Development of a low-power NB-IoT device
for remote environment monitorization

Sara Pereira

UMinho | 2023

**Universidade do Minho**
Escola de Engenharia

Sara Miranda Pereira

# Development of a low-power NB-IoT device for remote environment monitorization

janeiro de 2023

**Universidade do Minho**
Escola de Engenharia

Sara Miranda Pereira

# Development of a low-power NB-IoT device for remote environment monitorization

Dissertação de Mestrado
Engenharia Eletrónica Industrial e Computadores
Sistemas Embebidos e Computadores

Trabalho efetuado sob a orientação do
**Professor Doutor Jorge Cabral**

janeiro de 2023

## DIREITOS DE AUTOR E CONDIÇÕES DE UTILIZAÇÃO DO TRABALHO POR TERCEIROS

# Acknowledgements

This dissertation symbolizes a major milestone after more than five years of work at the University of Minho, and more importantly, a lifetime of growth.

I want to start by thanking my advisor, Professor Doctor Jorge Cabral, for his guidance and care. To Sofia Paiva and Rui Machado, a special thanks for guiding me this year with much patience and dedication. Also, I want to thank Rui Teixeira for helping me with his immense knowledge and experience with hardware.

Thank you to those who so generously welcomed me to this course and helped me countless times: Alexandra Ferrete, Isabel Silva, Nuno Rodrigues, João Carvalho, José Mendes, Hugo Carvalho, João Faria, and my teammate through many sleepless nights, my friend Mariana Duarte.

To those friends whom I got to know later on this journey, however no less important: João Relvas, Rui Costa, Rui Esteves, Pedro Duarte, Alexandre Mano, and Simão Leite. Thank you (and your girlfriends) for all the adventures and laughs shared.

To my friends from past journeys: Margarida Fertusinhos, Joana Telhada, Tânia Lopes, and Sara Gonçalves. Thank you for all these years of friendship.

To Zé, who showed me what it is really all about. Thank you for always understanding me, supporting me, and being my confidant. Thank you for spreading joy in my life and knowing the magic tricks to make me smile. It is a true blessing to have you in my life.

To my amazing cousins: Afonso, Maria, Rita, David, Eduardo, and Márcia, who took care of me growing up. And to the dearest aunts and uncles: Tia Clara, Tio Quim, Tia Ana, and Tio Marco. Thank you!

Finally, I would like to express my appreciation to those who have been there for me through the ups and downs for as long as I can remember. To my brother and sister, Guilherme and Bárbara, my father, João, and my dear mother, Carla. More than giving me opportunities through your efforts, you gave me unconditional support and a sense of certainty that everything would be just fine.

I will be eternally grateful to each one of you, as well as to many others who helped me in this journey.

This dissertation was developed in scope of the project ”(Link4S)ustainability - A new generation connectivity system for creation and integration of networks of objects for new sustainability paradigms [POCI-01-0247-FEDER-046122 | LISBOA-01-0247-FEDER-046122]” which is financed by the Operational Competitiveness and Internationalization Programmes COMPETE 2020 and LISBOA 2020 under the PORTUGAL 2020 Partnership Agreement, and through the European Structural and Investment Funds in the FEDER component.

## STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity.  I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

# Resumo

IoT é um conceito cada vez mais presente nas mais diversas áreas, sendo crescente o seu interesse para diferentes aplicações de monitorização. Por vezes, os ambientes que se pretendem monitorizar são remotos e de difícil acesso, precisando certos de cuidados para que seja exequível a sua monitorização. O projeto Link4S tem como um dos seus objetivos responder a esta necessidade da maneira mais eficiente possível, sendo imprescindível que haja uma boa cobertura de rede e que a duração da bateria seja longa. As LPWAN são uma boa solução para a problemática da cobertura de rede com bom trade-off entre o alcance e o consumo de energia. Mais concretamente, as soluções Narrowband-IoT (NB-IoT) de baixa potência surgem como possível resposta à necessidade de grande durabilidade de bateria. Posto isto, existem diversos modems comerciais que fazem uso da tecnologia NB-IoT e existem também diferentes MCUs que trarão diferentes vantagens e desvantagens, quer em termos de performance quer em termos de consumos de energia. Estas variantes e o seu impacto no desenvolvimento de um dispositivo IoT de reduzido consumo de energia com recurso a comunicação sem fios NB-IoT para aplicações de monitorização é o que se pretende avaliar nesta dissertação. Para isso é feito o refactoring do código Link4S já existente para funcionar com a nova placa e modem Murata Type1SE, que se diferencia por ter o STM32L462RE (MCU) integrado e também ter a possibilidade de trabalhar com eSIM. Este novo módulo é testado como parte de um Discovery Kit. Ainda no contexto desta dissertação, foi feito o design e de uma nova PCB customizada para acomodar este mesmo módulo, na qual funcionará a mesma aplicação, pois é esperado que o seu consumo de energia seja menor.

**keywords:** *IoT, NB-IoT, Modem, Low-power*

# Abstract

Internet of Things (IoT) is a concept that is increasingly present in the most diverse areas, and its interest in different monitoring applications is growing. Sometimes, the environments to be monitored are remote and difficult to access, requiring a series of steps to ensure monitoring is feasible. One of LinK4S' goals is to respond to this need in the most efficient way possible, since it is essential that there is a good network range and that the battery life is long. LPWANs are the solution to network problems with the best trade-off between range and energy consumption. More specifically, low-power Narrowband-IoT (NB-IoT) solutions emerge as a possible answer to the need for high-energy durability. That said, there are several modems that make use of NB-IoT technology, and there are also different MCUs that will bring different advantages and disadvantages, either in terms of performance or in terms of energy consumption. These variants and their impact on the development of a low-power NB-IoT device for monitoring applications are what we intend to evaluate in this dissertation. Regarding this, the existing Link4S code is refactored to work with the new Murata Type1SE modem, which is different from the previous work as it has the STM32L462RE (MCU) integrated and also has the possibility of working with eSIM. This new module is tested as part of a Discovery Kit. A new and customized PCB is designed to accommodate this same module and run the same application, in the context of this dissertation, as it is expected to be less energy consuming.

**keywords:** *IoT, NB-IoT, Modem, Low-power*

# Contents

# List of Figures

# List of Tables

# Code Snippets

# List of Abbreviations

| | |
|---|---|
| 3GPP | Third Generation Partnership Project. |
| AES | Advanced Encryption Standard. |
| BoM | Bill of Materials. |
| BOR | Brown Out Reset. |
| CIoT | Consumer Internet of Things. |
| CPS | Cyber-physical systems. |
| DL | Downlink. |
| EC-GSM-IoT | Extended Coverage GSM IoTs. |
| eDRX | Extended Discontinuous Reception. |
| ESD | Electrostatic Discharge. |
| eSIM | embedded SIM. |
| GPIO | General Purpose Input/Output. |
| GSM | Global System for Mobile Communications. |
| GSMA | Global System for Mobile Communications Association. |
| HAL | Hardware Abstraction Layer. |
| I2C | Inter-Integrated Circuit. |
| IIoT | Industrial Internet of Things. |

| | |
|---|---|
| IoT | Internet of Things. |
| LPWAN | Low Power Wide Area Network. |
| LTE | Long Term Evolution. |
| MCU | Microcontroller Unit. |
| NB-IoT | Narrowband Internet of Things. |
| PCB | Printed Circuit Board. |
| PSM | Power Saving Mode. |
| PTW | Paging Time Window. |
| PWPN | Power Wireless Private Networks. |
| QoS | Quality of Service. |
| RAT | Radio Access Technology. |
| RAU | Routing Area Update. |
| RC | Resistor–capacitor. |
| RFID | Radio Frequency Identification. |
| RTC | Real-time clock. |
| SC-FDMA | Single-Carrier Frequency-Division Multiple Access. |
| SOA | Service Oriented Architecture. |
| SPI | Serial Peripheral Interface. |
| TAU | Tracking Area Update. |
| TVSS | Transient Voltage Surge Suppressor. |

| | |
|---|---|
| UART | Universal Asynchronous Receiver-Transmitter. |
| UL | Uplink. |
| USART | Universal Synchronous Asynchronous Receiver Transmitter. |
| WLAN | Wireless Local Area Network. |

# Chapter 1: Introduction

This chapter begins by outlining the context, inspiration, and objectives of this dissertation. First, the contextualization offers a crucial viewpoint on the dissertation's overall structure. The reason for undertaking a study on this subject is then discussed, followed by the dissertation's goals. The final section of this chapter walks the reader through the document's structure.

## 1.1 Contextualization and Motivation

A vast variety of industrial, commercial, and military devices, from toys and vehicles to tanks and aircraft, contain embedded systems. The term ”embedded system” refers to a class of small, specialized computers that are embedded in their surroundings and have a variety of inputs and outputs.

Traditionally, embedded systems had little communication with other devices and were created for a specific purpose. Real-time processing of the information gathered from sensors in the real environment was the goal. When the internet and other communication means were introduced, the term Internet of Things (IoT) was added to the technology lexic. A networked sensor (or actuator) node is a term used to describe an embedded system that has communication capabilities, either wired or wireless, as well as different types of sensors or actuators [1].

One of the primary challenges with the rise of the Internet of Things (IoT) is developing an energy-efficient solution that allows devices in remote locations with no access to a power supply to operate for years without requiring maintenance (e.g. a battery replacement). Since sensing applications do not require large data transfers, a battery life of years in the deployment phase is achievable. For providing a wireless communication link, NB-IoT is a communication technology that enables IoT commercial devices to be deployed successfully. Given this, it is of interest to assess different modems, boards, and software stacks that will further optimize the IoT system's energy usage.

## 1.2 Objectives

The intention of this dissertation is to develop a low-power NB-IoT end device for monitoring applications. The device must research in remote locations. That's why a wide-range network is needed. The work will be focused firstly on migrating the current Link4S solution to work with the Murata Type1SE modem, on the BL462ECELL board. Later, a custom board will be specifically designed to accommodate the modem. The comparison of these new solutions with the already existing and the comparison of power consumption of the Murata Type1SE modem in comparison with the Quectel BC66 modem are two of the final goals. Thus, the proposed objectives for this dissertation are:

- **Migration from the current Link4S solution to a recent development board, BL462ECELL**
  The existent Link4S application has great level of portability among STM boards, however the code still needs to be refactored to function properly with the new hardware.

- **Interface development for Murata modem and BL462ECELL board sensors**
  The modem to be studied in this dissertation is different from the modems previously used with this application. This way, the modem module and some of the code associated with it will need to be developed.

- **Design of a PCB to accommodate the Murata Type1SE modem along with sensors and other peripherals**
  Since the BL462ECELL has a great number of unused peripherals, it is possible to achieve greater battery life by designing a board with only the necessary hardware.

- **Analysis and evaluation of system performance and energy consumption**
  Since the project's main goal is to minimize energy consumption, it is a priority to test this factor. Also, it is important o test if the application works as predicted and that there are no errors in run-time for both BL462ECELL Discovery Kit and the custom board.

- **Comparison of energy consumption**
  It is of interest to compare the energy consumption of the new two solutions with the Murata Type1SE modem with the existing solution (with Quectel BC66 modem and STM32L081 MCU)

## 1.3 Document Structure

There are six chapters in this document, which will be briefly discussed next. The concepts utilized in the dissertation are presented after this introductory chapter, in the state-of-the-art chapter.

Chapter 3 is related to the system specifications, which summarizes the hardware used in the dissertation and the software tools. It explains the different low-power saving modes available for STM32L462RE and overviews the different peripherals that were selected for the custom board.

Chapter 4 contains the design and implementation of both the hardware and software. First, it covers the schematics for the custom board, which is followed by the layout. Afterward, the chapter switches its focus to the software part of the dissertation, which involves the refactoring of the code from an Arm Cortex M0+ to an Arm Cortex M4 and from a Quectel modem to a Murata modem.

The tests and outcomes of the developed system, the dissertation's conclusions, and some recommendations for follow-up study are all presented in the final two chapters.

# Chapter 2: State of the Art

In this chapter, several research and projects relevant to this dissertation were discussed. The main covered topics are related to IoT, Service Oriented Architecture (SOA) applied to the IoT, Low Power Devices, LPWAN and NB-IoT.

## 2.1 The Internet of Things

IoT, or the Internet of Things, is a network of objects embedded with electronic circuits, software and sensors. These objects collect and/or exchange data. The IoT allows the remote control and sensorization of objects and environments, resulting in a greater digitalization of the physical world and, consequently, better data management and increased efficiency [2].

The conventional architecture of an IoT device consists of Wireless Communication Technologies, Actuators, Microcontroller Unit (MCU), Power Management, and Sensors. The application determines the sensors and actuators, as well as the wireless technology to use [1]. An illustration of this traditional architecture is presented in Figure 2.1.

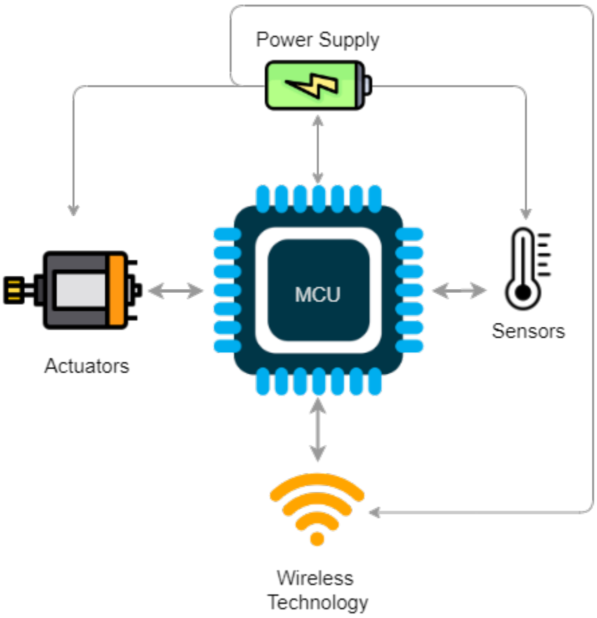


Figure 2.1: IoT Conventional Architecture.

### 2.1.1 IoT and Service Oriented Architecture

In designing the architecture of IoT, the extensibility, scalability, and interoperability among heterogeneous devices and their models should be taken into consideration.

Service Oriented Architecture (SOA) treats a complex system as a set of well-defined simple objects or subsystems. Those objects or subsystems can be reused and are maintained individually; therefore, the software and hardware components in an IoT device can be reused and upgraded efficiently. Due to these advantages, SOA has been widely applied as a mainstream architecture.

Trappey et al. [3] constructed a logical framework by layers to categorize IoT technology, which was then utilized to describe Cyber-physical systems (CPS). According to the publications [2] [4], the most frequent layering in an IoT network has four primary levels:

1. **Sensing Layer**
   To perceive the status of ”things” with a unique identity and to integrate, for example, actuators, sensors, RFID tags or others;

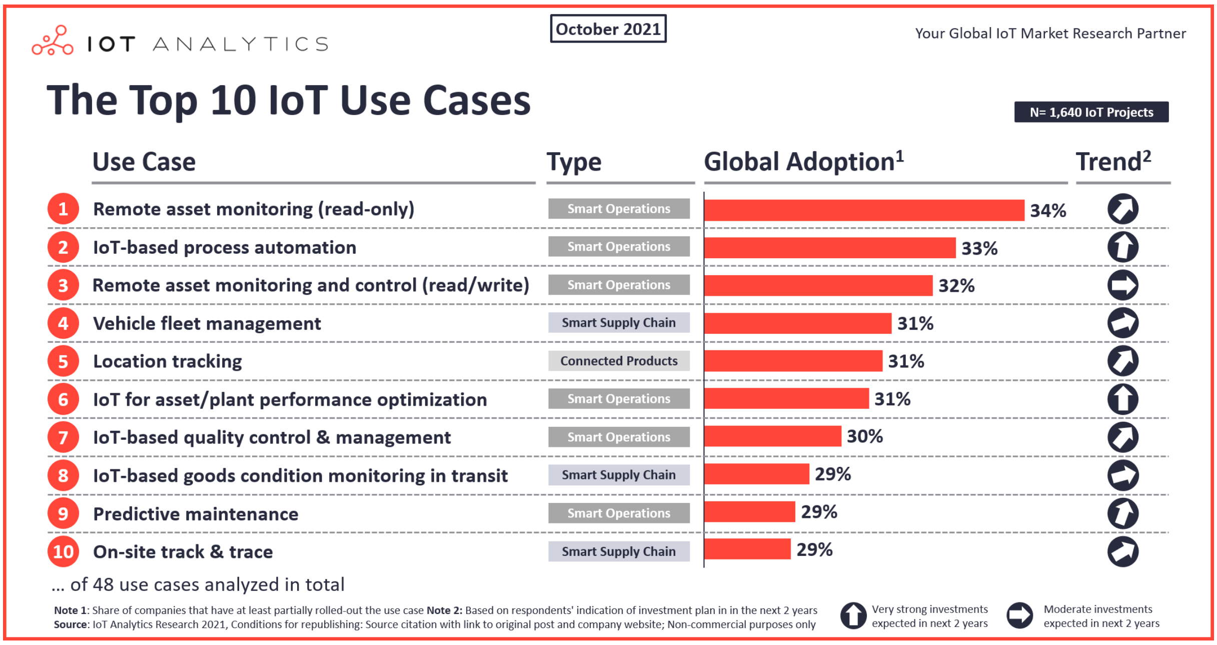
2. **Network Layer**
   To facilitate the flow of information from the sensing layer to the ”Service Layer” over a wired or wireless network. This layer automatically detects and registers devices in the network, allowing all devices to be connected for data sharing and exchange.

3. **Service Layer**
   Employs a middleware technology to support services and applications requested by users or applications. This layer ensures interoperability across heterogeneous devices by providing valuable functions such as information search engines and communication, data storing, sharing, and data management.

4. **Interface Layer**
   To support the interconnection and management of ”things” and to visualize data allowing the user to engage with the system in a simple and understandable manner.

### 2.1.2 IoT Use Cases

IoT is increasingly present in everyday life, leaving space for new terms according to the areas it is applied to, such as Industrial Internet of Things (IIoT) which refers to the industrial applications of IoT, or Consumer Internet of Things (CIoT), which refers to the Internet of Things in the context of consumer applications, use cases and devices [5]. Figure 2.2 [6] presents a survey made by IoT Analytics regarding the top 10 IoT use cases.



The Top 10 IoT Use Cases

N= 1,640 IoT Projects

| | Use Case | Type | Global Adoption[1] | Trend[2] |
|---|---|---|---|---|
| 1 | Remote asset monitoring (read-only) | Smart Operations | 34% | ↗ |
| 2 | IoT-based process automation | Smart Operations | 33% | ↑ |
| 3 | Remote asset monitoring and control (read/write) | Smart Operations | 32% | → |
| 4 | Vehicle fleet management | Smart Supply Chain | 31% | ↗ |
| 5 | Location tracking | Connected Products | 31% | ↗ |
| 6 | IoT for asset/plant performance optimization | Smart Operations | 31% | ↑ |
| 7 | IoT-based quality control & management | Smart Operations | 30% | ↗ |
| 8 | IoT-based goods condition monitoring in transit | Smart Supply Chain | 29% | → |
| 9 | Predictive maintenance | Smart Operations | 29% | ↑ |
| 10 | On-site track & trace | Smart Supply Chain | 29% | ↗ |

… of 48 use cases analyzed in total

**Note 1**: Share of companies that have at least partially rolled-out the use case **Note 2:** Based on respondents' indication of investment plan in in the next 2 years
**Source**: IoT Analytics Research 2021, Conditions for republishing: Source citation with link to original post and company website; Non-commercial purposes only

↑ Very strong investments expected in next 2 years
→ Moderate investments expected in next 2 years

Figure 2.2: Top 10 IoT use cases [6].

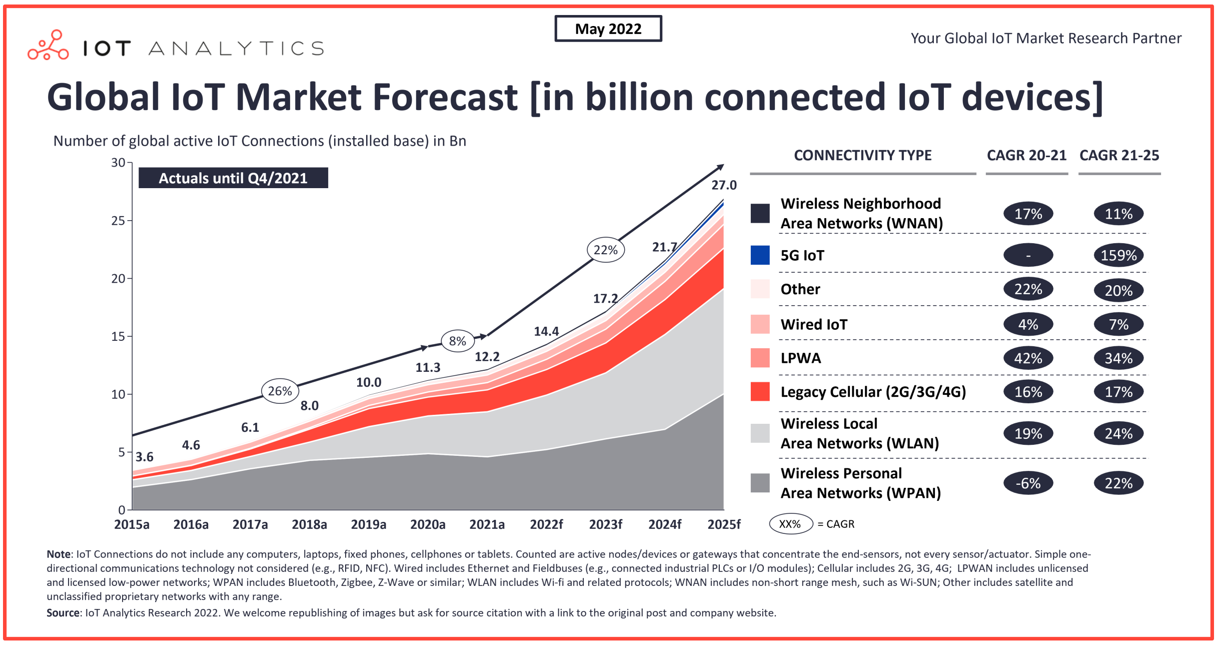
It is possible to see that, unsurprisingly, the main application of IoT is still the easiest and less complex to apply, which is "Remote asset monitoring (read-only)". Due to its simplicity, this use case is one of the least expensive to set up. Remote asset monitoring frequently replaces the labor-intensive, costly manual process of physically inspecting and recording asset conditions. The epidemic unmistakably speeded up adoption in 2020, and it is predicted that this trend will continue as 36 percent of interviewed companies said they intend to considerably increase their investment in this use case over the two years following the interview.

Other vastly adopted use cases are IoT-based Process Automation with 33%, and Remote Asset Monitoring and Control (read/write) with 32% of applications among the interviewed companies.

### 2.1.3 IoT Market

The intentions of investment are related to what might be the future market growth, however, it is not the only factor to take into consideration.

Even though there is currently a problem regarding the chip shortage and despite the long-term impact of COVID-19 on the supply chain, the internet of things industry is expanding. According to IoT Analytics, the worldwide number of connected IoT devices increased by 8% to 12.2 billion in 2021 and more than 27 billion IoT connections are expected by 2025, as represented in Figure 2.3 [7] [8] [9].



May 2022

Your Global IoT Market Research Partner

Global IoT Market Forecast [in billion connected IoT devices]

| CONNECTIVITY TYPE | CAGR 20-21 | CAGR 21-25 |
|---|---|---|
| Wireless Neighborhood Area Networks (WNAN) | 17% | 11% |
| 5G IoT | - | 159% |
| Other | 22% | 20% |
| Wired IoT | 4% | 7% |
| LPWA | 42% | 34% |
| Legacy Cellular (2G/3G/4G) | 16% | 17% |
| Wireless Local Area Networks (WLAN) | 19% | 24% |
| Wireless Personal Area Networks (WPAN) | -6% | 22% |

**Note**: IoT Connections do not include any computers, laptops, fixed phones, cellphones or tablets. Counted are active nodes/devices or gateways that concentrate the end-sensors, not every sensor/actuator. Simple one-directional communications technology not considered (e.g., RFID, NFC). Wired includes Ethernet and Fieldbuses (e.g., connected industrial PLCs or I/O modules); Cellular includes 2G, 3G, 4G; LPWAN includes unlicensed and licensed low-power networks; WPAN includes Bluetooth, Zigbee, Z-Wave or similar; WLAN includes Wi-fi and related protocols; WNAN includes non-short range mesh, such as Wi-SUN; Other includes satellite and unclassified proprietary networks with any range.

**Source**: IoT Analytics Research 2022. We welcome republishing of images but ask for source citation with a link to the original post and company website.

Figure 2.3: Global IoT market forecast (in billion connected IoT devices) [7].

IoT device actuals for 2021 and the current projection for 2025 are both lower than anticipated. The earlier forecast for 2025 predicted 27.1 billion connected IoT devices; the prior estimate for 2021 predicted 12.3 billion. The dip in the growth curve is caused by three fundamental factors:

- The COVID-19 pandemic has had an influence on both demand and supply. Supply was decreased since manufacturing was suspended at times, and supply chains and raw material access were disrupted. Budgets were frozen in the first half of 2020. Demand increased in the second half of 2020, but supply was often interrupted.

- The chip shortage began as a result of the COVID-19 impact on the supply chain. However, it has created its own challenge; there is insufficient supply capacity to fulfill global chip demand.

- As inflation increases in the majority of the world's countries, forecasts for global growth are becoming less optimistic. This raises expectations for rising interest rates and a subsequent slowdown in the economy.

In terms of connectivity, new technological standards such as fifth-generation (5G), Wi-Fi 6, and LPWA are propelling device connections. Also, satellite IoT has the potential to have a larger influence near the end of the projection period.

It is worth noting that, even though the total number of IoT connections predicted has been reduced for the reasons stated above and the growth of connected devices slowed in 2021, it is anticipated to pick up again in 2022 and beyond. The number of linked IoT devices is expected to rise to 14.4 billion by the end of 2022, despite the fact that new challenges for the sector have surfaced, including inflation and extended supply shortages.

## 2.2 Wireless Technologies

Even though IoT is now a growing technological revolution, it was an abstract concept for a long time. One of the factors that promoted this change were the several solutions developed with IoT in mind as for example the Low Power Wide-Area-Networks, as well as many other technologies that helped IoT through the years, such as:

- **Radio Frequency Identification (RFID)** A passive RFID system is made up of two major parts: a tag (radio signal transponder) and a tag reader. RFID technology is mostly used in commodities logistics. Recent research, however, is looking towards using the narrow RFID range for proximity detection and localization [10].

- **ZigBee** is a wireless personal area network protocol that is used to establish local, low-rate, and low-power networks [11]. The main advantages of ZigBee devices are their interoperability and ability to operate as part of a mesh network. As a result, the failure of a single device has no effect on the network as a whole.

- **Bluetooth** was designed to replace wiring in portable devices, but it has since expanded to be employed in a wide range of applications. BLE is a low-power variant of Bluetooth that is still intended for short-distance communications. BLE is envisioned as a connection option for IoT short-range connectivity [12].

- **Wi-Fi** is a wireless networking technology that is based on the IEEE 802.11 standard for radio Wireless Local Area Network (WLAN). Wi-Fi, which can handle large data speeds, is frequently utilized for mobile connections. Unfortunately, it has significant limitations for IoT because of its high energy consumption [13][14].

- **Cellular Networks** is a radio network that is spread out over land areas called cells and is supported by at least one base station, or fixed-location transceiver. Due to the high power consumption involved in the connection with the base station and the high cost per basis unit, cellular networks have always been a poor match for several IoT use cases. Nevertheless, 2G was chosen, in a number of IoT deployments, to connect with IoT devices using modest data transfers. This is changing as new IoT connectivity choices for cellular networks, such as Cellular LPWAN (covered in later in this section), become accessible.

- **Low Power Wide-Area-Networks (LPWAN)** are a new set of communication models designed to meet the different needs of IoT applications, complementing classic short-range wireless and cellular technologies. Some characteristics of LPWAN are: wide coverage; low bandwidth; potentially small packet and application-layer data sizes; extended battery life.

  The technologies are intended to facilitate communication between devices. Power consumption, battery life, cost, and bandwidth must all be balanced in restricted networks. By accepting substantial bandwidth and duty cycle limits, LPWANs emphasize power and cost benefits. Short-range wireless networks, such as Bluetooth, make other trade-offs, one of which is decreased coverage, which is restricted to a few hundred meters in an ideal case. The cellular networks Global System for Mobile Communications (GSM) and Long Term Evolution (LTE) provide extensive coverage. However, they do not reach the energy efficiency required for devices in remote locations that require a longer battery life. This is due to its features such as switching between base stations and frequent signaling, but primarily due to their high data rates [15] [16] [17].

  LPWANs can be divided into two categories:

  - **Unlicensed Low-Power Wide-Area-Networks** work in unlicensed spectrum, and various solutions exist, including SigFox and LoRaWAN.The unlicensed spectrum is used by anyone without any exclusivity. However, the usage of unlicensed spectrum has significant limitations due to short duty cycles and access procedures. As a result, unlicensed LPWAN technologies are frequently viewed as a supplement to current cellular networks.

- **Cellular LPWAN** are primarily designed to meet the connection needs of mMTC applications. These needs are, for example, wide coverage area, reduced device complexity, and great battery longevity. The Third Generation Partnership Project (3GPP) offered three LPWA solutions to accomplish this: LTE Cat-M1 (LTE-M), Narrowband Internet of Things (NB-IoT), and Extended Coverage GSM IoT (EC-GSM-IoT). The idea is to repurpose old cellular network infrastructure to handle IoT connection [14].

In Table 2.1, one can see a summary of the characteristics of the technologies mentioned above.

Table 2.1: Comparison between connectivity technologies [14].

| | Zigbee | BLE | Wi-Fi | LPWAN unlicensed | LPWAN licensed |
|---|---|---|---|---|---|
| Scalability | × | × | ✓ | × | ✓ |
| Reliability | × | ✓ | ✓ | × | ✓ |
| Low power | ✓ | ✓ | ✓ | ✓ | ✓ |
| Low latency | × | ✓ | ✓ | × | ✓ |
| Large coverage | × | × | ✓ | ✓ | ✓ |
| Low module cost | ✓ | ✓ | ✓ | ✓ | ✓ |
| Mobility support | × | × | × | × | ✓ |
| Roaming support | × | × | × | × | ✓ |

Figure 2.4 depicts how the different technologies behave regarding data rate and range. Low-Power Wide Area Network (LPWAN) technologies are ideal for low-end limited devices because of their long-range and low-power requirements, since these devices are typically power-restricted.

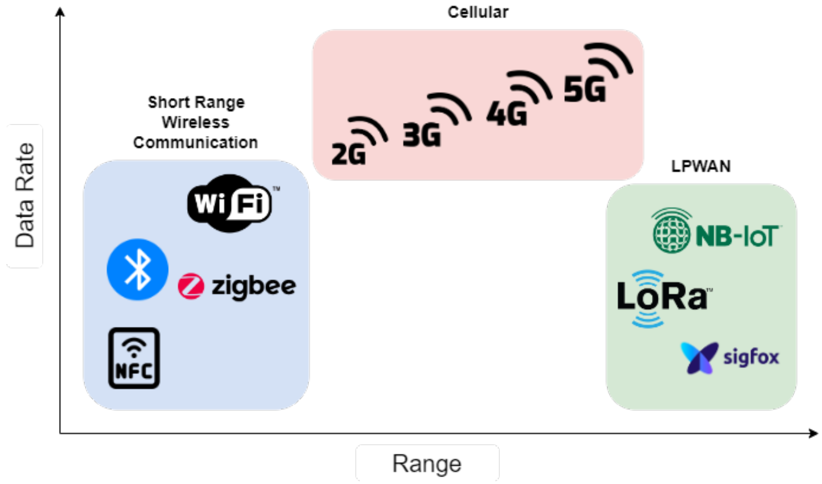


Figure 2.4: Different connectivity technologies regarding data rate and range (Adapted from [18]).

The dominant LPWAN technologies include Sigfox, LoRa, and NB-IoT, which will be presented and compared in the upcoming sections.

### 2.2.1 Comparison between Low Power Wide Area Networks

With the Internet of Things growing in popularity, there is a demand for devices that do not transfer a lot of data, are inexpensive, have long-range coverage, and have exceptionally small power requirements.

**Sigfox** is a company that saw the potential and developed a solution that fulfills the expectations of extremely low data rate and long-range use cases, with payloads of 12 bytes, a maximum of 140 transmissions per day, and a coverage range of 10km in urban and 40km in rural areas [19].

**LoRaWAN** (long range wide area network) was launched by LoRa-Alliance, with an increased payload length of 243 bytes and limitless messages per day for certain applications. Sigfox and LoRa use unlicensed radio bands, which may be prone to interference from external communication systems, among other constraints [20].

**NB-IoT** is a protocol designed to use the LTE network's licensed bands and can be implemented with a software upgrade in addition to the existing LTE infrastructure. 3GPP specifies NB-IoT in release 13. It has a substantially higher payload length than the preceding technologies, with a maximum payload length of 1600 bytes. Since Sigfox and LoRa operate in unlicensed bands, they are unable to provide the same level of Quality of Service (QoS) as NB-IoT, therefore any IoT application that requires guaranteed QoS should use NB-IoT [21].

Sigfox, LoRa, and NB-IoT create the groundwork for User Equipments (UEs) that may be in sleep mode for the majority of their lifetime without broadcasting or receiving data. This significantly decreases a device's average energy usage for an application that uses communications infrequently enough to stay inactive long enough to enter sleep mode. The access mode used by NB-IoT demands greater peak current than the Sigfox and LoRa access modes.

NB-IoT and LoRa class C have lower bidirectional latency than Sigfox and other LoRa classes, but they use more energy. As a result, NB-IoT and LoRa class C are superior candidates for applications that require low latency. Sigfox and LoRa allow for up to 50.000 end devices per cell, however NB-IoT is more scalable, allowing for up to 100.000. Other important IoT considerations are network coverage, range, and cost. Sigfox offers the most extensive base station coverage, with a range of more than 40 kilometers, LoRa up to 20 kilometers, and NB-IoT up to 10 kilometers. As NB-IoT is an evolution from LTE, its deployment is restricted to the range of LTE base stations [15].

In conclusion, Sigfox, LoRa, and NB-IoT are suitable for a variety of application demands. The primary characteristics of NB-IoT, include scalability, latency performance, payload length, QoS, and battery life. It is also evident that Sigfox and LoRa address separate aspects of IoT, as Figure 2.5 represents, which makes it possible for these technologies to co-exist.

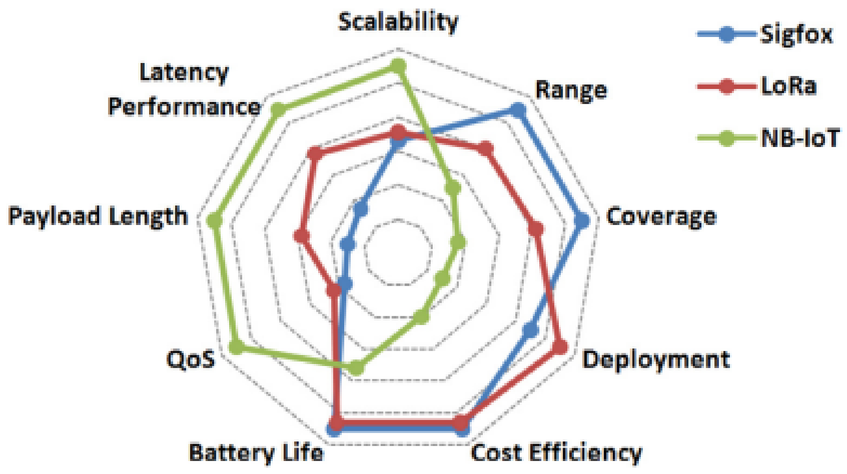


Figure 2.5: Respective advantages of Sigfox, LoRa, and NB-IoT in terms of IoT factors [15].

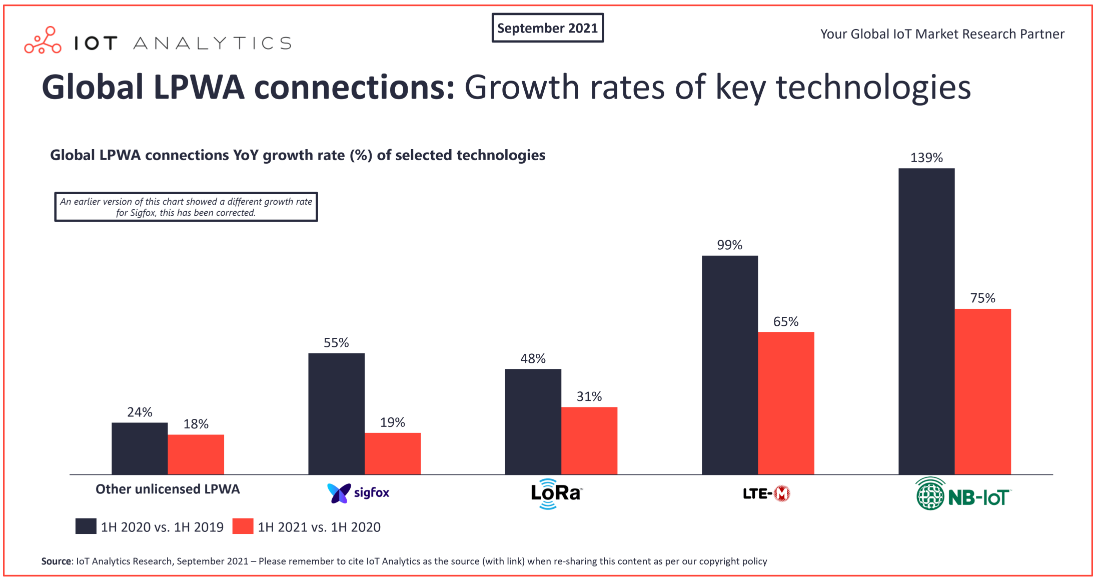
Table 2.2 summarizes the LPWANs discussed above, regarding some characteristics such as frequency, bandwidth, range, data rate, payload, among others. It is visible that each LPWAN can be more useful than the others depending on the specific use case or on the location [22].

Table 2.2: Predominant LPWAN techologies comparison (Adapted from [23]).

| | **LoRa-WAN** | **SigFox** | **NB-IoT** |
|---|---|---|---|
| **Licensed** | No | No | Yes |
| **Cellular** | No | No | Yes |
| **Standardization Group** | LoRa Alliance | SigFox | 3GPP |
| **Frequency (MHz)** | EU 868, EU 433, US 915, CN 490, etc. | EU 868, EU 433, US 915 | 700-900 |
| **Bandwidth** | 125 k, 250 k, 500k | 200 k (100 k each) | 180 k or 200 k |
| **Range (km)** | 3-10 (urban)<br>30-50 (rural) | 2-5 (urban)<br>10-20 (rural) | 1 (urban)<br>10 (rural) |
| **Payload (bytes)** | 0-243 | 12 (UL)<br>8 (DL) | 1600 |
| **Data Rate (bps)** | 300-50 k | 100 or 600 | 200 k |

### 2.2.2 LPWAN Market

Over 96 percent of the world's installed base of LPWAN-enabled active devices is made up of four different technologies: narrowband (NB)-IoT, long-range (LoRa), long-term evolution for machines (LTE-M), and Sigfox. One important factor is that NB-IoT connections increased by 75% year on year in the first half of 2021, as shown in Figure 2.6.



*An earlier version of this chart showed a different growth rate for Sigfox, this has been corrected.*

**Source**: IoT Analytics Research, September 2021 – Please remember to cite IoT Analytics as the source (with link) when re-sharing this content as per our copyright policy

Figure 2.6: Growth rates of global LPWAN connections (From IoT Analytics).

NB-IoT as a single technology currently has a 47 percent market share of the LPWA connection market, while LoRa has slid to second place with a 36 percent share of worldwide connections. These technologies held 94 percent of the market share in 2019; by 2021, the share had climbed by an additional two percentage points. [8].

The support from the top IoT vendors (such as Amazon, ARM, Cisco, Huawei, and Qualcomm) and network operators (such as Vodafone, Orange, and Telefonica), has contributed to the success of these four technologies and provided a wider range of products devices and applications, according to [8].

IoT Analytics also states China's digitalization initiatives rely on LPWAN more than those in other nations do, as reflected in Figure 2.7. As a result, China is the biggest user of the LPWAN, having the most NB-IoT and LoRa devices installed.

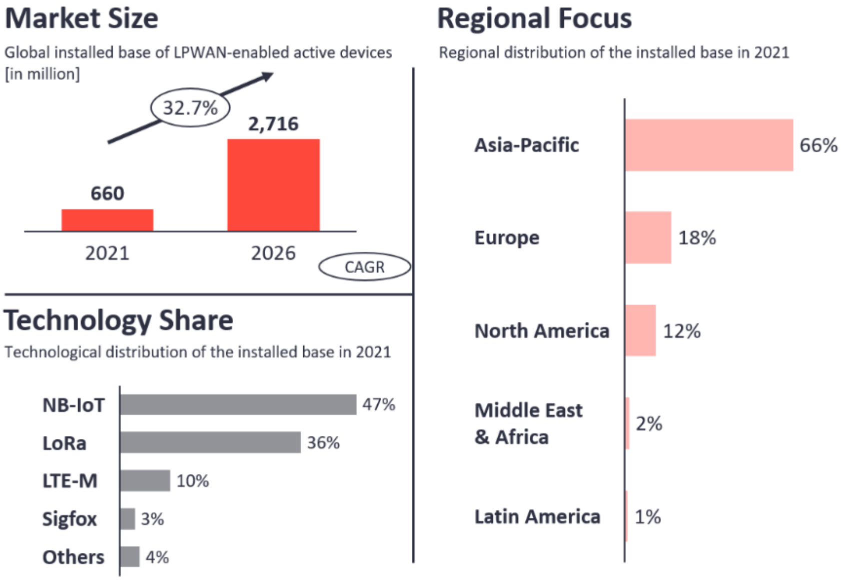


Figure 2.7: LPWAN Market Size (From IoT Analytics [8]).

Asset tracking and monitoring were the primary applications driving unlicensed LPWA growth in the previous 12 months, while monitoring systems, buildings and infrastructure business verticals drove NB-IoT growth. According to IoT Analytics, NB-IoT and LoRa/LoRaWAN will continue to dominate the industry over the next 5 years, with LTE-M and Sigfox behind in third and fourth place, respectively. While other technologies will continue to exist, it does not seem that they will play a substantial part in the broader global market at this time, albeit they remain appealing for specific applications [7].

### 2.2.3 Narrowband-IoT

The most notable new and established LPWANs technologies, as well as their various ways of meeting the demand for large area coverage while consuming little power were covered in the previous sections. Scalability, latency performance, payload length, and QoS were determined to be the characteristics where NB-IoT surpasses the other LPWAN options. This section will go through the link between LTE and NB-IoT, as well as major NB-IoT technologies and network architecture.

#### LTE architecture and its evolution

NB-IoT repurposes LTE (Long Term Evolution) capability for IoT devices by simplifying and optimizing it. When LTE was first established, the major needs for the new network were high peak data rates, quick round trip time, and flexibility in frequency and bandwidth. High peak data throughput in this context refers to throughput that is many orders of magnitude higher than that of the Universal Mobile Telecommunications System for 3G [24].

3GPP has specified various LTE upgrades in order to capitalize on the previously existing technology of LTE. The following were the objectives for IoT device connectivity [25]:

- Low-cost radios for devices;
- Thousands of devices per cell, transmitting a few bytes per day;
- Ultra-low power consumption (battery life of up to 10 years when transmitting a few bytes per day);
- Efficient support for devices with low data rates (in the order of tens and hundreds of kbits per second maximum throughput).

Because there are so many diverse IoT use cases with different needs, no solution can address them all. The most significant differences are in transmission frequency, bit-rate and coverage. Pure LTE channels with frequencies of 10MHz or 20MHz do not give enough inside coverage. In light of this, 3GPP has defined four LTE enhancements, as shown in Table 2.3: LTE Category 1 (LTE Cat-1), LTE Category 0 (LTE Cat-0), LTE Category M1 (LTE-M), and LTE Category NB1 (NB-IoT). The first three improvements mostly bring additional functionality to the current LTE interface.

Table 2.3: LTE Enhancements [26].

| | LTE Rel-8<br>Cat-1 | LTE Rel-8<br>Cat-0 | LTE Rel-12<br>Cat-M1 | Rel-13<br>NB-IoT |
|---|---|---|---|---|
| DL peak rate | 10Mbps | 1Mbps* | 1Mbps* | 0.2Mbps |
| UL peak rate | 5Mbps | 1Mbps* | 1Mbps* | 0.2Mbps |
| Duplex mode | Full | Half or Full | Half or Full | Half |
| UE bandwidth | 20 MHz | 20 MHz | 1.4 MHz | 0.18 MHz |
| Maximum transmit power | 23 dBm | 23 dBm | 20/23 dBm | 23 dBm |
| Relative modern complexity | 100% | 50% | 20-25% | 10% |
| *for full duplex mode | | | | |

**NB-IoT**

After discussing LTE advancements for IoT applications and introducing NB-IoT and some of its main characteristics, it is worth to note what makes the enhancement attractive for use cases requiring low energy consumption and extended coverage.

Since NB-IoT can coexist with GSM and LTE, it solves the congestion problem that arises in other LPWANs by inheriting the licensed bands of LTE, allowing it to utilise existing network gear and minimize implementation costs. The maximum NB-IoT data rate is lower when compared to other 3GPP technologies. However, cell coverage is improved and hardware complexity is lowered when compared to LTE, allowing NB-IoT to minimize costs and energy consumption.

The most visible change, as the name suggests, is the bandwidth, which is reduced when compared to Long Term Evolution Mobile (LTE-M). NB-IoT uses 180 kHz of spectrum at the physical layer, which is substantially less than the LTE spectrum of 1.4-20 MHz. NB-IoT has a subcarrier spacing of 15 kHz for downlink (DL), hence the 180 kHz channel is made up of 12 subcarriers. There are two distinct possibilities defined in the uplink (UL): It may employ a single carrier or a number of carriers. Single-tone: carrier spacing of 3.75 kHz (48 subcarriers) or 15 kHz (12 subcarriers). Multitone: 15 kHz carrier spacing for SC-FDMA (optional). On either side of the usable spectrum, a 10 kHz guard-band is occupied for both uplink and downlink [25].

As seen in Figure 2.8, the following operation modes are feasible with this frequency band selection [15]:

- In-band operation: utilizing resource blocks inside an LTE carrier.
- Guard-band operation: making use of the unused resource blocks in an LTE carrier's guard band.
- Stand-alone operation: one potential scenario is the usage of the GSM frequency bands now in use.

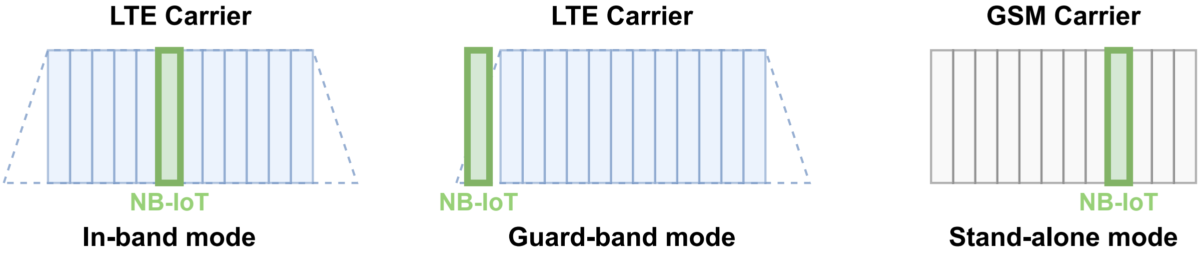


Figure 2.8: Operation modes for NB-IoT (Adapted from [15]).

**Cat-NB1** was the first version of NB-IoT. It was introduced in 3GPP Release 13. **Cat-NB2** is the most recent version of NB-IoT. It was introduced in 3GPP Release 14. Cat-NB2 increases NB-IoT's peak downlink data transfer speed and its uplink peak data rate. 3GPP Release 14 also introduces advanced positioning technologies for NB-IoT such as OTDOA (Observed Time Difference of Arrival) and Enhanced Cell ID, improving location accuracy. Also a new feature for NB-IoT devices is Radio Resource Control (RRC) connection re-establishment. This feature allows NB-IoT devices to transfer their cellular connection from one cell to another cell as the device moves between cells, without having to start this transfer process over again if the device experiences a radio link failure.

Table 2.4: Differences Between LTE Cat-NB1 and LTE Cat-NB2 (Adapted from [27]).

| | **NB-IoT** | |
|---|---|---|
| | **LTE Cat NB1** | **LTE Cat NB2** |
| **3GPP Release** | Release 13 | Release 14 |
| **Downlink Peak Rate** | 26 kbit/s | 127 kbit/s |
| **Uplink Peak Rate** | 62 kbit/s | 159 kbit/s |
| **Duplex Mode** | Half Duplex | Half Duplex |
| **Device Receive Bandwidth** | 180 kHz | 180 kHz |
| **Device Transmit Power** | 20 / 23 dBm | 14 / 20 / 23 dBm |

### 2.2.4 NB-IoT Applications

Power Wireless Private Networks (PWPN) in the power industries have employed NB-IoT to achieve broad and extensive coverage [28]. The authors' analysis further demonstrates that NB-IoT is appropriate for the latency tolerance services needed in PWPN.

According to [29], NB-IoT's power consumption feature has made it possible for aviation industries to use it, which will increase the efficiency of next-generation aircraft industries. Its application in smart cities is encouraging, and the coexistence of NB-IoT with the LTE infrastructure has been made possible by the incorporation of this technology into LTE functionalities [30].

More NB-IoT applications can be found in the utility sector, where smart meters and tracking are common [31].

In [32] concluded the use of NB-IoT-based e-healthcare systems is possible for a wide range of health-related professions, including caring for pediatric and elderly patients, the care of chronic illnesses and emergency response. Its research centered on the creation of a tool to automatically transmit the occurrence of a fall to people outside the patient's household, and to automatically communicate the location of the patient to EMS operators.

The benefits of NB-IoT for smart grids are numerous. It decreases energy usage and improves component compatibility. NB-IoT offers ideal qualities for a large scale deployment of sensors and motion-control devices for measuring, monitoring [33][34]. NB-IoT offers increased energy effectiveness and simple to deploy. The NB-IoT's sensors and actuators function for a longer amount of time than the other IoT variants. However, [34] also states that NB-IoT shouldn't be used for time-sensitive control and monitoring. Because a long latency may develop in the NB-IoT framework when the distance is significant and the transmitted power is constrained.

## 2.3 eSIM

The GSMA, the worldwide group responsible for creating the overall communications framework comprising about 800 mobile carriers, defines eSIM as an embedded universal integrated circuit card. When compared to traditional plastic, detachable SIM cards, the eSIM may be soldered directly into the device, providing greater flexibility because devices can immediately connect, regardless of where they are deployed or where they may go throughout their use. The advancement of the eSIM allows it to endure in harsh situations with severe temperatures, humidity, or vibration [35].

### 2.3.1 eSIM and IoT

Even though IoT has seen constant and exponential growth, it has struggled to match the market growth estimates set a decade ago. This is related to the complexities of project implementation and scale [36].

Swapping carrier profiles over the air is extremely critical when working with devices distributed all over the world, since switching real SIM cards is both expensive and time-consuming [37].

The fundamental capability of being able to transfer carriers over the air has various benefits, including:

- Manufacturing process has been streamlined.
- Resistance to manipulation and environmental factors
- Protection against network sunsets and pricing adjustments in the future Global LTE-M and NB-IoT solution for perpetual roaming constraints

The usage of eSIM throughout the manufacturing process makes it irrelevant whose country the device finally ends up in. Manufacturing gets more simplified and less complex. Once the device is in the deployment location, the most appropriate profile may be obtained, whether the selection is based on local roaming constraints or the profile provider's pricing.

The module may be entirely sealed throughout the production phase without the need to remove the eSIM from the handset. As a result, the product may be made more secure against physical manipulation by hostile people while also increasing tolerance to environmental conditions, as stated in [38].

Another key application for eSIMs is to circumvent roaming restrictions. Some countries impose these restrictions for 60-90 days. After that time, the devices are removed from the network. One workaround is to take the device out of the country and connect it to a different network. Transporting a great number of devices of different sizes is prohibitively costly. Loading a local profile where roaming limitations do not apply is the solution in such circumstances, and eSIM can readily support this.

The development of trust relationships resulting from the SIM's transition to the eSIM system is described in [39]. The authors demonstrate the standardization work that was done to keep the ecosystem's trust and reliability. It is shown through a case study that there are problems with responsibility distribution. There is the risk that in the future operators might be unable to foresee problems or pinpoint their core causes, resulting in difficulties in adequately explaining the issue to consumers or fulfilling their legal responsibilities to maintain the confidentiality and integrity of communications.

## 2.4 Power Consumption

IoT applications have distinct requirements and one of the requirements for this dissertation is to achieve low energy consumption in the device (> 10 years lifetime). Design considerations for low power include the main system elements such as the microcontroller, wireless communications, sensors and power management. Since one of the restrictions of this dissertation is to use NB-IoT, this will be one of the focus points.

### 2.4.1 Power saving modes of NB-IoT

IoT devices send and receive data on a regular basis. A device can sleep between periods of data transmission and reception to reduce power consumption and maximize battery life. The energy cost of entirely disconnecting from the network when sleeping and then re-attaching when awake is substantial, hence, in order to save power, the device stays attached during hibernation. The host network, on the other hand, will periodically page the device, which must wake up and reply. The device will then go back to sleep until it receives the next page or is required to wake up in order to transfer data.

Although the wake-respond-sleep cycle consumes a modest amount of energy, the total energy consumption of a device over its lifetime might be significant. Power Save Mode handles this by allowing IoT devices to agree with the network on an extended downtime. The network does not page the device during this time, so it only wakes up when it needs to deliver data or when the sleep period finishes.

The two main power-saving features for these technologies are Power Saving Mode (PSM) and Extended

Discontinuous Reception (eDRX), which are detailed below.

### PSM - Power Saving Mode

Power saving mode was first introduced in 3GPP Release 12. PSM is requested by the device via an attach, a tracking area update (TAU), or a routing area update (RAU). It offers the network two preferred timers, T3324 and T3412, respectively. The requested PSM period is the difference between them — T3412 minus T3324. The network selects the time values that will be used based on the keyword. It may accept or change the values requested by the device. The network saves connection state information, and the sleeping device stays registered with the network. A reattachment procedure is not required if the device awakens and sends data before the PSM period expires [40].

Many networks impose time limits on PSM periods. Each network has a maximum PSM span. When the device sends the values T3412 and T3324, the network applies the limit. If T3412 minus T3324 is greater than the maximum allowed, T3412 will be reset to the appropriate value.

During an active PSM period, the modem's radio is completely turned off, and the device cannot send or receive data. A simple illustration of how PSM works is shown in Figure 2.9.

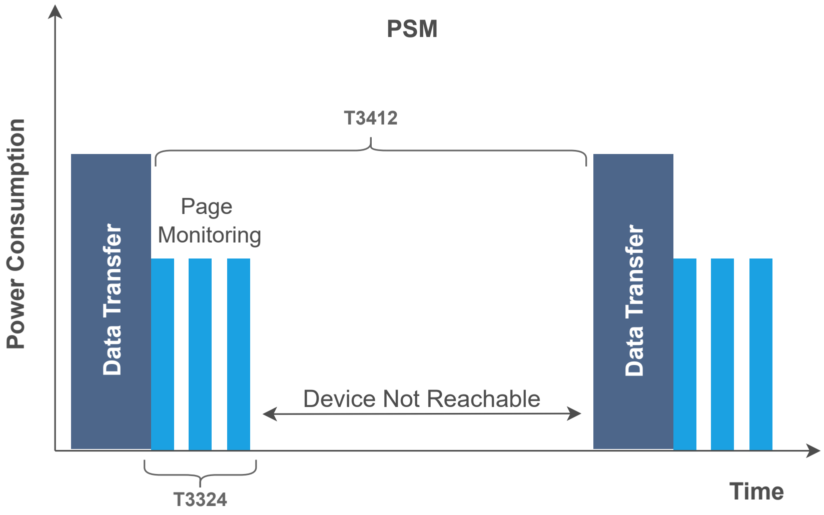


Figure 2.9: PSM (Adapted from [41]).

### eDRX - Extended Discontinuous Reception

Extended Discontinuous Reception (introduced in 3GPP Release 13) makes it possible to define a time period during which a device will remain in a low-power sleep state before waking to check for pending data, as illustrated in Figure 2.10. The device can monitor for pending data indications without requiring a full network connection. This consumes less power than a full network connection, and the time required to check for pending data is much less than the time required to establish a full network connection [40].

Depending on the application, it may be perfectly acceptable for a device to be inaccessible for several seconds or longer. eDRX allows the device to indicate that it will not be listening to the network for a longer period of time than usual. When eDRX is not enabled, the network determines how long a device can sleep before waking up. eDRX delegates that decision to the application. The device asks the network for a specific eDRX cycle; the network validates the eDRX cycle and provides a suitable Paging Time Window (PTW). Since only certain periods are permitted for eDRX, the network will reply with the closest official value, if the requested cycle time does not correspond to one of these [40].

eDRX has no effect on the module's ability to send data; it simply disables reception for the configured cycle. However, it does allow the application to choose a sleep time that is appropriate for its data flow, lowering radio use and thus power consumption.

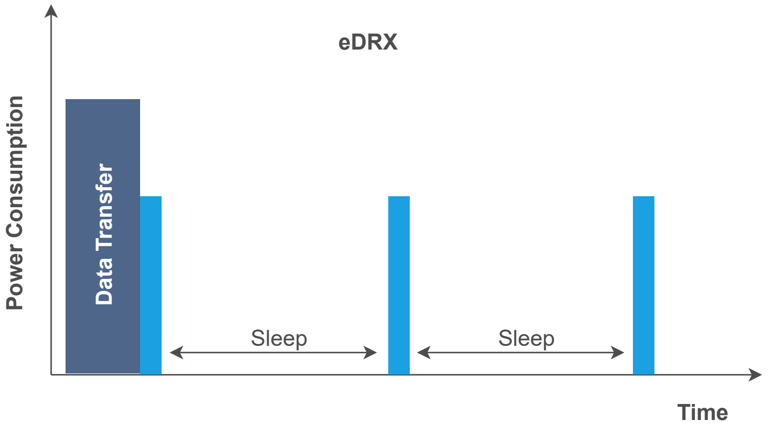


Figure 2.10: eDRX (Adapted from [41]).

### Comparing PSM and eDRX

Summarizing the information above, PSM and eDRX both allow energy saving [42].

- Power Saving Mode: allows the device to set sleep and active timers. After this timers are accepted by the network, the device does not need to be reattached every time it wakes up. During sleep phase, it is not available from the network side.

- Extended Discontinuous Reception consists in the extension of the time of the DRX that already existed. It allows the device to not listen to the network for a more extended period of time (eDRX inactive time), during the active time phase (Paging Time Window - PTW). In many applications of the IoT, it is acceptable to be unreachable for some time, seconds or even longer.

This makes the power consumption decrease, while the device is still reachable comparing to when the power saving mode is applied. However for the eDRX the energy saved is not as high as with PSM. These modes can be used simultaneously, as illustrated by Figure 2.11.

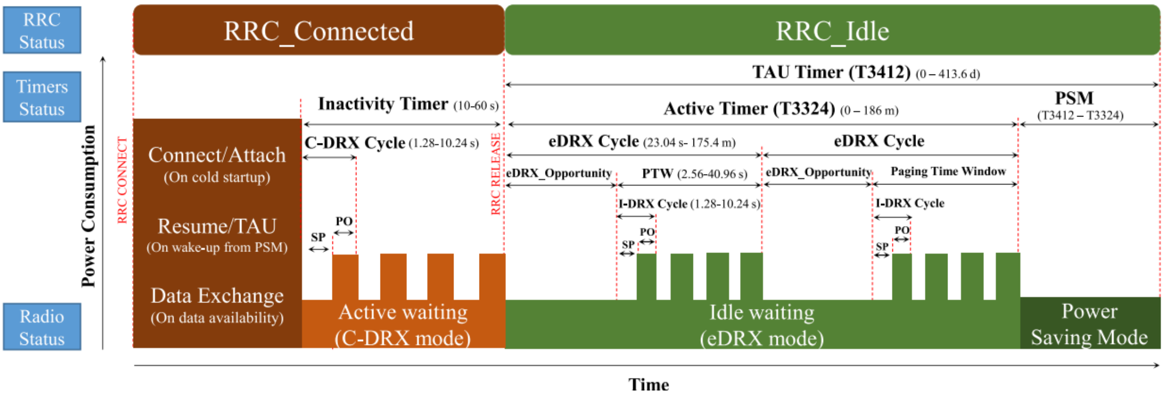


Figure 2.11: PSM and eDRX Power Saving Modes [43].

## 2.4.2 Power Saving Techniques

When dealing with wirelessly connected sensors, the most important problem to address is lowering energy consumption. Actually, a more accurate statement would be that the prime objective is to increase the operational lifetime of the network, but these two concerns frequently coexist. However, there are other techniques to increase the operating lifespan, such as lowering power consumption, extending sample and transmission intervals, employing renewable energy sources, or simply having larger batteries. Each of these techniques has merits and limitations. Using low-power optimization approaches is inexpensive since significant energy savings may be realized simply by making sensible software adjustments. The downside is that network performance may suffer since nodes spend most of their time in sleep mode [44].

Energy scavenging is an excellent method for increasing network longevity since a node that can harvest more energy than it consumes can exist indefinitely or until its hardware fails. The disadvantage is that the node cost and complexity increase. The alternative of just utilizing larger batteries can be quite costly, and it can increase the size of the nodes. Batteries scale poorly as well; doubling the capacity of a battery will double a device's operational lifetime, while low-power approaches can make a node able to operate several times longer.

The energy consumption of a sensor node may be separated into three major categories: the idle power consumption (the power consumed while in sleep mode); the energy spent by the device when sampling its sensors and processing sensor data; and wireless communication. These areas can be approached in a variety of ways in order to develop an energy-efficient sensor node.

The following are some common methods for reducing sensor node energy consumption.

- **Dynamic Voltage Scaling**

  The supply voltage of the microcontroller could be dynamically adjusted by a sensor node. Whenever running at low clock rates, modern microcontrollers may minimize the voltage. As a result, dynamic voltage scaling is frequently utilized in conjunction with dynamic frequency scaling. Because the power usage of a microcontroller is heavily reliant on the voltage supply, dynamic voltage scaling is an effective way to reduce total power consumption [45] [46].

- **Dynamic Frequency Scaling**

  A microcontroller's clock frequency could be dynamically adjusted to reduce power consumption based on the workload. Nevertheless, it is worthy to note that simply keeping the clock frequency low may increase energy usage because a lower clock frequency also means that some tasks may take longer to complete. In some instances, it may be advantageous to execute as rapidly as possible so that the microcontroller can revert to sleep mode. Because the energy consumption of a microcontroller is almost directly proportional to its operational clock frequency, dynamic frequency scaling is an appealing strategy for reducing power consumption [47] [46].

  In [47], a Dynamic Voltage and Frequency Scaling (DVFS) technique is implemented to reduce the power consumption of Virtex 5 FPGA. It was concluded that the proposed algorithm reduces 54.53% of total power.

- **Power down unused components**

  The power consumption of a system can be significantly lowered by shutting off components not in use. However, designers must keep in mind that some parts may have a high power-up cost in terms of time and energy. In some circumstances, putting a component to sleep rather than shutting it down may be more efficient. As a result, it is important to profile all components inside a system in order to properly explore their power consumption during different processes, startup times, and interactions with other elements of the system [44].

- **Duty-cycling (Using RTC)**

  Significant energy savings may be obtained by keeping a device in sleep mode for the majority of the time, with peripheral components switched off or into a low-power state. After turning on the relevant components, the device will periodically wake up and make measurements. After taking a measurement, the device can go back to sleep or activate its radio transceiver as needed. Wake-up can also be reactive, in the sense that system activation is not (only) regulated by the system's Real-time Clock (RTC). Instead, the system may be triggered by sensor responses or interruptions, as well as its radio transceiver and power supply.

A low-power scheduled alarm system using an LCD was developed in [48]. The power estimation of the automated bell system with and without sleep mode shows that, with the implementation of sleep mode using RTC for scheduling, 22.73% of total power is saved in the automated bell system.

Usually, a device will try to employ as many of these strategies as feasible to decrease power consumption; however, depending on the device hardware, the sensors, the application requirements, and the communication system, not all of the aforementioned methods may be accessible.

# Chapter 3: Specifications

This chapter describes the base system that lays the foundation on which this work builds upon. It is dedicated to the description of the system's hardware specifications. Also, at the end of the chapter, a brief software overview is presented in order to give context to the work done.

The device in use has a microcontroller that serves as the system's processing unit and is equipped with several sensors for monitoring environmental factors and an NB-IoT transceiver for wireless communication while minimizing power consumption. End-device management, end-device control, and data collection are all handled by the cloud. This system structure overview is depicted in Figure 3.1.

To improve energy efficiency, the device spends the majority of its operation in sleep mode, only waking up to read the sensors and eventually report an abnormality the moment any sensor wakes up the MCU. This is possible due to the selected sensors' capability to continuously sample and notify the MCU while operating in low-power mode.

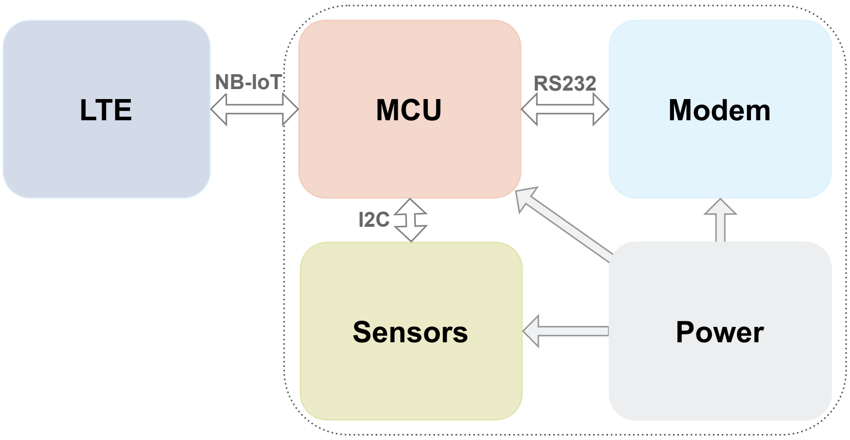


Figure 3.1: System Overview.

This dissertation explores a different alternative to the modem and MCU used in a previous work [49]. This alternative is based on a System-on-a-chip (SoC) that incorporates both the transceiver and the MCU, which may or not be advantageous in terms of power consumption.

## 3.1 Functional and Non-Functional Requirements

Since this is an ongoing project, it makes sense that many of the previous requirements still apply to the work reflected in this dissertation while adding some new requirements.

Usually, in order to produce a finished result, each project development has its own set of functional and non-functional requirements. Functional requirements specify what the system should accomplish, whereas non-functional requirements identify system properties. The prerequisites and limitations of our system are described below.

**Functional Requirements**

- Acquire environmental data;
- Send acquired data to the cloud;
- Signal unusual behavior;
- Receive cloud-based commands.

**Non-functional Requirements**

- Operate for at least ten years.

## 3.2 Hardware

Since the purpose of this dissertation is to use and assess the LBAD0ZZ1SE module (Type1SE), a B-L462E-CELL1 Discovery Kit was used. This kit served as a tool to prepare the code for the customized PCB to be developed, since both have the same MCU and transceiver (integrated into the LBAD0ZZ1SE module) and, therefore, can share a great part of the code for the application. The following sections will briefly describe the B-L462E-CELL1 Discovery Kit, the customized PCB and its components.

### 3.2.1 B-L462E-CELL1 Discovery Kit

B-L462E-CELL1 Discovery Kit [50] is well equipped for IoT applications that employ NB-IoT for wireless communications. The main reason this kit was chosen to work with was the fact that it offers the required tools and sensors that are needed to test and develop the application with the new MCU+Modem solution, integrated within the Type1SE [51].

The Discovery Kit, depicted in Figure 3.2, includes a global coverage antenna, and a low-power Discovery main board that is powered by an LBAD0ZZ1SE module. An ST4SIM-200M GSMA-certified eSIM, an LBAD0XX1SC-DM ultra-small LTE Cat M/NB modem, and an STM32L462REY6TR microcontroller are all included in the LBAD0ZZ1SE module. The ST-LINK debugger/programmer, extensive STM32Cube software libraries, and packaged software samples included in the B-L462E-CELL1 Discovery Kit enable end-to-end connection demonstration. Also, this Discovery Kit comes with a SW library (X-CUBE-CELLULAR) that hides the complexity of the management of cellular and related connectivity.

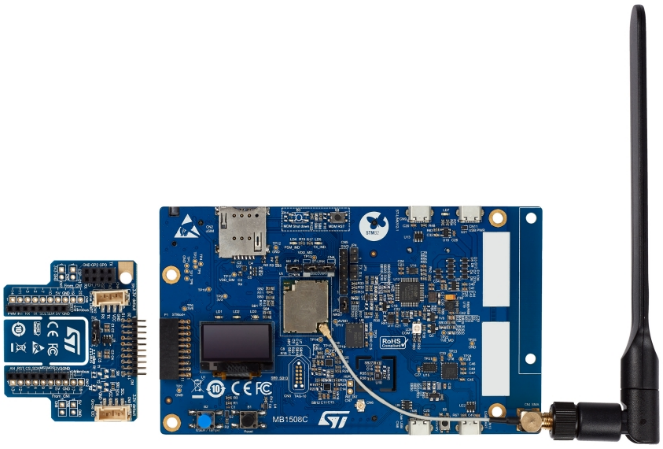


Figure 3.2: B-L462E-CELL1 Discovery Kit [50].

In Figure 3.3, it is shown a block diagram of the Discovery Kit hardware and connections that are intimately related to the Type1SE, where one can see the MCU and Modem part of Type1SE.

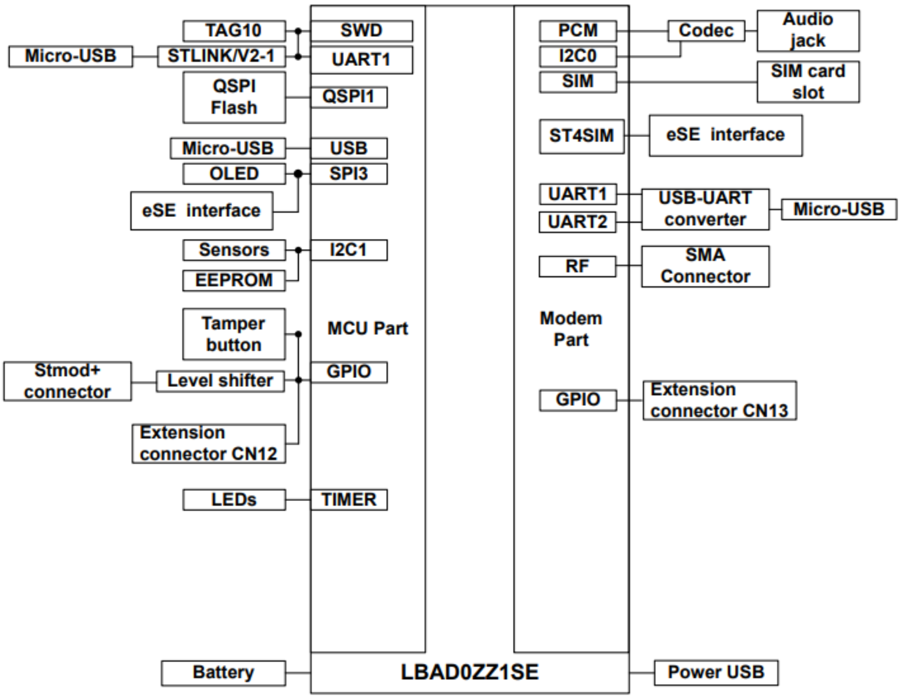


Figure 3.3: B-L462E-CELL1 Discovery Kit Hardware [52].

### 3.2.2 Customized Board

The customized PCB incorporates the LBAD0ZZ1SE module (Type1SE), the HDC2080, BMA400 and APDS-9306-065 sensors, an EEPROM, a SIM card slot, connnectors, battery support and other electronic parts. Its development had in mind the size of the board and power consumption, since the B-L462E-CELL1 Discovery Kit is expected to have greater power consumption due to the higher number of peripherals, despite both solutions having the LBAD0ZZ1SE (Type1SE) module as their MCU+Modem.

The block diagram of the board to be developed is depicted in figure 3.4. It is equipped with different sensors that are capable of measuring temperature and humidity (HDC2080), acceleration (BMA400), and light intensity (APDS-9306-065). For the purpose of communicating with the cloud the Murata Type1SE will be used, since it incorporates the NB-IoT Murata modem Type1SC. The Type1SE chip also integrates a Microcontroller Unit (MCU), which has many low power features.

The Saft LM17500 battery powers the system. The MCU and sensors are powered via a low quiescent TPS7A02 LDO, which regulates the supply voltage to 1.8 V.

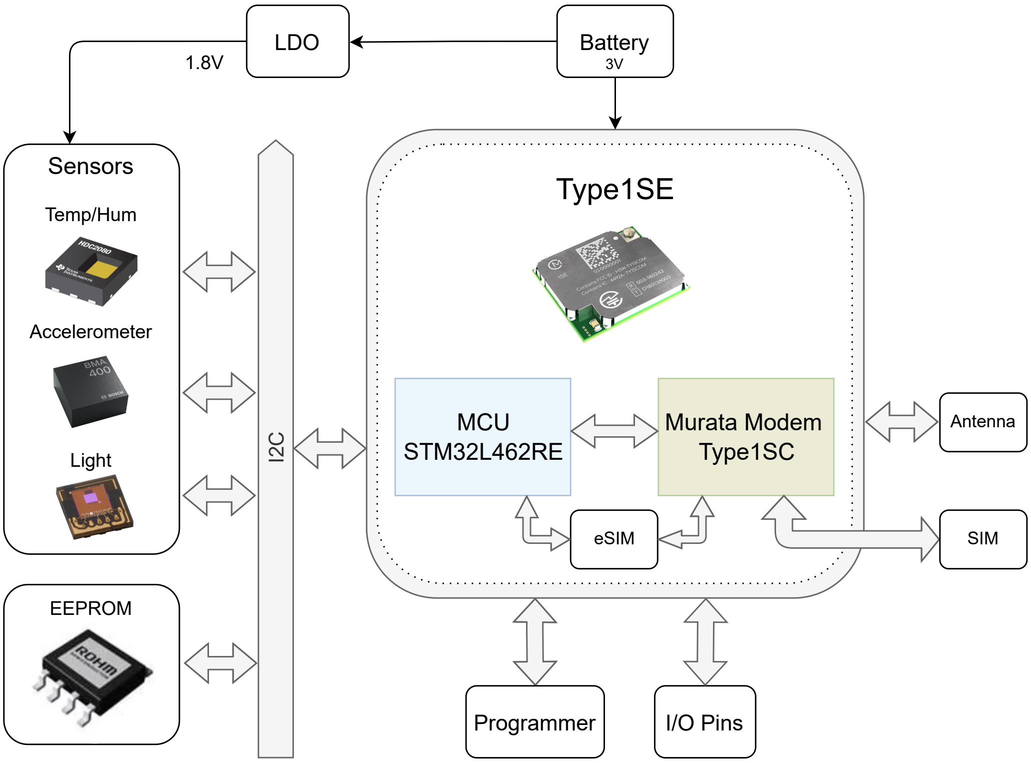


Figure 3.4: Customized Board System Overview.

### Type1SE

As mentioned above, the Type 1SE [51], LBAD0ZZ1SE, module is a Cat. M1/NB-IoT module. It is a device that incorporates MCU, eSIM (WLCSP) and LTE-M/NB1 modem. It has an antenna connection that enables many different types of IoT applications.

The appealing characteristics of LBAD0ZZ1SE for applications are: geolocation, its small size (15.4 x 18.0 x 2.5 mm) and its design for low current consumption, enabling up to 10-year battery life, and low cost. The module, presented in Figure 3.5 is equipped with:

- STM32L462REY6TR/Cortex M4 microcontroller w/512 KB Flash and 160 KB SRAM
- 1 MB on-board Serial Flash
- LBAD0XX1SC-DM ultra-small LTE Cat M/NB modem
- ST4SIM-200M GSMA-certified embedded SIM
- U.FL antenna connection

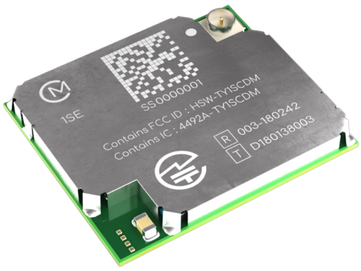


Figure 3.5: Type1SE [53].

Below, in Figure 3.6, it is possible to see a block diagram representation of the module.

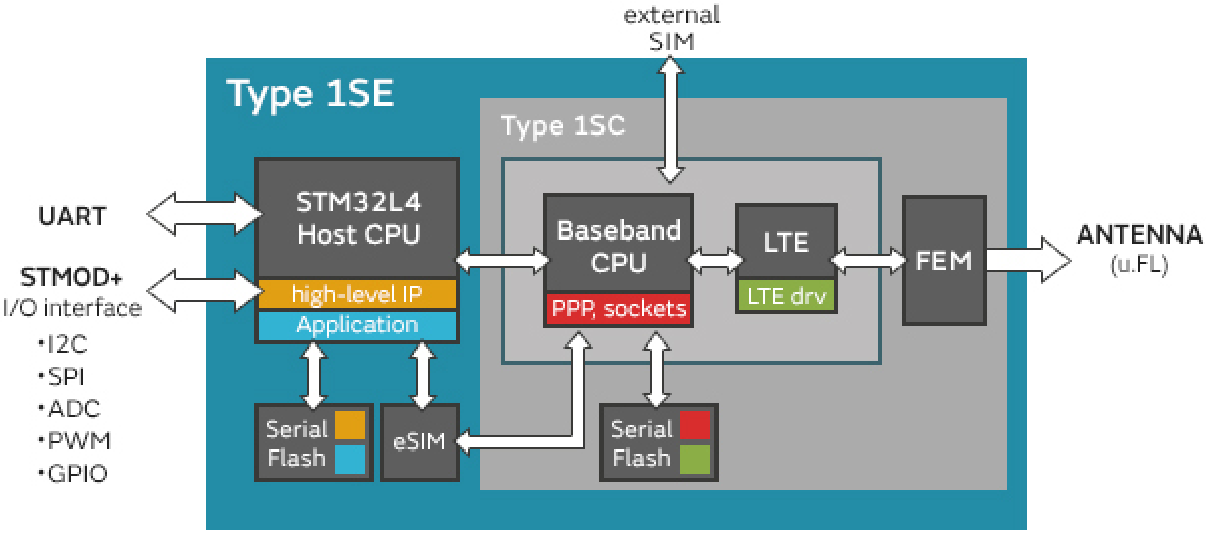


Figure 3.6: Type1SE Block Diagram [51].

### MCU

The STM32L462REY6TR [54] is an ultra-low-power microcontroller built around the 32-bit RISC Arm Cortex-M4 core that can operate at up to 80 MHz. A floating point unit (FPU) single precision is a characteristic of the Cortex-M4 core that supports all single-precision data-processing instructions and data formats. Additionally, it incorporates a memory protection unit (MPU) that increases program security. It also incorporates high-speed memories (Flash memory up to 512 Kbyte, 160 Kbyte of SRAM), a Quad SPI Flash memory interface, and a wide variety of enhanced I/Os and peripherals. The power supply can vary from 1.71 to 3.6 V.

The majority of the digital circuitries are powered by two built-in linear voltage regulators: the main regulator (MR) and the low-power regulator (LPR). Designing low-power applications is made possible by a wide range of power-saving modes that make different uses of the voltage regulators and clocks [55], as show in Table 3.1 and in the following text that summarizes the available low power modes:

Table 3.1: Variation of System Clock [55].

| Voltage Range | SYSCLK | MSI | HSI16 | HSE | PLL |
|---|---|---|---|---|---|
| Range 1 | 80 MHz max. | 48 MHz | 16 MHz | 48 MHz | 80 MHz |
| Range 2 | 26 MHz max. | 24 MHz | 16 MHz | 26 MHz | 26 MHz |
| Low-power run/sleep | 2 MHz max. | 2 MHz | Allowed | Allowed with divider | OFF |

- **Run mode**

  Dynamic voltage scaling is supported by the ultra low-power STM32L462RE to optimize its run mode power consumption. The maximum operating frequency of the system can be used to modify the voltage coming from the Main Regulator, which powers the logic (VCORE). There are two levels of power consumption:

  - Range 1, where the CPU can operate at up to 80 MHz;
  - Range 2, where the CPU can operate at a maximum of 26 MHz. The maximum frequency for all auxiliary clocks is 26 MHz. The main regulator can be turned off so that the low-power regulator can supply the VCORE. Then, the system is operating in low-power mode.

- **Sleep Mode**

  Only the CPU is turned off in Sleep mode. When an interrupt or event happens, all peripherals continue to function and potentially wake up the CPU.

- **Low-power run mode**

  To reduce the regulator's working current, when in Low-Power Run mode the VCORE is powered by the low-power regulator and the main regulator is switched off, allowing the CPU to function at up to 2 MHz, while the peripherals can be independently clocked by the HSI16. The programming can be performed from SRAM or Flash.

- **Low-power sleep mode**

  This mode is accessed by switching from the low-power run mode. Only the processor clock is turned off (uses LPR). When an event or an interrupt causes the system to wake up, it returns to low-power run mode.

- **Stop 0, Stop 1 and Stop 2 modes**

  Stop mode uses the least amount of power while still saving the contents of SRAM and registers.

All clocks in the VCORE domain are deactivated, including the PLL, MSI RC, HSI16 RC, and HSE crystal oscillators. The LSE or LSI continues to operate. There is the option for RTC to continue to function. Some peripherals with wakeup functionality can use the HSI16 RC to detect their wakeup situation while in Stop mode. Stop 0, Stop 1, and Stop 2 are the three stop modes offered. Most of the VCORE domain is put into a lower consumption state in Stop 2 mode. Stop 1 has the most active peripherals and wakeup sources, as well as a shorter wakeup time but higher energy consumption than Stop 2. The main regulator stays on in Stop 0 mode, allowing for a very quick wakeup time but significantly higher consumption. Depending on software settings, the system clock when exiting Stop 0, Stop 1, or Stop 2 modes can be MSI up to 48 MHz or HSI16.

- **Standby mode**

  Brown Out Reset (BOR) uses the Standby mode to achieve the lowest power usage. The internal regulator is turned off, which turns off the VCORE domain. The PLL, MSI RC, HSI16 RC, and HSE crystal oscillators are all turned off as well. The RTC can continue to function or not. In Standby mode, the brown-out reset (BOR) is always operational. Except for registers in the Backup domain and Standby circuitry, SRAM1 and register values are erased when entering Standby mode. SRAM2 can also be kept in Standby mode, powered by the low-power Regulator. An external reset (NRST pin), an IWDG reset, a WKUP pin event (configurable rising or falling edge), or an RTC event (alert, periodic wakeup, timestamp, tamper) or a failure on LSE causes the device to exit Standby mode (CSS on LSE). After waking up, the system clock is MSI up to 8 MHz.

- **Shutdown mode**

  The Shutdown mode allows for the most efficient power use. The internal regulator is turned off, which turns off the VCORE domain. The PLL, HSI16, MSI, LSI, and HSE oscillators are all turned off as well. The RTC can continue to function (Shutdown mode with RTC, Shutdown mode without RTC). In Shutdown mode, the BOR is unavailable. The transition to Backup domain is not supported since no power voltage monitoring is possible in this mode. Except for registers in the Backup domain, the contents of SRAM1, SRAM2, and registers are lost. When an external reset (NRST pin), a WKUP pin event (configurable rising or falling edge), or an RTC event happens, the device exits Shutdown mode (alarm, periodic wakeup, timestamp, tamper). After waking up, the system clock is MSI at 4 MHz.

The consumption of the above mentioned modes of operation is presented in Table 3.2.

Table 3.2: Power modes consumption [55].

| | Consumption | Regulator | Clocks | RAM |
|---|---|---|---|---|
| Run mode | 94 $\mu$A/MHz (Range 1)<br>85 $\mu$A/MHz (Range 2) | MR | Any | ON |
| Low-power run mode | 95 $\mu$A/MHz | LPR | Any, except PLL | ON |
| Sleep mode | 27 $\mu$A/MHz | MR | Any | ON |
| Low-power sleep mode | 38 $\mu$A/MHz | LPR | Any, except PLL | ON |
| Stop 0 | 125 $\mu$A/MHz | MR | LSE, LSI | ON |
| Stop 1 | 9.85 $\mu$A/MHz w/o RTC<br>10.5 $\mu$A/MHz w RTC | LPR | LSE, LSI | ON |
| Stop 2 | 2.05 $\mu$A/MHz w/o RTC<br>2.3 $\mu$A/MHz w RTC | LPR | LSE, LSI | ON |
| Standby | LPR ON<br>0.35 $\mu$A/MHz w/o RTC<br>0.52 $\mu$A/MHz w RTC<br>LPR OFF<br>0.1 $\mu$A/MHz w/o RTC<br>0.27 $\mu$A/MHz w RTC | LPR<br>OFF | LSE, LSI | LPR ON<br>SRAM2<br>LPR OFF<br>OFF |
| Shutdown | 0.02 $\mu$A/MHz w/o RTC<br>0.17 $\mu$A/MHz w RTC | OFF | LSE | OFF |

The MCU contains up to four I2Cs, three SPIs, three USARTs, one UART, a Low-power UART, one SAI, one SDMMC, one CAN. These interfaces allow the MCU to be connected to other components, such as sensors and transceivers. Additionally, it has a number of analogue features, including a 12-bit ADC, two ultra-low-power comparators, a number of timers, including two Low-power Timers (LPTIM), four general-purpose timers (1x 32-bit timer and 3x16-bit), and two basic timers, a RTC and a SysTick, which can be used as time bases. AES hardware encryption engine, two watchdogs, a window watchdog based on a bus clock, a watchdog with independent clock and window capability, and more are also included.

The use of the RTC is essential for the periodical tasks in the application in question, as for example sensor sampling and posterior remittance to the cloud server.

### Transceiver

The Type 1SC LTE-M module [56] is a small, low-cost LTE-M module that supports the cellular LTE-M and NB1 network protocols. It is manufactured in a silver resin-shielded box and measures only 11.1 x 11.4 x 1.5 mm (max). It contains an Altair ALT1250 transceiver, a FEM module, and external flash memory.

Type 1SC supports multi-band coverage to cover the majority of the world and low current consumption (through the use of eDRX and PSM). The 1SC module has the power modes shown in Table 3.3

Table 3.3: Type1SC power modes [56].

| Power Mode | Description |
|---|---|
| LS | Is mostly utilized for very brief periods of sleep and offers very quick entry and recovery times. It is utilized for CDRX mode. |
| DS | Offers quick entrance and recovery times and is mostly used in IDRX networking mode. |
| DH2 | This networking protocol, which is mostly utilized in the EDRX and IDRX networking modes, offers medium entrance and recovery times. |
| DH1 | The same as DH2, but without the IO logic. In this mode of operation, the average current is 48 muA |
| DH05 | Offers lengthy entrance and recovery times; primarily utilized for PSM and other extremely extended inactive intervals. In this mode, the IO output values are kept. |
| DH0 | Similar to DH05, but no IO output values are saved. In this mode of operation, the average current draw is 1.7 muA |

**PSM Current Draw**

The current draw will fluctuate because the PSM mode of operation is not currently optimum for this firmware release. The average current in the worst-case scenario is 33.76 mA.

**eDRX Current Draw**

This feature is significantly impacted by the SIM card. Different modes of operation are supported by network carriers. The current draw for this configuration might be as low as 86uA instead of 377uA under ideal circumstances when utilizing a SIM that can be deactivated.

Since this dissertation intends to use the NOS network, only PSM is available. Therefore, the mode of operation that is used is DH0 with PSM, which represents 1.7 $\mu$A when idle and 33.76 mA when active.

### Sensors

In order to monitor the environment, the device is equipped with several sensors, which are summarized in the Table 3.4. These sensors are responsible for measuring temperature, humidity, light and acceleration. Some of the chosen sensors have low power features.

Table 3.4: Sensors overview.

| Sensor | Power Supply | Consumption | Size | Interface | Low Power Modes |
|---|---|---|---|---|---|
| BMA400 | 1.72 V – 3.6 V | 3.5 $\mu$A - OSR-0 Normal mode<br>850 nA - OSR-0 Low-power mode<br>160 nA - Sleep mode | 2 x 2 (mm)2 | SPI<br>I2C<br>2 INT | Sleep<br>Low-power<br>Normal |
| APDS-9306-065 | 1.7 V - 3.6 V | 85 $\mu$ Normal mode | 2 x 2 (mm)2 | I2C<br>1 INT | |
| HDC2080 | 1.62 V – 3.6 V | 650 $\mu$A/550 uA - hum/temp measurement<br>0.55 $\mu$A - 1 measurement/second of hum/temp<br>160 nA - sleep mode | 3 x 3 (mm)2 | I2C<br>1 INT | Sleep<br>Triggered<br>Auto |

#### Light Sensor

The APDS-9306-065 [57] is a low voltage digital ambient light sensor that outputs digital signals with an I2C interface based on light intensity. It includes a photodiode, an ADC, an oscillator, and a power-on reset. The output count of the APDS-9306-065 is proportional to ambient light level, simulating the direct readout response of the human eye. In order to react to events, the APDS-9306-065 enables programmable hardware interrupt with hysteresis. The sensor can be used in space-sensitive applications thanks to its ultra-slim size, with a height of 0.65 mm.

#### Temperature and Humidity Sensor

The HDC2080 [58] device is a combined humidity and temperature sensor that offers extremely accurate readings in a compact DFN chip while consuming very little power. New integrated digital functions and a heating element to remove condensation and moisture are both included in the capacitive sensor. With the help of the customizable interrupt thresholds, alarms and system wake-ups can be provided without the need for a microcontroller to be constantly checking the system. The HDC2080 is made for battery-powered systems and features adjustable sampling periods, low power consumption, and support for a 1.8 V supply voltage.

### Accelerometer Sensor

The triaxial BMA400 [59] from Bosch Sensortec is a true ultra-low power acceleration sensor. The new acceleration sensor eliminates erroneous warnings by differentiating between crucial circumstances and false signals.

A clever on-chip motion and position-triggered interrupt feature makes this 12-bit digital triaxial acceleration sensor unique. Due to its auto low power mode and auto wake-up features, the BMA400 is particularly well suited for ultra-low power devices that demand a long battery lifespan. As a result, the BMA400 is particularly well suited for the case in study. Depending on the mode, the sensor's power consumption is different.

### EEPROM

Since the application uses an EEPROM to store information and the STM32L462RE does not have one, BR24G128FJ-3A was the chosen EEPROM to integrate the final board. It is a serial EEPROM with I2C BUS as its interface method. It has 128 Kbits with a bit format of 16K x 8 [60]. Some of its characteristics are presented in the Table 3.5.

Table 3.5: BR24G128FJ-3A EEPROM overview.

| EEPROM | Power Supply | Consumption | Size | Interface |
|---|---|---|---|---|
| BR24G128FJ-3A | 1.72 V – 5.5 V | Standby current - 2 $\mu$A<br>Supply current (Read) - 2 mA<br>Supply current (Write) - 2.5 mA | 4.9 x 6 (mm)2 | I2C |

### Power Supply

All the components are powered by the board power supply. There are two voltage ranges on this board: 1.8 V and 3.0 V. The Low-Dropout Regulator (LDO) TPS7A02 [61] is used to supply the 1.8 V that is required by the MCU and sensors. It is a very small, very low quiescent current (25 nA) LDO with good transient performance that can source 200 mA. The Saft LM 17500 battery [62] powers the system. This battery has a 3 V nominal voltage, a 3 Ah capacity, and a 1% leakage rate.

### 3.2.3 Software

#### Software Tools

For this dissertation, C was chosen as the programming language, since it was the language already used in previous versions of the application. Below are listed the tools/COTS used for the development of the software part of the project:

- **Keil $\mu$Vision5** - The Keil MDK IDE, a comprehensive software development environment for a variety of Arm Cortex-M based microcontroller chips, was used as the development environment for the end device [63].

- **ST-LINK** - Debugger on chip tool, that allows to follow the state of the board while debugging the code [64].

- **STM32CubeMX** - The STM32CubeMX was additionally utilized for project generation. It is a graphical utility that makes configuring STM32 microcontrollers simple. Generates code for the initialization and some configurations of the board's peripherals [65].

- **Azure RTOS ThreadX** - The Azure RTOS ThreadX is an advanced real-time operating system (RTOS) that is open-source. [66].

- **MATLAB** - MATLAB is used as a programming and numeric computing platform to analyze data, develop algorithms, and build models. In this project, it was used during the test phase to register results. [67].

- **Git** - Git is a distributed version control system that is free and open source and made to manage both small and large projects fast and effectively.

Figure 3.7: Software Tools.

### Previous Work and Alterations

The work in scope of this dissertation will have for its foundation the work developed in [49].The Link4S application will be ported and modified so that it can work with the new Type1SE module. This section is dedicated to briefly describing the work presented in [49] so that one can have a better understanding of the application organization and of the work developed in this dissertation in the next chapters.

### Software Stack

It is essential to cover the software stack, presented in Figure 3.8, if one wants to understand the structure of an application.

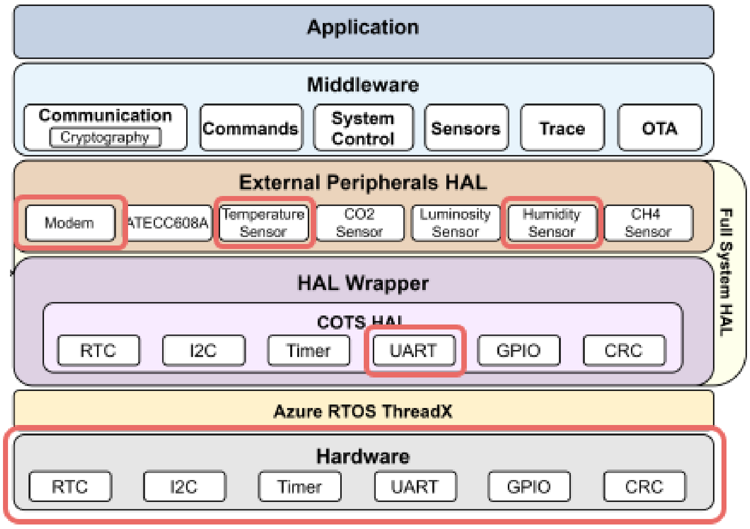


Figure 3.8: Software Stack (Adapted from [49]).

The stack comprehends six layers, where four of these have the threads presented in Table 3.6:

- **Hardware** - The hardware layer that will be changed according to the content of the section 3.2.
- **Azure RTOS ThreadX** - This layer includes all the ThreadX files that make Azure RTOS ThreadX work properly in the application.
- **HAL Wrapper** - This layer is responsible for abstracting, even more, the already abstracted hardware in Hardware Abstraction Layer (HAL), which is developed by ST.
- **External Peripherals HAL** - The external peripherals HAL includes all the drivers for the added peripherals, such as sensors, modem, etc.

- **Middleware** - The lower levels required by the Application are controlled by the modules in the middleware layer.

- **Application** - The application layer is responsible for coordinating the different parts of the application. At first, it initializes the application. Afterwards, it enters a loop where it waits for flags that will result in the execution of certain functions and where it does error correction.

Table 3.6: Link4S Application Threads.

| | Threads |
|---|---|
| Azure RTOS ThreadX | Timer |
| External Peripherals HAL | Message Receive |
| Middleware | Trace, Sensors Manager, Sample,<br>Emergency Manager, Commands Manager,<br>System Control Manager, Async Events,<br>Connection Manager |
| Application | Application Manager |

The layers of the system stack and most part of the application will remain the same. The objective is to port it to an entirely new Hardware. This will require that the parts highlighted in red in Figure 3.8 are reformulated.

# Chapter 4: Design and Implementation

This chapter reflects the hardware and software development stage, which was based on the system requirements with the goal of converting these into reality.

## 4.1 Hardware

In this section, the hardware development for a custom PCB is described. The purpose of this board is to have a device that is specifically designed for the application to lower power consumption. Altium Designer [68] was the chosen tool for the development of the customized board. It is one of the industry's leading PCB design software.

### 4.1.1 Schematic

The first step entails designing the circuit that will be integrated into the PCB while taking into account the previously established architectural layout (Figure 4.1). One needs to select the project-specific components while also checking the market for their availability. The majority of external devices used already have an Altium footprint available in the Altium Library Loader [69]. A footprint in PCB design is a layout for an electronic component that will eventually be soldered there. Every component that is soldered onto a printed circuit board requires a footprint, whether it is a through-hole connector, a surface mount capacitor, or any other component. The devices that lack a schematic symbol and a footprint need to be created. To accomplish this, it is necessary to verify the dimensions and pinout of the component and use Altium's functionalities in order to create the device's footprint.

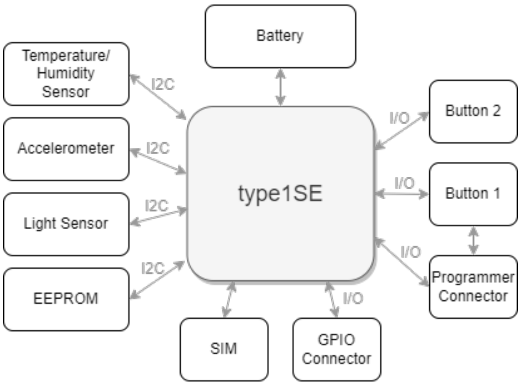


Figure 4.1: Architectural layout.

### LBAD0ZZ1SE

One of the main challenges of designing a custom board that would accommodate the Murata type1SE was the little information available, since it is a fairly recent chip. A few documents were consulted as a reference for the schematic design. These include ”Type1SC Hardware Design Guidelines” [56] and the Type1SE datasheet [70]. This datasheet is still very brief. The only information or guidance available was the short information about the GPIO in its datasheet and the hardware design guidelines for type1SC (which is incorporated in type1SE). However, there were no hardware design guidelines for Type1SE.

The document that gave more insight on how to design hardware for the module was the schematic of B-L462E-CELL1 Discovery from ST [71]. By consulting this document along with the two documents mentioned above, it was possible to reverse-engineer what the module needed regarding hardware protections versus which protections the module already had inside and what connections were needed.

In Figure 4.2, one can see the Type1SE pinout. It has 126 pins. The microcontroller and some other components were found in the Altium Library Loader.

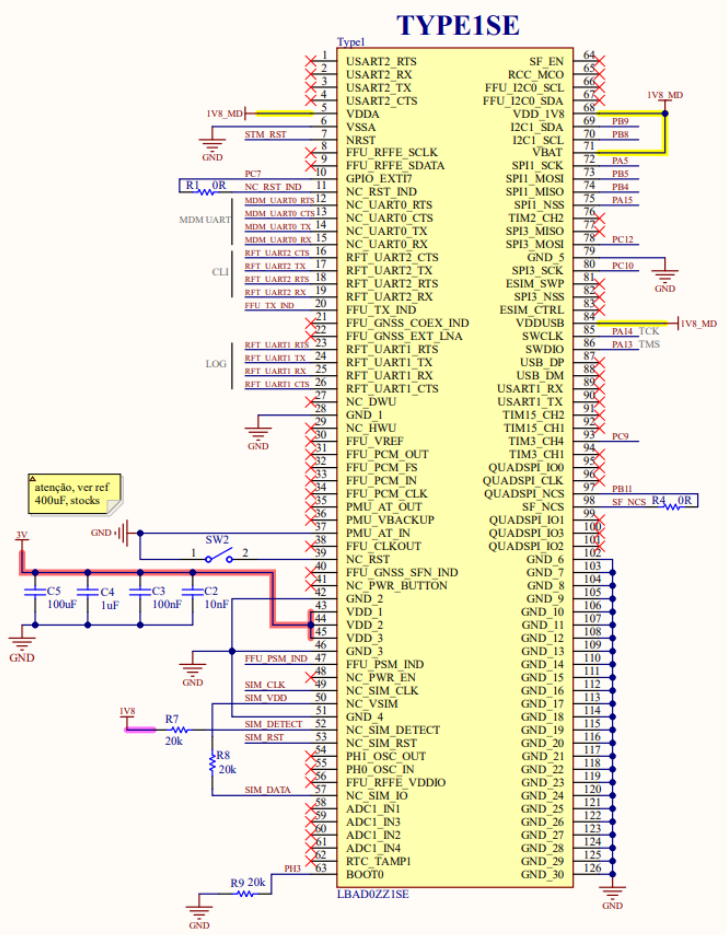


Figure 4.2: Type1SE.

A capacitor is positioned next to each source pin in every chip, so the microcontroller chip is not an exception; for example, Figure 4.3 shows four decoupling capacitors A decoupling capacitor is a passive part with the ability to store energy locally. It takes some time to charge and discharge due to the nature of the device. By supplying adequate DC supply, it prevents abrupt voltage changes, hence safeguarding the system or IC. The dimensions of decoupling capacitors come in standardized values.

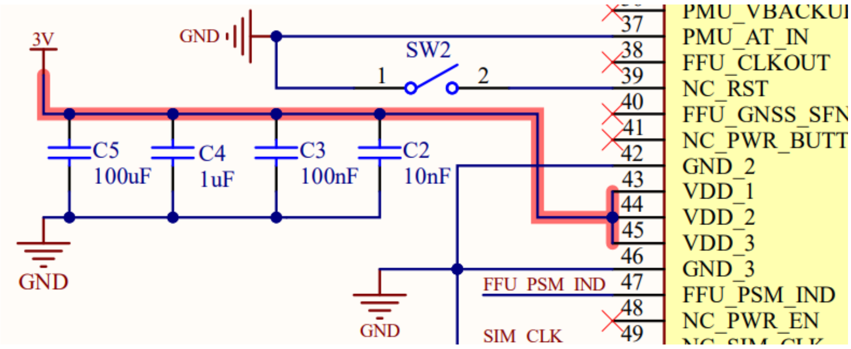


Figure 4.3: Capacitors.

In order to program and debug the device, a JTAG interface is needed. In Figure 4.4, one can see the interface for this purpose. It consists of a simple five pin connector: one for power, one for ground, two for TMS (Test Mode Select) and TCK (Test Clock), pins PA14 and PA13 respectively, and a last one for the STM reset that is also connected to a switch button.

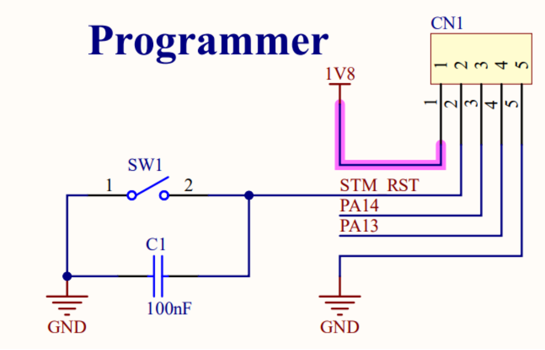


Figure 4.4: Programmer Interface.

### Sensors and EEPROM

All four sensors were chosen for their digital interface and low power consumption. The board design was simplified by using I2C as the sensor's communication protocol, requiring only the sharing of the two communication lines (SDA and SCL). The sensors and the EEPROM are powered by 1.8V and each one has an interrupt line. Any I/O pin will typically have pull-up resistors associated when they are open-drain. All open drain I/Os require the use of an external pull-up or pull-down resistor to keep the digital output in a defined logic state. The SDA and SCL lines are no exception so they have their corresponding pull-up resistors.

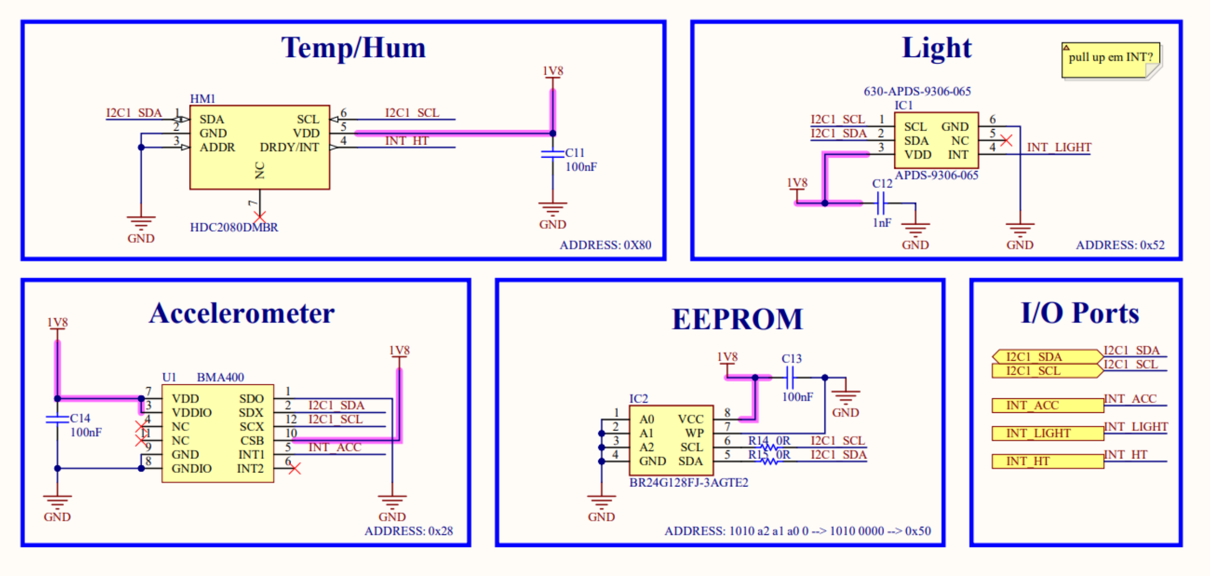


Figure 4.5: Sensors ans EEPROM Sheet.

### SIM Card

The application requires a socket for an external SIM card. A sim card socket is always exposed to electrostatic discharge (ESD), since it will be in contact with the user. This way, it is important that the device has the ability to resist these effects, avoiding damage. ESD can be carried as a transient voltage or radiated due to its fast rise time that can cause the application to behave erratically by interacting with other signals. Printed circuit boards must be shielded against electrostatic discharge. The protections for this purpose can be seen in Figure 4.6, highlighted in green, where the NSQA6V8AW5T2G is a Transient Voltage Surge Suppressor (TVSS) with a low leakage current of less than 1 $\mu$A and a RC circuit that is also meant to protect against ESD discharges.

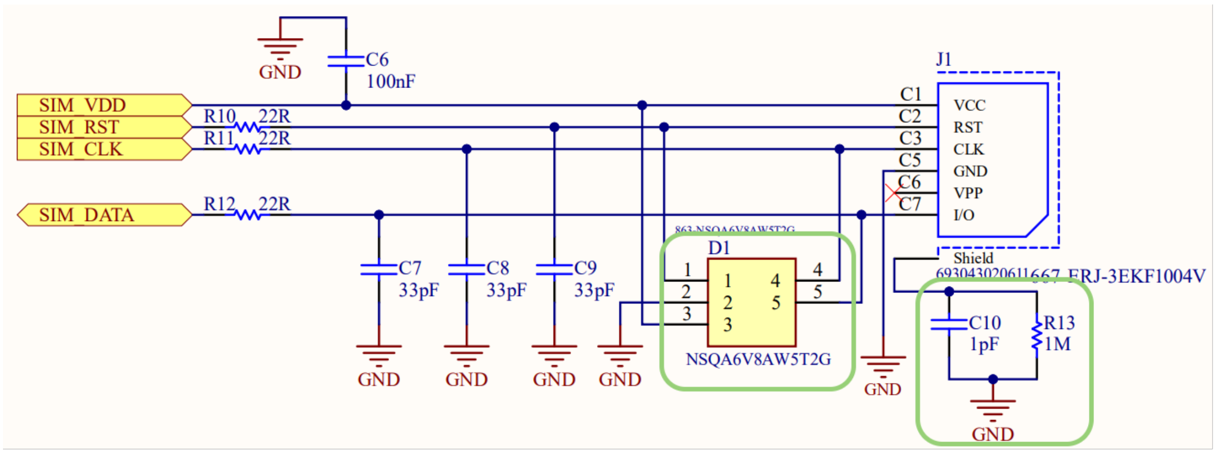


Figure 4.6: Sim card circuit.

### Power

In the power section of the schematics one of the main points is the TPS7A02. It is an ultra-small, ultra-low quiescent current low-dropout linear regulator (LDO). It will guarantee the 1.8 V part of the circuit has the intended source voltage.

The capacitors present in this circuit are meant to absorb noise in the signal. The jumper, JP2, is a testpoint for consumption measurements and to ensure the LDO stability.

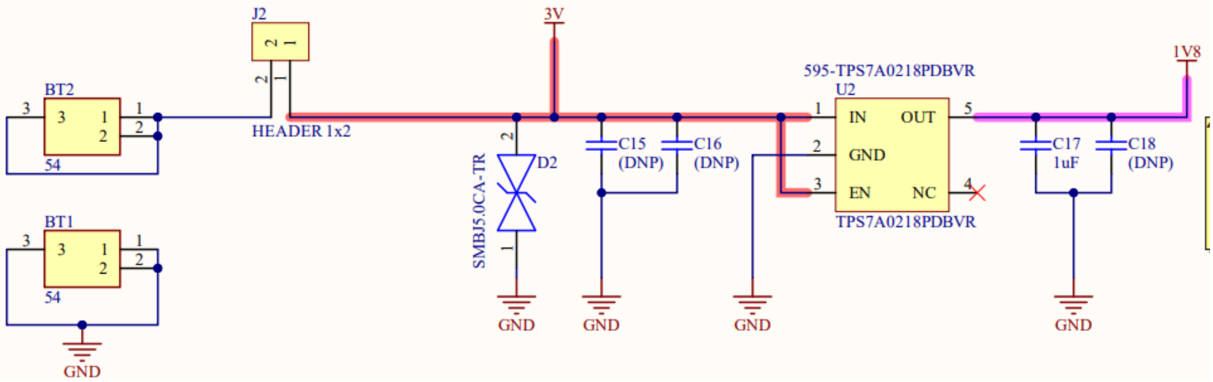


Figure 4.7: Power Sheet.

## 4.1.2 Layout

A good layout holds the most accountability of a good PCB design. Therefore, the layout stage is one of the most crucial stages of designing a PCB. The layout entails placing the devices in a defined area with the desired PCB dimensions. The devices are positioned inside the PCB's boundaries according to the designer's preferences, making it possible for different layouts to use the same components. Certain connections between the devices require to be placed close to one another, however even in this case, several layouts can be created.

### Signal Integrity

The integrity of a signal on a PCB can be compromised by a variety of interferences, and the higher the rise time of the signal, the more susceptible it is to these issues. These interferences include ground bounce, impedance mismatches, electromagnetic coupling or crosstalk, for example. These disturbances will reduce the signal's quality which may result in transmission errors if they are not managed. Inconsistent signals can manifest as difficult-to-diagnose occasional issues or complete system failure.

A high-speed signal may overwhelm a lower-speed signal if they are routed too closely together. Crosstalk between signals may lead to interference inside the system since the weaker signal may be influenced by the stronger one rather than conveying its intended waveform. This is not unique to traces that are routed next to one another; it can also occur between different layers of the same board.

The board was designed to have two copper layers. In this case, a circuit layout is imprinted on both sides of a two-layer PCB, making handling crossed signals simpler. This is the first option considered for manufacturing using bigger plated through holes and vias to install components. A higher number of layers was not needed for this application.

The back layer of the PCB is primarily a ground surface in this instance, with only a few short traces or localized power planes. The goal is to maintain a ground plane underneath every signal.

### Component Placement

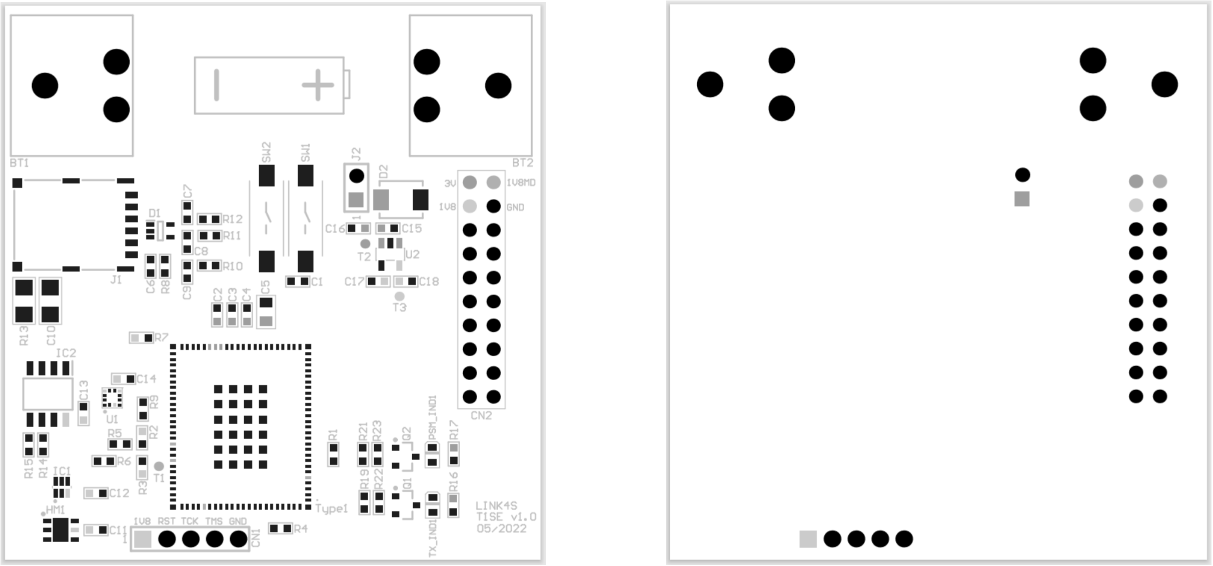
For the placement of the components the following aspects were taken in consideration:

- **Components should be arranged according to functionality.** For instance, as long as it was possible, communication related components were confined to their own space in the board and power management components were also placed in their own block. As a general rule, the loudest signals should not be close to the most delicate ones. Additionally, one has better control over the return path of components these are arranged according to their function.

- **There should be space for routing.** Smaller PCBs are necessary for many electronics applications, nevertheless many times there is an ideal size one must adhere to, or else routing every trace will prove to be difficult. When positioning components, especially those close to when with a lot of pins (as is the case of the Type1SE), it was important to make sure there was enough space for copper traces to pass through.

- Board-to-wire **connectors should be kept close to the board's edge**. Not only does it appear ordered, but positioning connectors close to the edge also avoids accidental contact with other PCB components and makes their use easier.

In Figure 4.8, it is possible to see a 2D representation of the board, more specifically of the board components placing and corresponding pads and holes.



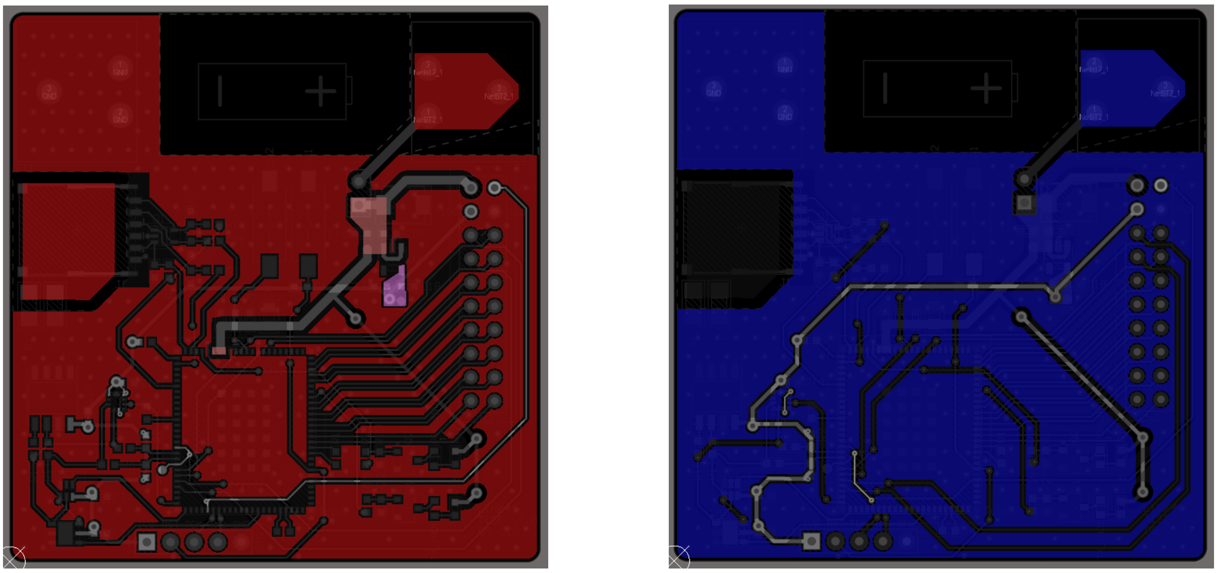
(a) Upper Layer.

(b) Bottom Layer.

Figure 4.8: Components and their holes and pads.

### Polygon Pours

Polygons give a straightforward technique to control power and ground distribution over a complex PCB. Polygons are sections of the PCB that are filled or flooded with copper. They are also known as copper pours or polygon pours [72]. This zone is poured around existing components and traces, resulting in a copper-filled region with predefined clearance. Polygons should not be left floating and should be connected to a ground net to ensure isolation. The polygons in the customized board are depicted in Figure 4.9.



(a) Top Layer.

(b) Bottom Layer.

Figure 4.9: Polygons.

In the two figures above, one can see both layers of the PCB filled with copper. This polygons are both connected to ground. Because of the presence of other tracks, pads, and so on, there may be sections of a polygon that are completely isolated from the connected net when it is poured. These areas are called dead copper and should be removed.

In Figure 4.10, it was needed to cut part of the copper plane in order to protect the circuit from ESD. By cutting the copper around the SIM socket it will further ensure the protection of the circuit from the possible discharges that will happen when the user touches the socket.

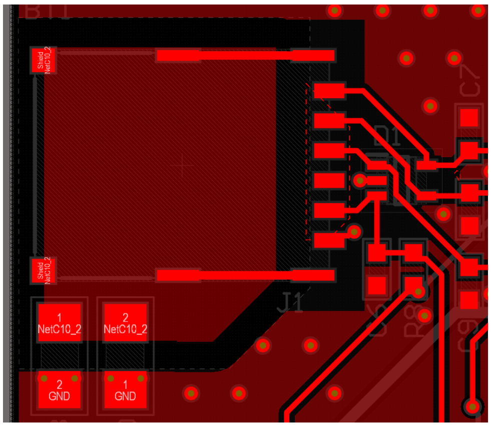


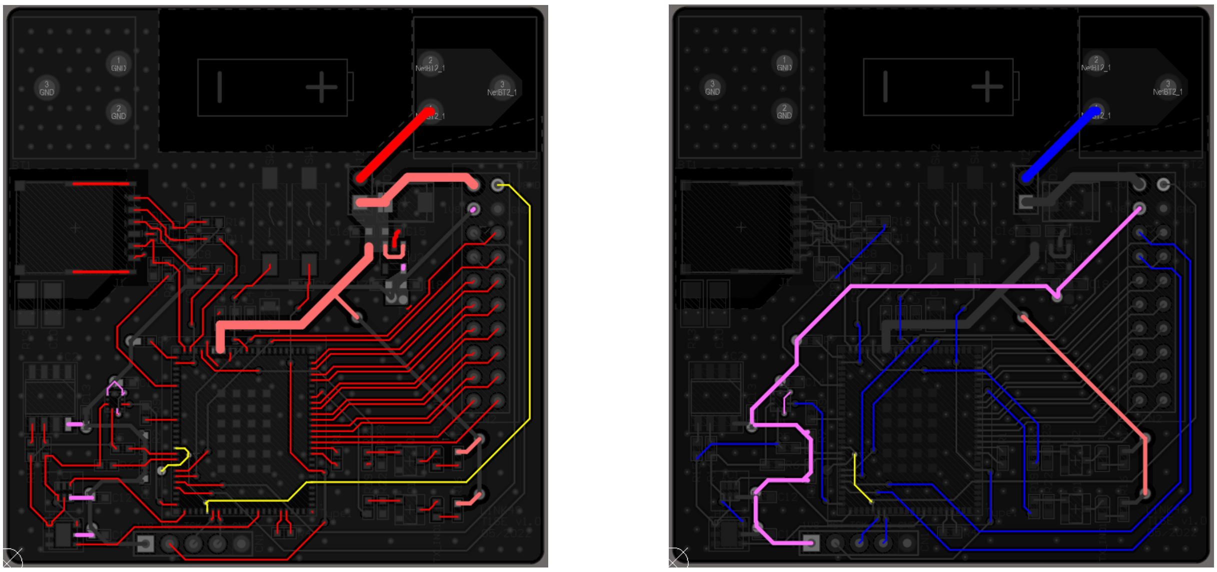
Figure 4.10: Copper Cutout Around Sim Card Socket

### Clearance Rules

Several clearance rules were created for this design depending on electrical characteristics of the signals. The general rule applied to the layout was a clearance of 8 mil. For the '3V' net and the 'NetBT2_1' net (linked to the battery) the clearance was set to 20 mil. For 'NetC10_2' net (around the sim card socket) the clearance was set to 50 mil.

### Routing

After all components are placed it is time to connect the pins. As shown in Figure 4.11, signals on the top layer are represented in red and on the bottom they are represented in blue. If a connection needs to be made between the top layer and the bottom layer, this connection between is enabled by a metal hole (via) that allows the conduction of electricity between the upper and lower tracks. Oftentimes, the connection between layers is necessary to prevent the overlap of the traces.



(a) Top Layer.

(b) Bottom Layer.

Figure 4.11: Traces.

A good practice is to make traces horizontal in the upper layer and mostly vertical in the bottom layer. This is to avoid a phenomenon called "**Crosstalk**". Crosstalk happens when there is an undesirable electromagnetic coupling between traces on a PCB. This coupling results in the overriding of one signal in a trace by another, even though the two traces are not physically connected. This can happen within the same layer or even between different layers. The bigger the overlap area, greater the interference is. This phenomenon is known as broadside coupling, and it occurs because the two signal layers are separated by only a very thin layer of core material. To prevent this, the orientation of the traces in upper and bottom layers should be different. If two traces run parallel to each other across two layers, the overlap is greater than that of two perpendicular traces. In Figure 4.12, it is illustrated how crosstalk happens (on the left) and how it can be avoided (on the right).

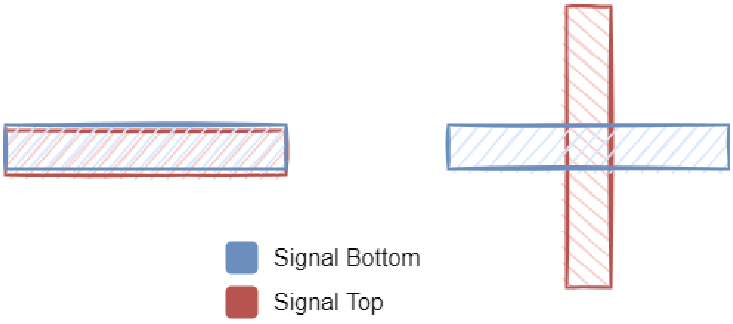


Figure 4.12: Cross Talking.

To avoid cross talking from occurring within the same layer, one should pay attention to the space between traces, as show in Figure 4.13.

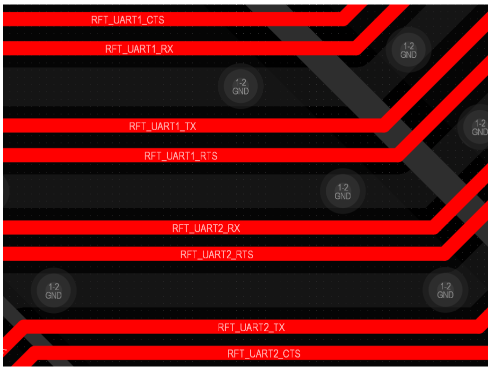


Figure 4.13: Trace Spacing.

### Proper Return Path and Via Stitching

A return path of current is the path current follows when it returns to its origin. When a signal is sent from its origin and arrives at its destination, it will desire to return to its source. This is because of the conservation energy principle. The return path is always the shortest trip back through the path of least resistance. This is a fundamental rule of circuit design.

Sometimes the return path can be affected as illustrated in the left side of Figure 4.14. What happens in this cases is that, even with a ground plane on both layers of the PCB, the path of return will be affected if there is no copper connecting the top and bottom ground plane, resulting in no proper path for the signal to return.

The solution in this cases is via stitching. Via stitching is a method for connecting bigger copper areas across multiple layers, effectively forming a solid vertical connection through the board structure and preserving low impedance and short return loops [73]. Areas of copper that may otherwise be cut off from their net can also be connected to that net via stitching.

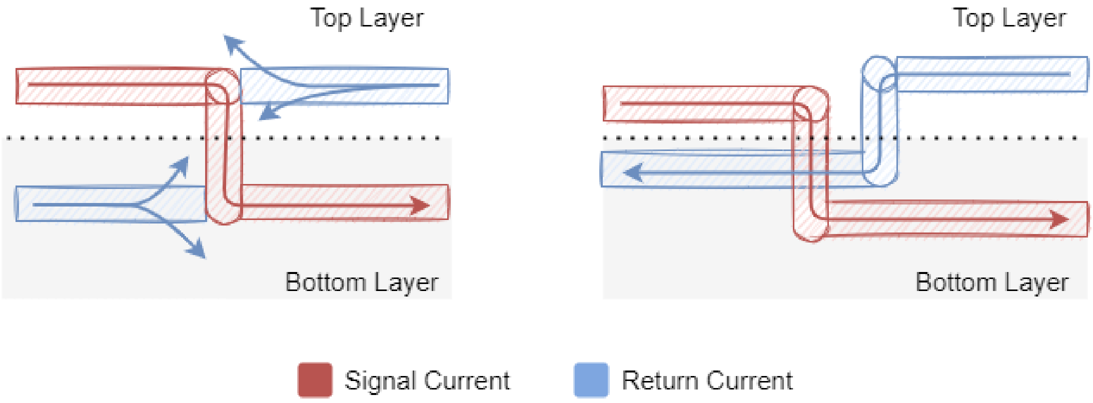


Figure 4.14: Proper Return Path.

Other problem that via stitching solves is ground plane discontinuity which can be observed in Figure 4.15. Because allowing return current to flow is the ground plane's primary function, there should be as few interruptions as possible.

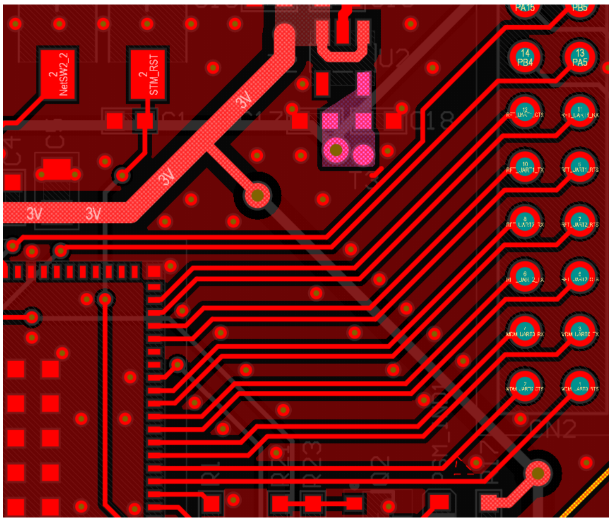
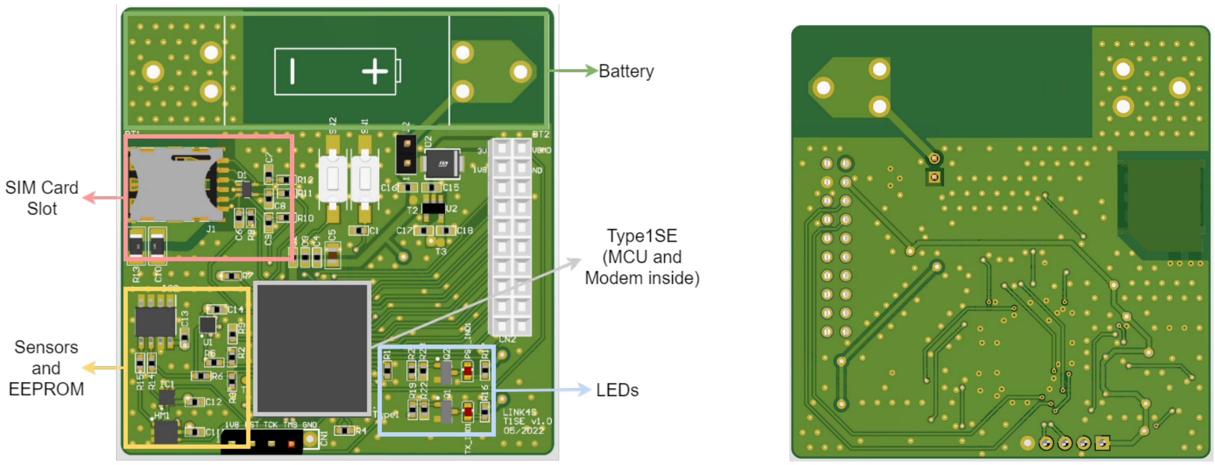


Figure 4.15: Via Stitching and Ground Continuity.

## 4.1.3 Implementation

A printed circuit board (PCB) was designed to produce an end device that is low-power, inexpensive, and has NB-IoT capabilities. In Figure 4.16, it is possible to see a 3D animated representation of what the final product looks like.



(a) Top Layer.

(b) Bottom Layer.

Figure 4.16: 3D PCB Representation.

The dimensions of the board are 57 mm × 59 mm with all modules integrated, not needing external connectors to function. Figure 4.17 shows the printed circuit board without any soldered components.

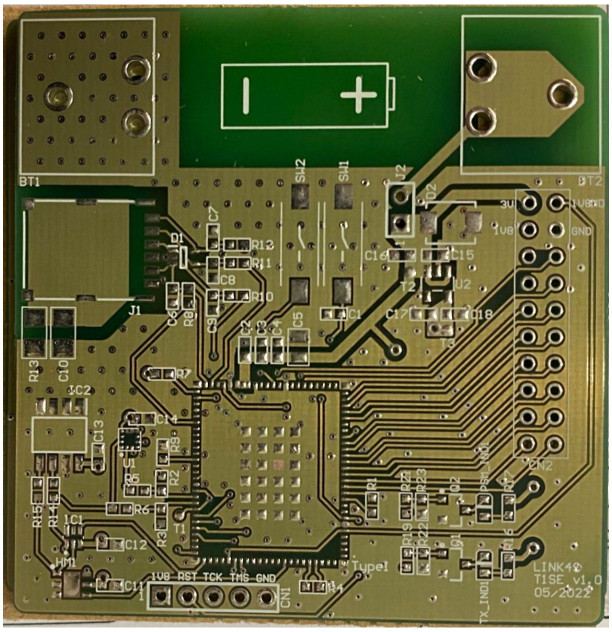


Figure 4.17: Printed Circuit Board.

Due to chip shortages and delivery delays of the Type1SE module it was not possible to implement and test the PCB on time, even though every other component arrived and was soldered. It was even attempted to remove the module from the discovery kit so that it could be soldered onto the PCB, however this attempt was unsuccessful.

### BoM

Bill of Materials (BoM) is a list of the materials used in the schematics. BoM is generated automatically by the software, with the manufacturer and a reference for each product added. This step is crucial for determining whether the chosen electronic devices are offered by common electronic service providers because, even if a device has an online user manual, it doesn't necessarily mean it is available on the market. By choosing the suppliers, it is possible to calculate the cost of the PCB with the components and create an order at the same time. The BOM, or list of the number of devices, suppliers, and references, may be found in Appendix B.

## 4.2 Software

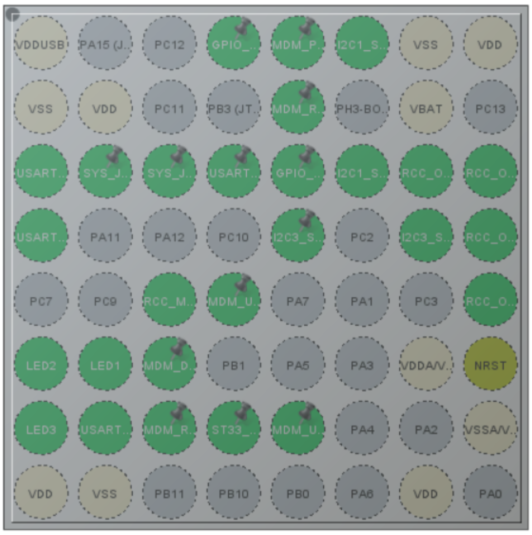
As already mentioned in the previous chapters, the software developed in the scope of this dissertation had for its basis the work developed in [49]. The main goal of this dissertation is to study the already developed Link4S application and port it to work with the new MCU and modem that constitute the Type1SE module.

Since the code has an abstraction layer, ”HAL Wrapper”, the portability between STM platforms is easier. This layer is responsible for abstracting even more the already abstracted hardware in Hardware Abstraction Layer (HAL), which is developed by ST. Despite the many benefits of this abstraction layer, there are still undeniable differences between the MCUs and the modems that need to be considered and new code will need to be developed for the application to function properly with the new hardware. Furthermore, the module used in this work is recent, which made the refactoring process more challenging.

This section is dedicated to the software developed to reach the goal of porting the application to the new hardware. The alterations mentioned in the section 3.2.3 will be further explained in this section.

### 4.2.1 Configurations of the MCU

A STM32CubeMX project serves as the foundation for the software developed. All project configurations are set through the STM32CubeMx tool. The board’s pin configurations are shown in Figure 4.18. The specific pins (such as power supply or BOOT) that are highlighted in yellow or light green are not accessible as peripheral signals. The pins in green are the ones that are set up.



| | | | | | | | |
|---|---|---|---|---|---|---|---|
| VDDUSB | PA15 (J.. | PC12 | GPIO_.. | MDM_P.. | I2C1_S.. | VSS | VDD |
| VSS | VDD | PC11 | PB3 (JT.. | MDM_R.. | PH3-BO.. | VBAT | PC13 |
| USART.. | SYS_J.. | SYS_J.. | USART.. | GPIO_.. | I2C1_S.. | RCC_O.. | RCC_O.. |
| USART.. | PA11 | PA12 | PC10 | I2C3_S.. | PC2 | I2C3_S.. | RCC_O.. |
| PC7 | PC9 | RCC_M.. | MDM_U.. | PA7 | PA1 | PC3 | RCC_O.. |
| LED2 | LED1 | MDM_D.. | PB1 | PA5 | PA3 | VDDA/V.. | NRST |
| LED3 | USART.. | MDM_R.. | ST33_.. | MDM_U.. | PA4 | PA2 | VSSA/V.. |
| VDD | VSS | PB11 | PB10 | PB0 | PA6 | VDD | PA0 |

Figure 4.18: STM32CubeMX MCU pins configuration.

The STM32CubeMX tool's clock configuration menu was used to modify the operating clock frequency as needed. The options available for the MCU selected are shown in Figure 4.19. The several clock sources for the system clock and the two multiplexers (mux) that let users choose the source clock for the RTC and the system clock are highlighted in green. MSI was chosen for System Clock. For the RTC the LSE clock was selected.

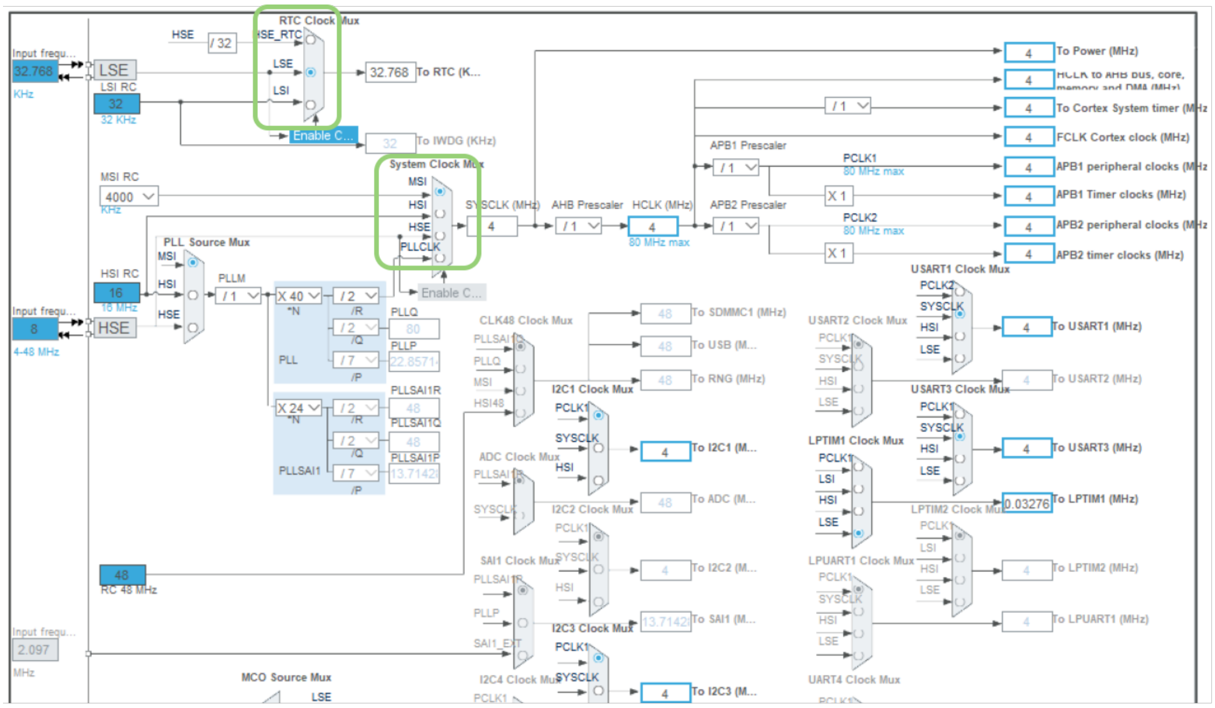
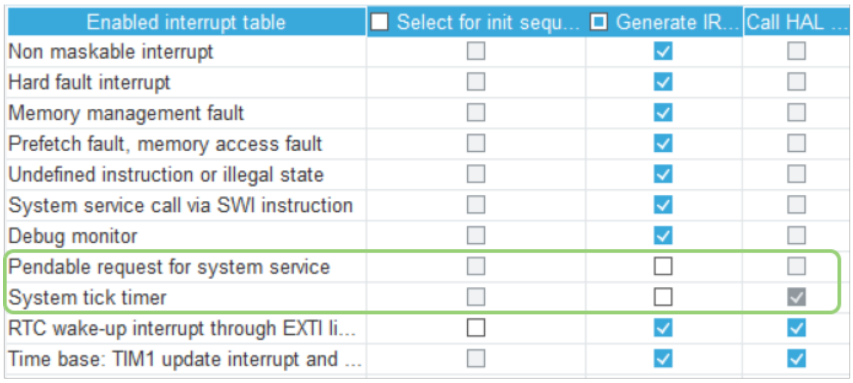


Figure 4.19: STM32CubeMX MCU clock configuration.

The C code project is generated once all settings have been established. Keil µVision was the chosen IDE because of prior work history with this tool and its extensive debugging capabilities for STMicroelectronics MCUs.

In Figure 4.20, one can see some of the system interrupts. The Pendable request for system service and the System tick timer interrupts were disabled for the integration of the ThreadX OS, which is further explained in section 4.2.2.



| Enabled interrupt table | ☐ Select for init sequ... | ☐ Generate IR... | Call HAL ... |
|---|---|---|---|
| Non maskable interrupt | ☐ | ☑ | ☐ |
| Hard fault interrupt | ☐ | ☑ | ☐ |
| Memory management fault | ☐ | ☑ | ☐ |
| Prefetch fault, memory access fault | ☐ | ☑ | ☐ |
| Undefined instruction or illegal state | ☐ | ☑ | ☐ |
| System service call via SWI instruction | ☐ | ☑ | ☐ |
| Debug monitor | ☐ | ☑ | ☐ |
| Pendable request for system service | ☐ | ☐ | ☐ |
| System tick timer | ☐ | ☐ | ☑ |
| RTC wake-up interrupt through EXTI li... | ☐ | ☑ | ☑ |
| Time base: TIM1 update interrupt and ... | ☐ | ☑ | ☑ |

Figure 4.20: STM32CubeMX MCU pins configuration.

### 4.2.2 Azure RTOS ThreadX

This section focuses on Azure RTOS ThreadX, the OS that was selected for this system. An overview of how it functions will be provided in the first part. The subsequent parts will then go over how ThreadX was validated and ported to the STM32L462RE.

#### Overview

ThreadX enters its thread scheduling loop after initialization is finished. All program execution that occurs between processor reset and the start of the thread scheduling loop is considered initialization. The scheduling loop searches for an application thread that is prepared to run. ThreadX gives control to a ready thread after it has been located. When a thread is ended (or another thread with a higher priority becomes available), execution switches back to the thread scheduling loop to identify the next available thread with the highest priority. The most prevalent method of program execution in ThreadX applications is the technique of continuously scheduling and running threads, as depicted in Figure 4.21.

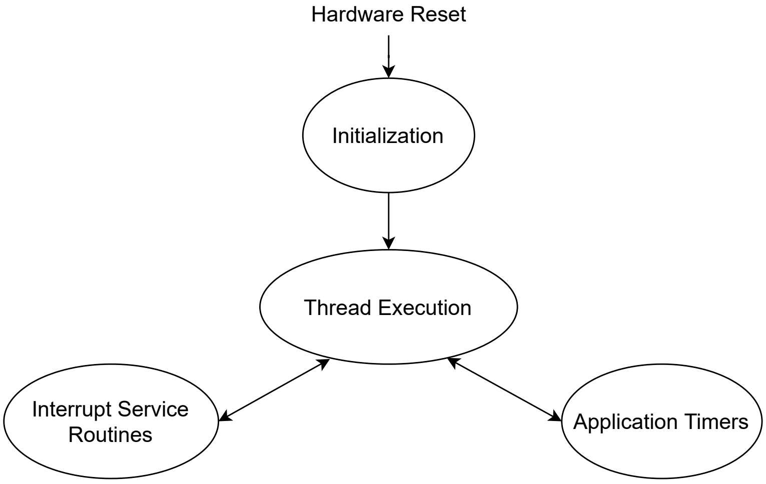


Figure 4.21: ThreadX Execution (Adapted from [74]).

### Initialization

It is essential to comprehend how an RTOS initializes, especially if it needs to be ported to a different architecture. As shown in Figure 4.22, the kernel's initialization procedure is broken down into six parts:

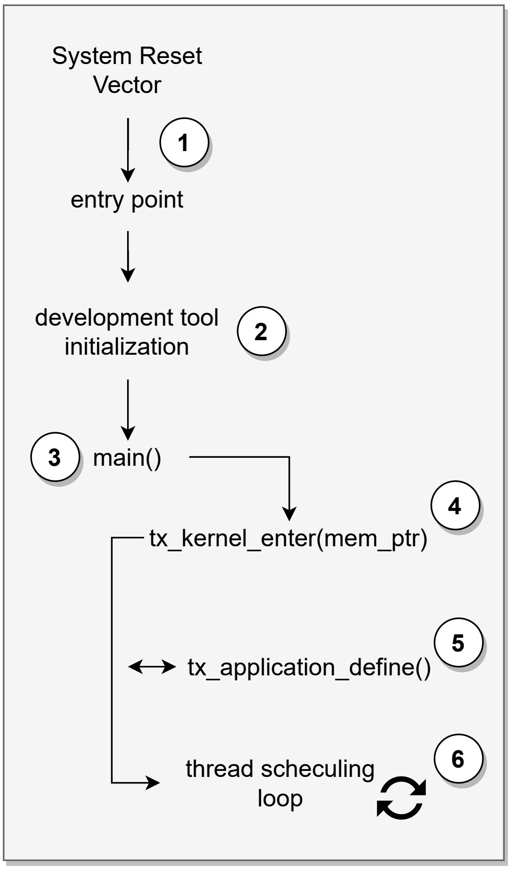


Figure 4.22: ThreadX Initialization (Adapted from [74]).

1. Reset logic is a function that is common to all microprocessors. The address of the application's entry point is acquired from a specific memory region when a reset takes place. The CPU transmits control to the entry point after retrieving it. The application entry point is typically provided by the development tools and is frequently written in native assembly code.

2. The development tool's high-level initialization takes over once the low-level initialization is finished. Usually, initialized global and static C variables are set up in this stage.

3. Control is passed to the main function after the development tool initialization is finished. The application is in charge of the following events at this stage.

4. The entry into ThreadX, tx_kernel_enter, is usually only called by the main function in most programs. Applications can, however, carry out preparatory processing before entering ThreadX (often

for hardware startup). Prior to invoking the tx_application_define function for the application, the entry function coordinates initialization of different internal ThreadX data structures.

5. All of the initial application threads, queues, semaphores, mutexes, event flags, memory pools, and timers are defined by the tx_application_define function. All resources for the initial application, nevertheless, are defined here. There is only one input argument for the tx_application_define function, which is the first RAM address that is available.

6. Control is passed to the thread scheduling loop after tx_application_define has finished returning. The initialization process is now complete.

### ThreadX Porting

Firstly, one needs to add the ThreadX source code [75] to the project. This source code is organized in several folders. There is a main folder called "threadx" and, inside this one, there are three subfolders: common, ports and samples. The code necessary to import to the project is the common folder and its contents and the port folder specific for the given case is cortexm4 > ac5 (AC5 compiler), as depicted in Figure 4.23.

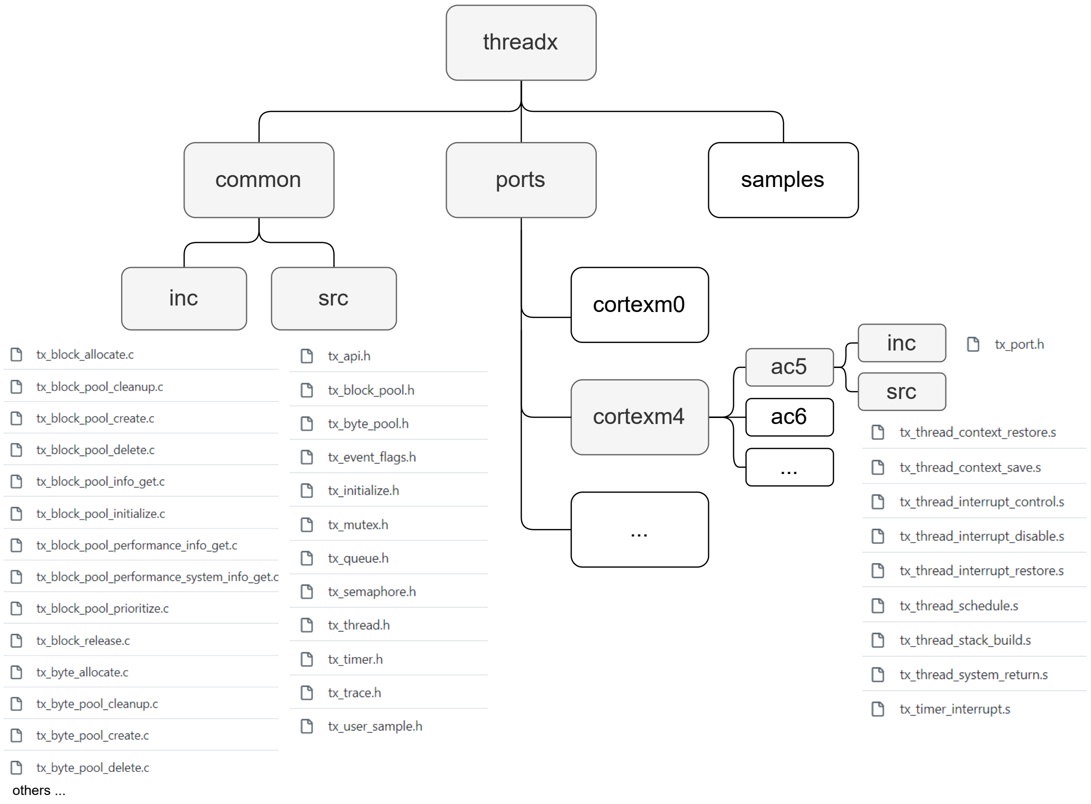


Figure 4.23: Threadx folder organization.

Unlike other similar RTOS, ThreadX is designed in a way that makes supporting multiple architectures considerably simpler. The API in ThreadX is totally unrelated to the hardware below. Therefore, just the kernel files and the tx_port header inside the ports folder need to be changed in order to migrate to a different architecture.

To port TreadX to the application one had to:

1. Remove the original stm32l4xx_it.c PendSV and Systick Interrupt service function, by commenting or by setting the configurations in MxCube, as seen in Figure 4.20, since these will be substituted by __tx_PendSVHandler and systick_cycles (declared in main as (SystemCoreClock / TX_TIMER_TICKS_PER_SECOND) - 1), respectively.

2. The **tx_initialize_low_level.s** file is used to define a portion of the interrupt processing function, establish the system pulse, initialize the stack address and vector table, and configure the interrupt priority. However, some of these functions are already in the starting file, **startup_stm32l462xx.s**. This way it is important to manage and eliminate any conflicts.

In Code 4.1, one can see some configurations implemented by _tx_initialize_low_level(), namely the configuration of the interrupt vector table, the adress of the system stack pointer and the SysTick.

```
;    /* Setup Vector Table Offset Register.  */
;
    MOV     r0, #0xE000E000                          ; Build address of NVIC
     registers
    LDR     r1, =__tx_vectors                        ; Pickup address of
   vector table
    STR     r1, [r0, #0xD08]                         ; Set vector table
   address
;
;    /* Set system stack pointer from vector value.  */
;
    LDR     r0, =_tx_thread_system_stack_ptr         ; Build address of
   system stack pointer
    LDR     r1, =__tx_vectors                        ; Pickup address of
   vector table
    LDR     r1, [r1]                                 ; Pickup reset stack
   pointer
    STR     r1, [r0]                                 ; Save system stack
   pointer

;    /* Configure SysTick.  */
    MOV     r0, #0xE000E000                          ; Build address of NVIC
     registers
    LDR     r1, =systick_cycles
  LDR     r1, [r1]
    STR     r1, [r0, #0x14]                          ; Setup SysTick Reload
   Value
    MOV     r1, #0x7                                 ; Build SysTick Control
     Enable Value
    STR     r1, [r0, #0x10]                          ; Setup SysTick Control
```

Code 4.1: Part of the startup_stm32l462xx.s file code.

### 4.2.3 Modem

One of the main changes to the architecture of the device was the modem. In the previous application the modem was a Quectel BC66 and in this application it is Murata Type1SC. While some of the commands and part of the structure associated with the Modem code remained the same, there were several changes that needed to be made.

#### Initialization

One of those changes was to the peripherals initialization in the main(), as shown in Code 4.2. In order for the modem to work, the functions "TYPE1SC_close_channel()" and "TYPE1SC_open_channel()" need to come after GPIO initialization and before the initialization of the USART responsible for the communications between the modem and the MCU, in this case USART3.

```
/* Initialize all configured peripherals */
   MX_GPIO_Init();
   TYPE1SC_close_channel();    // these two functions need to be
   TYPE1SC_open_channel();   // before MX_USART3_UART_Init();
   MX_USART3_UART_Init();
   MX_RTC_Init();
   MX_I2C1_Init();
   MX_AES_Init();
   MX_CRC_Init();
   MX_I2C3_Init();
   MX_LPTIM1_Init();
   MX_TIM6_Init();
   MX_USART1_UART_Init();
```

Code 4.2: Peripherals Initialization.

The "TYPE1SC_close_channel()" function is responsible for setting the data terminal ready (DTR), which is the pin PC8, to low. The corresponding fluxogram is shown in Figure 4.24.

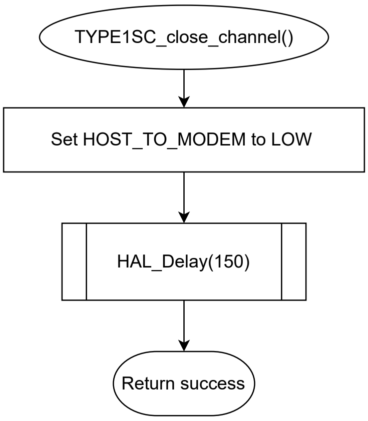


Figure 4.24: TYPE1SC_close_channel fluxogram.

The "TYPE1SC_open_channel()" function is responsible for setting the data terminal ready (DTR), which is the pin PC8, to high. The corresponding fluxogram is show in Figure 4.25.

On the front of many external modems are LED indications, one of which is TR (”terminal ready”). The DTR pin’s status is tracked by it. When DTR is high, the LED is on, and when it is low, it is off.

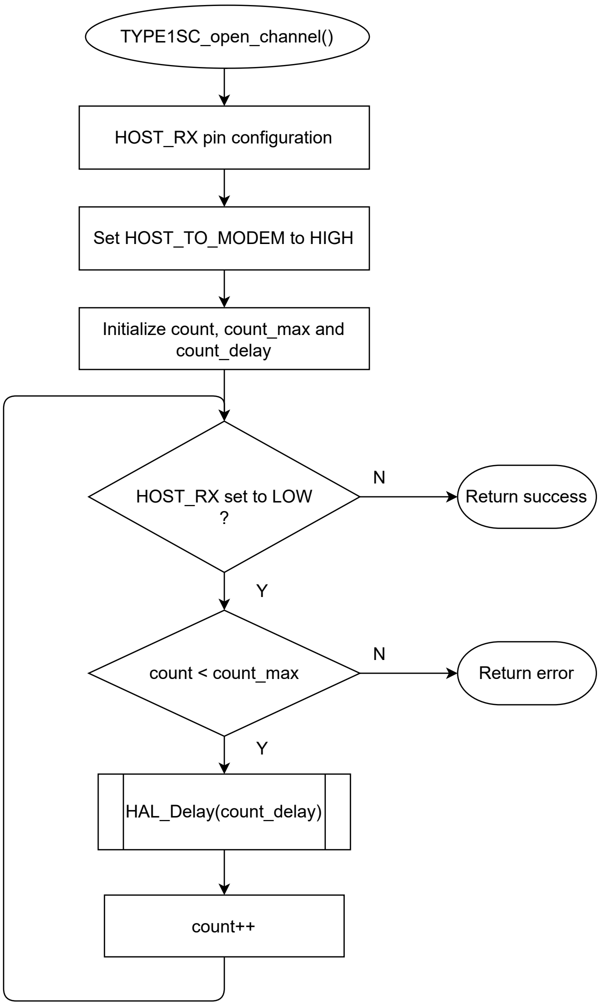


Figure 4.25: TYPE1SC_open_channel fluxogram.

### Modem Configurations

After being initialized, the modem needs to be configured to work according to the application’s needs. This happens in configModem_e function. In this function: the modem is configured to work at a 9600 bps baud rate; PDP context and authentication are defined; the preferred Radio Access Technology (RAT) list is set to ”NBIOT”; power saving methods specific to the modem are configured here, in the case of the network used only PSM is available.

### Sockets

In Figure 4.26, the three different socket states possible for this modem are represented. These states are: closed, allocated and connected.

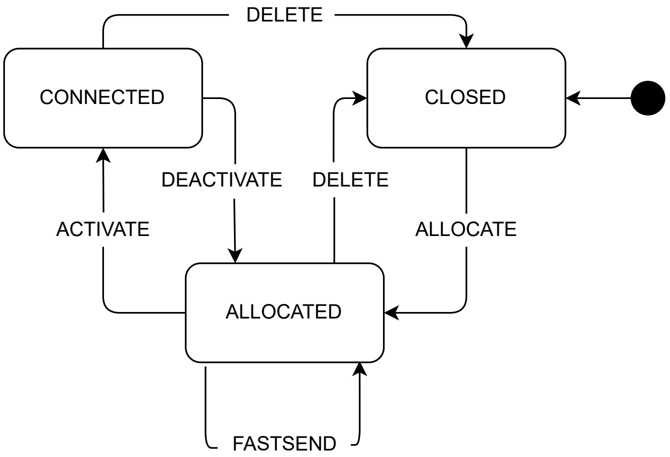


Figure 4.26: Socket States (Adapted from [76]).

Regarding the AT commands, the structure of the commands and their functionalities is in some cases vastly different from the previous modem, making it mandatory for the structure of the code to change, as for example the data processing. Some examples of these situations are the parsing of the date sent by the modem to the MCU to update the RTC or the parsing of the messages.

### Create and Open Socket

When using sockets for communication, there is the need to first create a socket. This socket can use either TCP or UDP. The creation of the socket is made through the AT command:

"AT%SOCKETCMD="ALLOCATE",<SessionID>,"UDP","OPEN",<IP>, <Destination UDP/TCP port>, <Source UDP/TCP port>".

After the socket is created, it is still not ready to send or receive data. It needs to be activated through the AT command "AT%SOCKETCMD="ACTIVATE",<id>", as shown in the Code 4.3.

```
status_et modemOpenSocket_e(uint8_t socket_u8, const char* ip_addr_cc,
                            const char* dest_port_cc, const char*
    src_port_cc){
     status_et status_e;
     unsigned long actual_flags_ul;

     /* Using status_e as command message length*/

    /* open UDP socket with local port */
     status_e = (status_et)snprintf(m_tx_ac, CMD_STR_SIZE, "AT%%SOCKETCMD=\"
    ALLOCATE\",0,\"UDP\",\"OPEN\",\"%s\",%s,%s", ip_addr_cc, dest_port_cc,
    src_port_cc);

     /* Clean event flags */
     tx_event_flags_get(&m_eventflags, socket_open_ef|error_ef,
                         TX_OR_CLEAR, &actual_flags_ul, TX_NO_WAIT);

     status_e = checkRcv_e(m_tx_ac, status_e, 2000);
     if (status_e != success){
         return status_e;
     }
     socket = nb_iot.last_msg[0];
     status_e = (status_et)snprintf(m_tx_ac, CMD_STR_SIZE, "AT%%SOCKETCMD=\"
    ACTIVATE\",%c", nb_iot.last_msg[0]);

     status_e = checkRcv_e(m_tx_ac, status_e, 2000);
     if (status_e != success){
         return status_e;
     }


     return success;
}
```

Code 4.3: Open Socket.

### Send and Receive Data

In order to interpret the AT commands that are received, the application has a switch case that checks if a keyword of the AT command was received and then proceeds to treat it accordingly. Some of the keywords are for example: ”OK”, ”ERROR”, ”PINGCMD”, ”SOCKETEV”, ”SOCKETDATA”, ”SOCKETCMD”, etc.

Unlike the modem used in the previous application, this modem has the same AT command for both sending and receiving data through a socket. The AT command ”AT%SOCKETDATA” is used in both situations. This creates the need to distinguish when an AT command response is to confirm data was sent and when it contains a message.

To solve this problem a flag was created. This flag is called ”msg_rcv_ef” and it is set every time the modem sends a request to read any available message it might have received. When the keyword ”SOCKETDATA” is detected, the application proceeds to check if this flag is active. If this is the case, it means a message was received and the program will save it. If this is not the case, it means it is information regarding the arrival of the data sent by the device itself. An illustration of this decision-making process is displayed in Figure 4.27. Also, the corresponding code snippet is shown in Code 4.4.

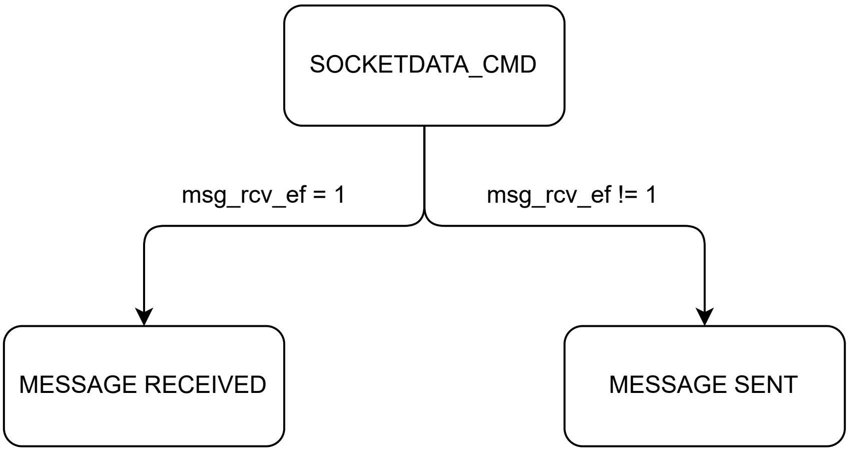


Figure 4.27: SOCKETDATA - AT command processing.

```
case SOCKETDATA_CMD:
  if ((circStrstr((const char*)m_rx_au8,"SOCKETDATA", begin_i_u32,
    M_RX_BUFFER_SIZE) != NULL)){

      /*CHECK IF "SOCKECTDATA="RECEIVE",..." WAS ALREADY SENT*/
      status_e = (status_et)tx_event_flags_get(&m_eventflags, msg_rcv_ef,
    TX_OR_CLEAR, &actual_flags_ul, MS_TO_TICKS(3000));

     if(status_e == success){
         //Murata modem answer --> \%SOCKETDATA:<socket_id>[,<rlength>,<more
     Data>[,<rdata>]] OK/ERROR

     char *ptr_pc;
     token_pc = (char*)&m_rx_au8[index_u32 + 3];

     uint16_t len_rcv_u16 = TO_UINT16(circStrtoul(CHAR_PTR(m_rx_au8), &
    ptr_pc, 10, begin_i_u32+(token_pc-CHAR_PTR(&m_rx_au8[begin_i_u32])),
    M_RX_BUFFER_SIZE));

     circStrtoul(CHAR_PTR(m_rx_au8), &ptr_pc, 10, begin_i_u32+(&ptr_pc[1]-
    CHAR_PTR(&m_rx_au8[begin_i_u32])), M_RX_BUFFER_SIZE);

     if(ptr_pc[0] == ',') {
       circMemcpy(nb_iot.cloud_dest_pau8 + nb_iot.data_rcv_u16, m_rx_au8,
    len_rcv_u16*2, begin_i_u32+(&ptr_pc[2]-CHAR_PTR(&m_rx_au8[begin_i_u32]))
    , M_RX_BUFFER_SIZE);
       //len_rcv_u16*2 because data is received in hexa, so it has double
    the size

       /*Convert hexa to text*/
       char buf = 0;
       for (int i = 0; i < (len_rcv_u16*2) && (i/2 + 1) <= len_rcv_u16; i++)
     {
            if (i \% 2 != 0) {
                nb_iot.cloud_dest_pau8[i/2] = hex_to_ascii(buf, nb_iot.
    cloud_dest_pau8[i]);
            }
            else {
            buf = nb_iot.cloud_dest_pau8[i];
            }
        }

        nb_iot.data_rcv_u16 += len_rcv_u16;

        /* Set event flag to alert URC received.  */
        tx_event_flags_set(&m_eventflags, rcv_data_ef, TX_OR);
     }
  }
  else
  /*ELSE --> SENT "SOCKECTDATA="SEND",..."*/
  tx_event_flags_set(&m_eventflags, send_ok_ef, TX_OR);
}
break;
```

Code 4.4: Switch case snippet (SOCKETDATA_CMD).

### Send Data

After the socket is open, the modem is ready to send a message. The modem sends and receives data in hexadecimal format. Because of this, it is important to convert text to hexa before sending the data. After the data is sent, the application waits for the confirmation that the operation was successful, otherwise it will return an error, as in Code 4.5.

```
status_et modemSendData_e(uint8_t socket_u8, uint8_t *data_pu8, uint16_t
   data_len_u16){
    unsigned long actual_flags_ul;
    status_et status_e;

    /* Clear send_ok_ef and error event flag */
    tx_event_flags_get(&m_eventflags, ok_ef|error_ef, TX_OR_CLEAR, &
   actual_flags_ul, TX_NO_WAIT);

    status_e = (status_et)snprintf(m_tx_ac, CMD_STR_SIZE, "AT%%SOCKETDATA
   =\"SEND\",1,%d,\"",data_len_u16);
    writeUART_v(MODEM, (uint8_t*)m_tx_ac, status_e);

    char data_pu8_hexa[2];
    for(int i = 0; i<data_len_u16; i++){
        sprintf(data_pu8_hexa, "%02X", data_pu8[i]);
        writeUART_v(MODEM, (uint8_t*)data_pu8_hexa, 2);
    }
    /* Send \r\n */
    writeUART_v(MODEM, (uint8_t*)"\"\r", 2);

    /* Waits for OK or ERROR event flag*/
    if(tx_event_flags_get(&m_eventflags, ok_ef|error_ef, TX_OR_CLEAR,
       &actual_flags_ul, MS_TO_TICKS(10000))!= TX_SUCCESS){
        return timeout;
    }

      if((actual_flags_ul & ok_ef) == 0){
        return failure;
    }
    return success;
}
```

Code 4.5: Send Data.

### Get Cloud Data

In some cases where the modem receives data, it will first receive an AT command ”AT%SOCKETEV” that will notify that there is data to be read. Afterward, the modem needs to send an AT command requesting to read the data: ”AT%SOCKETDATA=”RECEIVE”,...”. After making this request, the application activates rcv_data_ef, as in Code 4.6. This way the application will know that the next command the modem receives contains a message, as stated previously.

```
status_et modemGetCloudData_e(uint8_t *dest_pu8, uint16_t* len_pu16,
    uint16_t max_len_u16){
     char* token_pc = NULL;
     char *ptr_pc = NULL;
     uint8_t connect_id_u8;
     status_et status_e;
     unsigned long actual_flags_ul;

     nb_iot.cloud_dest_pau8 = dest_pu8;
     nb_iot.data_rcv_u16 = 0;

     status_e = (status_et)snprintf(m_tx_ac, CMD_STR_SIZE, "AT%%SOCKETDATA
    =\"RECEIVE\",%d,%d",socket,nb_iot.dest_max_len_u16);
         /* Clear rcv event flag*/
         tx_event_flags_get(&m_eventflags, rcv_data_ef, TX_OR_CLEAR, &
    actual_flags_ul, TX_NO_WAIT);

         status_e = checkRcv_e(m_tx_ac, status_e, 3000);
         if (status_e != success){
             return status_e;
         }

         status_e = (status_et)tx_event_flags_get(&m_eventflags, rcv_data_ef
    , TX_OR_CLEAR, &actual_flags_ul, MS_TO_TICKS(3000));

     return success;
}
```

Code 4.6: Get Cloud Data.

# Chapter 5: Tests and Results

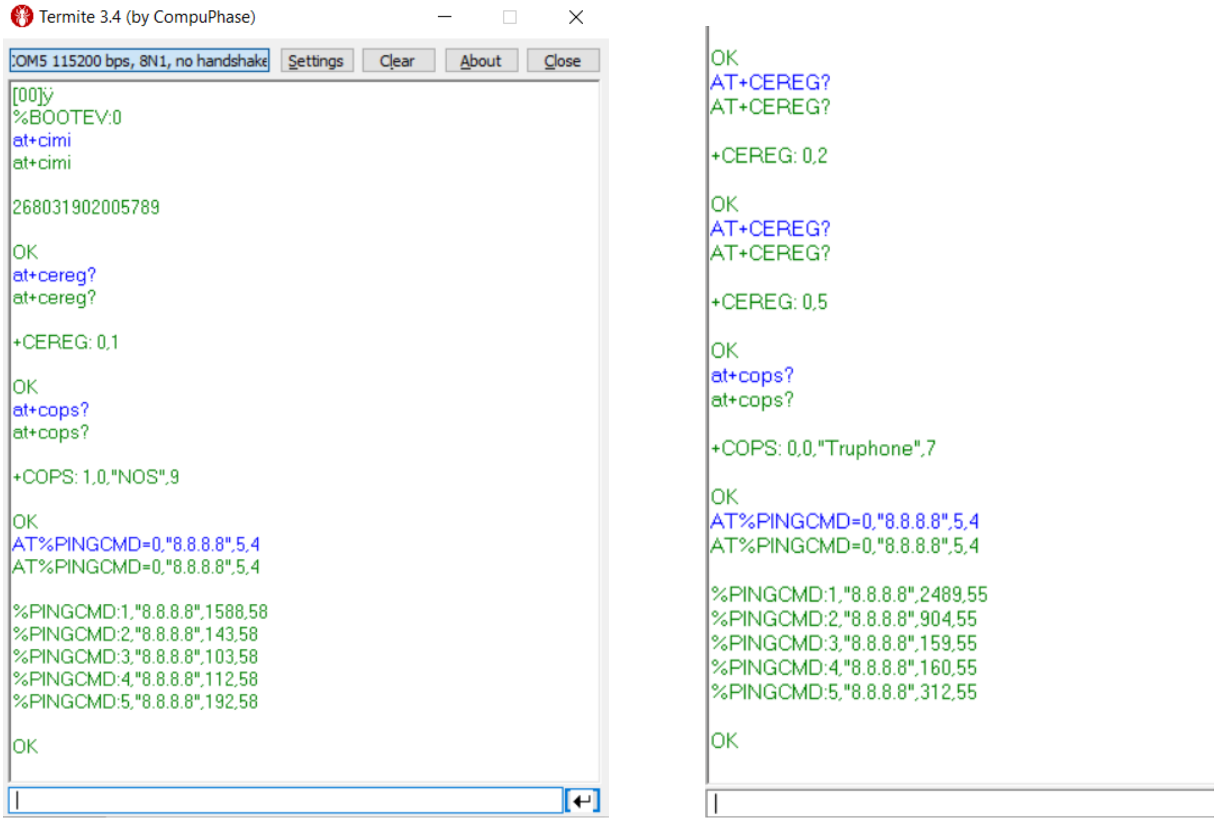
This section describes the results of some hardware and system functionality tests, as well as the power consumption measurement. First, the systems' functionality was tested. Following that, a brief explanation of the equipment utilized is provided. The most relevant results will then be analyzed.

## 5.1 Unit Tests

The basic purpose of unit testing is to separate the smallest portion of a system from the rest and assess whether or not it operates as expected. This is an important step in avoiding problems when merging all modules because it reduces the extent of such faults.

### 5.1.1 Modem

In order to test the communication with the modem and the connection to the server, a simpler bare-metal application was developed. This application consisted of two USARTS, USART3 (modem-MCU) and USART1 (mcu-user). This way one could communicate with the modem and test the different AT commands without the complexity of the final application. In Figure 5.1, it is possible to see the result of the application on the terminal.



```
:OM5 115200 bps, 8N1, no handshake
[00]ÿ
%BOOTEV:0
at+cimi
at+cimi

268031902005789

OK
at+cereg?
at+cereg?

+CEREG: 0,1

OK
at+cops?
at+cops?

+COPS: 1,0,"NOS",9

OK
AT%PINGCMD=0,"8.8.8.8",5,4
AT%PINGCMD=0,"8.8.8.8",5,4

%PINGCMD:1,"8.8.8.8",1588,58
%PINGCMD:2,"8.8.8.8",143,58
%PINGCMD:3,"8.8.8.8",103,58
%PINGCMD:4,"8.8.8.8",112,58
%PINGCMD:5,"8.8.8.8",192,58

OK
```

(a) NOS Sim Card - NB-IoT.

```
OK
AT+CEREG?
AT+CEREG?

+CEREG: 0,2

OK
AT+CEREG?
AT+CEREG?

+CEREG: 0,5

OK
at+cops?
at+cops?

+COPS: 0,0,"Truphone",7

OK
AT%PINGCMD=0,"8.8.8.8",5,4
AT%PINGCMD=0,"8.8.8.8",5,4

%PINGCMD:1,"8.8.8.8",2489,55
%PINGCMD:2,"8.8.8.8",904,55
%PINGCMD:3,"8.8.8.8",159,55
%PINGCMD:4,"8.8.8.8",160,55
%PINGCMD:5,"8.8.8.8",312,55

OK
```

(b) Truphone eSIM - LTE Roaming.

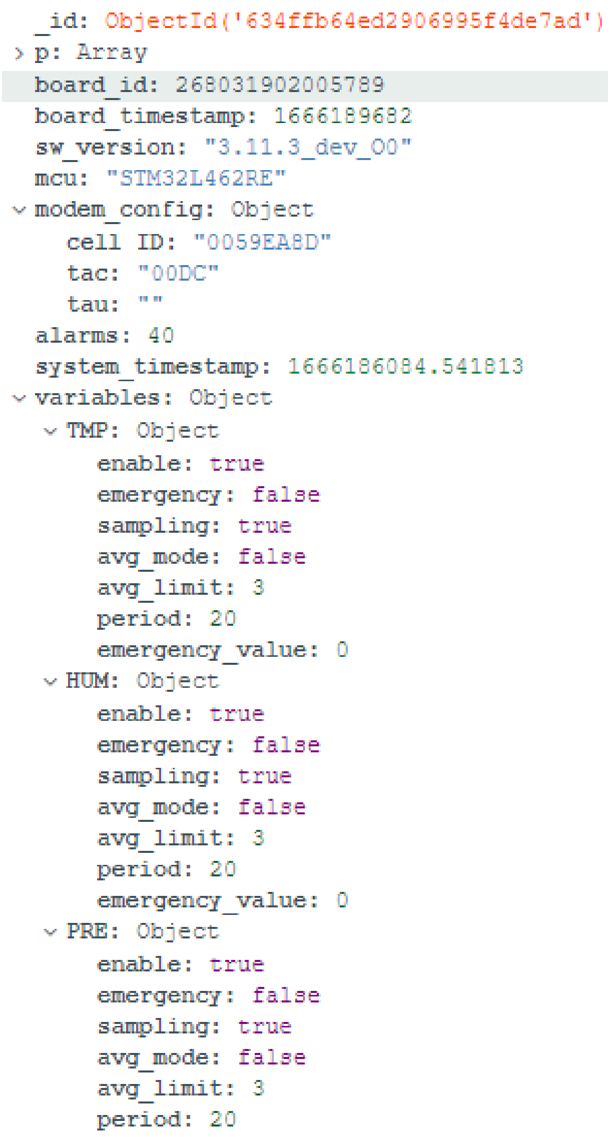
Figure 5.1: Unit Test: Modem - Terminal.

On the left, the ping command is being tested for NB-IoT using the NOS SIM card. On the right, the same test is done for LTE using the Truphone eSIM on Roaming.

### 5.1.2 Server Messages

Figure 5.2 and Figure 5.3 below show both messages in the server (which was developed out of the scope of this dissertation), proving that messages are being sent and that their content has adequate data.

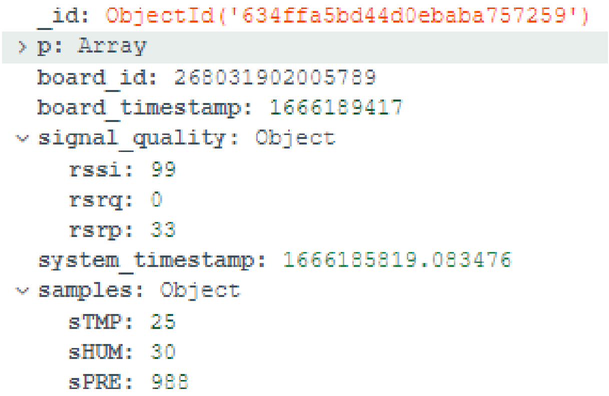
The first message shows the config message, that is sent after the modem is connected. Some of the information shown by this type of message includes, for example, the sensors that are sampling, if emergencies are on or off, the board identification and the RTC alarm interval. As one can see, the board_id is the same as the CIMI in Figure 5.1, proving that it is the same device communicating.



```
  _id: ObjectId('634ffb64ed2906995f4de7ad')
› p: Array
  board_id: 268031902005789
  board_timestamp: 1666189682
  sw_version: "3.11.3_dev_O0"
  mcu: "STM32L462RE"
⌄ modem_config: Object
    cell ID: "0059EA8D"
    tac: "00DC"
    tau: ""
  alarms: 40
  system_timestamp: 1666186084.541813
⌄ variables: Object
  ⌄ TMP: Object
      enable: true
      emergency: false
      sampling: true
      avg_mode: false
      avg_limit: 3
      period: 20
      emergency_value: 0
  ⌄ HUM: Object
      enable: true
      emergency: false
      sampling: true
      avg_mode: false
      avg_limit: 3
      period: 20
      emergency_value: 0
  ⌄ PRE: Object
      enable: true
      emergency: false
      sampling: true
      avg_mode: false
      avg_limit: 3
      period: 20
```

Figure 5.2: Config.

The second message (see Figure 5.3) shows the regular message, that is sent every time the RTC alarm completes its previously set interval. This message shows the board_id and the last sensor samples before the RTC alarm trigger, among other information.



```
  _id: ObjectId('634ffa5bd44d0ebaba757259')
> p: Array
  board_id: 268031902005789
  board_timestamp: 1666189417
v signal_quality: Object
    rssi: 99
    rsrq: 0
    rsrp: 33
  system_timestamp: 1666185819.083476
v samples: Object
    sTMP: 25
    sHUM: 30
    sPRE: 988
```

Figure 5.3: Regular.

## 5.2 Experimental Setup

In Figure 5.4 the setup used to test the power consumption is depicted. It was made up of an oscilloscope Tektronix MDO3012 (on the right) to capture MCU GPIO toggles from the application, and the precise Source/Measure Unit (SMU) Keysight B2901A (on the left). The oscilloscope data was obtained using Tektronix's OpenChoiceDesktop program. The end-device power consumption was measured using the Keysigh B2900 Quick IV Measurement Software, and the average current consumption was calculated, plotted, and post-processed using Matlab.

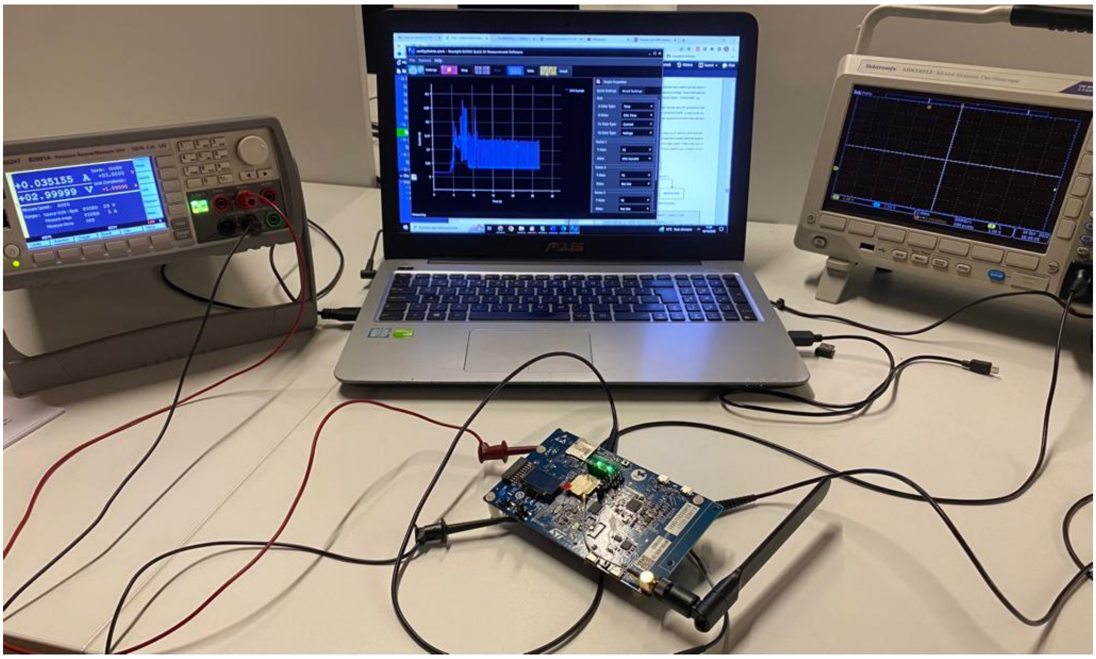


Figure 5.4: Experimental Setup.

## 5.3 Current Consumption Measurements

Current consumption measurements are shown in the present section. These consumption measurements were used to understand the consumption of different operation modes of the modem and the consumption of different application events and estimate the battery life in a later section.

### 5.3.1 B-L462E-CELL1 Discovery Kit

In this subsection, some consumption measurements are presented regarding the different power-saving modes and the application events running the application on B-L462E-CELL1 Discovery Kit.

#### LTE-M vs NB-IoT Default vs NB-IoT with PSM

In the three following figures, Figure 5.5, Figure 5.6, and Figure 5.7, one can see the current consumption of the discovery kit while running the application using different technologies and power-saving modes available for NOS, namely: LTE-M, NB-IoT and NB-IoT with PSM. These three tests and the other measurements shown later were executed regarding the full system running for 100 seconds on the B-L462E-CELL1 Discovery Kit at 4 MHz, with an RTC interval of 40 seconds, which means a regular message is sent from the modem to the server each 40s. The spikes of the graphics will be explained in greater detail in the next subsection.

Figure 5.5 displays the discovery kit's current consumption while the application uses LTE-M (or CATM).

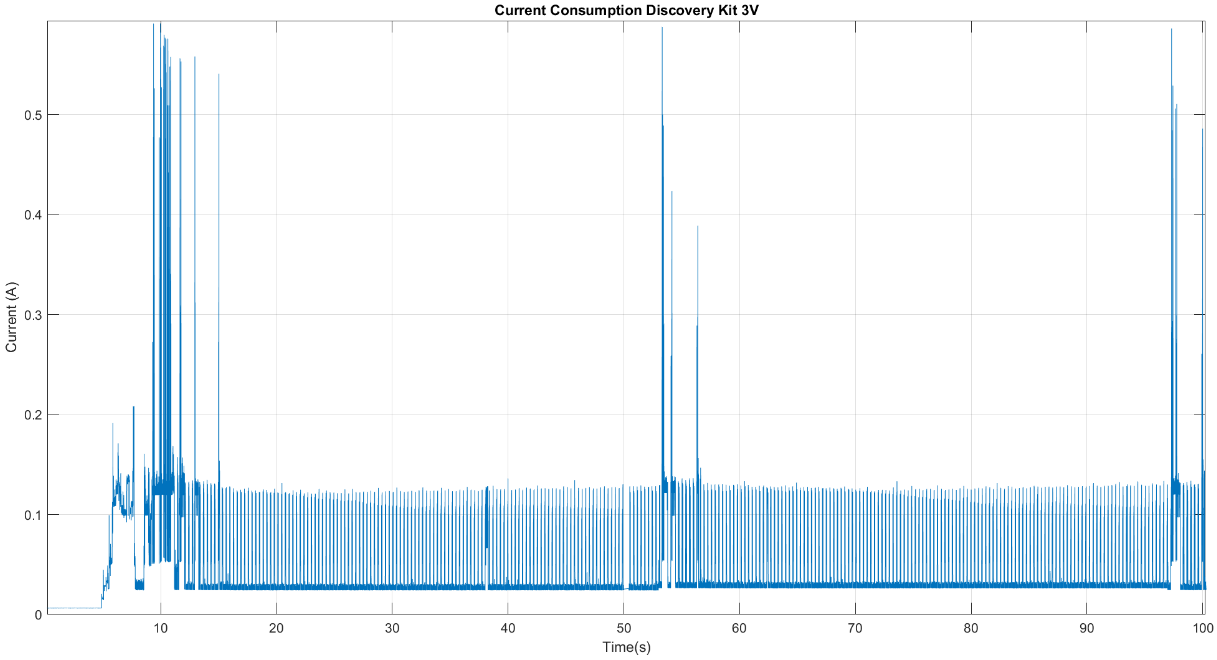


Figure 5.5: CATM1 w/o PSM.

Figure 5.6 shows the discovery kit's current consumption while the application is being used with NB-IoT, yet without PSM.

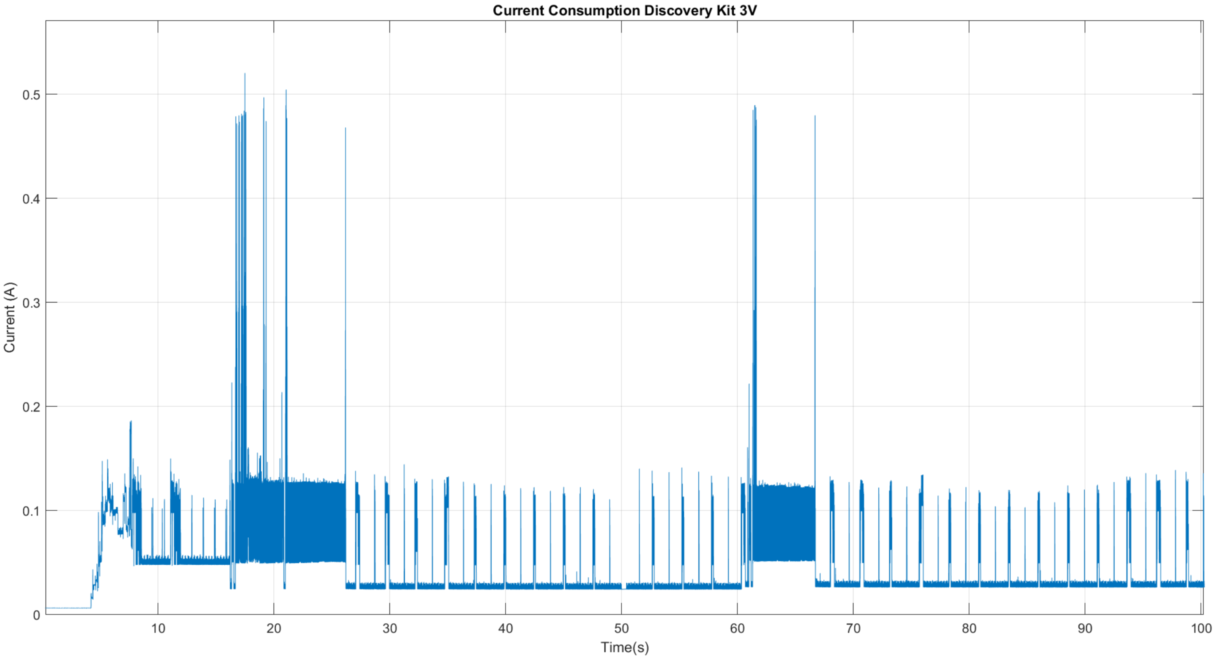


Figure 5.6: NB-IoT w/o PSM.

Figure 5.7 shows the discovery kit's current consumption while the application is being used with NB-IoT and PSM active.

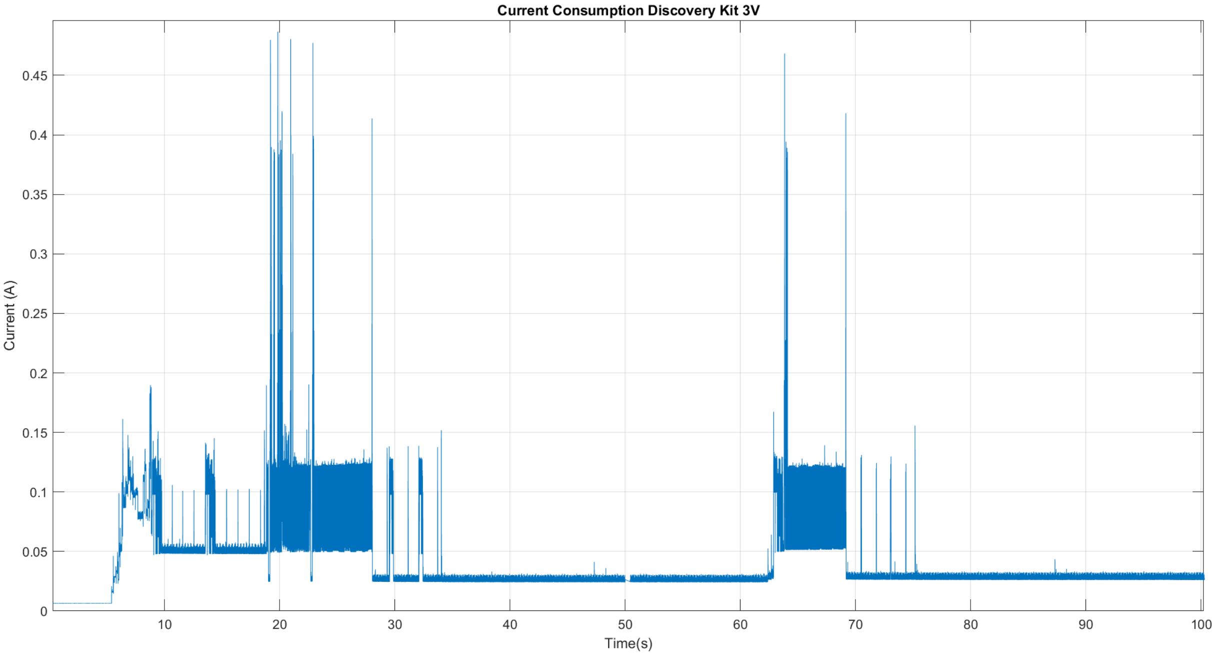


Figure 5.7: NB-IoT w PSM.

It is possible to see PSM's impact on power consumption, by comparing the three figures. If the time-space between regular transmissions is big enough the current draw of the application using PSM will be less than when the application runs without PSM. Also, the characteristics of PSM are clearly noticeable in Figure 5.7: the period of the DRX cycles is followed by an idling period (PSM) which drastically decreases the power consumption. The use of eDRX is not available in the NOS network, henceforth it was not possible to be tested.

### 5.3.2 Events Representation and Time Counting

In Figure 5.8, the orange dashed lines represent at first the initialization of the sensors and afterward (starting at 27.18 s) they represent the sensor's sampling every 5 seconds. It is worth noting that the current draw resulting from the sampling of the sensors goes unnoticed since the full system is being considered (as it is not possible to test the components separately in the Discovery Kit).

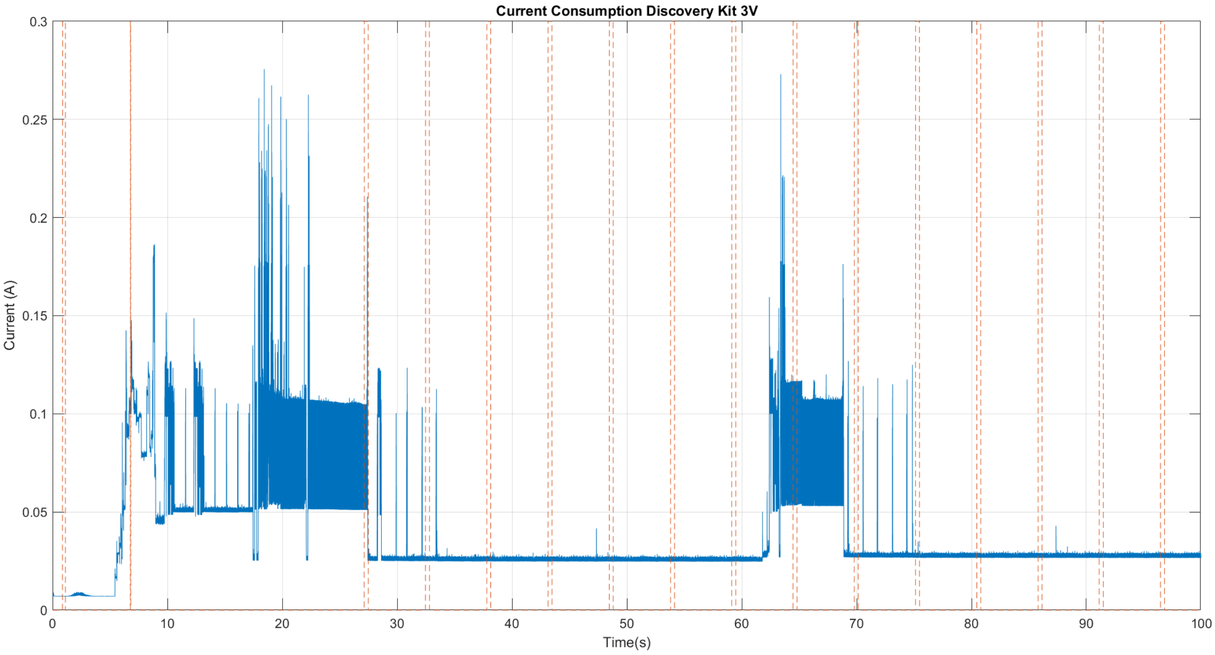


Figure 5.8: Sensors Sampling.

These figures show current spikes while the modem is being initialized, as well as the DateTime request and configuration are transmitted through the modem. Later, the modem enters a stage where it can receive data from the cloud and enters PSM. In Figure 5.9, the red dashed lines highlight the transmission of a regular message, set to happen every 40 seconds, as previously stated.

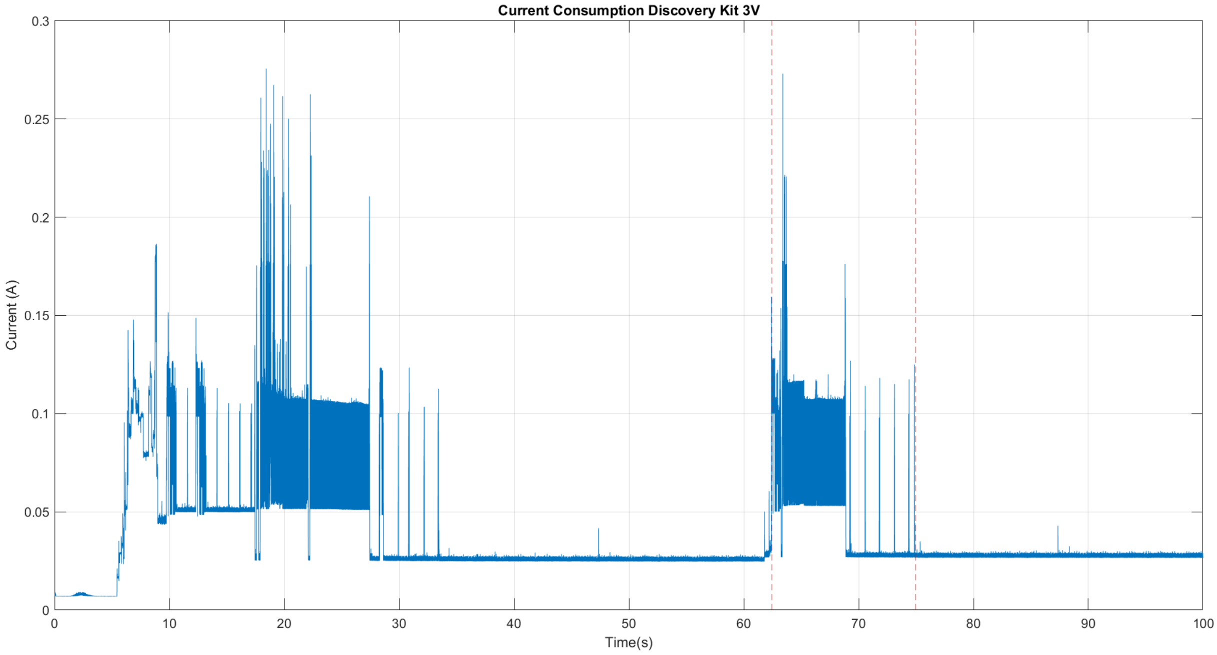


Figure 5.9: Regular Transmission.

## 5.4 Battery Life Estimation

To estimate the application power consumption, it is necessary to estimate the number of milliamps hours (mAh) per day that the system uses. As a result, it may be written as the sum of the modem's average power consumption, the MCU's and sensors' average power consumption, and the battery leakage, as shown in Equation 5.1. However, since it was not possible to test the components of the B-L462E-CELL1 Discovery Kit individually, the equation for the discovery kit is as in Equation 5.2. The possibility of individualized consumption measurement was one more reason for the design of the customized PCB.

$$Q_{Total} = Q_{Modem}[mAh] + Q_{MCU}[mAh] + Q_{Peripherals}[mAh] + Q_{Leakage} \tag{5.1}$$

$$Q_{Total} = Q_{Discovery\_Kit}[mAh] + Q_{Leakage} \tag{5.2}$$

The capacity of a battery in milliamps hours (mAh) is equal to the current drawn from the battery in milliamps (mA) times the time in hours (h) during which the current is drawn, as presented in Equation 5.3. Consequently, the battery life of a device can be calculated through Equation 5.4.

$$Q[mAh] = I[mA] \times t[h] \tag{5.3}$$

$$Battery\_life[h] = \frac{Q[mAh]}{I[mA]} \tag{5.4}$$

On table 5.1, one can see the full system consumption for each of the main events of the application running on the discovery kit. For each event, the average current consumption and duration are displayed. The resulting average current consumption is then converted to mA.h per day based on the number of occurrences (n) of each event and the amount of time the modem is sleeping (which is given by $24 - T_{reg} - T_{comp} - T_{samp}$ ). The board power consumption is the sum of each type of power consumption per day, which is the same as summing the values of the column on the right, (1)+(2)+(3)+(4). Each type of power consumption per day is calculated by multiplying the average current consumption (mA) by the time (h) per day.

Table 5.1: Full System Consumption for the B-L462E-CELL1 Discovery Kit.

| Event | Average Current (mA) | Time (s) | Time (h) per day | mA.h per day |
|---|---|---|---|---|
| Regular Transmission | 55.7 | 12.501 | $\frac{n_{reg}\times 12.501}{3600}$ | (1) |
| Command Transmission | 61.2 | 10.98 | $\frac{n_{com}\times 10.98}{3600}$ | (2) |
| Sensors Sampling | 25.2 | 0.34 | $\frac{n_{samp}\times 0.34}{3600}$ | (3) |
| Sleep | 25.1 | - | Remaining Time | (4) |
| | | | Total | $\sum_{i=1}^{4}(i)$ |

The battery used is a Saft LM 17500, which has a 1% leakage [62]. The battery capacity ($Q_B$) in mAh determines the average leakage current, and IL% is the annual self-leakage percentage. Equation 5.5 depicts the average leakage current consumption, which is 0.003425 mAh/day.

$$I_{leakage} = \frac{Q_B \times I_L\%}{24 \times 365} \tag{5.5}$$

Theoretically, the number of days that the system can run can be calculated by dividing the battery capacity by the daily average, as shown in Equation 5.6.

$$days = \frac{Q_B}{Avg_{Daily}} \tag{5.6}$$

As an example of a real-world application, the same usecase as in [49] was used for comparison. The temperature, humidity, and light sensors will do 72 samples each, resulting in 216 samples between transmission every 6 hours. Per day 864 samples, four regular transmissions and 0.33 command transmissions.

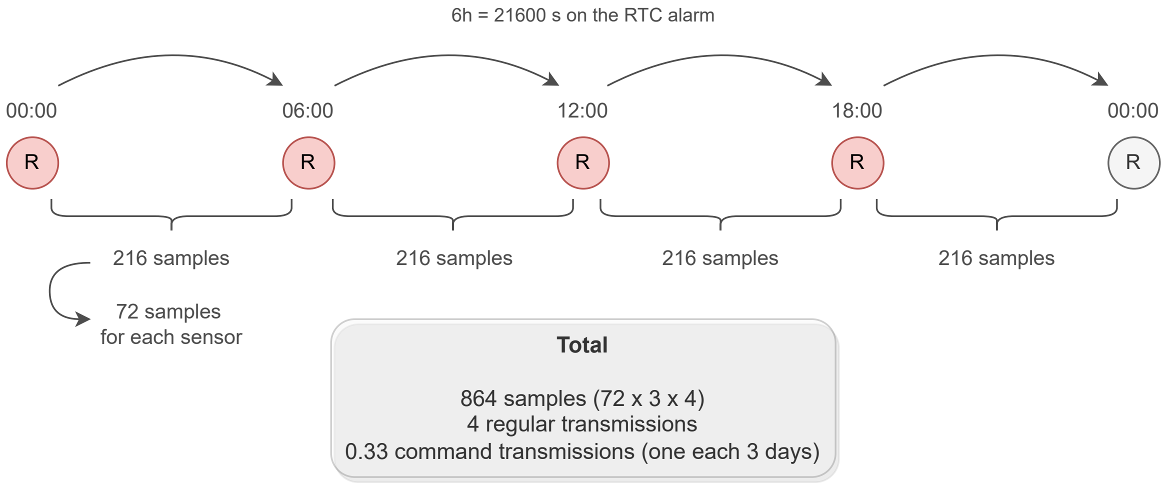
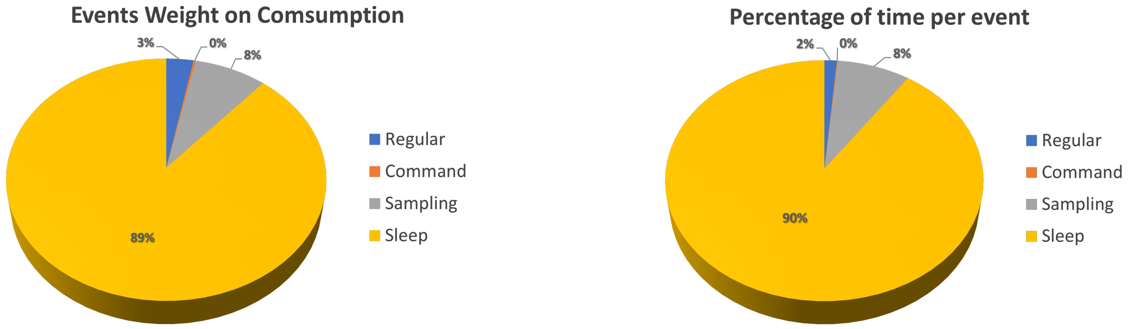


Figure 5.10: Real-world application use case

The average daily consumption for this usecase equals **602.8729** mAh, leading to a battery life of 4.97 days, considering Saft LM 1750 battery.

The weight of the events on the estimation of the average daily consumption of the discovery kit is represented in Figure 5.11a, while the percentage of time spent in each event is shown in Figure 5.11b.



(a) Weight of the events on power consumption.

(b) Weight of the events according to their time.

Figure 5.11: Analysis of the impact of the application events on the Discovery Kit.

It is possible to conclude that sleep is the most time-consuming event. At the same time, it has the most weight on the consumption of the discovery kit. This happens because even though Sleep consumes less than the other events, it still has high consumption considering it is supposed to be for a low-power application. One can see almost a proportion between the time spent by an event and the weight it has on the consumption of the device.

### 5.4.1 Battery Life Expected for Customized PCB

Even though it was not possible to measure the consumption of the customized PCB, it was possible to estimate it by consulting the datasheets of the different components.

In table 5.2, one can see an estimate for the consumption of the customized PCB. This table specifies the expected consumption for the various parts of the system, for example, the Type1SC modem (that is part of Type1SE), the MCU, and the sensors. For Type1SC the value for consumption in idle is 1.7 $\mu$A, while during transmission the PSM current draw is 33.7 mA in the worst-case scenario which will be the value considered for all types of transmissions. Meanwhile, the MCU's current draw when in Standby mode is 2.08 $\mu$A and 340 $\mu$A when in Run mode (Range 2) for a clock frequency of 4 MHz. As for the sensors, the consumptions mentioned in section 3.2.2 were considered.

Table 5.2: Expected average consumption for the customized PCB.

| **Event** | **Average Current (mA)** | | | | **Time** | **Time (h)** | **mA.h** |
|---|---|---|---|---|---|---|---|
| | Sensors | MCU | Type1SC | Total | **(s)** | **per day** | **per day** |
| Regular Transmission | 0.00232 | 0.34 | 33.7 | 34.04232 | 12.5010 | $\frac{n_{reg}\times 12.501}{3600}$ | (1) |
| Command Transmission | 0.00232 | 0.34 | 33.7 | 34.04232 | 10.98 | $\frac{n_{com}\times 10.98}{3600}$ | (2) |
| Sensors Sampling | 0.0864 | 0.34 | 0.0017 | 0.4281 | 0.34 | $\frac{n_{samp}\times 0.34}{3600}$ | (3) |
| Sleep | 0.00232 | 0.00208 | 0.0017 | 0.0061 | - | Remaining Time | (4) |
| | | | | | | Total | $\sum_{i=1}^{4}(i)$ |

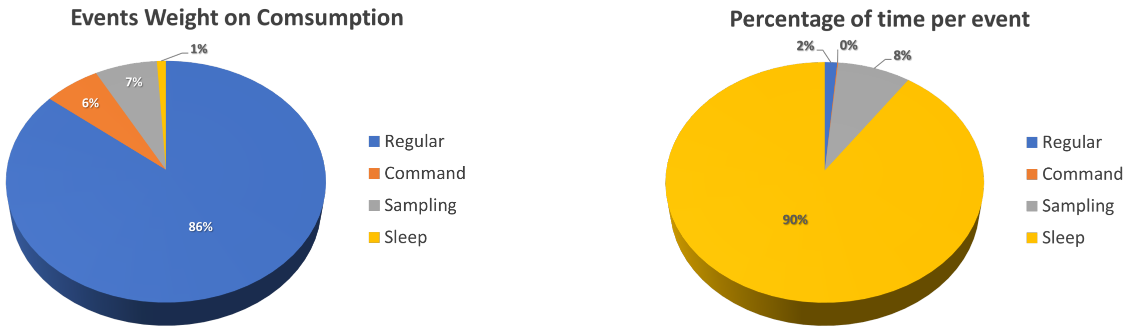
The average daily consumption for the use case presented before equals **0.6879** mAh.

Since the battery life estimation is based on the component's documentation, it is advisable to consider an error margin. In this case, a margin of 5% was considered. The total estimation of the average daily consumption includes not only the estimation based on the datasheets and its error margin but also the battery leakage. Equation 5.7 gives the formula to obtain the total estimation of the average daily consumption considering all these factors.

$$Avg_{Total\_daily\_estimated} = Avg_{Daily\_estimated} \times (1 + 0.05) + Avg_{Leakage} \tag{5.7}$$

According to this same equation, the total estimation of the average daily consumption obtained is **0.7257** mAh, which leads to a battery life of 4134 days or **11.33 years**.

The weight of the events on the estimation of the average daily consumption is represented in Figure 5.12a, while the percentage of time spent in each event is shown in Figure 5.12b.



(a) Weight of the events on power consumption.

(b) Weight of the events according to their time.

Figure 5.12: Analysis of the impact of the application events.

It is possible to conclude that the Regular Transmission has the greatest weight on the estimated power consumption even though it is one of the less time-consuming events of the application. At the same time, the Sleep event represents most of the application activity and is predicted to consume the least amount of energy, contrary to what can be seen in Figure 5.11.

# Chapter 6: Conclusions and Future Work

It has never been more crucial to employ IoT devices to collect data that can eventually be used to feed machine learning models or to monitor the environment. The current systems would benefit greatly from the addition of devices powered by batteries with expanded (>10 years) lifespan, while being independent of reticulated power, since they can collect data for many years without human intervention in remote locations. One of the several LPWAN technologies that enables achieving this lifetime is the NB-IoT. NB-IoT connects to the network via the existing LTE service. It can do so even in areas with low LTE coverage, like distant or even underground locations.

One of the goals of this dissertation was to refactor the previously developed application to work with a new MCU and modem, both integrated into Type1SE module, in order to evaluate its potential for low-power use cases. The software was developed and tested in B-L462E-CELL1 Discovery Kit, as it has the module Type1SE that is intended to be studied. However, as a discovery kit has several peripherals that are not needed for the use case, a customized PCB was designed to accommodate the Type1SE and only the necessary peripherals. Also, in order to better compare with previous versions, the PCB needs to be as similar as possible to the hardware of previous versions, using the same battery and, if possible, the same sensors.

The B-L462E-CELL1 Discovery Kit lifetime, when powered by a Saft LM 17500, is expected to be around 4.97 days, with an average daily consumption of 602.8729 mAh. This battery life is low, considering it is for a low-power IoT application. This is greatly due to the fact that the board has a high number of peripherals and electrical components that are unnecessary to the use case. Meanwhile, the estimated battery life for the customized PCB built in this work is 4134 days or **11.33 years**, with an average daily consumption of 0.7257 mAh. Even though the estimated battery life for the customized PCB is significantly higher than the battery life of the development kit, it is still shorter than the previously developed version [49]. This happens for several reasons: the MCU being tested is more powerful; the modem Type1SC consumes more energy than the previously used modem; the light sensor used in previous versions was not available, and this sensor is more energy-consuming. Nonetheless, the estimated battery life for the customized PCB exceeds the criteria for a battery life of more than ten years.

## 6.1 Future Work

Despite the work developed throughout this dissertation, there are still certain aspects that have room for improvement. Because of this, some areas for future development are listed below along with potential fixes:

- **Peripherals**
  Besides the I2C and the sensors, there are still many different peripherals that can be integrated into the application so that it can fulfill different purposes. Also, from a hardware standpoint, new and more optimized components are always entering the market, so it might be advantageous to change part of it in the future.

- **eSIM**
  Further analysis can be made regarding the power consumption and efficiency that results from using an eSIM. Operator-related issues must be cleared out before pursuing this solution.

- **Custom Board Soldering and Testing**
  Due to the chip shortages, there was a delay in the delivery of the Type1SE module, which made it impossible to test the custom board. This board, as stated before, would likely produce more interesting results and conclusions, regarding power consumption. A clear goal for the near future is to test the power consumption for the already designed custom board and compare it with the previous solutions.

# References

[1] J. K. Reena and R. Parameswari, “A Smart Health Care Monitor System in IoT Based Human Activities of Daily Living: A Review,” in *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, pp. 446–448, 2019.

[2] O. Bhat, P. Gokhale, and S. Bhat, “Introduction to IOT,” *International Advanced Research Journal in Science, Engineering and Technology ISO*, vol. 3297, 2007.

[3] A. J. Trappey, C. V. Trappey, U. H. Govindarajan, A. C. Chuang, and J. J. Sun, “A review of essential standards and patent landscapes for the Internet of Things: A key enabler for Industry 4.0,” *Advanced Engineering Informatics*, vol. 33, pp. 208–229, 8 2017.

[4] R. T. Tiburski, L. A. Amaral, E. D. Matos, and F. Hessel, “The importance of a standard security architecture for SOA-based iot middleware,” vol. 53, pp. 20–26, Institute of Electrical and Electronics Engineers Inc., 12 2015.

[5] S. B. S. Paiva and J. Cabral, “Design and Power Consumption Analysis of a NB-IoT End Device for Monitoring Applications,” *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*, pp. 2175–2182, 2020.

[6] I. Analytics, “The top 10 IoT Use Cases.” `https://iot-analytics.com/top-10-iot-use-cases/`. [Online; Accessed on 18-January-2022].

[7] I. Analytics, “State of IoT 2022: Number of connected IoT devices growing 18% to 14.4 billion globally.” `https://iot-analytics.com/number-connected-iot-devices/`, [Online; Accessed on 30-September-2022].

[8] I. Analytics, “5 things to know about the lpwan market in 2021.” `https://iot-analytics.com/5-things-to-know-lpwan-market/`. [Online; Accessed on 18-January-2022].

[9] V. Alcácer and V. Cruz-Machado, “Scanning the Industry 4.0: A Literature Review on Technologies for Manufacturing Systems,” *Engineering Science and Technology, an International Journal*, vol. 22, no. 3, pp. 899–919, 2019.

[10] M. Bolic, M. Rostamian, and P. M. Djuric, "Proximity detection with RFID: A step toward the internet of things," *IEEE Pervasive Computing*, vol. 14, pp. 70–76, 4 2015.

[11] Zigbee, "Zigbee | Solução completa de IOT - CSA-IOT." `https://csa-iot.org/pt/todas-as-solu%C3%A7%C3%B5es/zigbee/`. [Online; Accessed on 30-May-2022].

[12] "Bluetooth Technology Overview | Bluetooth® Technology Website." `https://www.bluetooth.com/learn-about-bluetooth/tech-overview/`. [Online; Accessed on 04-October-2022].

[13] M. R. Palattella, M. Dohler, A. Grieco, G. Rizzo, J. Torsner, T. Engel, and L. Ladid, "Internet of Things in the 5G Era: Enablers, Architecture, and Business Models," *IEEE Journal on Selected Areas in Communications*, vol. 34, pp. 510–527, 3 2016.

[14] P. Andres-Maldonado, *NB-IoT M2M Communications in 5G Cellular Networks*. Master's Thesis, University of Granada, 2019.

[15] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "A comparative study of LPWAN technologies for large-scale IoT deployment," *ICT Express*, vol. 5, pp. 1–7, 3 2019.

[16] U. Raza, P. Kulkarni, and M. Sooriyabandara, "Low Power Wide Area Networks: An Overview," *IEEE Communications Surveys and Tutorials*, vol. 19, pp. 855–873, April 2017.

[17] W. Ayoub, M. Mroue, F. Nouvel, A. E. Samhat, and J.-c. Prévotet, "Towards IP over LPWANs technologies: LoRaWAN, DASH7, NB-IoT," in *2018 Sixth International Conference on Digital Information, Networking, and Wireless Communications (DINWC)*, pp. 43–47, 2018.

[18] Embien, "Introduction to LoRa Technology - Embien Technologies Blog." `https://www.embien.com/blog/introduction-to-lora-technology/`. [Online; Accessed on 04-October-2022].

[19] Sigfox, "Sigfox - The Global Communications Service Provider for the Internet of Things (IoT)." `https://www.sigfox.com/en`. [Online; Accessed on 30-May-2022].

[20] "Homepage - LoRa Alliance®." `https://lora-alliance.org/`. [Online; Accessed on 30-May-2022].

[21] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "Overview of Cellular LPWAN Technologies for IoT Deployment: Sigfox, LoRaWAN, and NB-IoT," in *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pp. 197–202, 2018.

[22] Y. Chen, Y. A. Sambo, O. Onireti, and M. A. Imran, “A Survey on LPWAN-5G Integration: Main Challenges and Potential Solutions,” *IEEE Access*, vol. 10, pp. 32132–32149, 2022.

[23] E. M. Migabo, K. D. Djouani, and A. M. Kurien, “The Narrowband Internet of Things (NB-IoT) Resources Management Performance State of Art, Challenges, and Opportunities,” *IEEE Access*, vol. 8, 2020.

[24] “Lte.” `https://www.3gpp.org/technologies/keywords-acronyms/98-lte`. [Online; Accessed on 20-January-2022].

[25] J. Haukland, *Modelling the Energy Consumption of NB-IoT Transmissions*. Master’s Thesis, Norwegian University of Science and Technology, 2019.

[26] Ericsson, “Enhanced 4G LTE coverage for Machine-Type Communications and Internet of Things.” `https://www.ericsson.com/en/blog/2016/2/cellular-iot-alphabet-soup`. [Online; Accessed on 12-September-2022].

[27] R. Ratasuk, N. Mangalvedhe, Z. Xiong, M. Robert, and D. Bhatoolaul, “Enhancements of narrowband IoT in 3GPP Rel-14 and Rel-15,” *2017 IEEE Conference on Standards for Communications and Networking, CSCN 2017*, pp. 60–65, 10 2017.

[28] L. Bao, L. Wei, C. Jiang, W. Miao, B. Guo, W. Li, X. Cheng, R. Liu, and J. Zou, “Coverage Analysis on NB-IoT and LoRa in Power Wireless Private Network,” *Procedia Computer Science*, vol. 131, pp. 1032–1038, 1 2018.

[29] “Power Consumption Analysis of NB-IoT Technology for Low-Power Aircraft Applications,” *IEEE 5th World Forum on Internet of Things, WF-IoT 2019 - Conference Proceedings*, pp. 719–723, 4 2019.

[30] M. E. Soussi, P. Zand, F. Pasveer, and G. Dolmans, “Evaluating the Performance of eMTC and NB-IoT for Smart City Applications,” *IEEE International Conference on Communications*, vol. 2018-May, 7 2018.

[31] X. Cao, J. Chen, Y. Zhang, and Y. Sun, “Development of an integrated wireless sensor network micro-environmental monitoring system,” *ISA Transactions*, vol. 47, pp. 247–255, 7 2008.

[32] W. Manatarinat, S. Poomrittigul, and P. Tantatsanawong, “Narrowband-internet of things (NB-IoT) system for elderly healthcare services,” *Proceeding - 5th International Conference on Engineering, Applied Sciences and Technology, ICEAST 2019*, 7 2019.

[33] S. K. Routray, D. Gopal, A. Javali, and A. Sahoo, “Narrowband IoT (NBIoT) Assisted Smart Grids,” *Proceedings - International Conference on Artificial Intelligence and Smart Systems, ICAIS 2021*, pp. 1454–1458, 3 2021.

[34] S. K. Routray, D. Gopal, A. Pallekonda, A. Javali, and S. Kokkirigadda, “Measurement, Control and Monitoring in Smart Grids using NBIoT,” *Proceedings of the 6th International Conference on Inventive Computation Technologies, ICICT 2021*, pp. 229–234, 1 2021.

[35] Ericsson, “What is esim? learn more about esim technology.” `https://www.ericsson.com/en/esim`, url=https://www.ericsson.com/en/esim.

[36] Libelium, “What is an esim and why should your IOT project care about it?.” `https://www.libelium.com/libeliumworld/what-is-an-esim-and-why-should-your-iot-project-care-about-it`. [Online; Accessed on 12-January-2022].

[37] O. Apilo, P. Karhula, and J. Makela, “eSIM-Based Inter-Operator Mobility for Advanced Smart Products,” *IEEE Internet of Things Magazine*, vol. 5, pp. 120–126, 6 2022.

[38] C. Silva, J. P. Barraca, and R. Aguiar, “ESIM suitability for 5G and B5G enabled IoT verticals,” *Proceedings - 2021 International Conference on Future Internet of Things and Cloud, FiCloud 2021*, pp. 210–216, 8 2021.

[39] C. Gaber and P. Kaluza, “eSIM Adoption: Essential Challenges On Responsibilities Repartition,” *2022 1st International Conference on 6G Networking, 6GNet 2022*, 2022.

[40] A. K. Sultania, C. Blondia, and J. Famaey, “Optimizing the Energy-Latency Tradeoff in NB-IoT with PSM and eDRX,” *IEEE Internet of Things Journal*, vol. 8, pp. 12436–12454, 2021.

[41] I. Qualcomm Technologies, “Leading the LTE IoT evolution to connect the massive Internet of Things.” `https://www.qualcomm.com/media/documents/files/leading-the-lte-iot-evolution-to-connect-the-massive-internet-of-things.pdf`. [Online; Accessed on 10-May-2022].

[42] “How to enable power saving modes of nb-iot and cat-m and the energy consumption expected.” `https://www.digikey.com/en/articles/how-to-enable-power-saving-modes-of-nb-iot-and-cat-m`, 2020. [Online; Accessed on 15-January-2022].

[43] B. Buurman, J. Kamruzzaman, G. Karmakar, and S. Islam, “Low-Power Wide-Area Networks: Design Goals, Architecture, Suitability to Use Cases and Research Challenges,” *IEEE Access*, vol. 8, pp. 17179–17220, 2020.

[44] J. Eliasson, *Low-Power Design Methodologies for Embedded Internet Systems*. PhD thesis, Luleå University of Technology, 2008.

[45] M. Pedram, “Power optimization and management in embedded systems,” *Proceedings of the Asia and South Pacific Design Automation Conference, ASP-DAC*, vol. 2001-January, pp. 239–244, 2001.

[46] S. Akui, K. Seno, M. Nakai, T. Meguro, T. Seki, T. Kondo, A. Hashiguchi, H. Kawahara, K. Kumano, and M. Shimura, “Dynamic voltage and frequency management for a low-power embedded microprocessor,” in *2004 IEEE International Solid-State Circuits Conference (IEEE Cat. No.04CH37519)*, pp. 64–513 Vol.1, 2004.

[47] M. Ibro and G. Marinova, “DVFS Technique on a Zynq SoC-based System for Low Power Consumption,” in *2020 International Conference on Broadband Communications for Next Generation Networks and Multimedia Applications (CoBCom)*, pp. 1–5, 2020.

[48] R. Bhakthavatchalu, S. S. Mallia, R. HariKrishnan, A. Krishnan, and B. Sruthi, “Low power scheduled alarm system using embedded microcontroller with USB interface,” *2011 International Conference on Emerging Trends in Electrical and Computer Technology, ICETECT 2011*, pp. 610–615, 2011.

[49] J. Borges, *Robust Software Services for IoT Embedded Systems*. Master’s Thesis, Minho University, 2021.

[50] ST, “B-L462E-CELL1 Discovery Kit.” `https://www.st.com/en/evaluation-tools/b-l462e-cell1.html`. [Online; Accessed on 05-March-2022].

[51] Murata, “Type1se.” `https://www.murata.com/en-global/products/connectivitymodule/lpwa/overview/lineup/type-1se`. [Online; Accessed on 18-September-2022].

[52] ST, “Discovery kit for LTE Cat M/NBIoT with STM32L4 Series - User Manual,” 2021.

[53] “Murata type1se.” `https://www.google.com/search?q=murata+type1se&sxsrf=ALiCzsYDsSft-k7r-9hIg2ER5Cw6HNnQug:1665698228286&source=lnms&tbm=`

isch&sa=X&ved=2ahUKEwiFusy5md76AhUS1BoKHbPIAw0Q_AUoAnoECAEQBA&biw=1536&bih=722&dpr=1.25#imgrc=E-QmlKyr2IFAWM. [Online; Accessed on 3-May-2022].

[54] "STM32L462RE, Ultra-low-power Arm® Cortex®-M4 32-bit MCU+FPU, 100DMIPS, 512KB Flash, 160KB SRAM, analog, audio, AES." https://www.st.com/en/microcontrollers-microprocessors/stm32l462re.html. [Online; Accessed on 18-September-2022].

[55] ST, "STM32L462CE STM32L462RE STM32L462VE Datasheet: Ultra-low-power Arm Cortex-M4 32-bit MCU+FPU, 100DMIPS, 512KB Flash, 160KB SRAM, analog, audio, AES," 2020.

[56] I. Analytics, "Type1SC Hardware Design Guidelines." https://fccid.io/HSW-TY1SC/User-Manual/Manual-4340216.pdf. [Online; Accessed on 18-December-2022].

[57] AVAGO, "APDS-9306/APDS-9306-065 Miniature Surface-Mount Digital Ambient Light Sensor."

[58] T. Instruments, "HDC2080 Ultra-low-power digital humidity and temperature sensor," 2021.

[59] Bosch, "Acceleration sensor BMA400 sensor," 2021.

[60] ROHM, "BR24Gxxx-3A (128K 256K 1M)."

[61] T. Instruments, "Tps7a0218pdbvr, 200-ma, nanopower-iq (25 na), low-dropout (ldo) voltage regulator with enable," 2021.

[62] Saft, "Saft LM17500 Battery," 2021.

[63] Keil, "µVision IDE." https://www2.keil.com/mdk5/uvision/. [Online; Accessed on 06-June-2022].

[64] ST, "ST-LINK/V2 - ST-LINK/V2 in-circuit debugger/programmer for STM8 and STM32 - STMicroelectronics." https://www.st.com/en/development-tools/st-link-v2.html. [Online; Accessed on 06-June-2022].

[65] ST, "STM32CubeMX - STM32Cube initialization code generator - STMicroelectronics." https://www.st.com/en/development-tools/stm32cubemx.html. [Online; Accessed on 06-June-2022].

[66] "O que é Azure RTOS ThreadX? | Microsoft Docs." https://docs.microsoft.com/pt-pt/azure/rtos/threadx/. [Online; Accessed on 06-June-2022].

[67] "MATLAB - MathWorks - MATLAB Simulink." https://www.mathworks.com/products/matlab.html. [Online; Accessed on 06-June-2022].

[68] Altium, "Altium Designer." https://www.altium.com/altium-designer/. [Online; Accessed on 20-July-2022].

[69] "Altium Designer Loader Download." https://componentsearchengine.com/library/altium?gclid=CjwKCAjwkaSaBhA4EiwALBgQaIny6BXUGDu7BCWgqPk0afjUf-2YIUDoo_aTg5RDhjOC6UHaQZcsRxoCvNIQAvD_BwE. [Online; Accessed on 6-June-2022].

[70] "Lbad0zz1se datasheet." https://fcc.report/FCC-ID/HSW-TY1SCDM/5068107.pdf. [Online; Accessed on 27-August-2022].

[71] ST, "B-L462E-CELL1 Discovery." https://www.st.com/content/ccc/resource/technical/layouts_and_diagrams/schematic_pack/group1/a7/4e/96/96/6a/7d/41/2a/MB1508-L462RE-C03_Schematic/files/MB1508-L462RE-C03_Schematic.pdf/jcr:content/translations/en.MB1508-L462RE-C03_Schematic.pdf. [Online; Accessed on 06-June-2022].

[72] Altium, "Altium WP PCB: Polygon Or Plane, Which Is Better? | Altium." https://resources.altium.com/p/polygon-or-plane-which-is-better. [Online; Accessed on 20-July-2022].

[73] Altium, "Adding Via Stitching Via Shielding to a PCB in Altium Designer | Altium Designer 22 User Manual | Documentation." https://www.altium.com/documentation/altium-designer/via-stitching-via-shielding-pcb. [Online; Accessed on 20-July-2022].

[74] "Chapter 3 - Functional Components of Azure RTOS ThreadX | Microsoft Learn." https://learn.microsoft.com/en-us/azure/rtos/threadx/chapter3. [Online; Accessed on 18-July-2022].

[75] Azure, "Threadx." https://github.com/azure-rtos/threadx. [Online; Accessed on 06-June-2022].

[76] "Revision History Type 1SC/ Type 1SC-DM AT Commands Reference Guide." www.murata.com. [Online; Accessed on 27-August-2022].

# Appendix A: Schematics

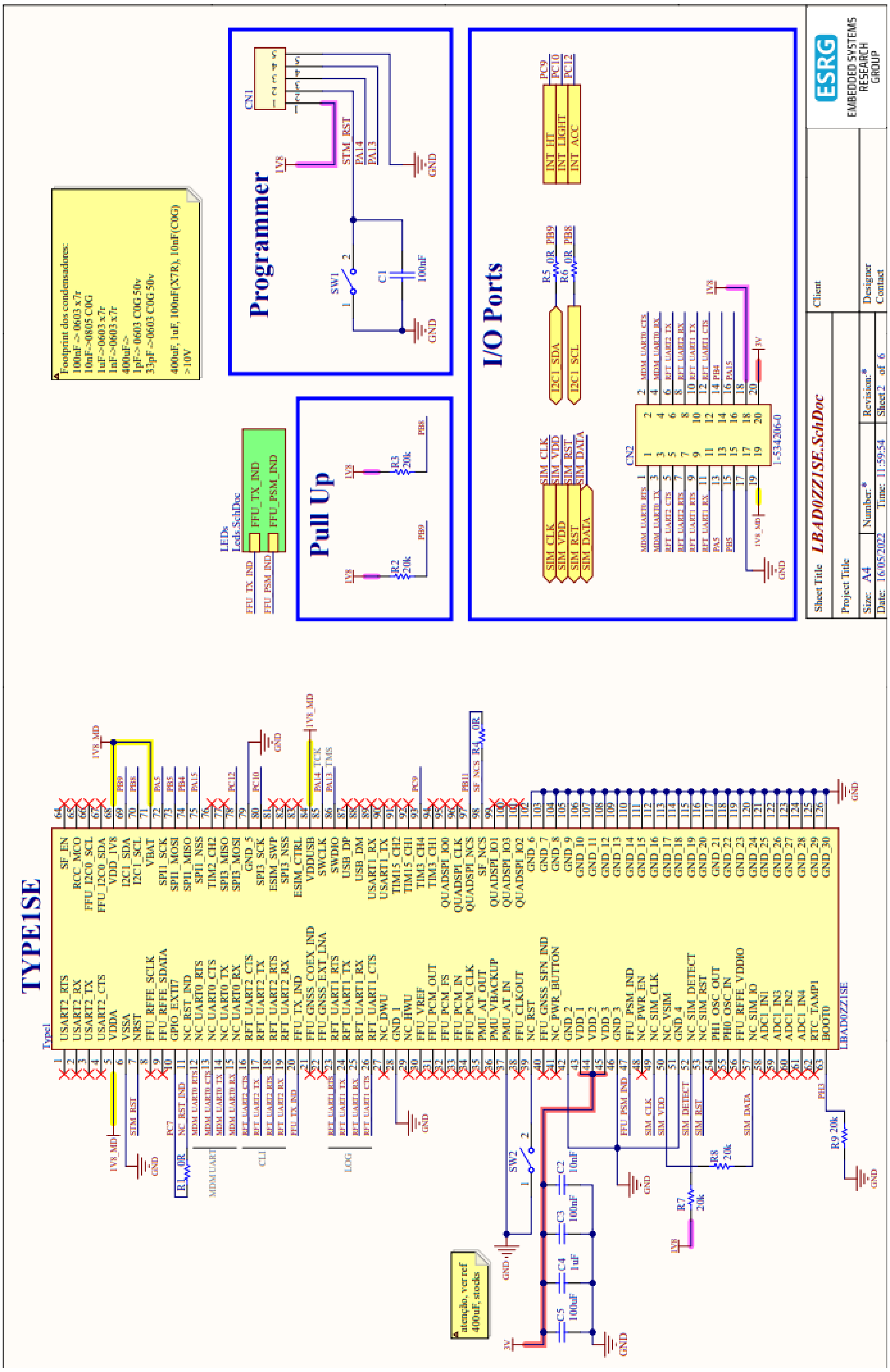


Figure A.1: Custom Board Schematics: Sheet 2.

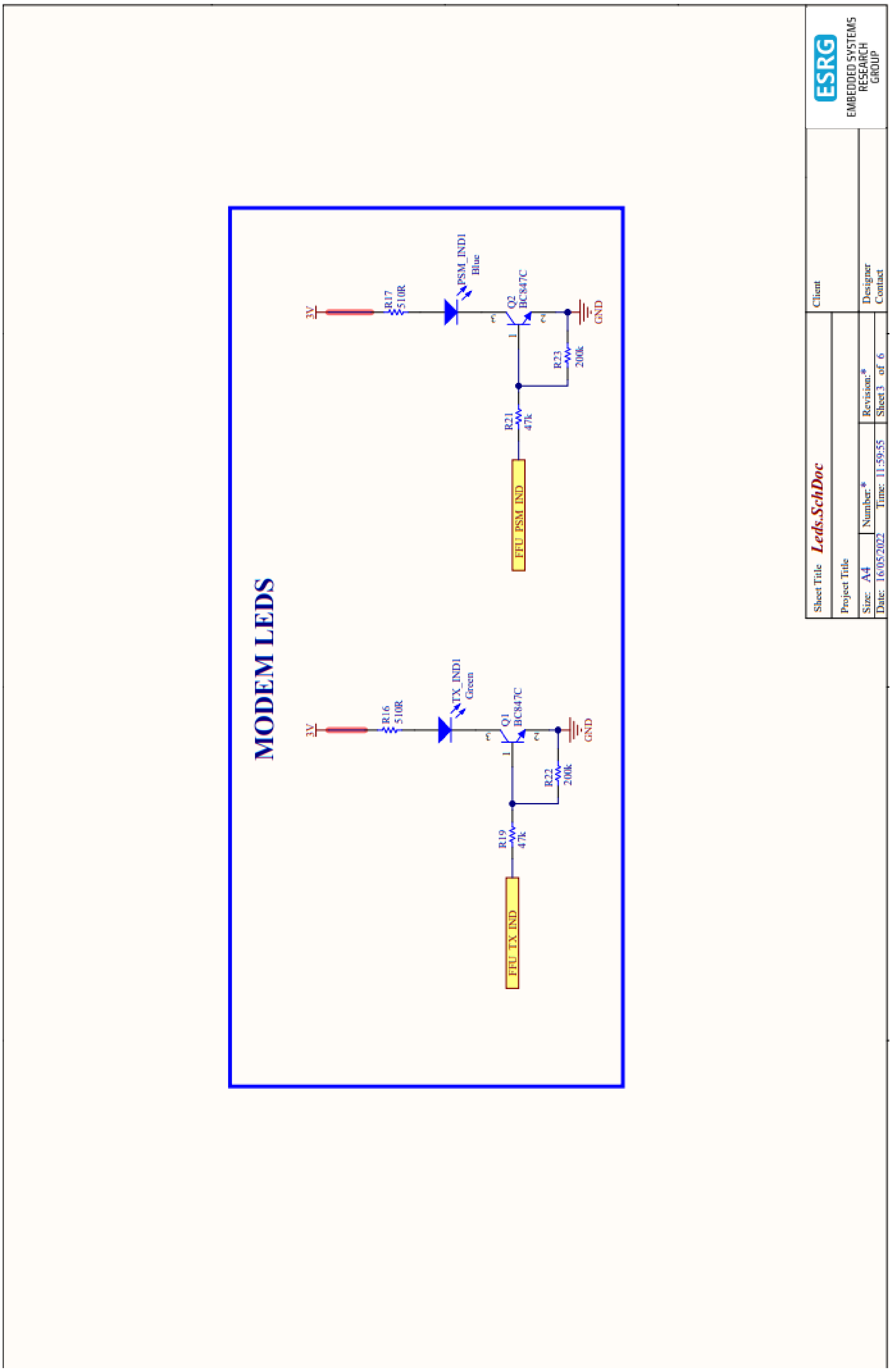


Figure A.2: Custom Board Schematics: Sheet 3.

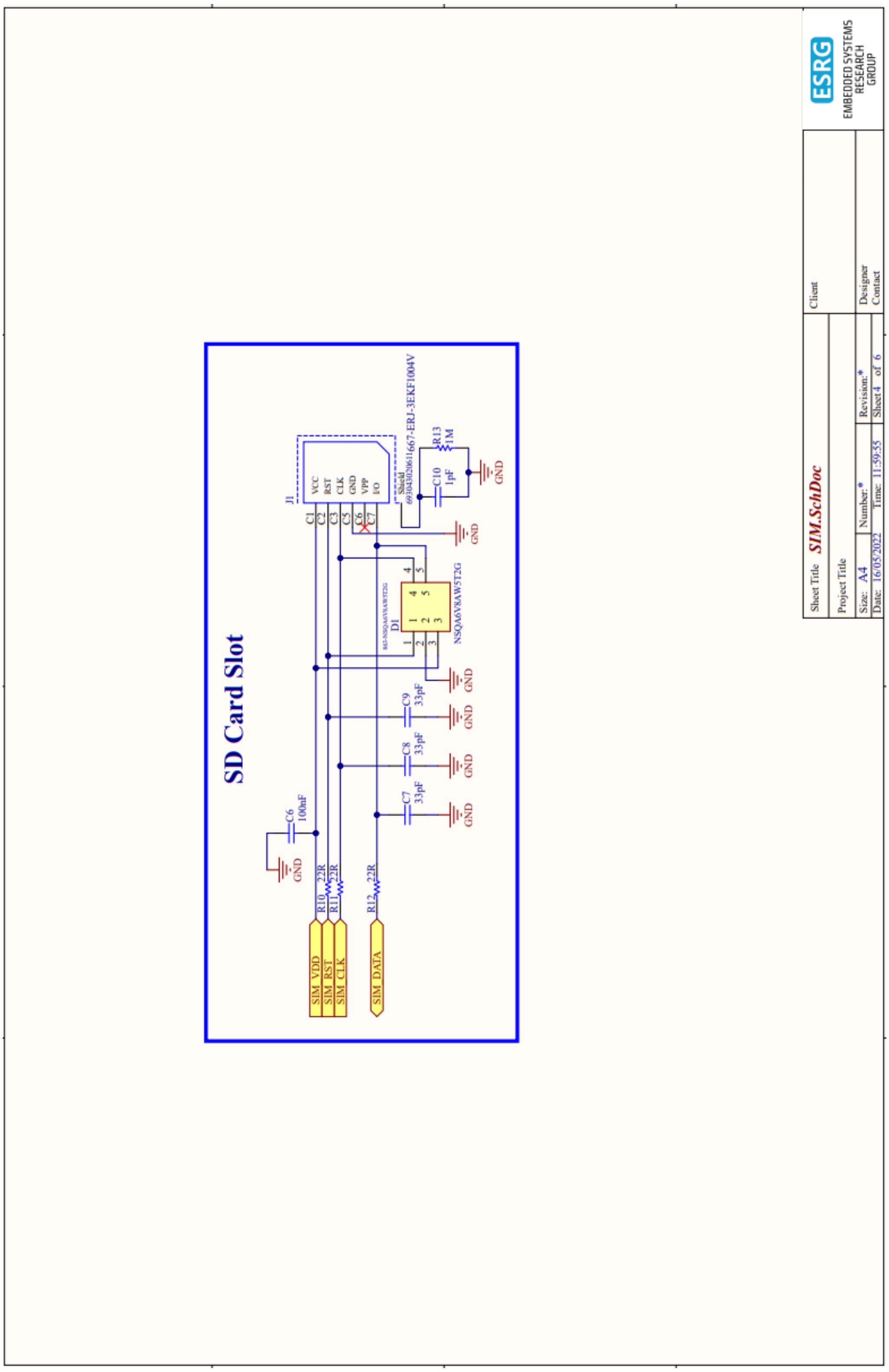


Figure A.3: Custom Board Schematics: Sheet 4.

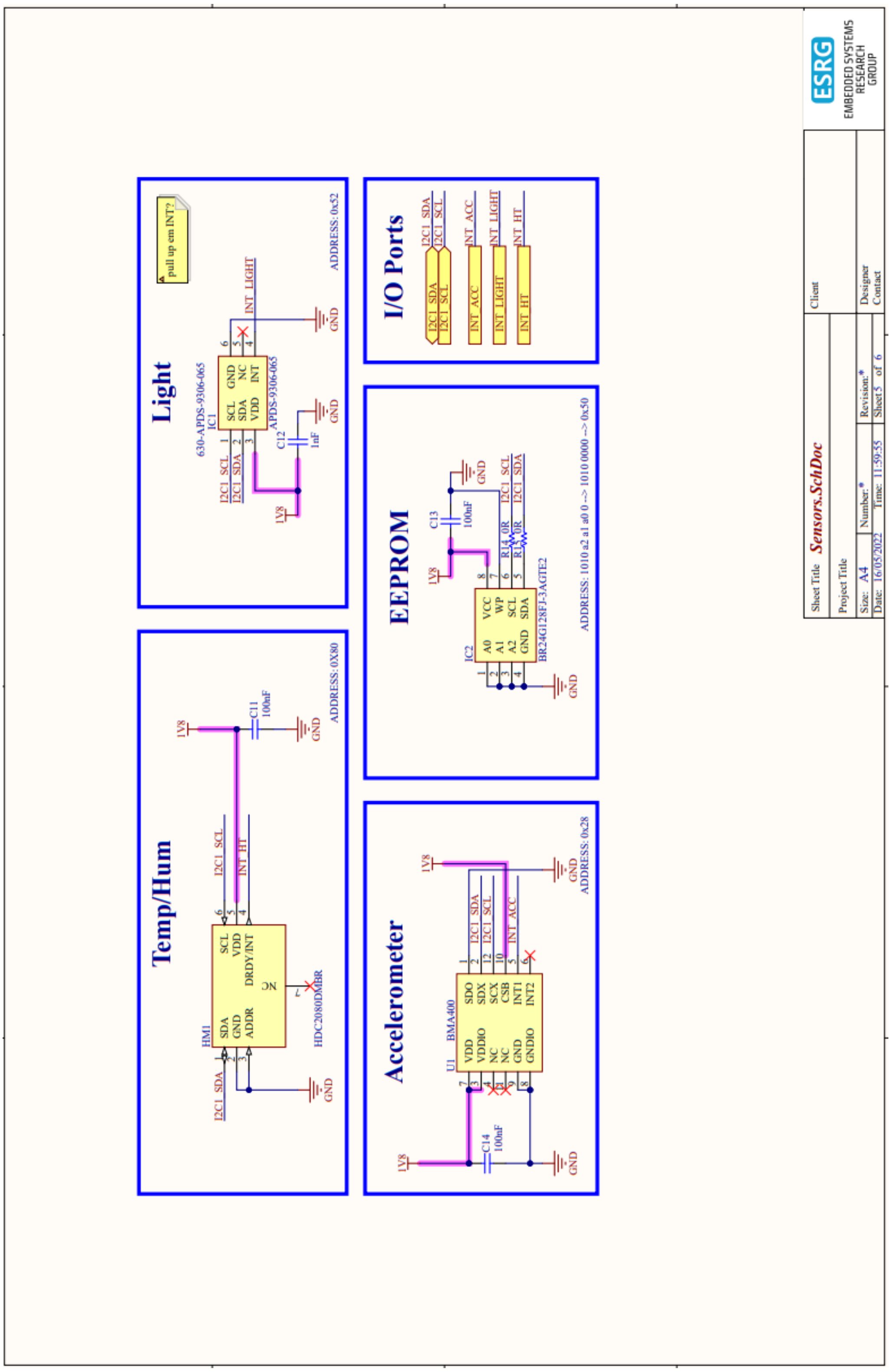


Figure A.4: Custom Board Schematics: Sheet 5.

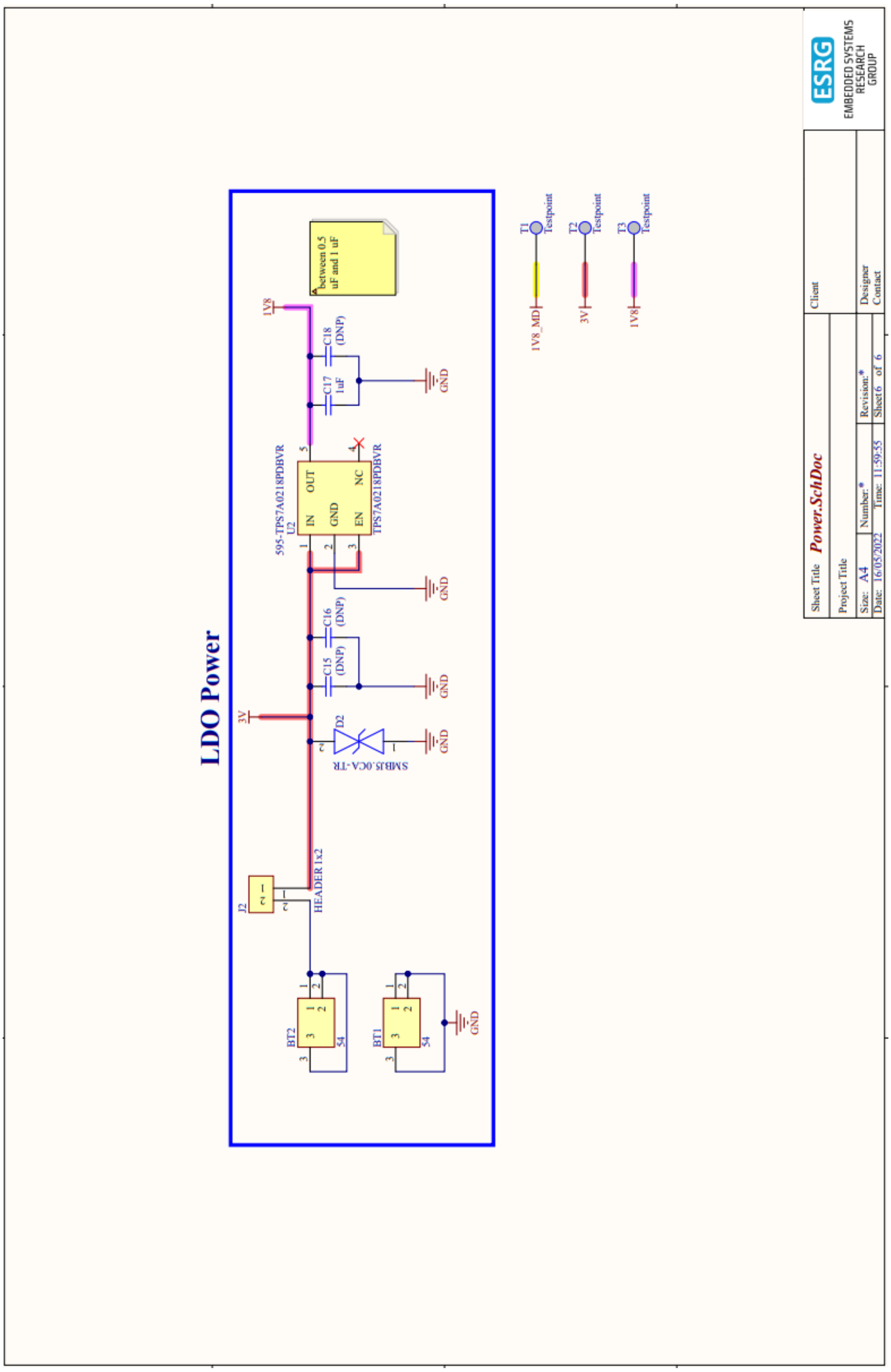
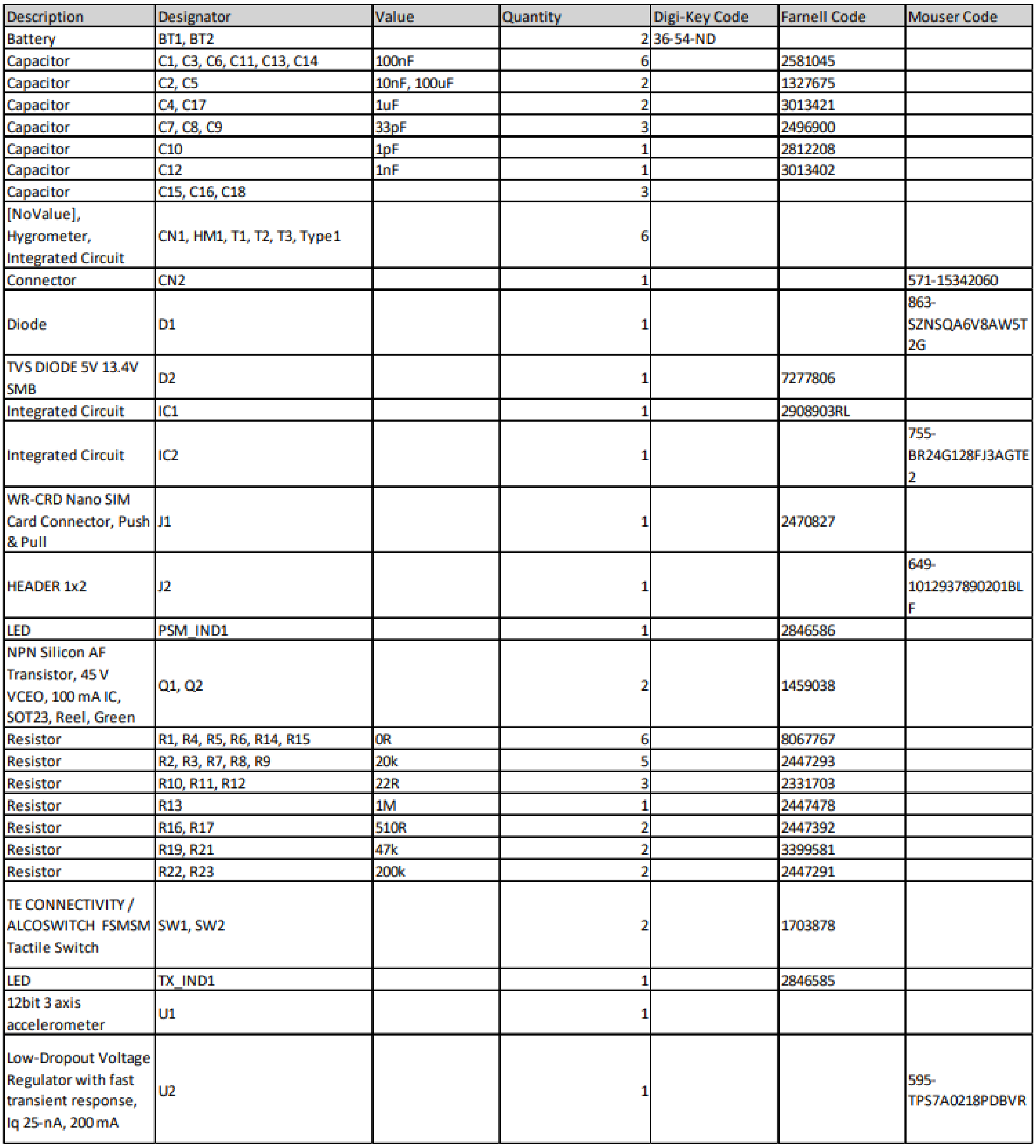


Figure A.5: Custom Board Schematics: Sheet 6.

# Appendix B: BOM



| Description | Designator | Value | Quantity | Digi-Key Code | Farnell Code | Mouser Code |
|---|---|---|---|---|---|---|
| Battery | BT1, BT2 |  | 2 | 36-54-ND |  |  |
| Capacitor | C1, C3, C6, C11, C13, C14 | 100nF | 6 |  | 2581045 |  |
| Capacitor | C2, C5 | 10nF, 100uF | 2 |  | 1327675 |  |
| Capacitor | C4, C17 | 1uF | 2 |  | 3013421 |  |
| Capacitor | C7, C8, C9 | 33pF | 3 |  | 2496900 |  |
| Capacitor | C10 | 1pF | 1 |  | 2812208 |  |
| Capacitor | C12 | 1nF | 1 |  | 3013402 |  |
| Capacitor | C15, C16, C18 |  | 3 |  |  |  |
| [NoValue], Hygrometer, Integrated Circuit | CN1, HM1, T1, T2, T3, Type1 |  | 6 |  |  |  |
| Connector | CN2 |  | 1 |  |  | 571-15342060 |
| Diode | D1 |  | 1 |  |  | 863-SZNSQA6V8AW5T2G |
| TVS DIODE 5V 13.4V SMB | D2 |  | 1 |  | 7277806 |  |
| Integrated Circuit | IC1 |  | 1 |  | 2908903RL |  |
| Integrated Circuit | IC2 |  | 1 |  |  | 755-BR24G128FJ3AGTE2 |
| WR-CRD Nano SIM Card Connector, Push & Pull | J1 |  | 1 |  | 2470827 |  |
| HEADER 1x2 | J2 |  | 1 |  |  | 649-1012937890201BLF |
| LED | PSM_IND1 |  | 1 |  | 2846586 |  |
| NPN Silicon AF Transistor, 45 V VCEO, 100 mA IC, SOT23, Reel, Green | Q1, Q2 |  | 2 |  | 1459038 |  |
| Resistor | R1, R4, R5, R6, R14, R15 | 0R | 6 |  | 8067767 |  |
| Resistor | R2, R3, R7, R8, R9 | 20k | 5 |  | 2447293 |  |
| Resistor | R10, R11, R12 | 22R | 3 |  | 2331703 |  |
| Resistor | R13 | 1M | 1 |  | 2447478 |  |
| Resistor | R16, R17 | 510R | 2 |  | 2447392 |  |
| Resistor | R19, R21 | 47k | 2 |  | 3399581 |  |
| Resistor | R22, R23 | 200k | 2 |  | 2447291 |  |
| TE CONNECTIVITY / ALCOSWITCH FSMSM Tactile Switch | SW1, SW2 |  | 2 |  | 1703878 |  |
| LED | TX_IND1 |  | 1 |  | 2846585 |  |
| 12bit 3 axis accelerometer | U1 |  | 1 |  |  |  |
| Low-Dropout Voltage Regulator with fast transient response, Iq 25-nA, 200 mA | U2 |  | 1 |  |  | 595-TPS7A0218PDBVR |

Figure B.1: Bill of Materials.