21:29 -> TC 1: 20.096 °C, TC 2: 20.108 °C, TC 3: 20.054 °C, TC 4: 20.084 °C, TC 5: 20.123 °C, TC 6: 20.051 °C, ADT7410: 20.172 °C
09:21:33 -> TC 1: 20.077 °C, TC 2: 20.06 °C, TC 3: 20.018 °C, TC 4: 20.047 °C, TC 5: 20.104 °C, TC 6: 20.013 °C, ADT7410: 20.141 °C
09:21:37 -> TC 1: 20.098 °C, TC 2: 20.081 °C, TC 3: 20.049 °C, TC 4: 20.071 °C, TC 5: 20.116 °C, TC 6: 20.051 °C, ADT7410: 20.164 °C
09:21:41 -> TC 1: 20.098 °C, TC 2: 20.089 °C, TC 3: 20.061 °C, TC 4: 20.081 °C, TC 5: 20.118 °C, TC 6: 20.041 °C, ADT7410: 20.172 °C
09:21:45 -> TC 1: 20.086 °C, TC 2: 20.101 °C, TC 3: 20.041 °C, TC 4: 20.054 °C, TC 5: 20.111 °C, TC 6: 20.029 °C, ADT7410: 20.172 °C
09:21:49 -> TC 1: 20.106 °C, TC 2: 20.096 °C, TC 3: 20.054 °C, TC 4: 20.081 °C, TC 5: 20.114 °C, TC 6: 20.037 °C, ADT7410: 20.18 °C
09:21:53 -> TC 1: 20.097 °C, TC 2: 20.097 °C, TC 3: 20.037 °C, TC 4: 20.062 °C, TC 5: 20.114 °C, TC 6: 20.047 °C, ADT7410: 20.188 °C
```

Figure 3.8: Text form data visualization

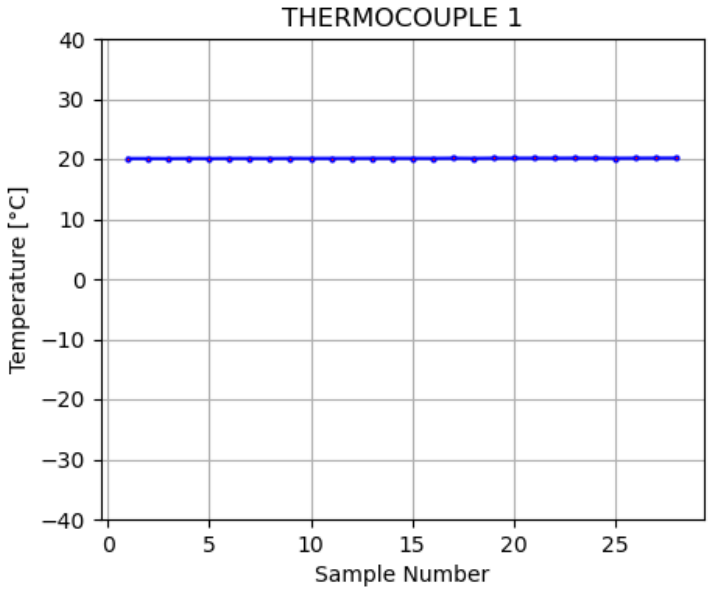


Figure 3.9: Graphical form data visualization

Chapter 4

System verification and metrological characterization

This chapter presents the experimental results obtained by testing the device's performance, specifically focusing on measurement accuracy and current consumption. Beyond these key aspects, various tests were executed to evaluate the transmission range achievable with different radio configurations, and to validate the correct functionality of the wireless charging and the reset circuit. All steps required for the computation of the uncertainty characterizing the data acquisition chain are also reported. Before starting the different tests the fabricated PCB needs to be assembled, therefore the preliminary step required soldering all the SMD components to the board. This was done with the aid of a microscope, a soldering tool for the majority of the components, while an heated plate was employed for the microcontroller due to some pins being situated beneath its footprint. Figure 4.1 shows the front side of the PCB where most of the modules are placed, while Figure 4.2 reports the back side with the circuitry required to reset the microcontroller. The final dimensions of the PCB are 7 cm for the length and 5 cm for the width

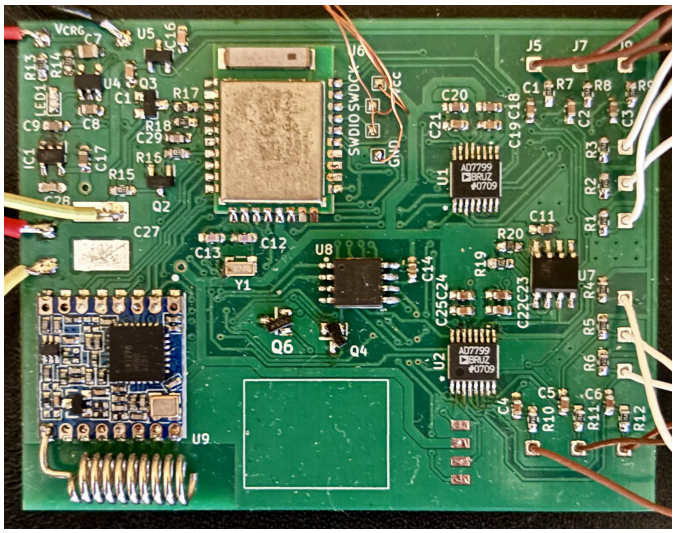


Figure 4.1: Assembled PCB front side

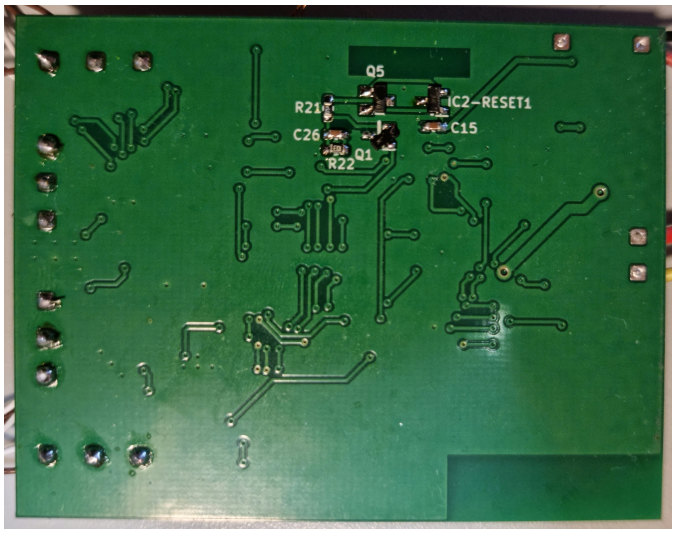


Figure 4.2: Assembled PCB back side

4.1 Uncertainty evaluation

The uncertainty of the data acquisition chain is evaluated exploiting the probabilistic model which represents a better alternative to the deterministic one which overestimates the final result. Considering a positive temperature range, the measurement model is defined as

$$\theta = \sum_{i=1}^{6} k_i \cdot e_\theta^i \tag{4.1}$$

where k_i represents the i^{th} coefficient of the polynomial function used to retrieve the temperature value θ from the voltage e_θ^i which represents the equivalent voltage obtained by the reference junction compensation procedure. In fact this voltage can be written as

$$e_\theta = e_{\Delta\theta} + e_{REF} \tag{4.2}$$

where $e_{\Delta\theta}$ is the voltage generated by the thermocouple while e_{REF} is the voltage corresponding to the reference junction temperature obtained by applying the direct polynomial equation. Now it is possible to expand these two terms in order to highlight the uncertainty contributions.

$$e_{\Delta\theta} = \frac{V_{REF}}{G} \cdot (\frac{D}{2^{N-1}} - 1) \tag{4.3}$$

$$e_{REF} = \sum_{j=1}^{8} h_j \cdot \theta_{REF}^j \tag{4.4}$$

the equation 4.3 is obtained starting from the expression of the output code of the ADC employed which uses an offset binary coding. It is possible to see that three terms generate uncertainty contributions namely the reference voltage of the ADC V_{REF}, the ADC itself with the output D and the gain of the PGA indicated with G even though the ADC's specification do not point out the actual uncertainty on the gain, as a consequence it is assumed to be already included in the term D. Equation 4.4 instead reports the direct polynomial expression required to get the voltage corresponding to the reference junction temperature θ_{REF} measured by the ADT7410. Putting all together leads the complete measurement model

$$\theta = \sum_{i=1}^{6} k_i \cdot \left[\frac{V_{REF}}{G} \cdot (\frac{D}{2^{N-1}} - 1) + \sum_{j=1}^{8} h_j \cdot \theta_{REF}^j \right]^i \tag{4.5}$$

From 4.5 it is possible to identify the main uncertainty contributions

- Thermocouple uncertainty associated with the coefficients $k_i \rightarrow$ It depends both on the type of the thermocouple and its class. In the case of class 1, T type thermocouple, the absolute uncertainty in the measurement range is $\pm 0.5\ ^{\circ}C$.

- Reference voltage uncertainty $\rightarrow$ the datasheet of the component states a relative uncertainty of 0.15 % as a consequence given the reference voltage of 1.25 V, the absolute uncertainty becomes 1.9 mV.

- Reference junction temperature sensor uncertainty $\rightarrow$ the datasheet of the component reports an absolute uncertainty of $0.5\ ^{\circ}C$.

- ADC uncertainty represented by the term D $\rightarrow$ the uncertainty of the ADC is typically expressed by the total unadjasted error (TUE) that is the total uncompensated error that includes Quantization, Offset, Gain and Non-Linearity Errors.

The computation of the TUE can be performed considering the separate uncertainty contributions reported in the datasheet of the ADC, in particular the AD7799. For simplicity they are reported in Figure 4.3 which is extracted from the datasheet of the component.

Table 1.

Parameter	AD7798B/AD7799B[1]	Unit	Test Conditions/Comments
ADC CHANNEL			
Output Update Rate	4.17 – 470	Hz nom	
No Missing Codes[2]	24	Bits min	AD7799: f_{ADC} < 242 Hz
	16	Bits min	AD7798
Resolution			See Table 5 to Table 8
Output Noise and Update Rates			See Table 5 to Table 8
Integral Nonlinearity	±15	ppm of FSR max	
Offset Error[3]	±1	µV typ	
Offset Error Drift vs. Temperature[4]	±10	nV/°C typ	
Full-Scale Error[3, 5]	±10	µV typ	
Gain Drift vs. Temperature[4]	±1	ppm/°C typ	

Figure 4.3: AD7799 uncertainty contributions

The important contributions are the Integral Nonlinearity which is stated as ±15 ppm of FSR max, the Offset Error of $\pm 1\ \mu V$ typical, the Full-Scale Error of $\pm 10\ \mu V$ typical and the quantization error that is computed as 0.5 LSB which is equal to 75.5 nV. Considering that the full scale range max is 1.25 V, the Integral Nonlinearity becomes equal to $18.75\ \mu V$. With these terms it is possible to compute the TUE as

$$\begin{aligned} \text{TUE} &= E_q + INL + E_{\text{offset}} + E_{\text{full-scale}} \\ &= \frac{1}{2} \cdot \frac{2 \cdot V_{REF}}{2^{24}} + \frac{15}{10^6} \cdot V_{REF} + 1\,\mu\text{V} + 10\,\mu\text{V} \\ &= 29.82\,\mu\text{V} \rightarrow 200.15\,\text{LSB} \end{aligned} \tag{4.6}$$

Now that all the uncertainty contributions are defined as absolute uncertainty, it is necessary to convert them in standard uncertainties to correctly apply the probabilistic model. A typical choice that it is done is to consider the measurand as a random variable uniformly distributed within a certain interval centered on the measurand value itself. This means that there is zero probability to find the measured outside the stated interval and equally distributed probability inside the interval. The square root of the variance of the random variable then is the standard uncertainty. By applying the definition of variance it is possible to demonstrate that the generic standard uncertainty is $\text{u(x)} = \frac{\delta(x)}{\sqrt{3}}$ where $\delta(x)$ is the absolute uncertainty. The standard uncertainty contributions for all the terms are reported below

- **Thermocouple** $\rightarrow$ $\text{u(TC)} = 0.29\,^{\circ}\text{C}$
- **Reference voltage** $\rightarrow$ $\text{u}(V_{REF}) = 1.1\,\text{mV}$
- **Reference junction thermometer** $\rightarrow$ $\text{u}(\theta_{REF}) = 0.29\,^{\circ}\text{C}$
- **Analog to digital converter** $\rightarrow$ $\text{u(ADC)} = 115.6\,\text{LSB}$

The standard uncertainty associated with the final temperature value is obtained considering the combined standard uncertainty expression

$$u_{\text{c}}^2(\theta) = \sum_{i=1}^{\text{N}} \left(\frac{\partial \theta}{\partial x_i} \right)^2 \cdot u^2\left(x_i\right) + 2 \sum_{\substack{j=1 \\ j \neq k}}^{\text{N}-1} \sum_{k=1}^{\text{N}} \frac{\partial \theta}{\partial x_j} \frac{\partial \theta}{\partial x_k} u\left(x_j\right) \cdot u\left(x_k\right) \tag{4.7}$$

due to the fact that there is no correlation between any variable, it is possible to further simplify the equation as

$$u_{\text{c}}^2(\theta) = \sum_{i=1}^{\text{N}} \left(\frac{\partial \theta}{\partial x_i} \right)^2 \cdot u^2\left(x_i\right) \tag{4.8}$$

Now the remaining part is the computation of the sensitivity coefficients namely the partial derivatives of θ with respect to all the variables present in the measurement model.

- Reference voltage contribution

$$\begin{aligned}\frac{\partial\theta}{\partial V_{REF}} &= \frac{1}{G}\cdot\left(\frac{D}{2^{N-1}}-1\right)\cdot\sum_{i=1}^{6} i\cdot k_i\cdot\left[\frac{V_{REF}}{G}\cdot\left(\frac{D}{2^{N-1}}-1\right)\cdot 1000+e_{REF}\right]^{i-1}\\ &= 0.0414\ \frac{^\circ C}{mV}\end{aligned} \tag{4.9}$$

- ADC contribution

$$\begin{aligned}\frac{\partial\theta}{\partial D} &= \frac{1000\cdot V_{REF}}{2^{N-1}\cdot G}\cdot\sum_{i=1}^{6} i\cdot k_i\cdot\left[\frac{V_{REF}}{G}\left(\frac{D}{2^{N-1}}-1\right)\cdot 1000+e_{REF}\right]^{i-1}\\ &= 2.43\times 10^{-5}\ \frac{^\circ C}{LSB}\end{aligned} \tag{4.10}$$

- Reference junction temperature contribution

$$\begin{aligned}\frac{\partial\theta}{\partial\theta_{\mathrm{REF}}} &= \sum_{i=1}^{6} i\cdot k_i\cdot e_\theta^{i-1}\cdot\left[\sum_{j=1}^{8} j\cdot h_j\cdot(\theta_{\mathrm{REF}})^{j-1}\right]\\ &= 0.9141\end{aligned} \tag{4.11}$$

The value of e_{REF} has been computed starting from the maximum temperature that the sensor is going to measure namely 60 °C and by applying the direct polynomial equation. The gain G instead is 128 while the the reference voltage V_{REF} is 1.25 V and the digital code of the ADC is 10510322 which is the maximum ADC code for the temperature range considered. Now putting all together yields to the final expression that allows to compute the standard uncertainty associated with the temperature measured by the device in the worst case of maximum temperature.

$$u_c(\theta)=\sqrt{S_{V_{REF}}^2\cdot u(V_{REF})^2+S_D^2\cdot u(D)^2+S_{T_{REF}}^2\cdot u(\theta_{REF})^2+u(TC)^2} \tag{4.12}$$

where the coefficients S are used to represent the sensitivity coefficients given by the partial derivatives computed above. Substituting all the numbers the final result is

$$u_c(\theta)=0.45\,^\circ\mathrm{C} \tag{4.13}$$

At the beginning it was stated that the computations are performed considering the positive temperature range namely from 0 to 60 °C. For the negative interval that goes from -60 to 0 °C the equations remain pretty much the same. The only changes are related to the number of coefficients and their values, used by the

polynomial equations that perform the conversion from voltage to temperature and vice versa and of course the values of the variables D, θ_{REF} and e_{REF}. For this reason it is reported only the final result that is

$$u_c(\theta) = 0.5\,^{\circ}\mathrm{C} \tag{4.14}$$

For completeness, the coefficients of the polynomial relations used in the computations are reported in the following tables. The first one reports the direct polynomial coefficients used to convert a temperature to a voltage while the second one reports the inverse polynomial coefficients used to convert a voltage to a temperature therefore it is the calibration function.

Coefficient	**Coefficient Value**
h_1	$3.8748106364 \times 10^{-2}$
h_2	$3.3292227880 \times 10^{-5}$
h_3	$2.0618243404 \times 10^{-7}$
h_4	$-2.1882256846 \times 10^{-9}$
h_5	$1.0996880928 \times 10^{-11}$
h_6	$-3.08157587720 \times 10^{-14}$
h_7	$4.5479135290 \times 10^{-17}$
h_8	$-2.7512901673 \times 10^{-20}$

Table 4.1: Direct polynomial coefficients valid from 0 to 400 °C

Coefficient	**Coefficient Value**
k_1	2.592800×10^{1}
k_2	-7.602961×10^{-1}
k_3	4.637791×10^{-2}
k_4	-2.165394×10^{-3}
k_5	6.048144×10^{-5}
k_6	-7.293422×10^{-7}

Table 4.2: Inverse polynomial coefficients valid from 0 to 400 °C

4.2 Temperature accuracy verification

To verify the performance of the device in terms of measurement accuracy, namely how close the temperature provided by the measuring system is to a specific reference temperature, different tests were conducted. The setup involves the use of a climatic chamber which is a special device used to simulate and maintain specific temperature and humidity conditions for testing purposes. It is equipped with a display from which temperature and humidity can be controlled, in addition it allows to generate specific profiles where the rate of change of the two quantities is accurately regulated. For the needs of this project, only the temperature control is exploited. Figure 4.4 reports a picture of the climatic chamber utilized

Figure 4.4: Climatic chamber

In addition the setup involves the use of a Dewar where fragmented melted ice cubes are inserted to generate a stable 0 °C point that acts as a reference temperature. To verify the measurement accuracy, the PCB as well as one of the six thermocouples were placed inside the climatic chamber while the remaining five thermocouples were brought outside thanks to a circular aperture in the machine and inserted in the Dewar that contains the ice bath. Therefore instead of changing the temperature measured by the thermocouples what is changed is the temperature experienced by the reference junction thermometer placed on the surface of the PCB. Using this technique provides one additional advantage with respect to directly inserting the thermocouples inside the climatic chamber namely the possibility

to test the device's resistance to significant temperature variations. Ideally the temperature readings of the five thermocouples inside the Dewar should be as close as possible to 0 °C for every temperature at which the device is exposed within the measurement range considered of -60 °C to +60 °C. To generated the most difficult conditions possible, every temperature transient is performed at the maximum rate intended as degree Celsius per second, allowed by the climatic chamber. Figure 4.5 shows the device inside the climatic chamber while Figure 4.6 shows the Dewar that maintains the thermocouples in the ice bath

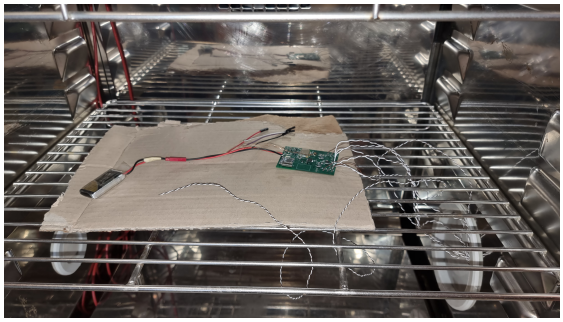


Figure 4.5: Wireless sensor inside the climatic chamber

Figure 4.6: Dewar used to create the 0 °C reference

4.2.1 Positive temperature transient

In the first test the temperature of the climatic chamber has been increased from the ambient temperature of about 22 °C to 40 °C in order to have a final temperature difference across the thermocouples' leads of 40 °C. Figure 4.7 reports the temperature values acquired by all the 6 thermocouples and the reference junction thermometer that is the ADT7410. As already anticipated before, one thermocouple was placed inside the climatic chamber, in particular the number 5

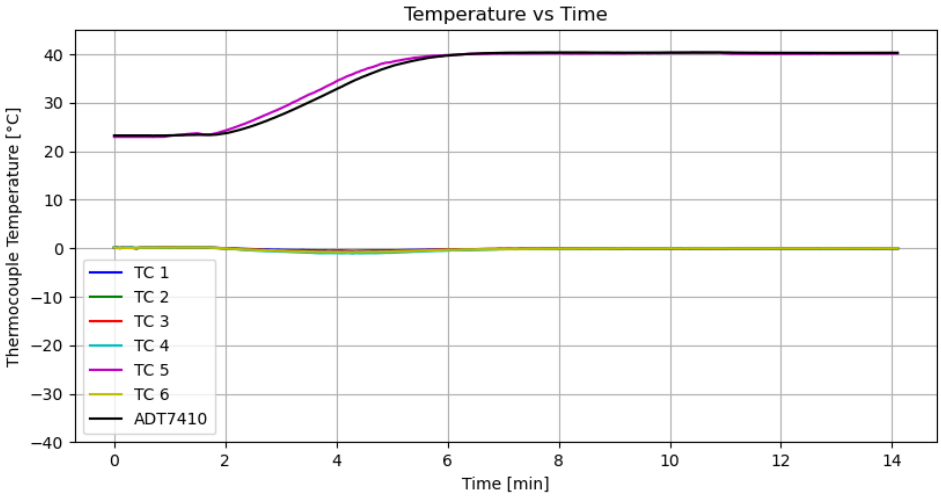


Figure 4.7: 22 °C to 40 °C temperature variation

while all the others were in the ice bath. The plotted data shows that the measured values by TC 1, 2, 3, 4, and 6 are consistently close to 0 °C, except during the transient period when the climatic chamber starts blowing hot air to reach the desired temperature. This discrepancy is primarily due to a slight temperature difference between the point where the reference junction of the thermocouples is connected and the temperature sensed by the ADT7410. In Figure 4.8, it's evident that there is a deviation towards negative temperature values. This indicates that the reference junction experiences a temperature increase before the thermometer, namely the ADT7410, detects the change. The thermocouple's measuring junction remains at 0 °C within the ice bath, while the reference junction's temperature increases due to the transient. Consequently, a more negative voltage develops across the reference junction, leading to an overall temperature decrease after the compensation procedure. With a peak value of -1 °C the temperature reached during the transient is not within the uncertainty obtained in the previous section but this is correct because the model used did not take into account the thermal gradient between the cold junction and the reference temperature sensor. It is important to highlight that the temperature transient generated by the climatic chamber is much higher than what will be experienced by the device during a freeze dryer recipe, as a consequence this tests are meant to put the system in the most difficult condition possible. Figure 4.9 reports the measurements in steady conditions namely when the transient is extinguished. In this case the temperature values obtained are quite accurate since there is a maximum deviation of 0.1 °C from the reference temperature of the ice bath.

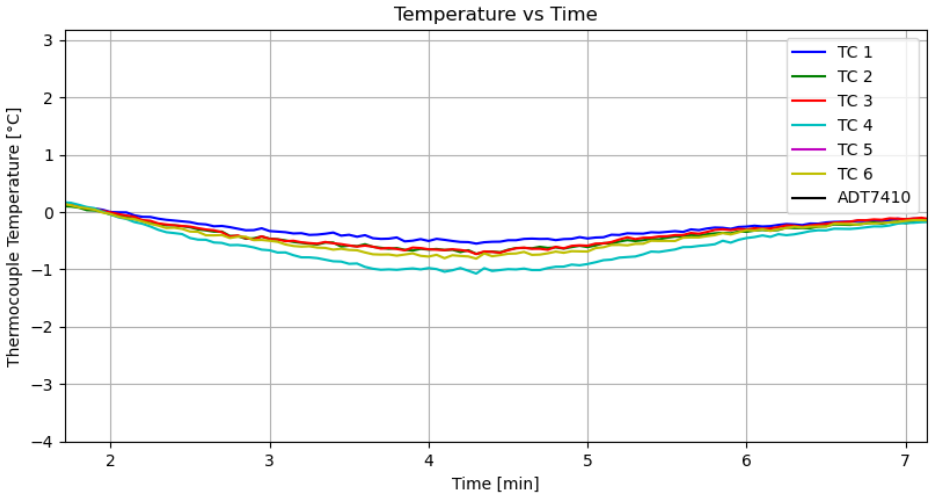


Figure 4.8: 22 °C to 40 °C temperature variation: transient detail

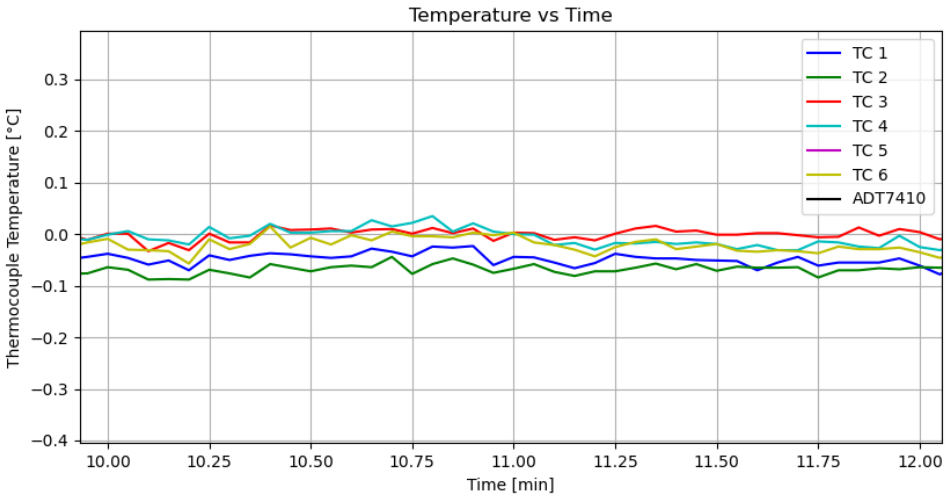


Figure 4.9: 22 °C to 40 °C temperature variation: steady state detail

4.2.2 Negative temperature transient

In order to test the accuracy of the device during negative temperatures, a different transient was generated. In this case, starting from 40 °C the temperature is decreased up to -35 °C which is close to the limit of the climatic chamber. The result is reported in Figure 4.10. Due to the very steep initial transient, it is expected to have a more significant deviation of the measured values with respect to the previous case during the first minutes of the test. Furthermore, the situation now is reversed since the deviation is toward positive temperatures instead of negative ones. This is due to the fact that the reference junction becomes colder at a faster rate with respect to the reference thermometer, therefore a more positive voltage is developed at the thermocouples' ends contributing to an overall temperature increment after the compensation procedure.

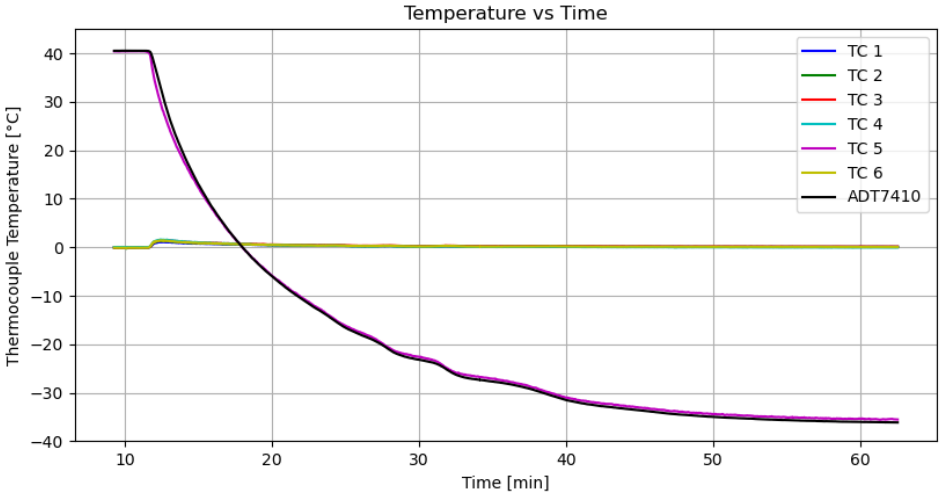


Figure 4.10: 40 °C to -35 °C temperature variation

Figure 4.11 confirms the positive and higher deviation from the 0 °C point during the negative transient which now reaches up to 1.5 °C. The situation at steady state is reported in Figure 4.12 where a zoomed version of the last minutes of the test is shown. Also in this case the accuracy of the measurements is well below the stated uncertainty. The small difference in terms of absolute temperature that is measured between the two tests at steady state can be related to the non homogeneity of the fragmented ice of the Dewar, indeed small adjustments in the position of the thermocouples, leads to small temperature variations of the order of 0.05 °C. Regarding the values obtained from TC 5 which is the thermocouple placed inside the climatic chamber, they are consistent with the values of the reference junction thermometer.

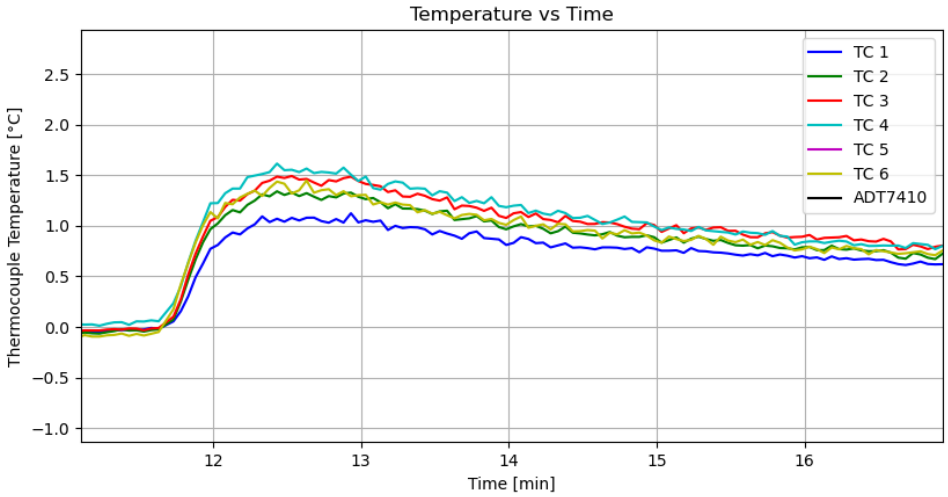


Figure 4.11: 40 °C to -35 °C temperature variation: transient detail

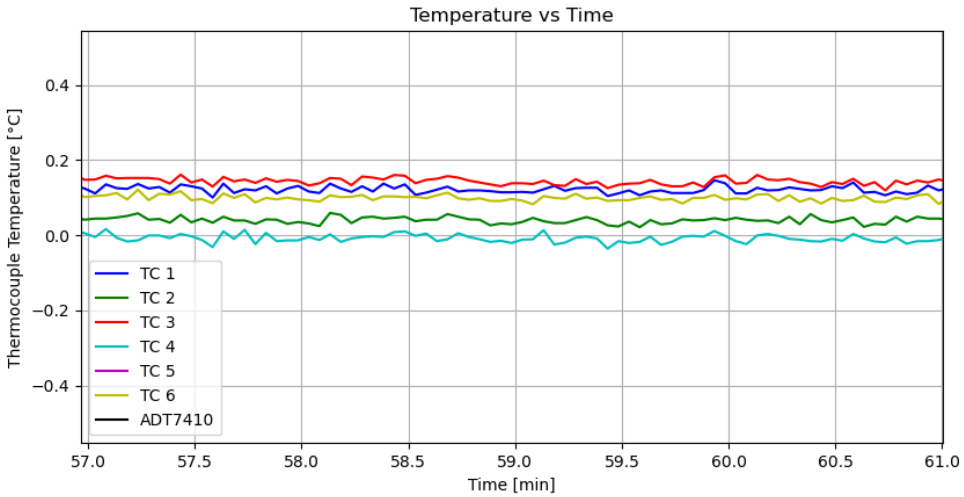


Figure 4.12: 40 °C to -35 °C temperature variation: steady state detail

In the last experimental verification, starting from -35 °C, the climatic chamber was stopped in such a way that the temperature could slowly rise. This allowed to verify the behaviour of the measuring system with a significantly less steep transient. In particular due to the reduced thermal gradient between the reference junction and its thermometer, no more evident variations should appear. Figure 4.13 reports the overall view where the focus is on the first 10 minutes where the slope of the black and magenta curves is less steep. By focusing on that first interval

as it is done in Figure 4.14, it is possible to see that the results are comparable to a steady state condition even though the temperature in the chamber is still changing. From all these experimental tests it is possible to state that the biggest uncertainty contribution is related to the thermal gradient between the reference junction and the thermometer used to measure its temperature. The temperature variations due the transients are expected to reduce quite significantly (at least one order of magnitude) after that the PCB will be resin coated, thanks to the lower thermal gradient between the thermocouples' reference junction and the digital thermometer. The resin in fact blocks air currents that now can circulate on the PCB.

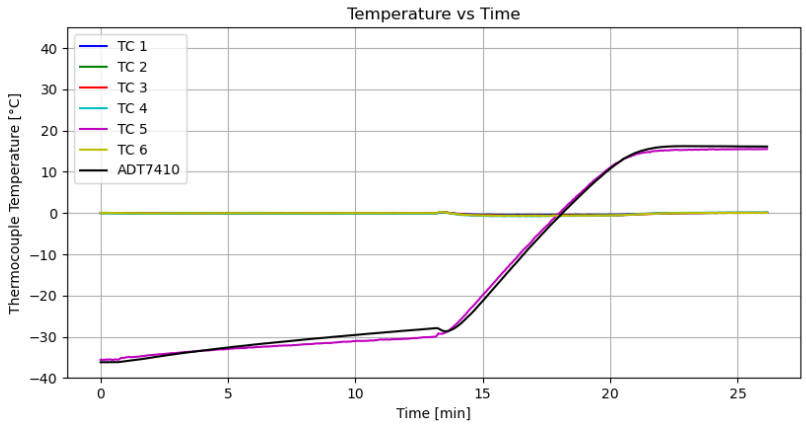


Figure 4.13: -35 °C to 18 °C temperature variation

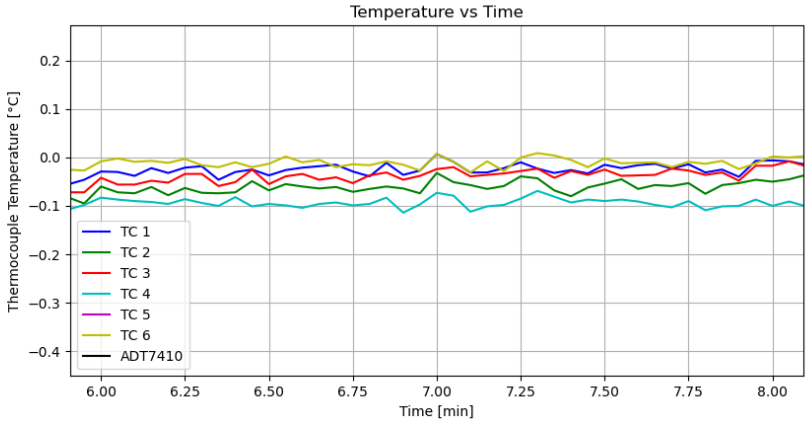


Figure 4.14: -35 °C to 18 °C temperature variation: transient detail

4.3 Current consumption evaluation

The current consumption of a battery powered device represents one of the most important parameters that determines the amount of time the device will operate, for this reason it needs to be the lowest possible compatibly with the characteristic and performance of the device itself. There are different possibilities that can be exploited to measure a current but not all of them can always be employed. In fact it depends whether the values under analysis are high, low, constant or not. For the evaluation of the current consumption of the wireless sensor, two setup are implemented depending on the configuration of the device. The first type of measurements is performed when it is in sleep mode, namely it is not measuring, all the modules are in low power mode or directly disconnected from the supply and the only activity derives from the advertising packets sent by the Bluetooth transceiver every 10 seconds. The currents involved during this process are expected to be very low, of the order of the micro Ampere, and with periodic peaks for this reason the method used employs a big capacitor and an oscilloscope to measure the voltage across it. The capacitor is a 2200 μF and it is hooked up in parallel with the battery, then after some seconds the battery is disconnected leaving the capacitor as the only energy source of the circuit. By looking at the voltage discharge transient it is possible to retrieve the amount of charge provided by the capacitor (integral of the voltage) in a specific time interval to eventually get the current.

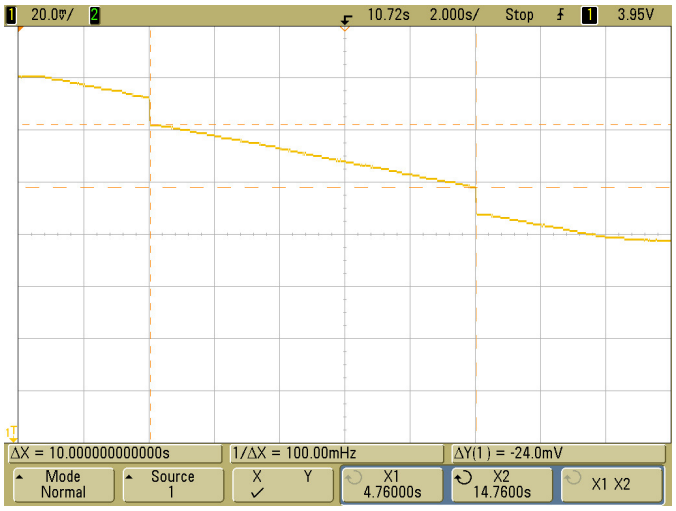


Figure 4.15: Capacitor discharge transient during sleep mode

Figure 4.15 shows the signal captured by the oscilloscope during the sleep mode. It can be seen that in 10 seconds, the voltage drops of 24 mV therefore the charge provided by the capacitor can be computed as reported is 4.15.

$$\Delta Q = C \cdot \Delta V = 52.8\,\mu\text{C} \tag{4.15}$$

The current is defined as the charge per unit of time therefore

$$I = \frac{\Delta Q}{\Delta t} = \frac{52.8\,\mu\text{C}}{10\,\text{s}} = 5.28\,\mu\text{A} \tag{4.16}$$

This value represents the amount of current that is drawn by the circuit without considering the advertising packets. When they are considered the voltage variation become 43 mV therefore the current increases up to 9.46 μA. The current consumption's obtained are still not the correct ones due to the parasitic elements of the capacitor used to indirectly measure that current and the oscilloscope load. In particular the parallel resistance also known as leakage resistance is responsible for increasing a little bit the current consumption. To quantify its effect, the setup remains the same with the capacitor connected in parallel with the battery, the difference is that, after it is charged up, the positive lead is disconnected from the circuit. The oscilloscope now acquires the voltage transient on the capacitor due only to the self discharge. Figure 4.16 reports the waveform acquired where there are two lines visible with different slopes: the more steep one represents the contribution of both the self discharge of the capacitor and the current drawn by the device whereas the other one represents the only self discharge contribution. The latter has been computed and it is 1.94 μA therefore the actual current consumption of the device during the sleep state without advertising packets and with advertising packets is 3.34 μA and 7.52 μA respectively. Considering a Lithium battery with a capacity of 400 mAh, the device is expected to operate for a few years in low power mode.

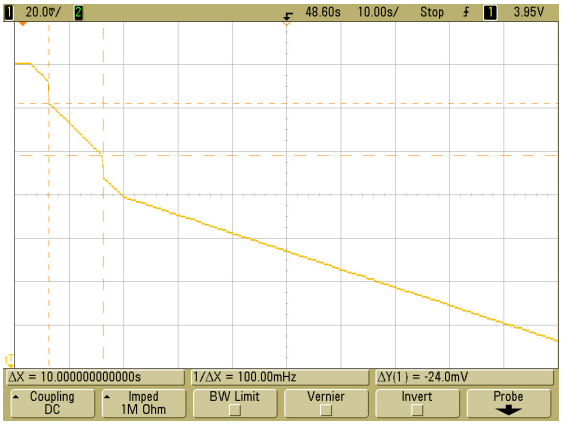


Figure 4.16: Capacitor self discharge

The next set of measurements is performed when the device is active. The measuring setup is changed since the current values are higher than during sleep mode therefore measuring the voltage drop across a capacitor is no longer the best approach due to the high amount of charge that would be required. The alternative is based on another indirect measurement based on a 10 Ω resistor placed in series with the battery. When the circuit is powered, the current flowing through the resistor generates a voltage drop across it that can be analyzed with the oscilloscope. The current values are obtained considering the value set for the vertical scale and dividing it by the resistor's value (10 Ω in this case). Figure 4.17 reports the voltage waveform across the resistor generated by the current drawn by the advertising packet

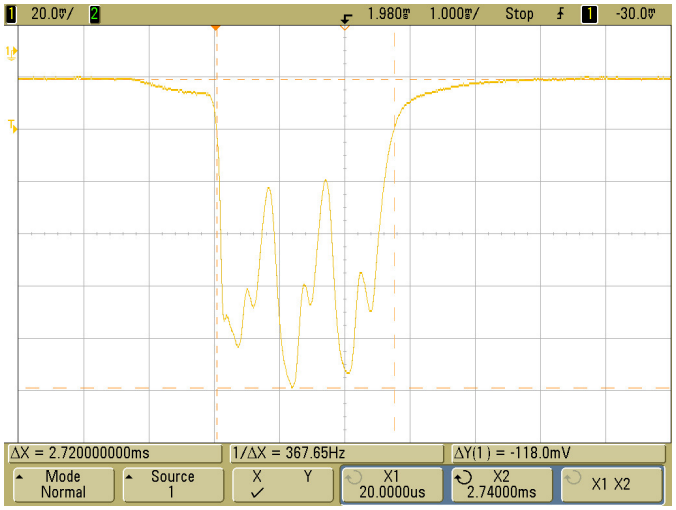


Figure 4.17: Advertising packet

It can be noted that the vertical scale of the oscilloscope is set to 20 mV/div therefore dividing it by 10 Ω provides 2 mA/div. As a consequence it is possible to compute the current peak that is 11.8 mA.

4.3.1 Measuring phase: LoRa configuration 1

The acquisition reported in Figure 4.18 represents the voltage waveform obtained when the device is measuring and sending data through the long range radio. This test was conducted with the LoRa module having a spreading factor (SF) of 7, a bandwidth (BW) of 125 kHz and a coding rate (CR) of 4/5. These parameters allow to reach an effective data rate 5.4 kbps and a theoretical receiver sensitivity of -121 dBm. The time interval between two consecutive measurements for this test was set to 5 seconds. From the waveform reported, it is possible to identify three

intervals. The first one with a period of 3.25 seconds where the current consumption is 3.3 μA, measured also with the help of a digital multimeter. The second one with a period of 1.69 seconds and a current consumption of 8 mA. This is the time interval needed to acquire the 6 thermocouples' channels, the temperature of the ADT7410 and to perform the write operation to the non volatile memory. The last interval has a period of 62 ms that is the time required by the long range radio to send each packet to the receiver for the real time data visualization. Each packet is formed by a preamble, an header, 25 byte for the payload and 2 byte for the CRC check. This is also called time on air and its duration is visible in Figure 4.19. For this interval the current consumption reaches 134 mA.

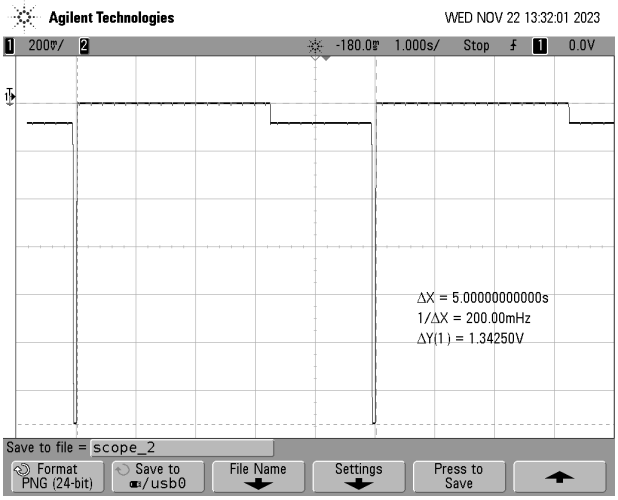


Figure 4.18: Voltage waveform during measurement phase, configuration 1

From these data it is possible to compute the average current consumption during the period as

$$I_{\text{avg}} = \frac{1}{T}\int_0^T i(t)\,dt \tag{4.17}$$

$$\begin{aligned} I_{\text{avg}} &= \frac{1}{T}\left(\int_0^{T_1} I_1\,dt + \int_{T_1}^{T_2} I_2\,dt + \int_{T_2}^{T_3} I_3\,dt\right) \\ &= \frac{1}{T}\left(I_1 \cdot T_1 + I_2 \cdot T_2 + I_3 \cdot T_3\right) \\ &= 4.4\ mA \end{aligned} \tag{4.18}$$

Considering a 400 mAh Li-Ion battery, the duration in these conditions, namely six thermocouples acquiring every 5 seconds, is expected to be of 90 hours which

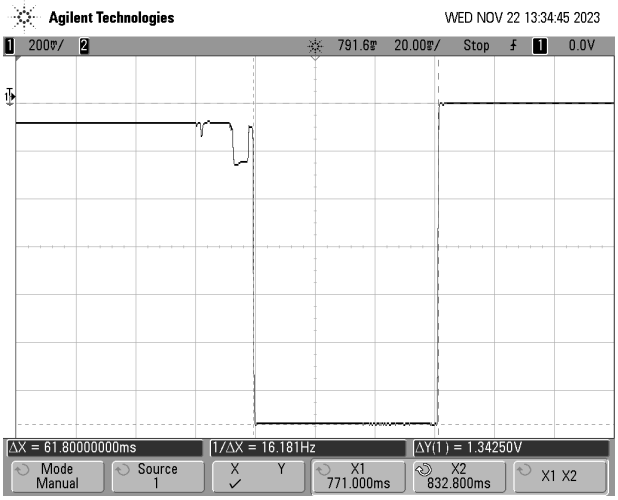


Figure 4.19: Time on air, configuration 1

corresponds to 3.75 days. Extending the time interval between two successive measurements enhances the overall battery lifespan. Table 4.3 provides a summary of the current consumption, the corresponding duration and the estimated duration of the battery for some of the possible different configurations of the device namely different working states and different measurement intervals. The battery is assumed to have a capacity of 400 mAh which is a common value for Li-Ion battery.

Mode	Time Interval	Average Current	Duration
Active	5 s	4.4 mA	90 hours
Active	60 s	0.37 mA	37 days
Active	5 min	76 μA	219 days
Sleep	/	3.3 μA	few years

Table 4.3: Battery duration for different device configurations

4.3.2 Measuring phase: LoRa configuration 2

Another LoRa configuration tested uses a spreading factor of 9, a bandwidth of 62.5 kHz so half than before and the same coding rate of 4/5. In this conditions the effective data rate reduces to 878 bps but the theoretical receiver sensitivity increases up to -130 dBm. This means that the transmission takes longer in terms of time on air but the overall range increases. Figure 4.22 reports a portion of the voltage waveform during the measurement phase while Figure 4.23 shows the

time on air required to send a single packet. It can be noted that the time on air increases from 62 ms to 410 ms therefore the average current consumption, for a 5 seconds time interval between measurements, increases from 4.4 mA, to 18.5 mA.

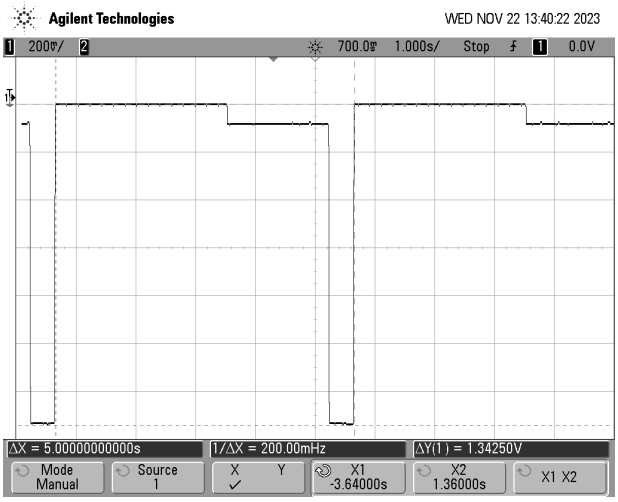


Figure 4.20: Voltage waveform during measurement phase, configuration 2

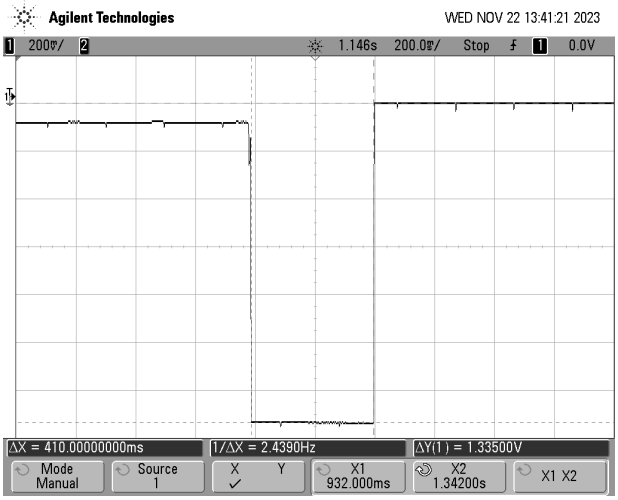


Figure 4.21: Time on air, configuration 2

4.3.3 Measuring phase: LoRa configuration 3

The last configuration tested uses a spreading factor of 11, a bandwidth of 62.5 kHz and the same coding rate of 4/5. The data rate further reduces to 268 bps while

the receiver sensitivity increases to -134 dBm. Figure 4.20 reports a portion of the voltage waveform during the measurement phase while Figure 4.21 shows the time on air required to send a single packet. As it can be seen, the time on air increases up to 1.65 seconds leading to an average current consumption of 47 mA so more than 10 times greater than the first configuration. From these tests it is evident that a balance between current consumption and transmission range should be considered depending on the type of application. In fact for environmental temperature monitoring, where the variations are much lower than during lyophilization, these parameters could be used since the acquisition of new values is less frequent.

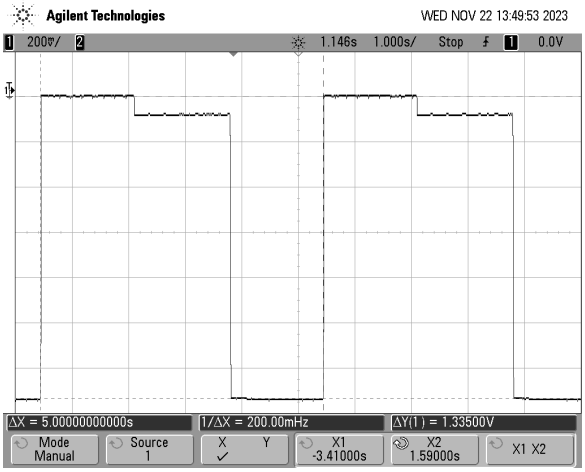


Figure 4.22: Voltage waveform during measurement phase, configuration 3

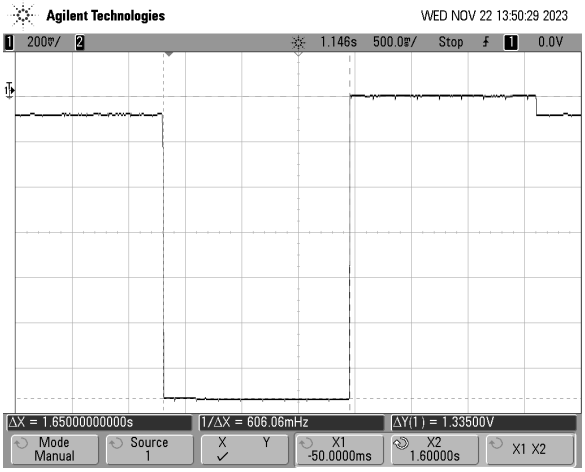


Figure 4.23: Time on air, configuration 3

Table 4.4 provides a summary of the current consumption and the estimated

battery duration for some acquisition intervals and the LoRa parameters of configuration 3. The battery capacity is considered 400 mAh

Mode	Time Interval	Average Current	Duration
Active	5 s	47 mA	8.5 hours
Active	30 s	7.8 mA	51 hours
Active	5 m	0.79 mA	21 days
Active	15 min	264 μA	63 days
Active	30 min	134 μA	124 days
Active	60 min	68.5 μA	243 days

Table 4.4: Battery duration for different device configurations

The following table reports the amount of charge required by the different operations

Function	Charge
Stand-by (1 h)	12 mC
BLE advertising	17.7 μC
Measurement 6 TCs	13.5 mC
Transmission LoRa config. 1	8.3 mC
Transmission LoRa config. 2	55 mC
Transmission LoRa config. 3	221 mC

Table 4.5: Charge values

4.3.4 Current consumption during reading operation

The last current measurement was performed during the data log functionality namely when data stored in the memory are transmitted to the central receiver. The measurement method used was the indirect current measurement by means of the voltage developed across a 10 Ω resistor. Figure 4.24 displays the voltage waveform obtained. The process started with the device in a sleep state, transitioning to an active state upon receiving a read command. Subsequently, the memory is read and all the data are transmission in burst mode. In fact it can be seen a first interval that depends on the amount of data that need to be read, where the current consumption reaches 9.5 mA, then it further increases when the long range radio is activated for the transmission.

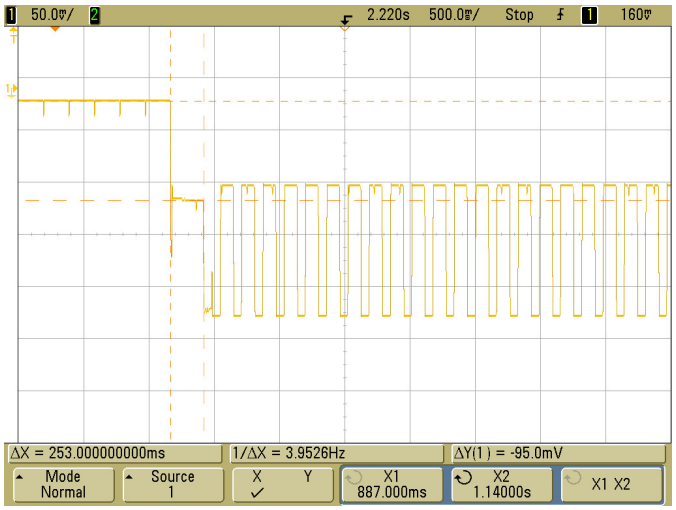


Figure 4.24: Voltage waveform during memory reading operation

4.4 Radio transmission range test

In order to characterize the maximum distance achievable between the transmitter and the receiver, different tests were conducted. The first one has been performed maintaining the device inside the climatic chamber with closed door, in order to simulate the working conditions of a freeze dryer. The receiver, visible in Figure 4.25 was then moved, on the same floor, from the room hosting the climatic chamber up to when the signal became too weak. The parameter used to analyze the strength of the received signal was the RSSI which stands for received signal strength indicator and it measures the received power in dBm. At one meter from the climatic chamber the registered value was -45 dBm while it decreased increasing the distance with the transmitter up to -100 dBm at the maximum distance of 75 m. The LoRa parameters used were SF 7, BW 125 kHz and CR 4/5.

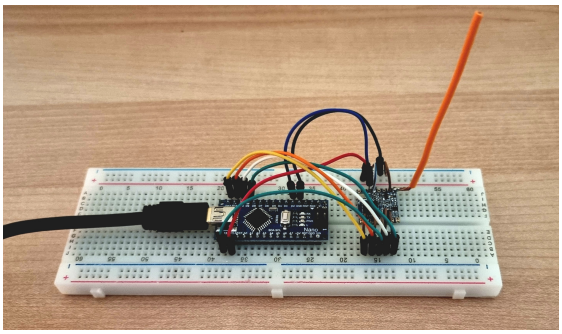
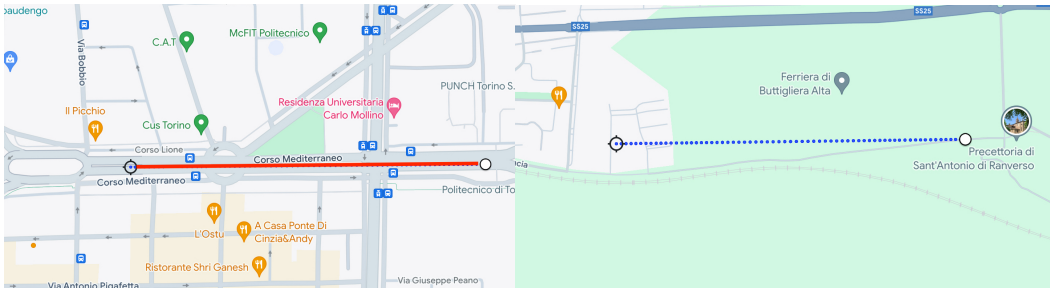


Figure 4.25: LoRa receiver

It is important to note that the environment where the tests were conducted includes thick walls and a lot radio sources that could create interference. Still the distance measured is sufficient for a freeze dryer application. Another test was performed in the same conditions but changing the LoRa configuration to SF 11, BW 62.5 kHz and CR 4/5. The maximum distance before loosing the communication, in this case increased to 90 m. The drawback though, is the significant increase in the current consumption as already pointed out in the previous section. A different test wanted to verify the maximum range achievable in an urban environment with the minimum amount of obstacles between the transmitter and the receiver and with the same LoRa parameters used before namely SF 11, BW 62.5 kHz and CR 4/5. For this test the wireless sensor has been positioned close to a window at an altitude of about 15 meters above a long street. The receiver was moved along the street up to the point where the connection was interrupted. The distance reached improved a lot with respect to the indoor test in fact the maximum distance registered was 530 m with an RSSI of -110 dBm. The previous result was obtained in an urban environment, therefore a considerable amount of interference, made

the demodulation of the signal at the receiver much more difficult. For comparison another test with the same LoRa parameters, has been performed in a rural area where the amount of electromagnetic interference is significantly lower. Always maintaining the line of sight between transmitter and receiver, the maximum distance reached before loosing the connection in this case increased to 850 m with an RSSI of -118 dBm.



(a) Outdoor range test: urban area **(b)** Outdoor range test: rural area

Figure 4.26

The following table summarizes the results obtained

Ambient	Distance	RSSI
Inside climatic chamber	75 m	-100 dBm
Outdoor: urban area	530 m	-110 dBm
Outdoor rural area	850 m	-118 dBm

Table 4.6: Maximum transmission distance in different environments

4.5 Reset circuit validation

To validate the correct behaviour of the reset circuit based on the hall effect switch, the device needs to be connected to the wireless charger. This choice has been made to increase the reliability of the system in response to strong magnetic field that could be present in the working environment. In this conditions, approaching the back of the PCB with a small magnet should be sufficient to change the voltage level of the magnetic switch that in turns, resets the microcontroller. The test has been performed with the magnet reported in Figure 4.27 which is a very small one.

Figure 4.27: Magnet used for the test

The magnetic field that it generates is sufficient to activate the switch at a distance of 2.5 cm therefore this is more than enough even when the circuit is resin coated.

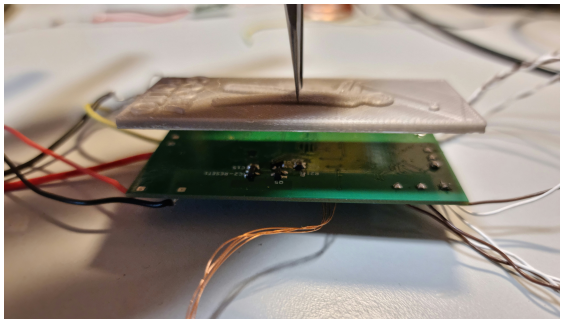


Figure 4.28: Test hall effect switch

Chapter 5

Conclusions

This project focused on the design, realization and validation of a wireless sensor for the temperature monitoring of different points inside freeze dryers and more generically harsh environments. The work started from the design requirements necessary to create a system able to work in a difficult environments characterized by low temperature, low pressure and non negligible amount of water. Among them there are the necessity to work with a battery powered solution or the requirement of a wireless communication due to the impossibility to introduce cables inside the freeze dryer chamber. After the selection of the best suited components for the application, the PCB has been designed. The next part of the work has been reserved for the realization of the firmware running on the microcontroller which is in charge of acquiring the voltage from the thermocouple sensors, communicate with all the modules (ADC, memory, etc..) and manage the wireless communication. In addition two other applications have been developed for data visualization and device configuration. The last part was reserved for the different tests conducted to validate the functionalities of the device. Among them, the most important are the temperature accuracy and the current consumption. Regarding the temperature accuracy, all the values registered were compliant with the derived uncertainty of 0.5 °C. Also the current consumption showed very good results thanks to the low power approach used. In fact reading values every 5 minutes guarantees a battery life of some months with a 400 mAh battery.

Appendix A

Temperature sensing technologies

This chapter provides an overview of the existing temperature measurement sensors. For each sensor both advantages and limitations are presented, as well as their basic working principle, contributing to a comprehensive understanding of temperature measurement techniques.
The typical parameters that are considered for the sensor selection are:

⇒ **Range**: it represents the lower and upper bound of temperature that can be withstand by the sensor or it can be the range that guarantees a stated uncertainty.

⇒ **Accuracy**: it is the uncertainty of the sensor that expresses the not perfect knowledge of the measurand.

⇒ **Size**: it is the footprint of the sensor

⇒ **Non-linearity**: it represents the deviation of the output characteristic of the sensor form a specific straight line.

A.1 IC bandgap temperature sensor

The first type of sensor that is considered is the one that relies on the temperature-dependent behavior of the base-emitter voltage of a BJT. MOSFETs based sensors are also possible but due to a longer calibration procedure involving two know temperatures instead of one, BJTs are preferred. As reported in Figure A.1 [10] these type of sensors are typically embedded in an integrated circuit composed by three main blocks that are the sensor itself, its bias circuit, and an ADC. In

addition they are typically equipped with some registers and a logic that manages the communication (SPI, I2C) with a microcontroller. In order to perform the temperature measurements two signals are required: a first one that is temperature dependent and a second one that is used as a reference for the analog to digital converter and thus must have a temperature coefficient as low as possible, ideally zero. The first is provided by the base-emitter voltage of the BJT while the second is generated by a bandgap reference voltage still obtained from the same circuit.

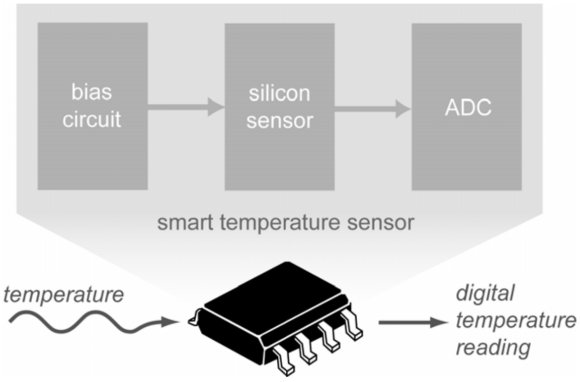


Figure A.1: Block diagram of smart temperature sensor

From the Ebers Moll equations, the collector current of a BJT can be expressed as

$$I_c = I_s \left(e^{\frac{qV_{BE}}{kT}} - 1 \right) \tag{A.1}$$

where I_s is the inverse saturation current, q is the electron charge, k is the Boltzmann's constant and T is the absolute temperature. Manipulating the previous equation to isolate the base-emitter voltage V_{BE} gives

$$V_{BE} = \frac{kT}{q} \ln \left(\frac{I_c}{I_s} + 1 \right) \tag{A.2}$$

being $I_c \gg I_s$ the 1 is typically omitted. The equations shows a logarithmic relation between the V_{BE} voltage and the ratio between the two currents. If I_c and I_s were constant then a linear relation would be available between the base-emitter voltage and the temperature since k and q are constants. Unfortunately the current I_s is characterized by a strong temperature dependence, roughly doubling its value for every 10 °C of temperature increment resulting in a temperature coefficient of about -2 mV/°C, as well as a process dependent behaviour related to the geometry of the transistor. To deal with this problem, an architecture involving two identical

transistors fabricated on the same die is exploited. One BJT is biased with a constant collector current I_c while the other with the same current multiplied by a factor p. Now the focus is on the difference among the base-emitter voltage of the two transistors ΔV_{BE} that allows to eliminate the dependence on I_s. It is important to remark that the two transistors must be identical and at the same temperature otherwise the saturation current would be different. The base-emitter voltages are defined as

$$V_{BE_1} = \frac{kT}{q} \ln\left(\frac{I_c}{I_s}\right) \quad V_{BE_2} = \frac{kT}{q} \ln\left(\frac{pI_c}{I_s}\right) \tag{A.3}$$

and considering the difference between them yields to

$$\Delta V_{BE} = V_{BE_1} - V_{BE_2} = \frac{kT}{q} \ln(p) \tag{A.4}$$

where now, after the selection of the multiplication factor p, a linear relation is obtained between the difference of base-emitter voltage of the two transistor and the absolute temperature. In the case of a factor p = 10, the temperature coefficient becomes 198.35 μV/°C. ΔV_{BE} is typically called V_{PTAT} which stands for voltage proportional to absolute temperature. A typical implementation of bandgap temperature sensor is represented in Figure A.2 [10]

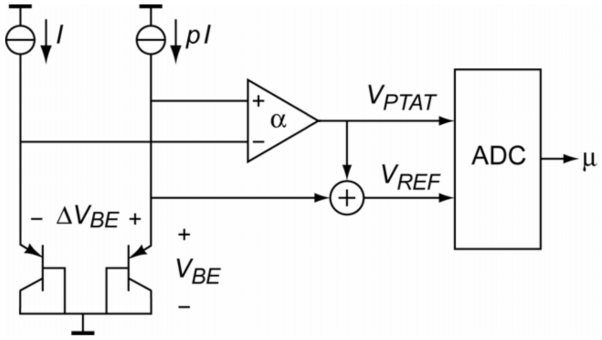


Figure A.2: Block diagram of an IC bandgap temperature sensor

The two voltages that are required to perform the temperature measurement are the temperature dependent ΔV_{BE} and the temperature independent reference voltage V_{REF}. The latter is obtained by summing ΔV_{BE} multiplied by a factor α with the base-emitter voltage of one of the two transistor indicated as V_{BE} in the scheme. The sum is almost temperature independent, in fact V_{BE} is the base-emitter voltage of a transistor that has a temperature coefficient of -2 mV/°C, so it is temperature dependent. ΔV_{BE} is also temperature dependent but with a positive temperature coefficient that compensates the negative one of V_{BE}. In this

way a stable and temperature independent reference for the ADC can be generated. Through a differential amplifier it is possible to amplify by a factor α the voltage ΔV_{BE} in such a way that the temperature coefficient of ΔV_{BE} becomes similar to that of V_{BE}.
Figure A.3 [10] reports the behaviour of the voltages described so far. Regarding the conversion to the digital domain, due to the quite long thermal time constants and high resolution required in packaged bandgap temperature sensors, Sigma-Delta ($\Sigma\Delta$) ADCs are typically exploited.

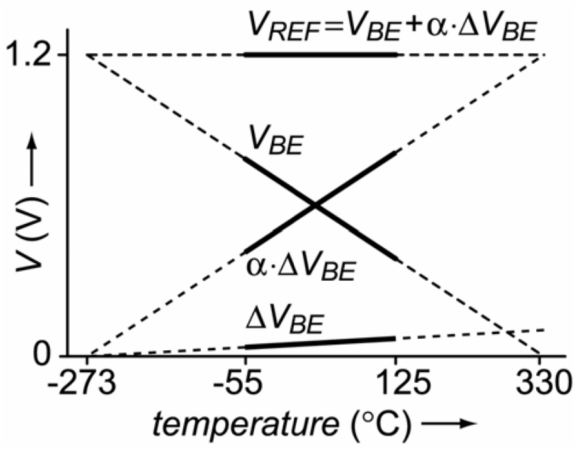


Figure A.3: Voltages variation in bandgap temperature sensor

In the equations presented at the beginning of the section, the focus was on the collector current, but in Figure A.2, PNP transistors are used. As a consequence it is the emitter that is biased with a constant current and not the collector. This is not a problem since it is possible to design the BJTs in such a way that the the ratio between the two currents is unitary yielding to $I_e = I_c$.
One additional aspect that is worth pointing out is the need of calibration in order to reach uncertainties of the order of $\pm 0.1°C$. Good calibration of each sensor requires up to some minutes so it becomes a quite expensive and time consuming operation on a large scale. Different techniques are used to reduce time and costs such as the calibration at wafer level, even though is is less accurate, or the calibration of just few sensors from which are extracted the parameters to calibrate all the other sensors of the batch [10].
In conclusion, IC bandgap temperature sensors are a valid alternative that offer high linearity and low uncertainty over a large temperature range, in addition since they are completely integrated, they are simple to use and no extra conditioning circuit is required. The main sources of uncertainties for this type of sensor comes from the slightly non linear behaviour of V_{BE}, its process spread, the errors affecting

the ADC and the non perfect temperature independent response of the voltage generated by the bandgap reference; indeed it is characterized by a bow-shaped drift curve.
The architecture of the system presented in the first chapter uses the ADT7410, which is an IC bandgap temperature sensor in SOIC package with an uncertainty of $\pm 0.5\,^{\circ}\mathrm{C}$, to measure the temperature of the cold junction of the thermocouples, necessary to perform the procedure called cold junction compensation. The component has been selected due to its low cost and sufficiently low uncertainty over the entire range required that is from $-50\,^{\circ}\mathrm{C}$ to $40\,^{\circ}\mathrm{C}$.

A.2 Thermocouples

Thermocouples are one of the most used temperature sensors due to their extended range spanning form $-270\,^{\circ}\mathrm{C}$ to $2500\,^{\circ}\mathrm{C}$ depending on the metals used and their robustness to harsh environment.

A.2.1 Seebeck effect

The basic physical principle that is behind thermocouples is called Seebeck effect, discovered for the first time in 1822 by a German physicist named Thomas Seebeck. When a bar of metal is subjected to a temperature difference between its two ends, then a voltage develops across the length of the metal and this voltage is called Thermoelectric voltage. The generation of this potential difference can be better understood in a very simplified way by looking at the movements of electrons inside the material. A metal, being a conductor, is characterized by the partial overlapping between the valence and the conduction band meaning that is can easily conduct electricity thanks to the large number of free electronic states. As the temperature of one end of the metal bar is increased, thermal energy is provided to those electrons that will acquire more kinetic energy causing their diffusion toward the colder region, generating a positive charge on the hot end and a negative charge on the cold end. The electron migration continues until the potential difference across the conductor becomes sufficiently large to counteract the temperature-induced diffusion of electrons.
The strength of the Seebeck effect is quantified using the Seebeck coefficient, denoted as $S = dV/dT$, with V representing voltage and T representing the temperature. Each metal is characterized by a different Seebeck coefficient, in addition its value it is not constant but is temperature dependent as visible form Figure A.4 [11].

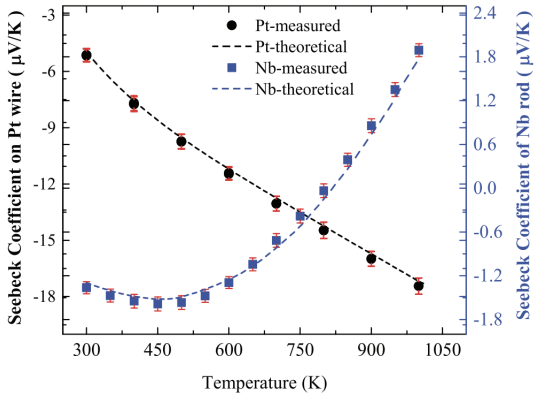


Figure A.4: Temperature dependence of Seebeck coefficient

A.2.2 Thermocouple's structure

Having defined the physical phenomena that is at the base of thermocouples, it is now time to define the sensor itself. A thermocouple is obtained by joining together two dissimilar metals at one end that is called hot junction or better measuring junction, while the other end is called cold junction or reference junction. The latter builds up from the connection of the two leads of the thermocouple with the copper traces that go to the conditioning circuitry. The reference junction has to be placed on an isothermal block, that is typically realized with a large metal mass or with a copper fill in a PCB, in such a way that the same voltage develops along the two copper traces giving a null contribution to the voltage generated by the thermocouple itself. Figure A.5 reports an example of thermocouple circuit [12].

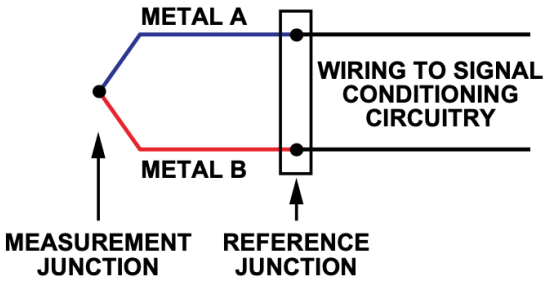


Figure A.5: Thermocouple

When a temperature difference in present between the two junctions, a thermoelectric voltage, relative to the temperature difference, can be measured across the reference junction. This voltage is given by the difference of the voltages developing across the two metal wires. The thermocouple thus generates a voltage that is

proportional to a temperature difference rather that an absolute temperature. So it becomes mandatory to know the temperature of the reference junction in order to perform a correction, called cold junction compensation, that allows to retrieve the absolute temperature from the measurement of the differential voltage of the thermocouple, when the reference junction is at a temperature different from $0°C$. In addition to the compensation, other factors introduce challenges in achieving accurate readings. In first place, thermocouples generate a small voltage that needs to be highly amplified to match the input dynamic of the ADC. Moreover, the noise picked up by the thermocouple's leads can easily superimpose to the measuring signal making impossible to distinct the two, that's why instrumentation amplifiers are typically used, as they are able to reject the common-mode component of the noise present at both the wires performing a differential measurement. A large number of commercial thermocouples are available, each one identified by individual letters and composed of varying metal combinations, thereby providing distinct operational ranges. Table A.1 reports the most commonly used types along with their characteristics

Thermocouple Type	Lead Metal A (+)	Lead Metal B (–)	Temperature Range (°C)	Voltage over Temperature Range (mV)	Seebeck Coefficient (µV/°C at 0°C)
J	Iron	Constantan	–210 to 1200	–8.095 to 69.553	50.37
K	Chromel	Alumel	–270 to 1370	–6.458 to 54.886	39.48
T	Copper	Constantan	–200 to 400	–6.258 to 20.872	38.74
E	Chromel	Constantan	–270 to 1000	–9.385 to 76.373	58.70

Table A.1: Thermocouple types and characteristics

A.2.3 Cold junction compensation

As already mentioned, the cold junction compensation is needed to get the absolute temperature of the measuring junction. In the early stages of thermocouple development the reference junction was kept inside an ice bath to have a well defined reference temperature of 0 °C. The typical setup was similar to that reported in Figure A.6 [12].

It is quite evident that this solution is only applicable for laboratory experiments and characterization while it is impractical for applications in the field. Due to the fact that almost all the thermocouple reference tables or polynomial equations have been derived from a reference junction in an ice bath, cold junction compensation is mandatory. To understand the procedure, the first thing to do is to write the voltage across the reference junction; referring to Figure A.5, this voltage can be written as

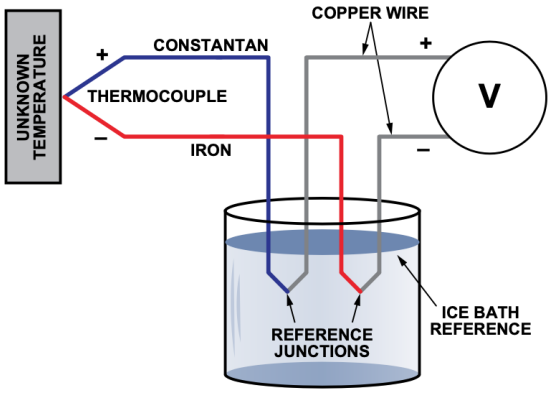


Figure A.6: Thermocouple ice bath setup

$$V_T = V_{metalA} - V_{metalB} = S_{ma} \times (T_m - T_r) - S_{mb} \times (T_m - T_r) \tag{A.5}$$

where V_T is the voltage generate by the thermocouple, S_{ma} and S_{mb} are the Seebeck coefficients of the two metals and T_m and T_r are the temperature of the measurement and the reference junction respectively. The equation can be also rearranged as

$$V_T = (S_{ma} - S_{mb}) \times T_m - (S_{ma} - S_{mb}) \times T_r \tag{A.6}$$

where now the difference of the Seebeck coefficients $(S_{ma} - S_{mb})$ is the Seebeck coefficient of the specific thermocouple (T, K, E etc). From this equation it is possible to define the sequence of steps required for the cold junction compensation:

1. Measurement of the voltage V_T at the thermocouple output.

2. Measurement of the reference junction temperature by means of another temperature sensor that can be for instance a thermistor, and IC sensor or an RTD. For this measurement it is important to employ high accuracy sensors since all the errors introduced in this phase will reflect to the final result.

3. Exploiting the thermocouple's reference tables or the polynomial expressions, the temperature value of the reference junction is converted into an equivalent thermoelectric voltage.

4. The voltage V_T and the equivalent voltage of the reference junction are summed.

5. The sum of the two voltages is then converted to a temperature value using again tables or polynomial expressions.

It is important to note that a simple sum between the equivalent temperature of the voltage V_T and the temperature of the reference junction can not be done due to the non linearity of the thermocouple's voltage response versus temperature.

A.2.4 Non-linearity and polynomial expressions

Thermocouples are non linear sensors as it can be seen from Figure A.7 [12]

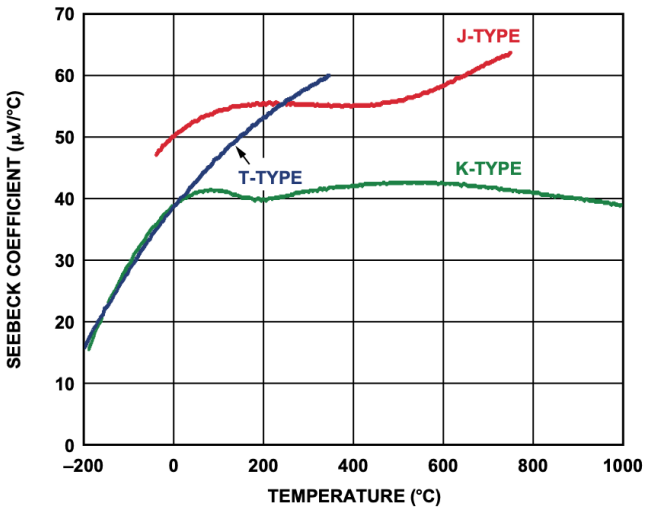


Figure A.7: Thermocouple sensitivity

due to their behaviour there are different methods used to perform voltage to temperature or temperature to voltage conversions mainly based on the computational capabilities of the hardware utilized for the calculations. A first method to deal with the non linearity is to consider a sub range of temperatures, along which the sensitivity exhibits a relatively flat behaviour. This of course limits the operational range of the sensor but in some applications could be enough. Another approach is based on the use of lookup tables; both the thermocouple voltage and the corresponding temperature are stored into a memory, then the temperature is computed through the interpolation between the two closest voltage values. The last method is also the one that provides the better and most accurate results since it is based on the use of polynomial equations that represent the thermocouple response. There are two of them, the first is the direct polynomial expression used to calculate the thermoelectric voltage from a know temperature, while the second in the inverse polynomial expression used to compute the temperature form a thermoelectric voltage. Based on the thermocouple type and the range of interest the order of the polynomial changes. The direct polynomial expression has the

form

$$E = \sum_{i=0}^{n} c_i \cdot (t_{90})^i \tag{A.7}$$

while the indirect polynomial expression can be written as

$$t_{90} = \sum_{i=0}^{n} d_i \cdot (E)^i \tag{A.8}$$

For both the equations E is the voltage in microVolt, t_{90} is the temperature in degree Celsius while c_i and d_i are the polynomial coefficients. The computational processing required for accurate calculations of these high order equations can require a significant amount of resources, particularly when dealing with high resolution floating-point numbers, that's why they should be used only when the the processor is equipped with a floating-point unit that can speed up the operations.

A.2.5 Construction types and uncertainty

Thermocouples come in three in different construction types:

- Exposed thermocouple
- Insulated thermocouple
- Grounded thermocouple

The exposed thermocouple does not have any protective sheath on it, so it is only formed by the two different metals joined together. This construction allows for the better heat transfer between the object and the sensor providing the faster response. Insulated thermocouples instead are equipped with a protective sheath, typically made by metal, all around the sensor that provides an additional protective layer when working in difficult environments. The extra protection though limits the response time. Grounded thermocouples are similar to the insulated ones, but now the tip is welded to the protective sheath. This allows for an increased heat transfer with respect to the previous type. Having a direct connection of the measuring circuit with the object that is under measurement in exposed and grounded thermocouples provides the faster temperature response but increases the susceptibility of the system to ground loops. Indeed the potential of the object that the thermocouple is touching may not be the same of the measurement circuit leading to wrong results in the readings.
In terms of uncertainties, the different thermocouple types are characterized by a certain number of classes valid for different temperature ranges defined by the

IEC-EN 60584 standard. Table A.2 [13] reports an extrapolation of tolerances for the types J, K and T.

Type	Tolerance Class	Range (°C)	Uncertainty (°C) (Larger between the two values)
J	Class 1	$-40 < T < 750$	± 1.5°C or $\pm(0.004 \cdot \lvert T \rvert)$
	Class 2	$-40 < T < 750$	± 2.5°C or $\pm(0.0075 \cdot \lvert T \rvert)$
K	Class 1	$-40 < T < 1000$	± 1.5°C or $\pm(0.004 \cdot \lvert T \rvert)$
	Class 2	$-40 < T < 1200$	± 2.5°C or $\pm(0.0075 \cdot \lvert T \rvert)$
	Class 3	$-200 < T < 40$	± 2.5°C or $\pm(0.015 \cdot \lvert T \rvert)$
T	Class 1	$-40 < T < 350$	± 0.5°C or $\pm(0.004 \cdot \lvert T \rvert)$
	Class 2	$-40 < T < 350$	± 1.0°C or $\pm(0.0075 \cdot \lvert T \rvert)$
	Class 3	$-200 < T < 40$	± 1.0°C or $\pm(0.015 \cdot \lvert T \rvert)$

Table A.2: Thermocouple uncertainty

In conclusion thermocouples are versatile sensors with a large number of advantages such as the extremely large temperature range, their robustness and resistance to shock and vibrations and the rapid temperate response especially in the case of exposed thermocouple. Of course they are not perfect in fact among the disadvantages there are the uncertainty that can not go below 1 or 2 $°C$ and the small output voltage that have to be properly collected by careful design of the conditioning circuit.

A.3 Resistive Temperature Detectors

Resistive Temperature Detector, or RTD, are a family a temperature sensors that exploit the temperature dependence of the electrical resistance of a metal to perform temperature measurements. Given their well characterized resistance variation, they can achieve uncertainties of the order of 0.1 $°C$ when proper signal conditioning is implemented. The metal used to realize the sensor has to possess a set of characteristics that make it suitable for RTDs implementation, the most important are

- High resistivity in order to generate, after proper excitation, a sufficiently large voltage for the downstream blocks that will process the signal
- High sensitivity to have the largest possible resistance variation in response to a temperature variation
- Linear response of the resistance-temperature characteristic
- Small material volume to have fast heat transfer that, in turn, generate a faster response
- Stability over time

The characteristics of the sensor are determined by the the type of metal that is employed for its realization. Among the most commonly used there are Platinum which is by far the most utilized thanks to its large measurement range, high sensitivity and linearity and high resistivity then there are some Platinum-iridium alloy especially used for high temperature. Other metals are Copper which has a lower cost compared to Platinum but it also has lower sensitivity and it is subjected to oxidation and Nickel with a resistivity in between Platinum and Copper and no oxidation problems but it is characterized by poor linearity. Figure A.8 [14] reports the different resistance-temperature response for Platinum, Copper and Nickel RTDs.

A.3.1 Physical realization

RTDs are typically constructed in two different ways: thin film or wire-wound. In the first type a thin layer of metal is deposited onto a ceramic substrate followed by very high-temperature annealing and stabilization, as well as a thin protective glass layer to cover the entire element. The deposited film has a region called resistance trimming loop that is used to adjust the value of the resistance R_0, which is the resistance of the RTD at 0 $°C$. This construction method provides the faster response time and the lowest cost but it is characterized by poor stability and

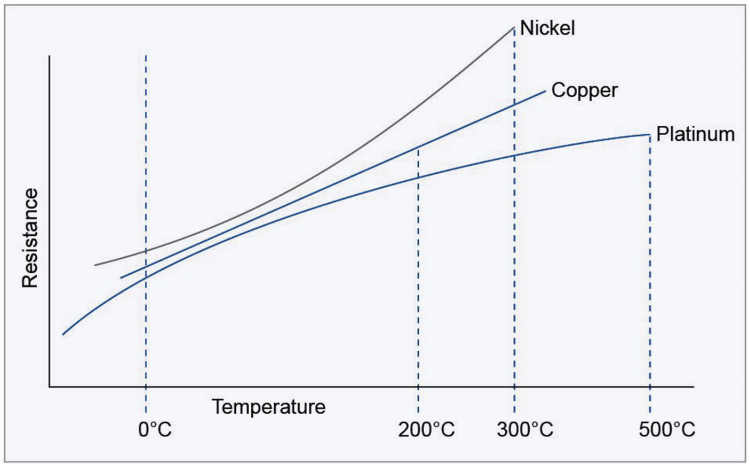


Figure A.8: Resistance vs temperature response of RTDs

can not be used for high temperature. The second type is realized by winding the metal wire around a core of insulating material that can be ceramic or glass. Next, lead wires are attached to the resistance wire, and then a glass or ceramic coating is applied over the wire for protection. Also in this case the length of the wire is properly trimmed to achieve a specific R_0 resistance. For example the pt100, that is a platinum RTD, has $R_0 = 100\ \Omega$. This construction type provides the highest accuracy and stability over time as well as the possibility to be used for very high temperature but is is expensive and provides slower response time with respect to thin film. Figure A.9 and A.10 report the two types

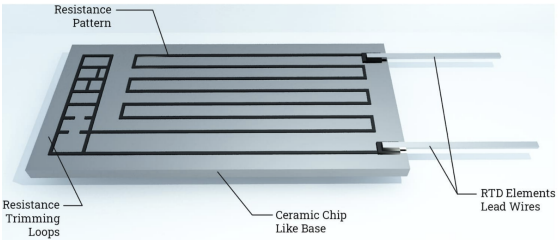


Figure A.9: Thin film RTD

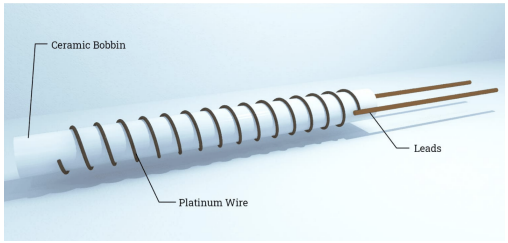


Figure A.10: Wire-wound RTD

A.3.2 Callendar-Van Dusen equations and uncertainty classification

The relation between the resistance of the RTD and the temperature is described by the Callendar-Van Dusen equations. There are two different relations, one for temperatures lower than 0 $°C$ and the other for temperatures greater than 0 $°C$

$$\text{For } T < 0 : R_{\text{RTD}}(T) = R_0 \cdot \Big\{1 + (A \cdot T) + (B \cdot T^2) + \Big[(C \cdot T^3) \cdot (T - 100)\Big]\Big\} \quad \text{(A.9)}$$

$$\text{For } T > 0 : R_{\text{RTD}}(T) = R_0 \cdot \Big[1 + (A \cdot T) + (B \cdot T^2)\Big] \quad \text{(A.10)}$$

The term R_0, as already mentioned is the resistance of the RTD at a temperature of 0 $°C$ while T is the temperature. The coefficients A, B and C are defined by the IEC-60751 standard and they depend on the metal that is used to realize the RTD. The uncertainty of RTDs is defined by the IEC 60751 international standard. Different tolerance classes are present based on the accuracy of the sensor.

Tolerance class	Temperature range of validity (°C)		Tolerance values (°C)
	Wire wound resistors	Film resistors	
AA	-50 to +250	0 to +150	$\pm(0.1 + 0.0017\|\text{T}\|)$
A	-100 to +450	-30 to +300	$\pm(0.15 + 0.002\|T\|)$
B	-196 to +600	-50 to +500	$\pm(0.3 + 0.005\|T\|)$
C	-196 to +600	-50 to +600	$\pm(0.6 + 0.01\|\text{T}\|)$

Table A.3: Tolerance values for RTDs

A.3.3 Conditioning circuit

RTDs are passive resistive sensors meaning that they are based on output resistance changes. In addition, passive means that proper excitation is required to measure the sensor's output variation. On the contrary thermocouples are active sensors because the voltage at the thermocouple's leads can be measured without external excitation sources.
When designing the conditioning circuit for RTDs it is important to take into consideration different aspects:

- Lead wires used to connect the sensor to the conditioning circuit are a source of systematic errors since their resistance can not be distinguished from the resistance of the RTD. This effect can be mitigated considering, instead of the classical two wire configuration, a three or four wires configuration. The latter is the one that reduce to the minimum the lead wires resistance effect.

- The current that flows through the RTD, required to generate a voltage drop that can be read by an ADC, must be below a certain value to avoid self-overheating. When the RTD temperature becomes higher than the surrounding one, the resistance of the sensor assumes a value that is no longer representative of the correct temperature and so it becomes an additional uncertainty source.

One conditioning circuit that is largely employed to perform temperature measurement with RTDs is the ratiometric circuit. A typical setup is reported in Figure A.11 [15]

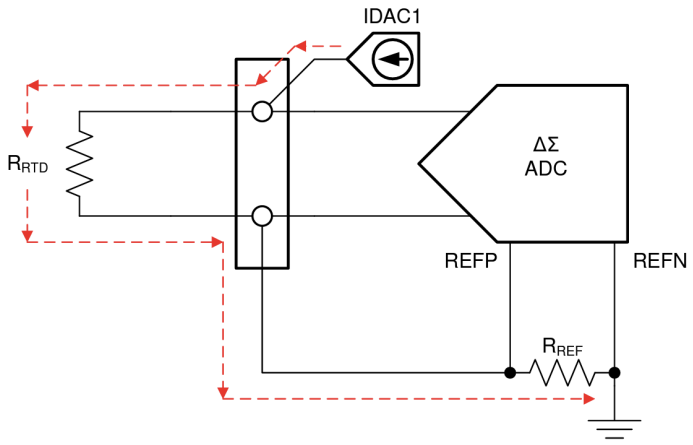


Figure A.11: RTD ratiometric measurement circuit

IDAC1 is the current excitation source that flows through both the RTD and the reference resistor R_{REF}. As already mentioned, its value choice is crucial to avoid self-overheating of the sensor. R_{REF} provides the reference voltage to the ADC and the DC bias to the RTD itself in such a way that the the analog input signal is in the input range of the PGA (inside the ADC in this case). The ADC output code can be written as

$$\text{Output code} = 2^{23} \cdot \frac{V_{RTD}}{V_{REF}} = 2^{23} \cdot \frac{I_{IDAC1} \cdot R_{RTD}}{I_{IDAC1} \cdot R_{REF}} \tag{A.11}$$

from which it can be noted that the excitation current cancels out, leading to an RTD resistance only function of the reference resistance and the ADC output code. Therefore the accuracy of the current source is not important. What is important is the selection of R_{REF} that has to be very accurate.

A.4 Thermistors

Thermistors are another category of devices that can be used to perform temperature measurements. They are typically limited to a temperature range that is between -100 °C and 300 °C, limiting their applications with respect to RTDs or thermocouples. They rely on the variation of the electrical resistance of a semiconductor in response to temperature changes and they are classified in two different families: NTC and PTC.
NTCs are negative temperature coefficient thermistors and they get the name due to their resistance reduction in response to a rising temperature. PTCs instead are positive temperature coefficient thermistors and they have the opposite property of NTCs namely they show higher resistance for higher temperatures. These devices are not only used for temperature measurements, indeed due to their behaviour, other common applications are self-resetting fuses, self-regulating heating elements and inrush current limiters. PTCs can be further divided in two other categories that are linear, also known as silistor, and non linear depending on their resistance versus temperature response but only the linear PTC is used as temperature sensing element. The difference in response between the two is determined by the material involved: ceramic for the non linear PTC and silicon for the linear. Figure A.12 shows the different resistance-temperature response of the two thermistors used as temperature sensors [16]

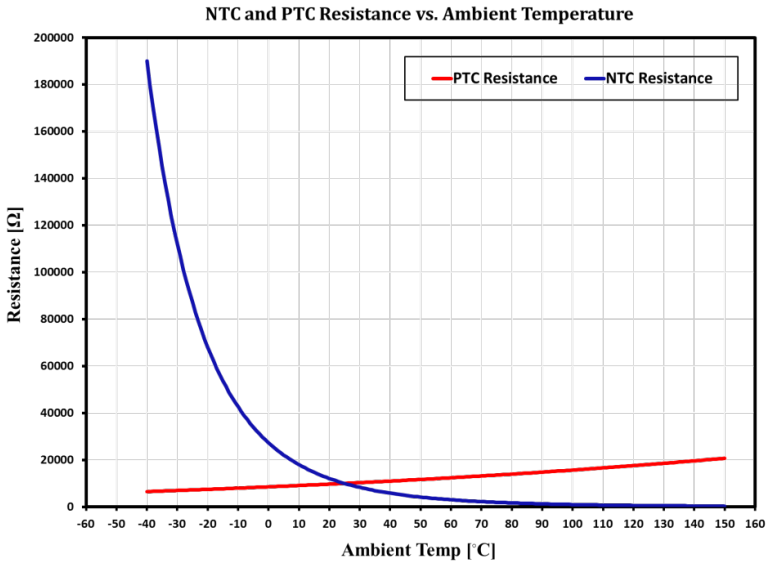


Figure A.12: NTC and linear PTC response

from the figure it can be seen that NTC thermistor exhibits higher sensitivity

(especially at low temperature) and higher resistance with respect to PTC but the main drawback is the strong non linearity. Despite the non linearity, NTC thermistors are the most used for temperature sensing applications.

A.4.1 Relation between temperature and resistance

One common expression that is used to describe the relation between the resistance and the temperature of an NTC thermistor is the Steinhart–Hart equation

$$\frac{1}{T} = A + B\ln(R) + C(\ln(R))^3 \tag{A.12}$$

the coefficients A, B and C are called Steinhart–Hart coefficients and they depend on the specific NTC thermistor that is used while T is the absolute temperature and R is the resistance of the thermistor.
Another possibility to express the temperature-resistance dependence is the β parameter equation, reported below

$$\frac{1}{T} = \frac{1}{T_0} + \frac{1}{\beta}\ln(\frac{R}{R_0}) \tag{A.13}$$

where R_0 is the resistance of the thermistor at the temperature of $T_0 = 298.15$ K. Isolating T provides the calibration function

$$T = \frac{1}{\frac{1}{T_0} + \frac{1}{\beta}\ln(\frac{R}{R_0})} \tag{A.14}$$

the value of β is provided by the manufacturer of the thermistor and it typically has a certain range of validity. Talking about sensor uncertainty, thermistors do not have a standard classification like RTDs or thermocouples, but there are non standard classes for certain temperature ranges. The most accurate ones can reach 0.1 $°C$ of uncertainty.

A.4.2 Conditioning circuit

Thermistors like RTDs are passive resistive sensors meaning that they exhibits an output resistance variation. In order to be able to read their output, proper excitation is necessary to convert the resistance change into a voltage change that can be easily collected by an analog to digital converter before being processed by a microcontroller or a PLC. As a consequence ratiometric measurements describes for collecting RTDs outputs or unbalanced Wheatstone bridge are common circuits employed.

Appendix B

BLE communication

B.1 Introduction

The Bluetooth Low Energy communication has been introduced by the Bluetooth SIG (Special Interest Group), to address all the devices that require a low power consumption due to the fact that most of the time they are battery powered. Bluetooth LE achieves low energy consumption by sacrificing data rate, using smaller data packets (27 to 251 bytes) and by sending data less frequently avoiding long radio-on times, which is a significant factor in power consumption. Some of the key features of BLE are reported below

- Operating band: 2400 MHz – 2483.5 MHz (2.4 GHz)
- Channel bandwidth: 2 MHz
- Number of RF channels: 40
- Maximum transmit power: 20 dBm (0.1 W)
- Maximum application data throughput: 1.4 Mbps
- Maximum range at reduced data rates (125 & 500 kbps): 1000 m

B.2 BLE protocol stack

The protocol stack is an high level representation of all the layers that enable the BLE communication. Figure B.1 reports a typical structure where it is possible to see the division between the host and the controller. The first one is the higher-level software component responsible for managing the overall Bluetooth communication

in addition it manages the interaction with Bluetooth devices, handles application-specific tasks, and initiates connections. The second one instead is the lower-level hardware or firmware component responsible for the physical layer and link layer operations. It handles the low-level radio communications, modulation, and demodulation and it is typically implemented on dedicated Bluetooth chips or modules.

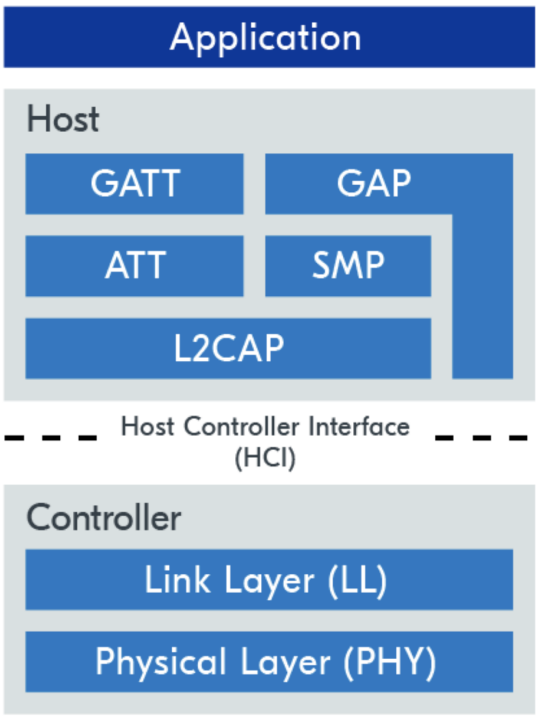


Figure B.1: BLE protocol stack

Each layer is fundamental for the correct operation of BLE devices for example the GAP layer is used to define specific device roles for nodes in a Bluetooth LE network. These roles determine important aspects such as how the device advertises its presence, or how it scans and connects to other nodes. In particular two roles can be defined: central and peripheral. The central is a device that scans and initiates connections with peripherals. The peripheral on the other hand is a device that advertises and accepts connections from centrals.
Other two important layers that work after the connection phase are the ATT and GATT that define how data are structured and exchanged between BLE devices. The ATT layer is based on a client-server architecture where the server (typically the peripheral device) holds the data and can either send it directly to the client (the central device) or the client can poll the data from the server. Then it defines a data structure called the attribute, which is used by the GATT server to store

data. The server and the client role are used by the GATT layer as a consequence they are typically referred as GATT server and GATT client. The GATT layer, as it can be seen from Figure B.1, is built on top of the ATT layer and it hierarchically classifies the attributes into profiles, services and characteristics. Figure B.2 reports an example of a generic profile with services and characteristics.

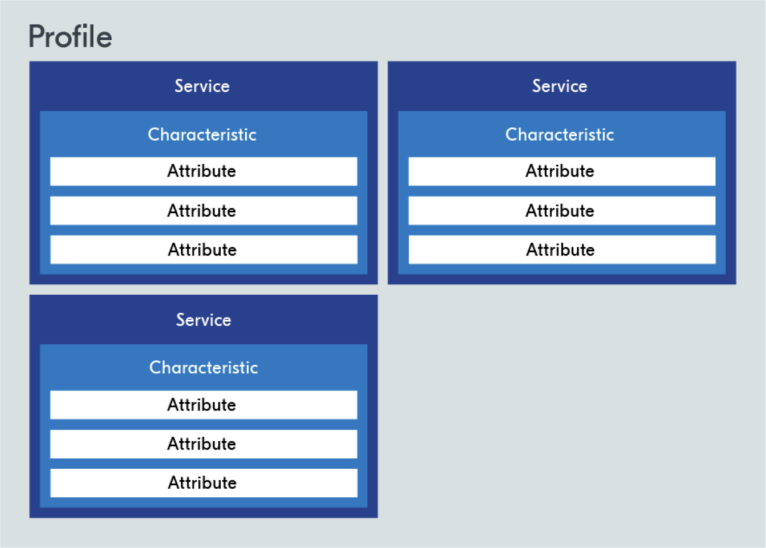


Figure B.2: Generic profile

Each characteristic in a given service is formed by at least two attributes that are the characteristic declaration attribute which holds metadata about the characteristic value attribute and the characteristic value attribute that stores the actual user data (for example the temperature measured by a temperature sensor). Optional attributes are the characteristic descriptors that hold additional metadata about the characteristic. A specific type of descriptor is the client characteristic configuration descriptor (CCCD). The CCCD is a vital element for characteristics that offer server-initiated operations like notify and indicate. It serves as a writable descriptor enabling the GATT client to control notifications or indications for that characteristic. Through the CCCD, the client can subscribe to specific characteristics, choosing whether to receive updates via indications or notifications.
Multiple characteristics are contained in a service. Service definitions always start with a service declaration attribute. This attribute holds metadata about the service, it also indicates the beginning of a service in the sequence of services stored on a GATT server. Then above this, a profile is one or more services that address the same use case.

B.3 BLE advertising procedure

In Bluetooth Low Energy, devices use advertising to announce their presence and potentially connect to other devices. Advertising packets are sent periodically at specific intervals called advertising intervals (from 20 ms to 10.24 s). Smaller intervals mean more frequent advertising but higher power consumption. BLE devices use 40 channels, including three primary advertisement channels for advertising and 37 secondary channels mainly for data transfer after a connection is established. The frequencies of the three channels used during advertising are selected in order to reduce the risk of interference from other technologies using the ISM band. Similar to the advertising interval, the scan interval is used by the central to define how often it will scan searching for advertising packets. The scan interval is used in conjunction with another parameter called scan window that represents the actual time the device is scanning inside the scan interval. Thus tuning the advertising interval and the scan interval it is possible to perform a trade off between power consumption and discovery time.

B.4 Connection process

The process of establishing a connection between BLE devices involves two key roles: the peripheral and the central. The peripheral device actively advertises its presence by broadcasting advertising packets, while the central device scans for these advertisements. When the central detects an advertising packet from a peripheral, it decides whether to initiate a connection based on the contents of the advertisement. Upon receiving a connection request from the central, the peripheral accepts it, leading to the establishment of a bi-directional connection channel. Once the connection request is successfully received, the two devices are considered to be in a connected state. During this phase, they transition from using the advertising channels (channels 37, 38, and 39) to data channels (channels 0 to 36). To minimize interference and enhance data throughput, BLE employs channel hopping, where the channel used for data transmission changes frequently. This approach ensures that if certain channels are noisy in the environment, the devices can switch to a cleaner channel in the next connection interval, reducing the chances of packet loss. For data integrity, all packets transmitted over Bluetooth LE are retried indefinitely until an acknowledgment is received or the connection is terminated. This reliability ensures that data is accurately delivered between devices. A unique characteristic of Bluetooth LE connections is their low-power nature. In a connection, both the peripheral and central devices spend the majority of their time in a sleep state. They agree on specific intervals, known as connection intervals, during which they wake up to exchange data. These sleep intervals help

conserve power, making Bluetooth LE a suitable choice for low-power, battery-operated devices. The connection event, which occurs at each connection interval, marks the time when both devices wake up to communicate.

B.4.1 Connection parameters

Upon a connection between two BLE devices, a set of parameters is exchanged called connection parameters:

- **Connection interval**: it determines how often two connected devices communicate. It's the agreed-upon time interval at which they wake up, exchange data, and then go back into a low-power, idle state.
- **Supervision timeout**: it is the parameter that decides how long it should take, after the last successful data exchange, for the devices to consider the connection lost.
- **Peripheral latency**: it is a feature that allows the peripheral device to skip waking up for a certain number of connection events.

B.5 Data exchange

Data exchange in Bluetooth LE refers to the operations that are executed between the server and the client to learn about attributes and exchange their values in accordance with attributes permissions. As pointed out in the previous sections, the communication is based on a client-server architecture where the client can ask data to the server or the server can directly send data to the client. Thus the classification in client-initiated operation or server-initiated operation. In the first case, the operations are

- **Read**: if the client wants to read a value from an attribute contained in the GATT server, it sends a read request command to the server that will respond with the wanted value.
- **Write**: If the client wishes to write a certain value to an attribute, it sends a write request and provides data that matches the same format of the target attribute. If the server accepts the write operation, it responds with an acknowledgement.
- **Write without response**: in this case the client does not wait an acknowledgment from the server after the writing operation

The second category of GATT operations consists of server-initiated operations, in which the server transmits information directly to the client without waiting for a prior request. Within this category, the server has the option to either notify or indicate. Among them there are

- **Notify**: A Notify operation is employed by the server to transmit the value of a specific attribute to the client, without requiring the client to initiate the request. For instance, this can be utilized to keep the client updated on recent changes in sensor readings.
- **Indicate**: it is similar to the notify but in this case an acknowledgment from the client is required [17]

Bibliography

[1] Stefan Schneid and Henning Gieseler. «Evaluation of a New Wireless Temperature Remote Interrogation System (TEMPRIS) to Measure Product Temperature During Freeze Drying». In: *AAPS PharmSciTech* (2008) (cit. on p. 4).

[2] Analog Devices, Inc. *AD7799 Datasheet* (cit. on p. 10).

[3] Walt Kester. «ADC Architectures III: Sigma-Delta ADC Basics». In: (Jan. 2011) (cit. on p. 11).

[4] Nordic Semiconductor. *nRF52 Series reference manual - Nordic infocenter* (cit. on p. 14).

[5] Lorenzo Vangelista. «Frequency Shift Chirp Modulation: The LoRa Modulation». In: *IEEE Signal Processing Letters* 24.12 (2017), pp. 1818–1821 (cit. on p. 15).

[6] Winbond. *3V 32M-BIT SERIAL FLASH MEMORY WITH DUAL/QUAD SPI AND QPI* (cit. on p. 16).

[7] Analog Devices. *ADT7410, ±0.5°C Accurate, 16-Bit Digital I2C Temperature Sensor* (cit. on p. 17).

[8] Microchip Technology Inc. *MCP73831/2, Miniature Single-Cell, Fully Integrated Li-Ion, Li-Polymer Charge Management Controllers* (cit. on p. 18).

[9] Texas Instruments. *DRV5032 Ultra-Low-Power Digital-Switch Hall Effect Sensor datasheet* (cit. on p. 31).

[10] K.A.A. Makinwa. «Smart temperature sensors in standard CMOS». In: *Procedia Engineering* 5 (Dec. 2010), pp. 930–939 (cit. on pp. 88, 90, 91).

[11] Mehmet Co-Advisor and Mehmet Parlak. «2014 An instrument for the high temperature M Gunes». In: (2015) (cit. on p. 92).

[12] Matthew Duff and Joseph Towey. «Two Ways to Measure Temperature Using Thermocouples Feature Simplicity, Accuracy, and Flexibility». In: (Oct. 2010) (cit. on pp. 93, 94, 96).

[13] Joseph Wu. «A Basic Guide to Thermocouple Measurements». In: (Sept. 2018) (cit. on p. 98).

[14] TE Connectivity. *Understanding RTDs, application notes* (cit. on p. 99).

[15] Joseph Wu. «A Basic Guide to RTD Measurements». In: (June 2018) (cit. on p. 102).

[16] Hadi Ebrahimi-Darkhaneh. *Error analysis in temperature sensing with NTC and silicon-based PTC thermistors: Comparing the Ratiometric and Absolute methods* (cit. on p. 103).

[17] Nordic Semiconductor. *Bluetooth Low Energy Fundamentals*. URL: `https://academy.nordicsemi.com/courses/bluetooth-low-energy-fundamentals/` (cit. on p. 110).