POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Performance analysis of Bluetooth Low Energy Networks for Wireless Sensors

Tesi di Laurea Magistrale in
Automation and Control Engineering -
Ingegneria Dell'automazione Industriale

Author: **Elia Epis**

Student ID: 871634
Advisor: Prof. Marco Mauri
Academic Year: 2022-23


Cosa hai da dire di tutti questi anni?
...

Abstract

Nowadays rail transport is gradually gaining more and more relevance in freight and passenger transport thanks to its low environmental impact. However one of its major drawback is that even though railway transport cover 18% of total freight traffic in the European Union, it is not as technologically advanced as passenger transport [3]. This type of transport is two to four times more efficient than conventional trucks and its use avoids delays caused by traffic jams and accidents, it is experiencing a growing trend and foreseeing no sign of regression. So we decided to further improve and optimize them.

This is why a real-time monitoring system able to perform condition-based maintenance was studied. To also eliminate cables encumbrance for an easier integration a solar-powered wireless sensor node board was developed. However In days where faster response and wider data variety are continuously pursued, the need for a fast, reliable and low power consumption wireless protocol is extremely important.

This master thesis work aims to analyzes the performances of the BLE communication protocol while transmitting with two placed sensor nodes. We will also discuss its architecture and functionality with the aim of improving its performance, reliability and power consumption.

The proposed system is composed of two nRF-52840 transceivers, with them simulating transmission and reception: transmission's job is to continuously collect pressure values and send them; on the other hand the receiver is in charge of requesting data and storing it to perform condition analysis.

BLE protocol has been tested using Power Profile Kit II, nRF52840 BLE Sniffer and Wireshark with 1 m distance between devices to measure power consumption, throughput and compare them with theoretical ones.

A study of devices' positioning and BLE wave propagations has also been carried out to understand the best feasible positions.

Abstract in lingua italiana

Al giorno d'oggi il trasporto ferroviario sta gradualmente assumendo sempre più importanza nel trasporto merci e passeggeri grazie al suo basso impatto ambientale. Tuttavia, uno dei suoi principali svantaggi è che, sebbene copra il 18% del traffico merci totale nell'Unione Europea, non è tecnologicamente avanzato come il trasporto passeggeri [3]. Questo tipo di trasporto è da due a quattro volte più efficiente dei camion convenzionali e il suo utilizzo evita i ritardi causati da ingorghi e incidenti, sta vivendo una tendenza in crescita e non prevede alcun segno di regressione. Abbiamo quindi deciso di proporre un modo per migliorarlo e ottimizzarlo ulteriormente.

Per questo motivo è stato studiato un sistema di monitoraggio in tempo reale in grado di eseguire una manutenzione basata sulle condizioni. Per eliminare l'ingombro dei cavi e facilitare l'integrazione, è stata sviluppata una scheda wireless alimentata a energia solare. Tuttavia, in un'epoca in cui si cerca di ottenere risposte più rapide e una maggiore varietà di dati, la necessità di un protocollo veloce, affidabile e a basso consumo energetico è prioritaria.

Questo lavoro di tesi si propone di analizzare le prestazioni del protocollo di comunicazione BLE durante la trasmissione tra due nodi. Discuteremo anche l'architettura e le funzionalità del protocollo con l'obiettivo di migliorarne le prestazioni, l'affidabilità e il consumo energetico.

Il sistema proposto è composto da due ricetrasmettitori nRF-52840, che simulano la trasmissione e la ricezione: il compito della trasmissione è quello di raccogliere continuamente i valori di pressione e inviarli; il ricevitore, invece, ha il compito di richiedere i dati e memorizzarli per eseguire l'analisi delle condizioni.

Il protocollo BLE è stato testato utilizzando Power Profile Kit II, nRF52840 BLE Sniffer e Wireshark con una distanza di 1 m tra i dispositivi per misurare il consumo energetico e il data-rate e confrontarli con quelli teorici.

È stato inoltre effettuato uno studio del posizionamento dei dispositivi e della propagazione delle onde BLE per capire quali siano le migliori soluzioni possibili.

Introduction

This thesis aims is to study the feasibility of wireless communication in a freight train's brake pressure data acquisition and monitoring system based on Condition Based Maintenance. BLE protocol has been implemented and reviewed, then the total performance is evaluated taking into account various parameters, such as energy consumption, efficiency and throughput. Last but not least wave propagation is studied for a path loss characterization and sensors' placement analysis.

In Chapter 1, *State of art*, a brief overview of the already existing technology is shown.

In Chapter 2, *Introduction to Bluetooth*, the BLE protocol and its performances are introduced.

In Chapter 3, *Materials* , the hardware and software used are listed.

In Chapter 4, *Firmware*, firmware written and implemented on boards is detailed.

In Chapter 5, *Results*, results for both power consumption and transmission characterization are listed.

Contents

Abstract **iii**

Abstract in lingua italiana **v**

Introduction **vii**

Contents **ix**

1 State of the Art **1**
1.1 Wireless Sensor Network . . . 3
1.1.1 Wireless Protocol . . . 4
1.2 Application . . . 6
1.3 Hardware . . . 8
1.3.1 Micro-controller . . . 9
1.3.2 Sensor . . . 9
1.3.3 Communication Module . . . 11
1.3.4 Power Section . . . 12
1.4 Software . . . 14
1.5 Experimental Set-Up Description and Results . . . 16
1.5.1 Results . . . 18

2 Introduction to Bluetooth and BLE **23**
2.1 Specifications . . . 24
2.2 Network Types . . . 25
2.3 Architecture and Protocol Stack . . . 26
2.3.1 Physical Layer . . . 27
2.3.2 Link Layer . . . 28
2.3.3 Host Controller Interface . . . 29
2.3.4 L2CAP . . . 29
2.3.5 Security Manager . . . 29
2.3.6 Generic Attribute Profile . . . 30
2.3.7 Attribute Protocol . . . 30
2.3.8 Generic Access Profile . . . 32
2.4 Timestamp . . . 33
2.4.1 Advertisement . . . 33
2.4.2 Transmission . . . 35
2.5 Performances . . . 35

2.5.1 Range . . . 36
2.5.2 Throughput . . . 37
2.5.3 Energy consumption . . . 38

3 Test Kit Introduction **39**
3.1 nRF52840-DK . . . 40
3.2 SoftDevice . . . 41
3.2.1 Event Handlers . . . 42
3.2.2 Nordic Uart Service . . . 42
3.3 Power Profiler Kit II . . . 43
3.4 Wireless InSite . . . 44

4 Firmware **47**
4.1 Transmitter . . . 47
4.1.1 Configuration . . . 48
4.2 Receiver . . . 50
4.2.1 Configuration . . . 51
4.3 Command Analysis . . . 52
4.4 Communication Process . . . 53
4.5 Wireless InSite . . . 54

5 Results **59**
5.1 Power Consumption and Throughput . . . 59
5.1.1 Advertisement . . . 59
5.1.2 Connection . . . 63
5.1.3 Transmission . . . 64
5.2 Propagation Simulation . . . 71

6 Conclusion and future developments **79**

Bibliography **81**

List of Figures **83**

List of Tables **87**

Ringraziamenti **89**

1 | State of the Art

In the recent years the need to further increase safety and efficiency obliged train manufacturers to equip passenger trains with tons of sensors allowing them to constantly monitor and control vehicle's conditions and security.
Knowledge of system's condition can prevent accident, component's fault and reduce maintenance cost and downtime avoiding unnecessary operations.
With the objective of efficiency becoming higher and higher the used approaches were becoming older and awful. Relative to every application two of the fundamental aspects are maintenance and security.
Until today systems' conditions checks and maintenance were calendar based occurring on a fixed schedule forcing downtime also in case they were not needed.
With this optic a new philosophy called Condition Based Maintenance (CBM) is slowly gaining more and more interest, its advantage is that it bases decisions to repair or replace assets depending on their current and/or future predicted conditions (fig.1). More specifically with visual inspection, scheduled tests and sensor measurement the equipment performance is monitored to determine the most cost-efficient and secure time to perform maintenance.

Nowadays different systems implement this technology: vehicles, infrastractures, power

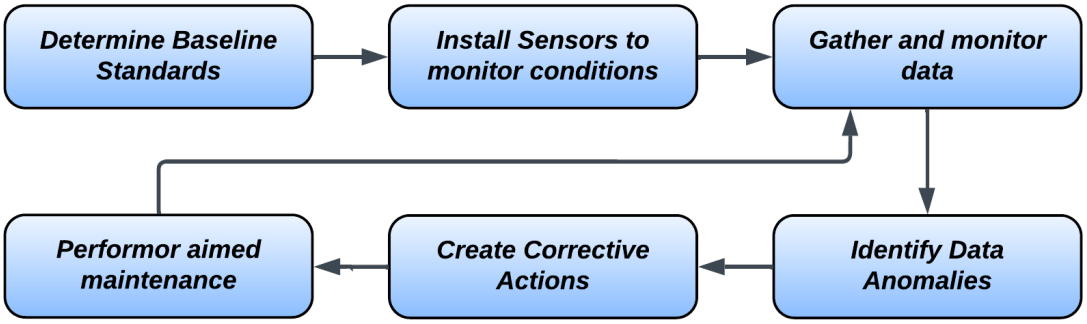


Figure 1: Condition Based Maintenance Workflow

pylons and much more. The hardest challenge is that every single application is obviously susceptible to different solicitations with respect to the other ones so different sensors must be used for every case study.
Regarding train condition's monitoring two different approaches can be implemented: on-board or wayside monitoring. The second one is not feasible since it measures forces transmitted to the rail when a train is passing but obviously requires a huge number of devices over the entire line. The other solution is then clearly the best and cheapest one.
In this chapter we first discuss the applications on which we could have worked, the chosen one and later the sensor board developed that will be the focus of this thesis work.
Moving on we deeply analyze the hardware mounted: a variety of sensors, a Bluetooth communication module, a Microprocessor and an energy harvester.
In the end we will have a closer look to the communication part, talking about the flow of the program and the working method.
There are different types of condition monitoring systems: Vibration Analysis, Infrared Thermography, Ultrasonic Analysis, Oil Analysis, Electrical Analysis, Pressure Analysis. We focused on vibration or pressure monitoring.
The first one is called Structural Health Monitoring (SHM) and used for infrastructure like bridges, wind turbines, transmission towers. It can help detect problems like imbalance, bearing failure, mechanical looseness, resonance, bent shafts and more [4].
Regarding this type of structure low frequency vibrations (0-40Hz) [19] are typical of first modes natural frequencies and directly connected to structure damage and problematics [17].
While the second type can be implemented in vehicles like freight train since they are still not efficient such as passengers' ones and recurrent failures with a high impact on reliability and service are related to the braking system in which, if a malfunction occurs, it results in complexity in finding the cause of the fault within the system.
Luckily there aren't much differences between every application, so if a basic standard board then we can just change the sensor used to gather information. Specifically if we want to monitor vibration we mount an accelerometer, if we seek pressure variations we use a pressure sensor and so on...
However the biggest common problems regard the impossibility of implementing a wired connection and the necessity to reassemble the convoy at the beginning of every trip. That's why we developed a battery powered wireless sensor board, equipped every sensor node with a renewable source of energy and implemented a wireless transmission protocol with a particular focus on low-power consumption.
There are a variety of different energy source (solar, thermal, wind...) but not all of them are achievable.

Between the feasible ones many types of energy sources have been tested, but among them all solar energy has been chosen as the most reliable and widely available source for its abundant presence in all outdoor applications and because it allows placing the sensor in any position on the wagon. This choice is also justified by the fact that previous studies has demonstrated that solar source is the most affordable energy source with respect to mechanical vibrations and/or wind energy generated by the motion of the vehicle.[2, 10]

1.1. Wireless Sensor Network

A wireless network refers to devices that performs communication between nodes in the Radio Frequency (RF). Thanks to their technology and, as the name tells, the capacity to communicate without the need of a wired connection their are becoming increasingly popular in homes, businesses and telecommunication making information easily shareable. In contrast to wired technologies, wireless ones make it possible to create ad-hoc low cost networks, in part because of the modest installation work required, providing good system scalability and easy reconfiguration.
Sensor networks that normally consists of sensors such as seismic, magnetic, thermal, pressure, visual, acoustic and much more to monitor a wide variety of ambient conditions. They can be used for sensing and controlling actuators.
When sensors implement wireless technology they create a **Wireless Sensor Network** (WSN) leading to the pros and cons with respect to wired ones outlined below:

- The number of wireless sensor node can easily be of higher magnitude and lower cost than wired ones due to minor installation work.
- The topology of a wireless sensor network can change very quickly and frequently.
- Sensor nodes mainly use broadcast communication instead of point-to-point.
- Sensors can be directly placed close to the actual measured phenomena when using a wireless communication.
- Wireless sensors have limited battery life.
- Wireless sensor communication is less reliable than the wired one.

Obviously this type of sensors don't feature a wired power source because this would spoil one of their best characteristic: mobility. This lead to one of the biggest challenge when developing a wireless sensor network: the power consumption management and optimization part.

Radio Waves propagation

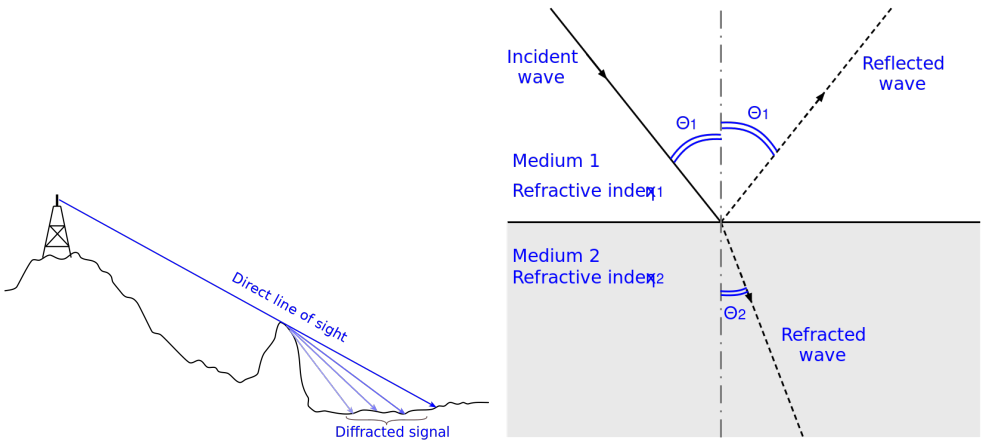
Radio frequency waves propagate through air from the transmitter to the receiver depending on the environment they work in. In each case it is mandatory to study the single situation with the goal to find the best possible and feasible Transmitter and Receiver position for lowering transmission power and receiving sensitivity. The study of radio waves propagation must take into account the environment they work in, obstacles dimensions, transmitter and receiver positions.
In our considerations we are considering the usage of radio frequency close to the GHz, in this scenario the wavelength is small compared with the geometry of usual object and the propagation of electromagnetic waves can be approximated by geometric optics, and it is believed that electromagnetic waves travel along a straight line, following the laws of geometric optics.
There are three possible ways a signal can behave when facing obstacles: *reflection*, *transmission*, and *diffraction*. They are shown in figure 2.

- When an electromagnetic wave encounters a change in medium, some or all of it may propagate into the new medium and the remainder is reflected. The part that enters the new medium is called the *transmitted* wave while the other one is *reflected*.
- The signal will *diffract* around the edge of an obstacle and start to fill the void behind it. It is found that diffraction is more pronounced when the obstacle becomes sharper and more like knife edged. The meaning knife edge depends upon the frequency, and hence the wavelength of the signal.

A large number of metal equipment in the train structure would become the main factor affecting the wireless signal. It has a shielding effect on electromagnetic waves making reflection and diffraction as obstacles for the wireless signal. Calculate the propagation paths between two points is difficult within a complex structure. However, when ignoring those arrival paths with relatively small amplitude which have little effect on signal transmission stability, the propagation path can be identified with the ray-tracing method.

1.1.1. Wireless Protocol

There are many different protocols used in wireless networks implementing a variety of technologies, some are used in long range transmission while others are specifically



(a) (b)

Figure 2: Diffraction **(a)**, reflection and transmission **(b)** example

developed for low-energy communication.
We will focus only on the second ones: in detail we talk about ZigBee, LoRa and Bluetooth Low Energy (BLE). Every single protocol has a relative maximum range, performances and power consumption. We will first analyze all of their characteristic and then show the decision process for the feasible one.

LoRa

LoRa (Long Range) is a physical wireless communication standard, implementing the chirp spread spectrum (CSS) modulation technique. LoRa defines only the physical layer above it a protocol like LoRaWan must work. LoRa uses carrier radio frequencies of 169, 433, 868 MHz (in Europe) and 915 MHz with a maximum achievable data rate of 50 kbps and a communication range that can reach 10 km.
Devices work in an asynchronous mode and transmit when there is data ready to be sent. Packets sent by a transmitter are received by one or more gateway that are in charge of re-transmitting them to a central server. The server filter for duplicated packets, implement security controls and manages the network.
This technology is fits like a glove when there isn't need for a high data rate, but the main objective is long range communication. Its biggest disadvantages are acknowledgment management.

ZigBee

Zigbee standard is a wireless protocol based on norm IEEE 802.15.4 treated by the ZigBee Alliance. it is a short range wireless communication protocol developed for low power

applications. ZigBee devices work in 868 MHz, 915 MHz and 2.4 GHz ISM frequency bands using a Direct Sequence Spread Spectrum (DSSS) modulation technique. with a maximum achievable data rate of 250 Kbps and an operating range that can reach 100 m.
It works with three different types of devices: End Devices (ED) can only sends packets to a router or a coordinator; Router (ZR) are used to extend the network coverage; Coordinator (ZC) manages the network topology and security.

Bluetooth Low Energy

Bluetooth Low Energy (BLE) is a protocol developed for short range low power wireless communication using Gaussian Frequency Shift Keying (GFSK) modulation technique. It is a completely new technology derived from Classic Bluetooth (BT). They share many features but BLE is specifically studied for low power applications and it cannot operate with Classic BT devices. BLE operates in the 2.4 GHz Industrial Scientific Medical (ISM) band with a maximum achievable data rate of 1.4 Mbps and 100m communication range.

Analyzing all the protocols listed above we found that: LoRa is normally used for long range application and it had a too much low throughput; ZigBee had problems in terms of communication range, in fact with our environment being fulfilled with metal object path loss is very high with respect to open field application and transmitters couldn't create a path reaching the gateway; with Bluetooth Low Energy being the most feasible we will only analyze this protocol inside this work thesis.

1.2. Application

From now on we are going to focus and deeply analyze just the specific freight train problem.
A recurrent failure with high impact on reliability and efficiency is related to the breaking system where also a small fault implies a damaging action on the travelling wagon and the infrastructure itself.
The breaking system plays also a major role in traffic safety and has a great influence on anomalies of other components meaning greater cost in case of breakdown.
Luckily most train models present on the market share a very similar structure where every freight wagon is connected to the locomotive with a main pipe and a weighing valve

allowing regulation of the pressure proportional to the load carried on the single bogie. It is therefore evident that monitoring the pressure trends in various crucial points allows identification of malfunctions.
One of the problem is that despite the presence of the mini photovoltaic panel, it is important to save the most energy possible due to the difficulty in obtaining a good sun exposure when the sensor is mounted on wagon, or when the solar panels are covered for any reason during travel for example in tunnels or underground sections.
In fig.3 we can appreciate the network topology of a single wagon, to mount them we used the test points already present on-board and decided to equip the weighting valve and one wheel for every bogie with a sensor communicating via Bluetooth with a Central Unit mounted on the back of the carriage. This gateway has a solar panel but since there is the possibility it is also wired to the electrical line.
In figure 10 we can see the positioning of the boards. The central control unit (schema-

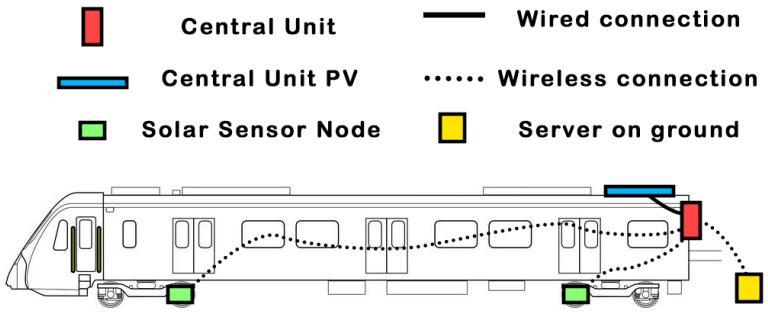


Figure 3: Scheme of wireless connection layout fot freight train

tized in fig.4) implements a 4G/LTE technology so it can transfer data to the cloud to store it. Specifically it consists in an industrial PC with a BLE dongle and, 4G/LTE modem, a GPS receiver for position information and a power management board to handle component's power supply.
In order to save energy, increase the system efficiency and limit the collected and stored data one bogie's axle is equipped with an axle-box synchronous generator that signals wagon's movement via CAN interface to the PC to start the acquisition when the train overpass 10 km/h. Additionally when the train does not move for approximately 5 minutes the gateway is switched off.
Every collection of pressures measurement is associated to a GPS track to completely understand the fleet positioning.

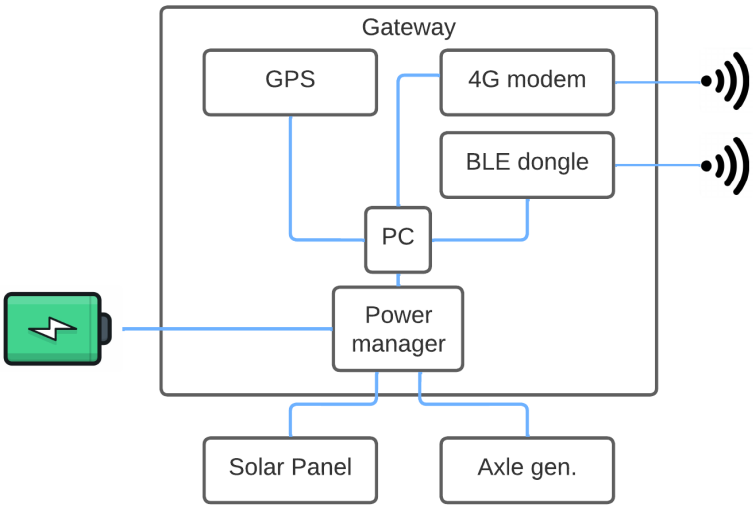
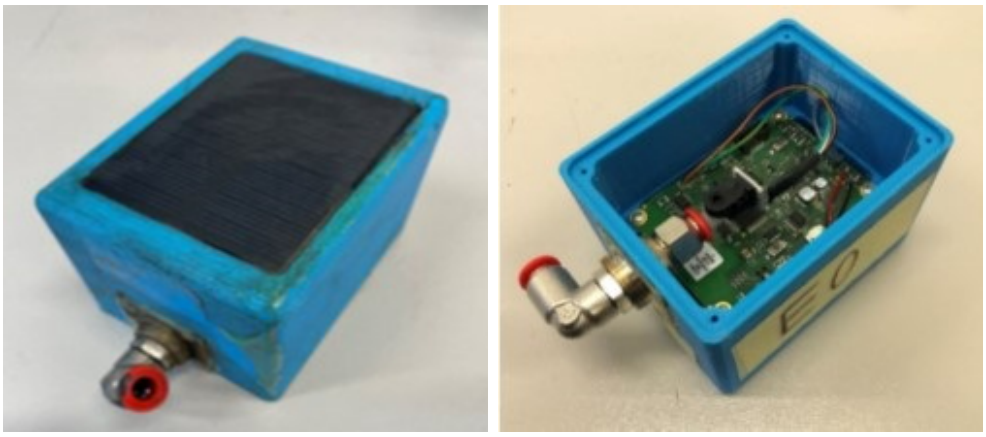


Figure 4: Gateway architecture

1.3. Hardware

The sensor nodes are basically composed of an ad-hoc developed electronic board with a Li-Po Battery mounted inside a 3d printed case [1]. The material chosen is a Chlorinated Polyetylene (CPE) that is a robust plastic material resistant to shocks and environmental agents. On top of the enclosure is placed a photovoltaic panel recharging batteries with sunlight.
A functional scheme of the board layout can be found in fig.6.
With power supply representing a critical issue the hardest challenge we faced was power management forcing a hardware and software system based on low-power consumption. This led us to choose an operating voltage of 3.3V Since most low-power processors and sensors work in this voltage range.
The main components presents are:

- Micro-controller (STM32L4Q5)
- Sensor
- Communication Module (BT840e)
- Power Harvester and Manager



(a) (b)

Figure 5: External **(a)** and internal **(b)** view of the sensor node

1.3.1. Micro-controller

To meet the power consumption and harvesting requirements the chosed processor is produced by ST-Microelectronics and part of the L4+ series created to exploit the limits of processing capabilities in the ultra-low power world. Specifically the one mounted is the STM32L4Q5 mounting a 32 bit Cortex-M4 CPU featuring a Digital Signal Processing (DSP) and a Floating Point Unit (FPU) speeding up all the on board calculation required. The choice was easily addressed to this processor due to its low power capability, in fact its current consumption can lower down to 41uA/MHz in Run Mode.

1.3.2. Sensor

The sensor present obviously depends on the application and consequently the parameters that need to be monitored. In this specific case the goal was monitor the brake system by performing pipes' and valves' pressure measurement.
To choose the perfect sensor we first had to understand the range of values that we could have inside the break system. By doing a research we found that there are federal regulations stating that the feed valve must be at least at 70 psi and most trains use a 90 psi main brake pipe pressure.
With this specifications we opted for a SSCDANN150PAAA3 that is an extremely low power sensor fully calibrated and temperature compensated for sensor offset. An important feature of this model is that it is an absolute pressure transducer.
This sensor has various version with different output interfaces: I2C, analog, SPI. However the analog with output voltage proportional to the pressure value one has been chosen because it consume 2.1mA while the serial interface version absorbs approximately 3.3

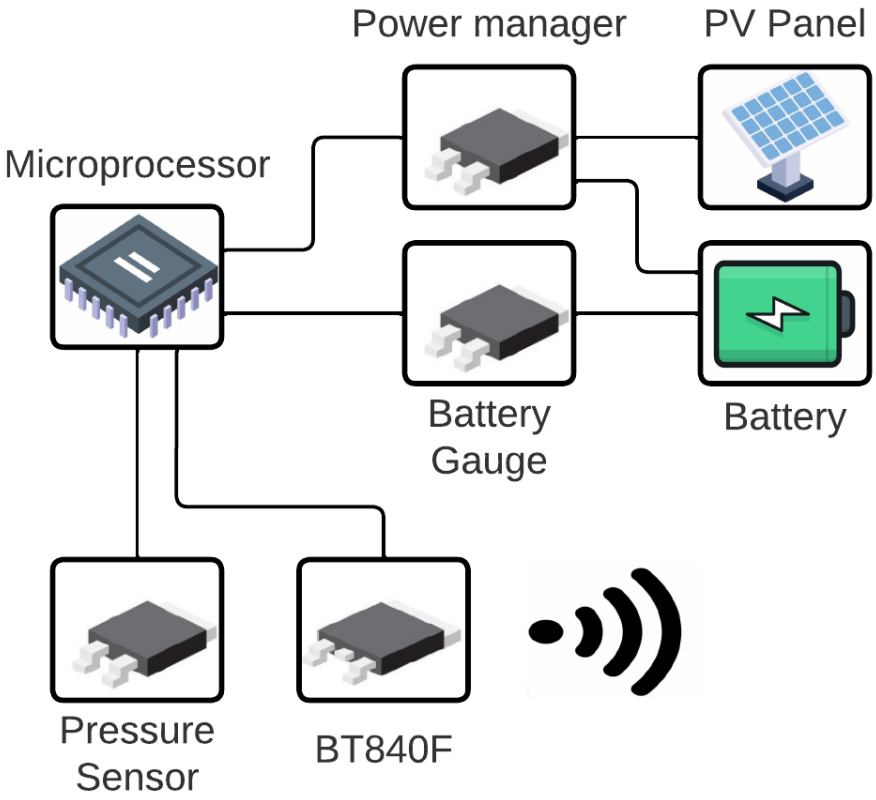


Figure 6: Hardware Architecture

mA.
Thanks to its low-power specification and fast boot time ($5ms$) we can power it through one of the μC pin so it can be supplied only when a measurement is needed and during the remaining time the current consumption of the pressure transducer is zero.

SSCDANN150 - Main Feature	
Supply Voltage	$3.3V$
Current Consumption	$2.1mA$
Measurement Range	$\pm 150psi$
Sensitivity	$0.375psi$
Temperature Range	$-40 \div 85°C$
Max ODR	$1000Hz$

Table 1: Pressuremeter main feature

1.3.3. Communication Module

As communication protocol the BLE technology has been chosen because of its good communication range, low power consumption, high data rate and optional security.
Since it would have been too much challenging and energy consuming to implement both the acquisition, processing and communication part on the same micro-controller we decide to simply divide them and use the mounted main processor to acquire and convert data and a BT840e to send the results.
This Fanstel module is perfect for our application because it is just an adaptor for Nordic nRF52840 providing a UART interface and, unlike BT840F, an external connector for an optional antenna useful in this type of environment with huge iron bodies and many shielding to guarantee a communication range increment. The RF internal circuit is already tuned to make it easier to insert it inside a board.

The nRF52840SoC is cortex M4F MCU, with a 1MB Flash and 256 KB RAM. Some of its key features are that it supports Multiple Protocols (BLE, ANT, Thread and Zigbee) and its antenna Output power is scalable from $+8dBm$ down to $-20dBm$ and Sensitivity for BLE is $-96dBm$.

BT840e - Main Feature	
Tx Power Level @1 Mbps	$+8dBm$ @ $1000m$
Tx Power Level @125 Kbps	$+8dBm$ @ $2300m$
Max BandWidth	$2Mbps$
Broadcast Length	$31 \div 255$ *bytes*

Table 2: BT840e main feature

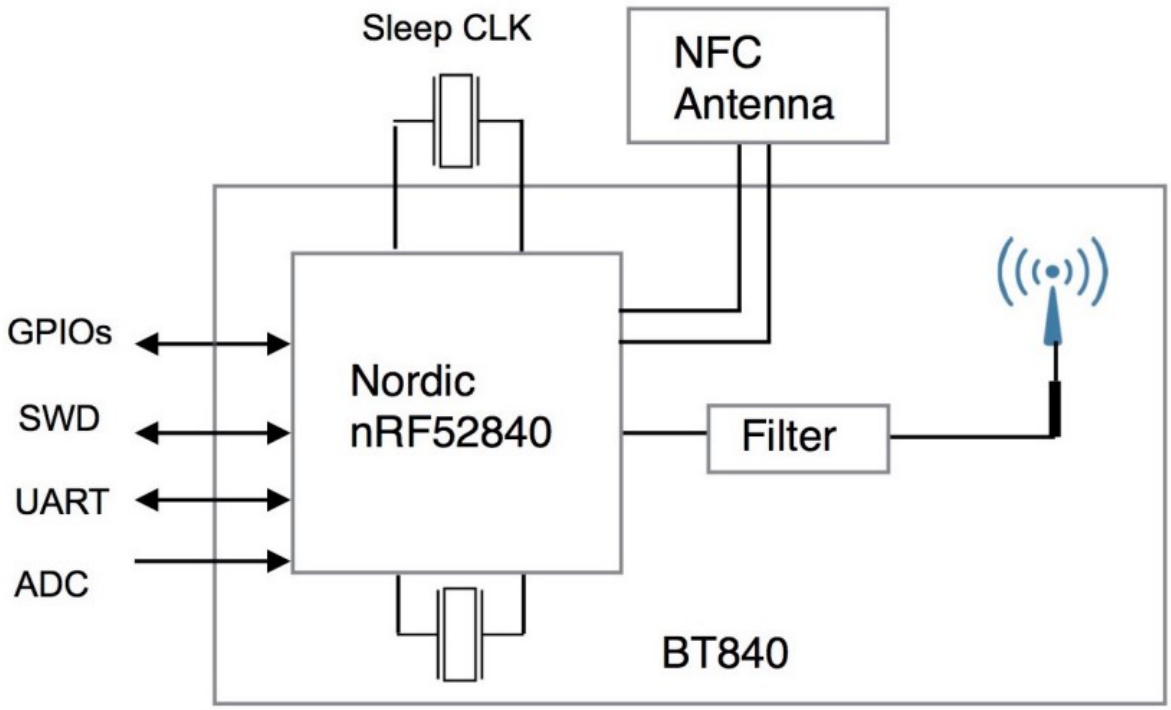


Figure 7: BT840e Block Diagram

1.3.4. Power Section

The main idea behind the implementation of an energy harvester is to first use the solar produced energy to supply the board and when energy in excess is produced recharge the batteries.
The batteries charging current is 10 mA way lower with respect to the maximum supportable with the solar Panel to prevent battery damage.

Harvester

Many kinds of harvesters have been studied and tested and among them all this type of scenario forced us to chose the solar energy: found as the most reliable and widely available one.
Obviously to gather energy we needed a Power Management IC, in this case we used the LTC3331. It is a Nanopower DC/DC Buck-Boost with Energy Harvesting Battery Charger that can manage the battery charge as function of the power coming from the panel. To extend the battery life it also implements a shunt battery charger limiting the current to a maximum of $10mA$ and a function protecting the battery from deep discharge. It has a programmable output voltage between 1.8V $\div$ 5V limited to 50 mA.

LTC3331 - Main Feature	
Max Output Current	$50mA$
Battery Voltage Range	$-0.3V \div 6V$
LDO Output	$1.8V \div 5V$

Table 3: LTC3331 main feature

Photovoltaic Panel

The photovoltaic panel is made of monocrystalline sylicon, to lower production cost a commercial solution was adopted for solar panels, with a dimension of 70 mm x 55 mm.

PV Panel - Main Feature	
Peak Power	$55mW$
Open circuit Voltage	$8.2V$
Voltage at Peak Power	$5.5V$
Current at Peak Power	$100mA$
Length	$70mm$
Width	$55mm$
Weight	$17g$
Efficiency	17%

Table 4: Photovoltaic Panel characteristics

Battery Gauge

The battery gauge LTC2942 is used to measure the current flowing through the battery and its voltage, it mounts a chip temperature sensor and is perfectly suited for single-cell Li-Ion batteries.
A precision coulomb counter measures current through a sense resistor during load and discharge phase and stores it with temperature and voltage in internal registers accessible via the onboard SMBus/I2C interface. These measures are fundamental to estimate the battery State of Charge (SoC).

Battery

A lithium polymer (LiPo) battery has been chosen for this application since they provide greater energy density with respect to other battery types.
A single LiPo cell is used since the node's voltage is lower then 3.7V (nominal battery voltage), normally a battery cell range from 4.2V (fully charged) to 2.8V (fully discharged). The problem with this type of battery is that their state of charge (SoC) and charging current have to be precisely controlled to prevent delamination reducing its reliability and life cycle. [16]

Battery - Main Parameters	
Technology	Lithium Polymer
Nominal Voltage	$3.7V$
Capacity	$2000mAh$
Dimensions	$44x72x7mm$
Weight	$40g$

Table 5: Battery characteristics

1.4. Software

The firmware is divided between two different micro-controllers: STM32L4Q5 and BT840F. Where the first one is used for data acquisition from the sensor and the second one is responsible of the BLE communication. It can be seen in fig.8.

The acquisition software is based on a state machine that can be depicted into three main state [18]: **"Wake"**, **"Run"** and **"Sleep"**:

- **Wake:** During this phase the micro is in sleeping mode while the BT module is sending a Wake message every 30s and waiting for a synchronization response by the gateway. When it is received the BT goes to low power mode switching off the antenna and the micro-controller begins working by starting the acquisition.
- **Run:** During the Acquisition phase the node acquires pressure data obtaining a measure every second (1 Hz) by a mean over 4 samples collected at 200Hz. Acquisition parameters were chosen based on pressure propagation velocity calculated analyzing system dynamics. Every sample's acquisition phase lasts about 28 ms (collecting 4 samples with a sampling time of 5 ms each takes 20 ms and 8 ms for other operations) after which the micro-controller goes to sleep to wake up when a

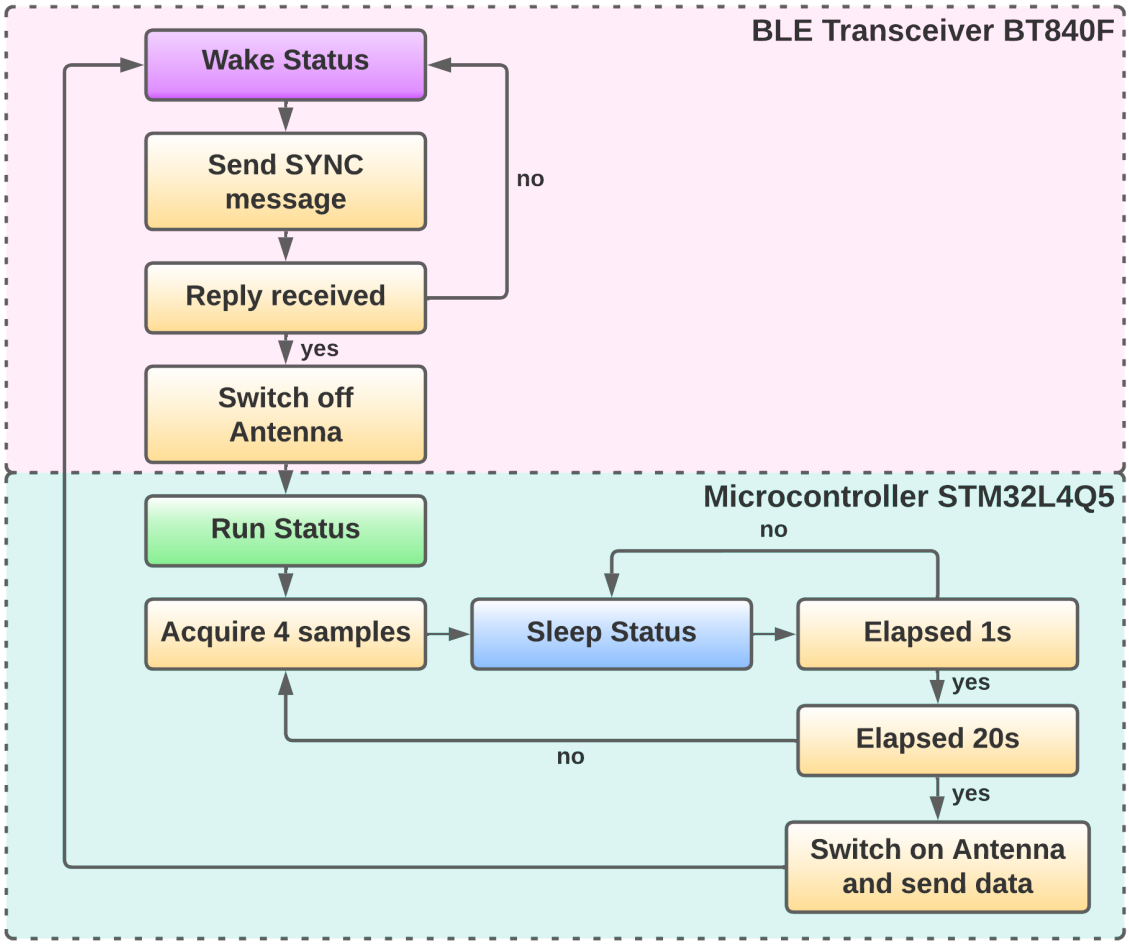


Figure 8: Software architecture. We can notice the BT840F part highlighted in blue and the STM32 part in red

second is elapsed and restart the acquisition. After 20 seconds 20 samples are collected and it's now time to send them to the BT module to transmit values trough BLE taking approximately 3s.
With this approach to always power off the antenna when there is no data to be sent to the gateway a huge energy saving is observed. After all the necessary calculations were done the micro starts to send the, now much shorter, data to the BT840F so it can be sent through BLE to the gateway.

- **Sleep:** This is a sort of "special" state in which the node can be put by the gateway to reduce the power consumption setting the micro-controller in low power mode and the BLE transceiver in reset state.

As it has been described above 20 samples are continuously collected and sent every 3 secs

but to better understand and correlate data during post processing phase the sensor node also transmit useful diagnostic data such as battery voltage, environment temperature and the RSSI value (index of the quality of the communication).
To send data the chosen method was to use BLE modules as a sort of "bridge" just to rely UART communication via wireless using Nus-Nordic available service.
Is important to note that while writing the node firmware special attention was dedicated to the most crucial aspect: power consumption optimization.
A theoretical estimation of sensor node consumption has been carried out to check battery durability and solar energy recharging feasibility. Every IC's maximum operating and sleeping current was read and used in combination with the decided operating time to calculate total current consumption using (1.1):

	Pressure I_p	**Micro I_μ**	**Sleep I_s**	**Comm. I_c**	**Total I_t**
Mean curr. (mA)	1.7	7.24	0.69	8	3.72
Time (s)	20	0.02 every 1 s	0.98 every 1 s	3	20

Table 6: Node Current Consumption

$$I_{\text{mean}} = \frac{I_p * t_p + I_\mu * t_\mu + I_s * t_s + I_a * t_a}{T_{\text{cycle}}} \tag{1.1}$$

After the calculation the battery life was calculated using the capacity of the equipped one (2000mAh) considering that the node sleep current is 0mA and an active time of 8 hours per day. This led to an estimated working time of 67 days without any inflow power from the Photovoltaic panel [18].

1.5. Experimental Set-Up Description and Results

To test the developed sensor nodes Mercitalia Intermodal SpA let us install them on a T3000e freight wagon durign a 5-month field campaign. This wagon is usually employed in long travels for transportation of trailers and/or containers, Table 7 sums up track sections and number of travels performed by the train's wagons.

Track Section	Estimated Distance (km)	Number of Travels
Milano-Munchen	610	2
Munchen-Milano	610	2
Verona-Munchen	450	2
Munchen-Verona	450	2
Verona-Rotterdam	1120	9
Rotterdam-Verona	1120	9
Total	24400	26

Table 7: T3000e Travels Distances

However during our field test the wagons mainly travelled along the Verona-Rotterdam section (figure 9) because nowadays it represents one of the most important route across Europe for goods transportation.

This type of wagon has been chosen because it is composed of two semi-wagons letting

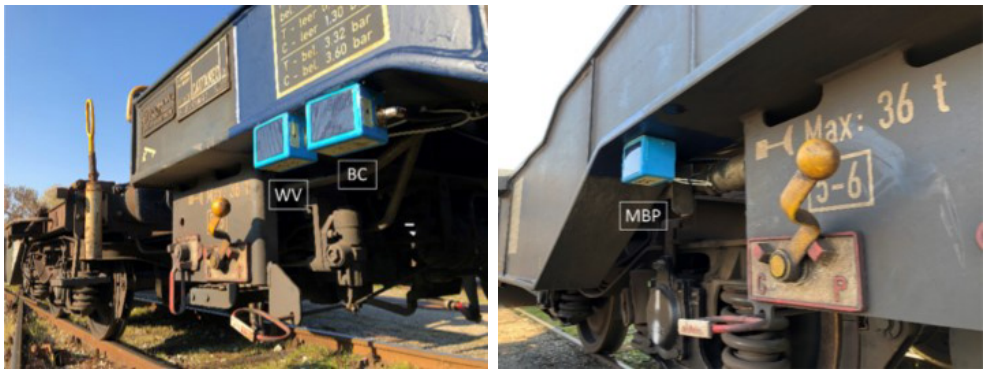
Figure 9: GPS Data recorded during field experiments

us use the first one for our sensor node, while the second one for wired pressure sensors in order to use the collected data as benchmark.

Luckily the T3000e is endowed of various test points placed in strategic positions to be used during maintenance and to perform tests. So it was chosen to take advantage of these test points to install our boards connecting them through universal connectors.

Three sensor nodes were mounted on the three main points of interest (Figure 10): the Main Pipe (MBP), the Breaking Cylinder (BC) and the weighting valve (WV).

Obviously it is mandatory to position them on the external side of the wagons to guarantee



(a) (b)

Figure 10: Position of sensor nodes on Breaking Valve (BC), Braking Main Pipe (BMP) and Weighting Valve (WV)

the best possible sun exposure for the solar panel and for an easier accessibility in case of maintenance.

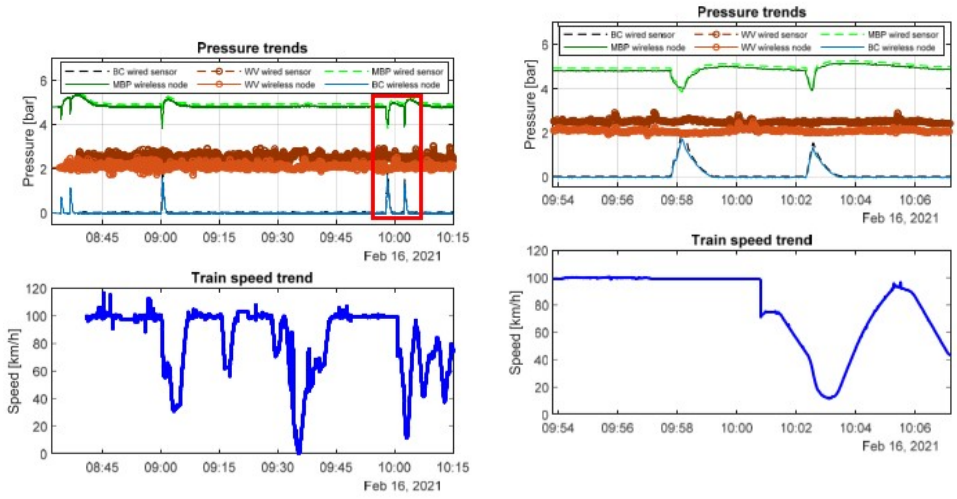
1.5.1. Results

Thanks to the field campaign we could test the functioning of our sensor node and collect a wide amount of data for experimental usage and allowing us to perform data analysis. Additionally we used wired sensors' data to validate the measurement performed by our custom board.

Pressure Measurements validation

To collect a validation of the measured device first of all we had to acquire data with our sensor boards and compare them with the ones gathered from wired sensors.
In Figure 11 we see the various pressure trends referring to Verona-Rotterdam track section and we can appreciate the correlation between both method's measurement.

To complete the validation and verify system's repeatability different travels along the same track were imposed. In figure 12 we can see an example of collected pressure values during four different travels. Despite, during different travels, braking phases can occur in different situations there was a great coherence between braking events and speed trends. With this procedure we could also analyse the different machinists' driving styles in every travel along the same track section.



(a) (b)

Figure 11: Pressure data during travel **(a)** and example of breaking event **(b)**

Energy Harvester Performance

One of the most important aspect to analyze is the power consumption and harvesting part to measure the system efficiency and calculate battery life ensuring energetic independence by any wired power source. Laboratory tests have previously been done to estimate power performances, however it's still not doable to reproduce every possibility undermining our harvesting capability (debris, cloud, temperature, dust). Only the field campaign can assure the functioning of our system. We will now analyze some examples of data collected.

In Figure 13 battery voltage and temperature trends of the three mounted sensor are represented. In almost every scenario the energy gathered through the PV panel and the low energy consumption allowed a continuous measurement without discharging batteries. In fact we can see that the voltage trends stayed almost constant during several hours of working activities, then, the blank voids are owing to the fact that when the train is stopped the whole system is put to sleep to save energy.

Nevertheless some problems came out. Indeed, analyzing voltages together with time spans, we see a steep decreasing trend during night hours. Reaching under 3V generates a major problem representing almost the limit for the sensor nodes to stay awake.

This limitation probably wouldn't have shown in passenger trains, but freight trains frequently travel at night, which means that the nodes are working without sunny conditions. And if the train is stopped in a warehouse or depot the PV panel is probably covered by an adjacent wagon not being able to charge until a new journey is started.

Anyway as soon as the panel is directly hit by sunlight the boards were able to recharge

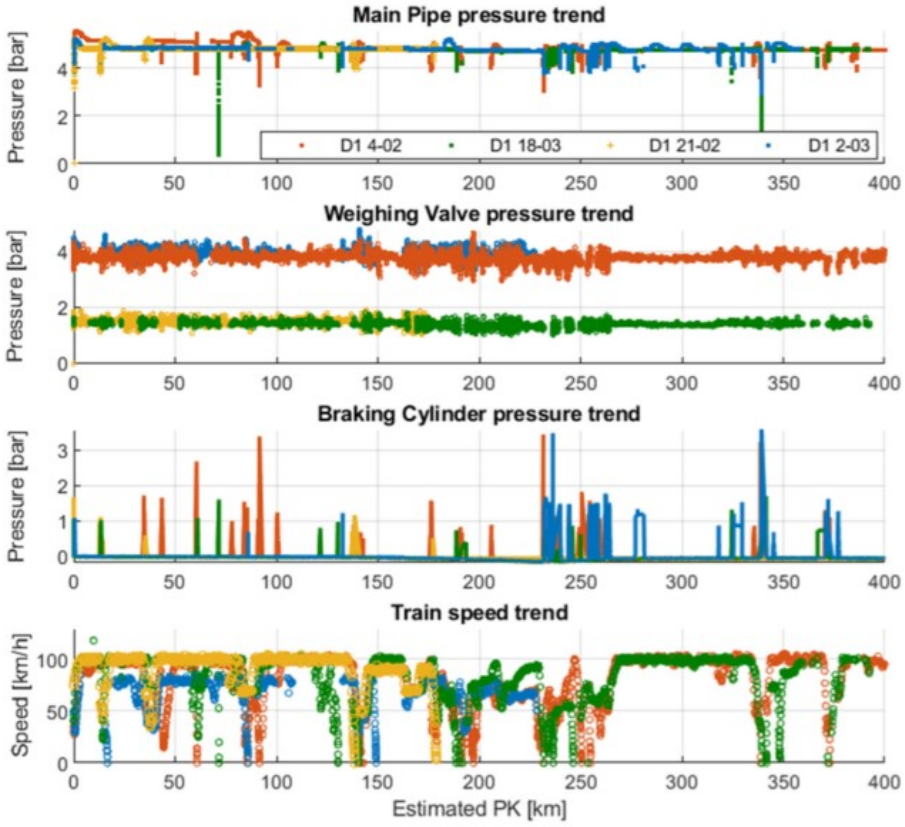


Figure 12: Example of collected pressure values

batteries allowing the sensor to start the measuring phase with aged batteries (Figure 14).

Diagnostic Analysis

In Figure 15 we can see the pressure time history along the whole track section Verona-Rotterdam and with a focus on a single breaking event. In the second graph we can understand that the utilized sampling frequency is perfectly enough to characterize braking pressure trends.

As we said before thanks to the gathering of all these parameters a predictive maintenance approach for the breaking system was possible. The contra is that CBM is not a real time procedure, so we decided to propose a new solution to immediately identify possible faults inside the breaking system. The method was to develop a simple and efficient algorithm running on-board the gateway performing diagnostic activities.

Deeply analyzing Figure 15b we can appreciate pressure variations and find three possible mode to implement real-time system moitoring:

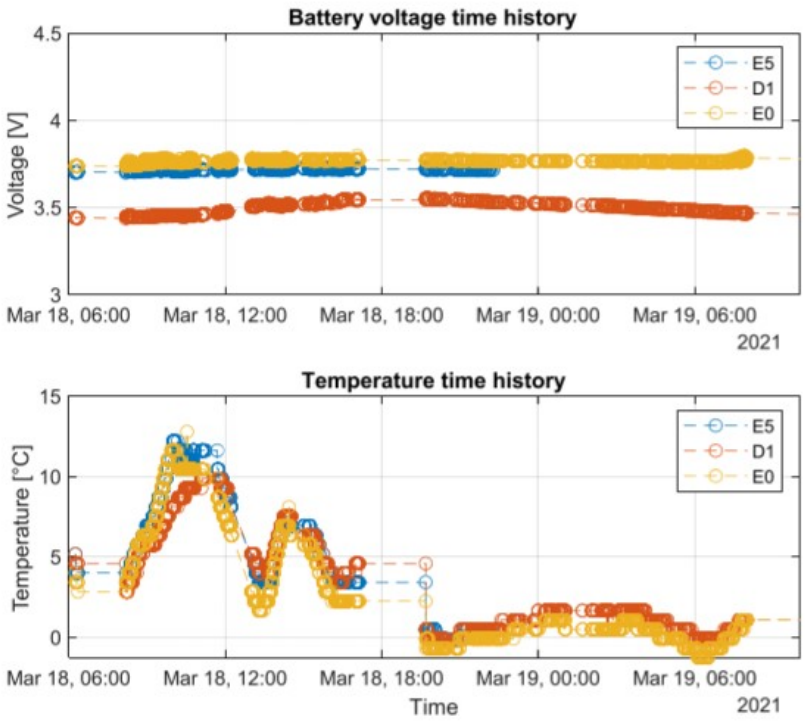


Figure 13: Battery and Temperatures Data

- in every single test-points seeing that at the beginning of a braking phase corresponds a decrease in main pipe's pressure and an increase in the cylinder. If this is not happening we are admiring a fault and the bogie will not brake.
- Another mode is to measure the braking phase duration, compare it with the chosen one before the travel start (selected with a leverage on freight or passenger mode) and if it is not compatible some problems occurred.
- The last method can be use the weighting valve as a method of paragon, obviously it adequate the brake pressure to the weight of the wagon, the more the weight, the more brake force is needed. Therefore monitoring the weighting valve and the braking cylinder pressure can lead to fault detection and distinguish loaded and unloaded wagons.

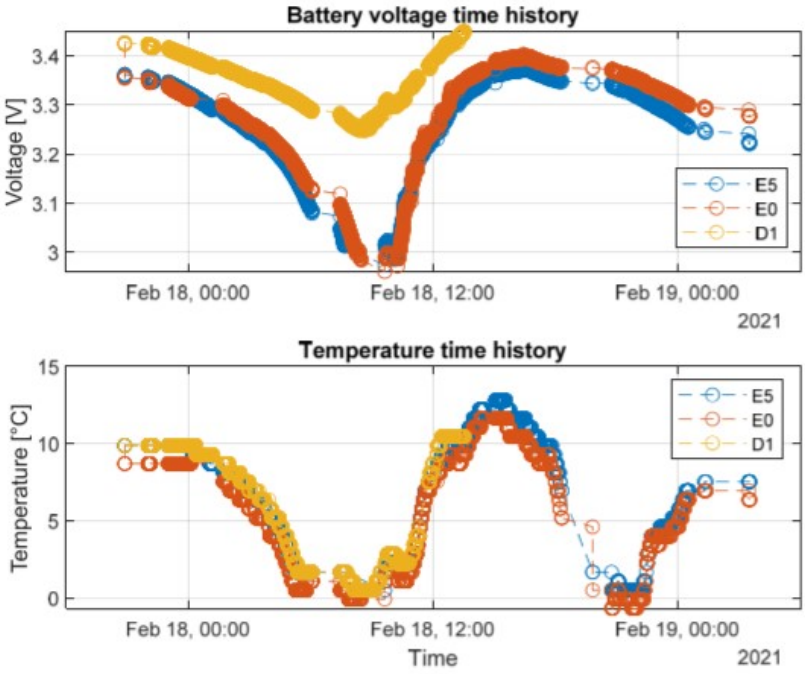
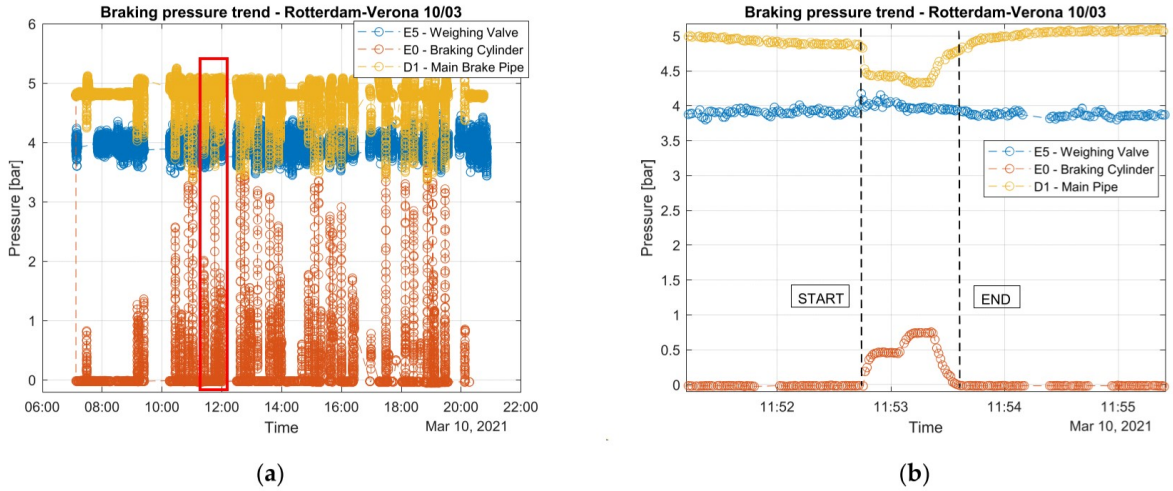


Figure 14: Battery and Temperatures Data



(a) (b)

Figure 15: Pressure Trends along the whole track section **(a)** and during a breaking event **(b)**

2 | Introduction to Bluetooth and BLE

In this chapter Bluetooth Technology is explained with a special focus on its Low Energy Version.

Bluetooth is a data transmission standard used in WPAN (Wireless Personal Area Network) providing a cheap and secure way to exchange data between devices over a short distance (range up to 100 m). Its pros are robustness, low power-consumption and cost. During its development different versions have been created starting from 1.0 up to 5.0:

- **Basic Rate (BR)**: 1.0 version
- **Enhanced Data Rate (EDR)**: v2.0
- **High Speed (HS)**: v3.0
- **Low Energy (LE)**: v4.0

The improvement during the first 3 versions was just focused on security and data rate enhancement, the first major upgrade came with version 4.0 and introduced a completely new standard: the BLE (Bluetooth Low Energy) protocol. It is not just an upgrade, but a whole new technology suitable for IoT application and low power consumption systems with ultra-low-power characteristic differently to Bluetooth.
Since it is a completely new technology it is not compatible with Bluetooth Classic, in fact a BLE device can communicate with a Bluetooth device only if at least one of them implement both technologies.
It was first created, monitored and standardized by the BSIG (Bluetooth Special Interest Group): a group of companies including Ericsson, IBM Intel, Microsoft, Apple, Toshiba and much more.
Bluetooth has been introduced in IEEE standard norm 802 including Wireless Speciality Networks in 2002: specifically norm 802.15.1 describes BT physical layer (PHY) and Media Access Control (MAC).

In BLE there are two types of devices:

- **Central**: Usually a PC or a smartphone
- **Peripheral**: Normally something small such as wearable, smartwatch, button and other IoT type devices

Since normally Central have a larger battery with respect to peripheral, BLE is an asymmetric technology putting much of the responsibility and heavy lifting part on the Central device allowing the peripheral to sleep, turn off the ratio and consume less power.
In fact in Classic Bluetooth master device must request to start the communication and begin a transmission while in BLE standard slaves can communicate without a permission. With this approach it is possible to fasten Data communication and optimize power management.

2.1. Specifications

BLE operates in the same frequency range of Bluetooth Classic and Wi-fi: 2.4GHz ISM (Industrial, Scientific and Medical) free band (2,402 GHz to 2,480 GHz). It implements 40 RF channels spacing 2MHz (center to center) wher three of them (37, 38, 39) are exclusively used as Advertising Channels. Devices communicates with a technology called Frequency Hopping Spread Spectrum (FHSS) switching between channels using a pseudo-random order pre-decided during the master-slave connection initialization.
The modulation used is the GFSK (Gaussian Frequency Shift Keying) with an index between 0.28 and 0.35 where bit "1" correspond to a positive frequency variation and "0" to a negative one.
There are 3 types of Bluetooth classes depending on the transmitter output power:

Class	**Max Output Power**	**Range**
Class 1	$100mW$	ca. 100 m
Class 2	$2.5mW$	ca. 4 m
Class 3	$1mW$	ca. 1 m

Table 8: Bluetooth Classes

Most commercial devices are Class 2, usually Class 1 are utilized in industrial application. It is important to remind that normally to achieve the best communication performance devices should be in eye-to-eye range because every obstacle between them drastically

reduces transmission range and efficiency. To improve them some parameters and configuration can be changed, in particular: Tx Output Power, battery conditions, antenna configuration and receiving device sensibility.
Bluetooth uses a Clock signal to be able to synchronize every device together, it has a period of $312.5\mu s$ and it is synchronized during connection initialization.
The maximum transmission velocities available depends on the Bluetooth version utilized:

Version	Max Data Rate
2.0 EDR	$1-3Mbps$
4.0 BLE HS	$1Mbps$
5.0 BLE	$2Mbps$

Table 9: Bluetooth Physical Rate Speeds [11]

2.2. Network Types

Bluetooth technology supports different type of network such as *point-to-point* (only two devices) or *multipoint* (more than two devices). Two or more units connected together create a **piconet**, in every piconet it is mandatory to only have one Master present which goal is to manage the connection, while the other devices are called Slave. The maximum number of Devices connected to the same piconet is 8 (1 Master and 7 Slaves). It is important to remind that Slaves cannot be connected together without a master.
Every master can also be a slave of a different piconet and a slave can be connected to more than one master. A net build this way is called a **Scatternet**.
With this method Bluetooth potentially has a capability of reaching network with infinite device population, however in reality there is a maximum number of 10 piconet connected together.
Both network configurations are shown in figure 16.

The main problem in a scatternet configuration is that devices contained in a piconet cannot relay data to other piconets. This aspect drastically reduced the number of reachable devices inside a network, this is the reason that with version 4.0 a new technology called **mesh** has been introduced. This allows every node to support the relay feature with which a message can traverse the entire mesh network hopping between devices. Bluetooth mesh does not require hardware changes, it does only require a software update in

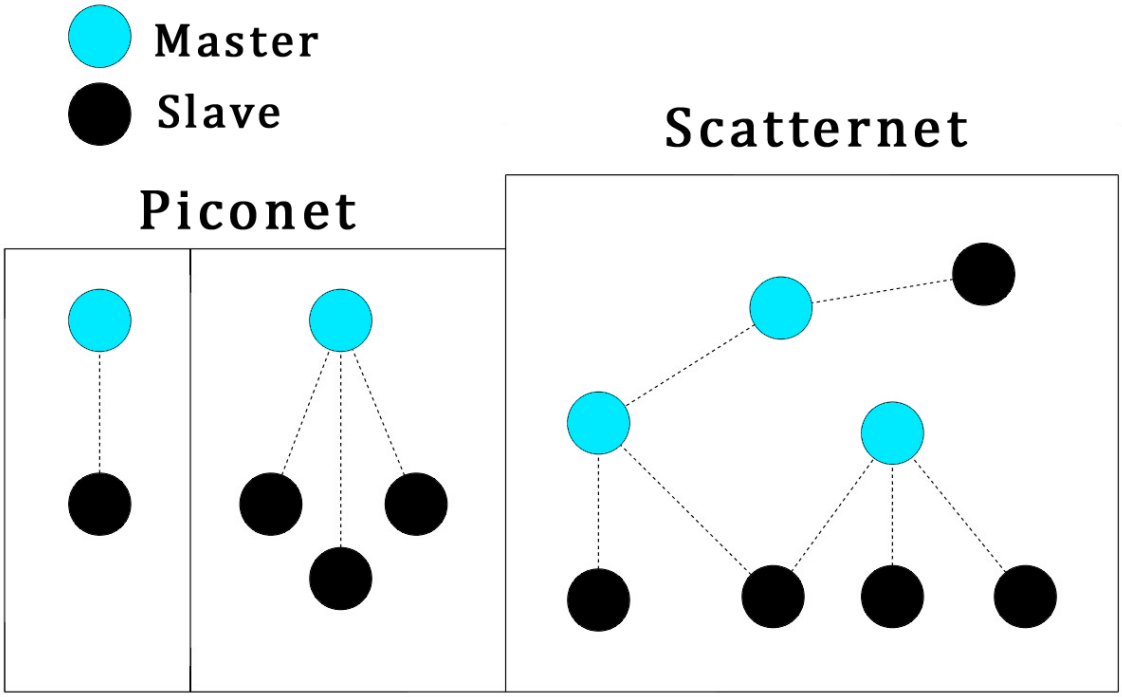


Figure 16: Different Network Types

order to support this standard.

2.3. Architecture and Protocol Stack

Bluetooth comes as a hardware-based and software stack radio system specifying the linkage between every layer so that it supports flexibility and implementation across a variety of different devices and platforms [13].
In this section we'll talk about:

- **Bluetooth protocol stack:** The protocol stack is the core of the Bluetooth specification that defines how the technology works.This is the core of Bluetooth specification allowing to ensure interoperability of Bluetooth devices and encouraging the adoption of Bluetooth technology.
- **Bluetooth profiles:** The profiles define how to use Bluetooth technology to accomplish specific tasks

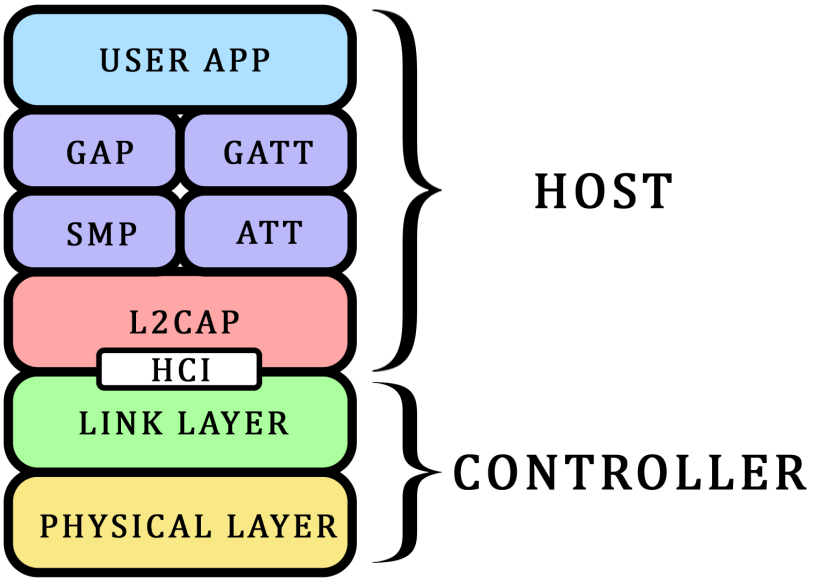


Figure 17: BLE Protocol Stack

2.3.1. Physical Layer

This layer is hardware based and responsible for modulation on demodulation of data into RF signals to be transmitted in the air. The Physical Layer describes the characteristics that a Bluetooth Transceiver must have.
Bluetooth devices use AFH (Adaptive Frequency Hopping) algorithm to choose the frequency channels between the 79 available ones contained inside the ISM band. They are selected and changed at a speed of $1600 hops/s$ with a pseudo-random order. The switch sequence is decided depending on master's preference while its timing coincide with the clock signal.

Transmission works with blocks called slot. A slot is composed of 2 clock cycles: during even slots master transmits while the slave receives, on the other side during odd slots slave transmits while master receives. The transmission of a whole packet can reach a maximum of 5 slots.
Unfortunately in some Bluetooth environment, due to interference, certain radio channels might not be working very well while others are working reliably. In this scenario the key point of AFH with respect to simple FH takes in hand: the Master during a connection maintans a "channel map" classifying each channel and its reliability. This channel map is then shared with every slaves using a Link Layer Procedure and it's used as base to decide which channel should be used and which one avoided during frequency hopping.
The channel map is kept updated during every connection and always shared with the other devices to be sure that every devices use the most reliable available channels. A

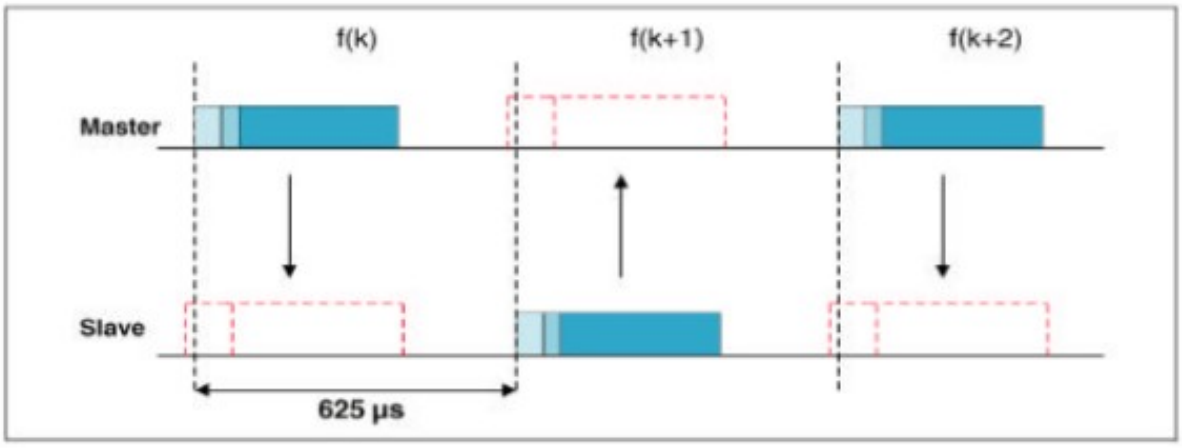


Figure 18: BLE Timing Slots

sort of channel maps created by a master device during a connection and transmission sequence is shown in figure 19.

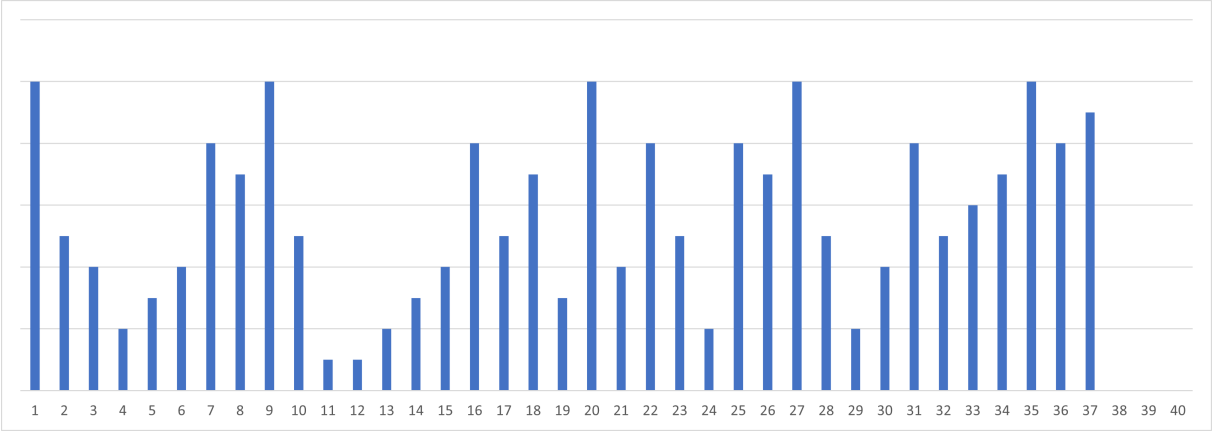


Figure 19: Example of BL channels reliability map

2.3.2. Link Layer

The Link Layer connect the Host Controller Interface (HCI) and the Physical Layer, it is responsible for establishing links and managing power-change requests.
It defines and manages the two different types of connection available depending on the type of data exchanged.

- **SCO Link** (Synchronous Connection Oriented): Usually used for voice services, it is a point-to-point connection between one master and one slave. It defines the timing of the transmission by reserving one slot in one direction and the consecutive

slot for the other direction. One master can support a maximum number of 3 SCO connection with the same slave or with different devices, while a slave can have a maximum of 3 SCO Link with the same master. Lost packets are not re-transmitted. Forward Error Correction is applied for data reliability.

- **ACL Link** (Asynchronous Connection-less): there are not reserved Time Slot, but both symmetric and asymmetric traffic are supported, the master controls the bandwidth of the ACL Link. Packet re-transmission is allowed with FEC and Backward Error Correction to ensure data integrity.

Link Layers also defines the following states:

- **Standby:** a device that is neither sending nor receiving packets.
- **Advertising:** a device that is advertising packets to announce its presence to other devices in the vicinity.
- **Scanning:** a device scanning to incoming advertising packets.
- **Initiating:** a device that is trying to establish a connection.
- **Connection:** a device that is connected to another one and can exchange data.

2.3.3. Host Controller Interface

The Host can't directly change or edit settings in the controller, to allow interoperability between them there is a mediation layer known as HCI (Host Controller Interface).
In case the Controller and the Host stacks don't share the same processor this layer can also be implemented through a serial port, a USB or other communication protocol .

2.3.4. L2CAP

L2CAP is responsible for Data integrity, Protocol Multiplexing, Segmentation and desegmentation. In case a packet doesn't reach the destination this layer is in charge of the re-transmission procedure. The L2CAP Layer adds a header to the higher-layer payload and passes the PDU to the Link Layer just below it.

2.3.5. Security Manager

The SMP (Security Managaer Protocl) can apply, if required, security algorithms to encrypt and decrypt data packets. This layer implements protocols and algorithms for security keys' generation and their exchange between connected devices.

It implements five main features:

- **Pairing**
- **Bonding**
- **Authentication**
- **Encryption**
- **Message Integrity**

2.3.6. Generic Attribute Profile

To cover the functions of GATT we first need to talk about three important concepts:

- **Services:** a service is a group of attributes to expose server characteristic (for example in a SIG-adopted humidity service a characteristic called humidity level is contained).
- **Characteristics:** they are held inside a service and carry information and/or data that wants to be exposed. Characteristics normally include more than just a value:
  - **Properties:** defines characteristic's properties: read, write, notify, indicate...
  - **Descriptors:** usually describes format, the unit of the values.
- **Profiles:** they define the behaviour of the client and the server.

GATT defines two Roles in which a device can act:

- **Server:** This device exposes the data it controls or contains. It accepts commands from other devices that can be responses, indications and notifications.
- **Client:** It interfaces with the server to read data and/or control its behaviour. it sends commands, requests and reads incoming notifications adn indications.

All devices can act in both roles.

2.3.7. Attribute Protocol

ATT describes how a server exposes data to a client, how it is structured, the available properties relative to every characteristic: if it can be read and/or written and the permission required to do so.
ATT performs data organization into **attributes** as shown in figure 20.

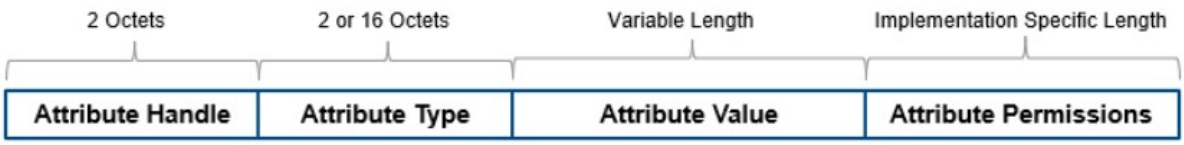


Figure 20: Structure of ATT packet

Attributes are defined by:

- **Attribute Handle**: a 16-bit identifier (0x0001 - 0xFFFF) value used by the server so that a client can refer to a single specific attribute.
- **Attribute Type**: a Universally Unique Identifier (UUID) that can have a length of 16-bit if a SIG-adopted one is used or 128-bit if a custom one is generated by the developer.
  For Example a SIG-adopted UUID can be: **0x15A3**
  This value is written inside the 128-bit base UUID instead of the bold zeros:
  0000**0000**-0000-1000-8000-00805F9B34FB
  On the other hand a custom UUID can be completely based on developer's choice:
  FBA1287E-227D-4C9E-AD2C-11124EDED640
- **Attribute Value**: is the field containing the value related to the attribute, it is of variable length.
- **Attribute Permissions**: specify read and write permission relative to an attribute and the security level required to perform operations.

To understand more deeply attributes and how we can interact with them we first have to talk about the available different ways to perform attribute operations:

- **Commands:** sent by the client and do not require a response
- **Requests:** sent by the client and require a response, can be of two types:
  - Information Requests
  - Read Requests
- **Responses:** sent by the server after a request has been received
- **Notifications:** sent by the server to notify a change of a characteristic's values after the client has enable them for a chosen characteristic. It doesn't require a response.

- **Indications:** sent by the server and require a Confirmation as a response, they are similar to notifications
- **Confirmations:** sent by the client to acknowledge an Indication

Bluetooth 5 Packet Examples

In figure 21 we can see a BLE 5.0 packet example, it maintains the same structure of Bluetooth 4.2.

The actual data transmitted is included inside the payload field, the length field is 8-bits

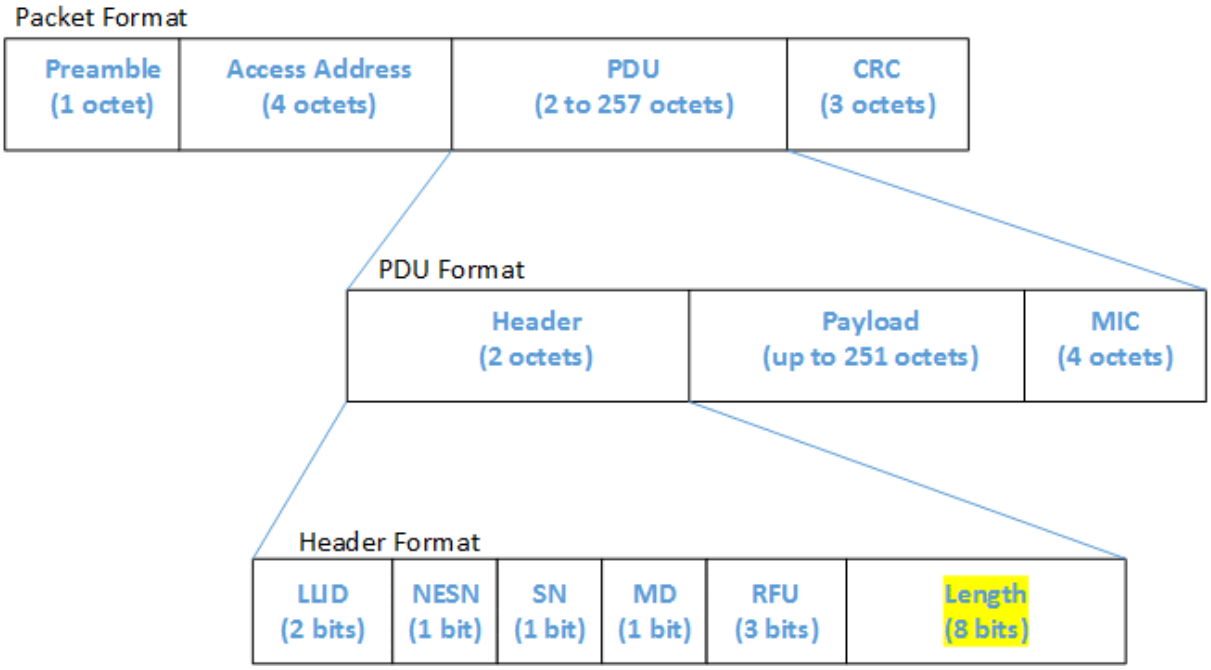


Figure 21: BLE packet construction

in length going up to 255, representing the length of the Payload and MIC if included. The Payload field shall be less than or equal to 251 octets in length because the MIC is 4 octets in length, 255 – 4 = 251.

2.3.8. Generic Access Profile

The Generic Access Profile provides access to the link layer operations including: the definition of the devices' roles in BLE communication, managing advertisement, connection establishment and security. Since it is a profile layer it only defines the way to initialize and establish a connection.

In particular thanks to this layer a device can act in four available roles:

- Broadcaster
- Observer
- Peripheral
- Central

The first two have the advantage that devices can exchange data without first establishing a connection, the broadcaster is the device that sends the information while the observer is listening for it. This type of communication is called connection-less.
The two last ones exchange data only after a connection has been established between peripheral and central. This type of communication is called uni-cast. It is slightly more complex with respect to the one we talked about before because there are some steps to fulfill to make it happen:

1. The peripheral must be announcing its presence by advertising.
2. The central have to be searching for these announcements.
3. Central device must send a connection indication to the peripheral.
4. Once connection has been accepted data can flow in both directions.

2.4. Timestamp

2.4.1. Advertisement

In figure 22 advertising and scanning processes are shown.
Regarding the peripheral the most important parameter is the *Advertising interval*, it indicates the time between the start of two different events: during this period the three advertising packets are sent and then a pause happens until the next connection interval begins.
Instead the central side defines its scan window based on two important parameters: *Scan Window* and *Scan Interval*.
The first one indicate the length of a scanning events while the second one the time between two different windows.

During scanning process the central starts searching on the first selected channel and switch to a different one every scan interval. If an advertising device is found on the same channel a connection may occur.

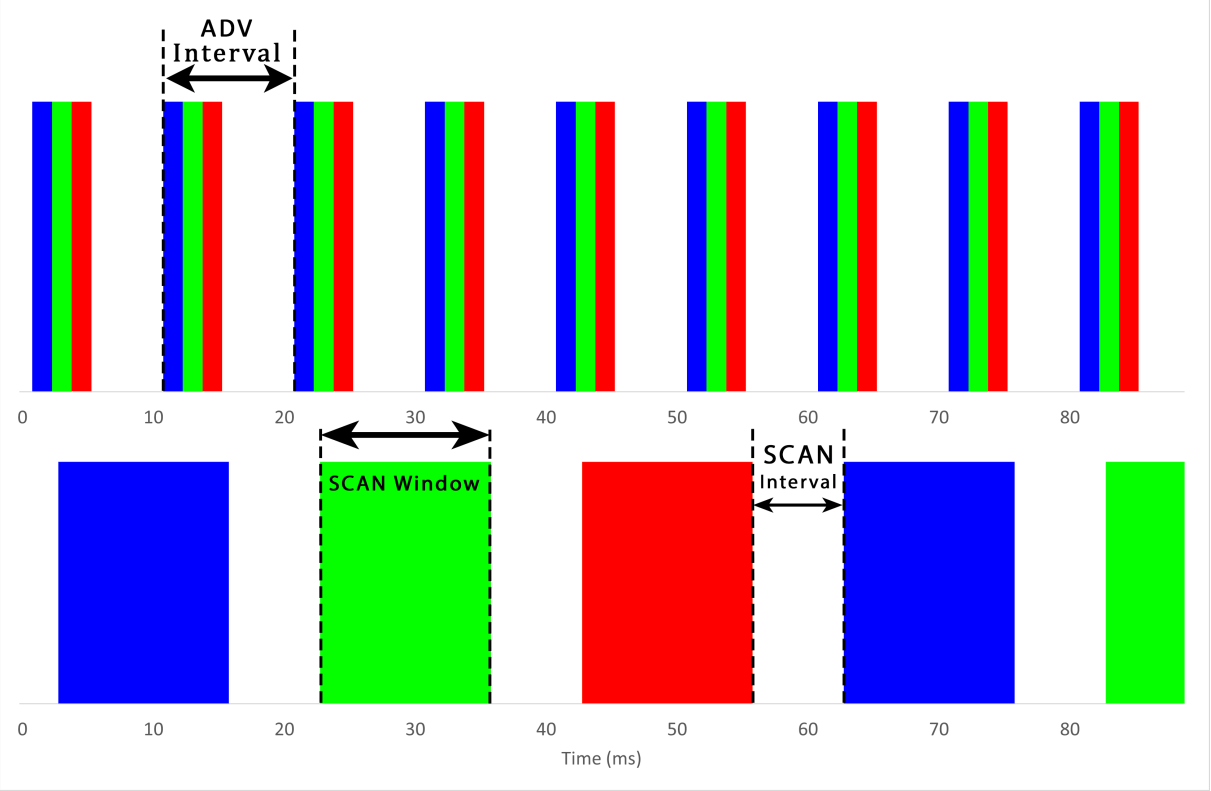


Figure 22: Advertising and Scanning time intervals example with a advertising interval of $10ms$, scan window $12.5ms$ and scan interval $6.5ms$

2.4.2. Transmission

When a connection is established the two devices start exchanging data (Fig. 23). Fist central ask for the start of a data transmission every *Connection Interval*, this phase is highlighted in red. They both proceed to wait an interframe space after which the peripheral decides to send or not the requested packets (can be empty), they wait again an IFS to end the transmission. This whole cycle repeats every connection interval.
The slave can decide to respond or not to a central's request: it can skip a defined number of packets but then the response is needed in order to keep the connection alive.This parameter is called *Slave Latency* and the bigger it is the lower the power consumption becomes.
The length of the transmission event is mainly dependent on two parameters: *Data Length* and *Physical Rate*: the first one affects packets (T) length while the second one changes R and T duration.

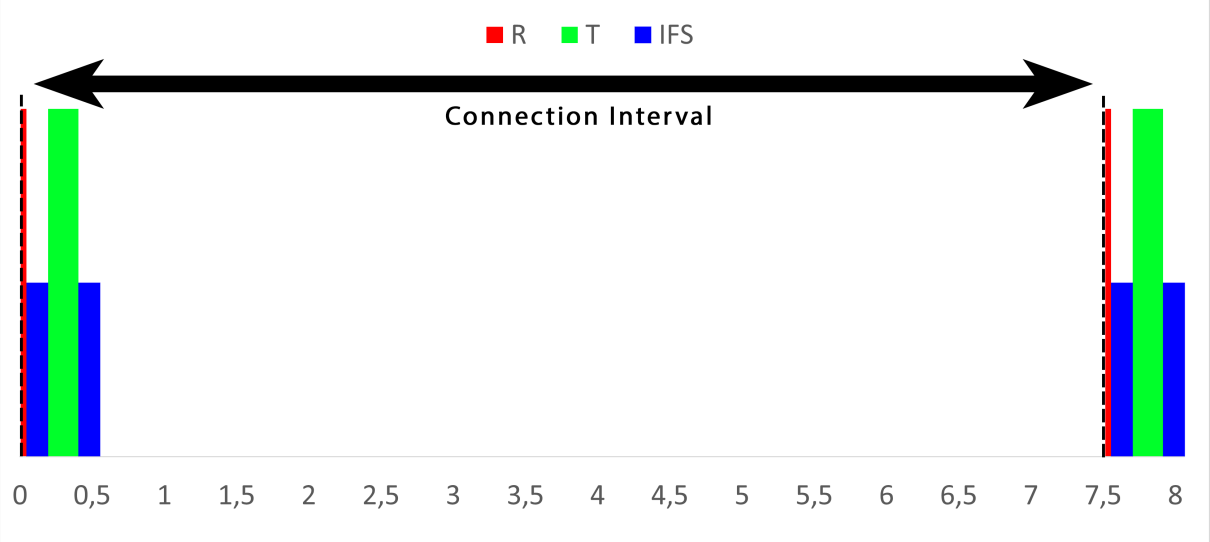


Figure 23: Connection time intervals example with a connection interval of 7.5ms, IFS 150μs, R 40μs and Tx packets 30$byte$ using a 2$Mbps$ physical speed

2.5. Performances

BLE Performances and Efficiency can be measured and described using different parameters. In this chapter we will only analyze the three most important ones in our test case: Throughput, Range and Energy consumption. [12]

2.5.1. Range

The communication range is influenced by conditions that can be controllable and not. One of the major disadvantage is that it operates in the same frequency range as IEEE 802.15.4, ANT+, Thread increasing the disturbance presence probability.
Range is mostly affected by:

- **Transmission Power**: offering a low-energy consumption protocol lead to a low transmit power creating a big challenge in covering long communication ranges. Obviously the higher the transmission power the greater distance can be covered. By law the maximum reachable power is 100mW (20 dBm) but it depends on the type of transceiver used. It can be selected choosing between different Levels, that in our application range from -40dBm to 8 dBm. Nordic nRF52840 can achieve a maximum of 4 dBm in normal and high-speed mode and 8 dBm in long range mode.
- **Receiver Sensitivity**: it plays a large role in determining the range, it is chosen depending on the selected mode: long-range is -103 dBm, high-speed is -92 dBm and normal is -93 dBm.
- **Environmental Conditions**: the ambient in which BLE is operating has an impact on the range.
  Indoor scenario are the worst ones with this optic, building's materials and barriers like walls, ceilings, people, furniture and much more strongly limit the range.
  Outdoors BLE could technically reach up to 100 meters with the line of sight between transmitter and receiver devices. Outdoors there are obstacles such as trees, weather conditions (rain or snow) that negatively affect the range.
  Contrary to popular belief communication range is not only reduced if there are obstacles on the line of sight between the two devices, but also the Fresnel zone should be free from obstacles. This situation is shown in Figure 24.

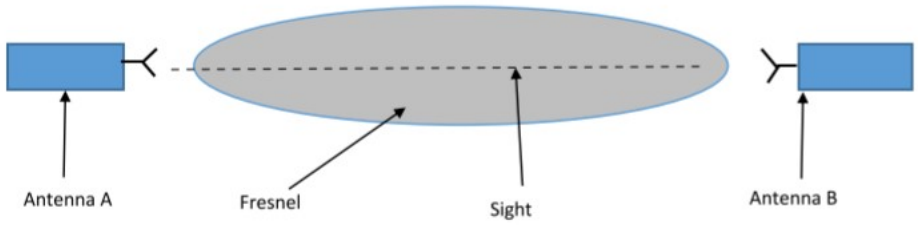


Figure 24: The Fresnel zone is the grey one, wider in the middle between the two devices

2.5.2. Throughput

The throughput refers to the final application's data rate intended as bits per second (bps or bit/s). The physical bit exchange velocity is way higher then throughput but the second one is calculated at the application layer so it is decreased by every packet header added.
Throughput is influenced by a wide variety of parameters, we'll see some of them:

- **Modulation Frequencies**: As seen above classic Bluetooth was capable of physically transmitting 1 Million bits per second, but with 5.0 it reaches velocity of 2 Mbps almost doubling the final data throughput.
- **Connection Interval**: It is the time between two consecutive connection events between central and peripheral. It can range from 7.5 to 4000 msecs, obviously higher connection interval will lower data throughput while shorter ones will make it faster.
- **Interference**: The present of disturbances is counterproductive because if a transmission error happens a resend is needed.
- **Inter-Frame Spacing**: This parameter represents the delay between two consecutive packets on the same channel (see Figure 25.

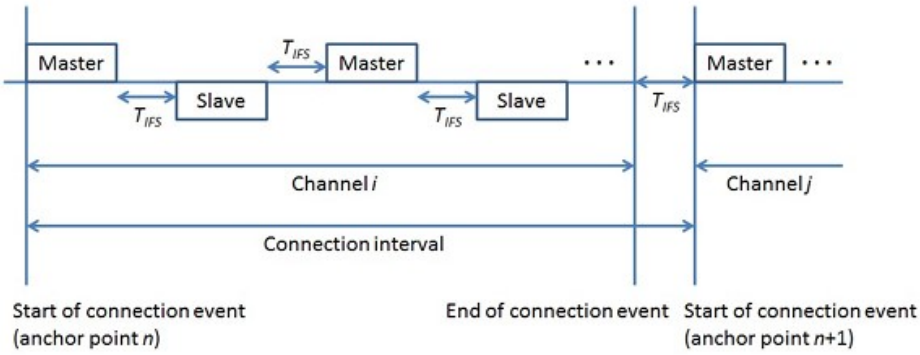


Figure 25: Data Exchange between master and slave

- **Packet Size**: This value describe the maximum bytes number that can be processed inside 1 packet, it is between 23 and 247 bytes. The bigger the size the faster the transmission will be.

2.5.3. Energy consumption

Since energy usage is a major concern in battery-powered products and so in our application. The factors that can optimize power consumption are:

- **Tx Power and Rx Sensitivity**: The power available to the transmitter is the most important parameter affecting Energy Consumption and range.
- **Transmission Duration**:The power consumed is clearly related to the transmission duration. It is the amount of time that a radio is active and it is determined by how often the radio must transmit or receive and the length of time required to transmit or receive.
  Clearly this factor is influenced by almost all the parameters seen above: Connection Interval, Interference, Packet Size, IFS.
- **Physical Rate**: The frequency modulation used also affect Energy consumption since it can drastically fasten up transmission phase.

3 | Test Kit Introduction

In our test case our goal is to exchange information through BLE protocol between two devices: one simulating the sensor node and working as a peripheral and the other one (normally a dongle or a PC) representing the receiving node and working as central.
To supervise the communication we connected both devices to a "manager" represented by a PC that is in charge of receiving logging message and sending commands using UART interface. We can see a diagram of the whole system in figure 26.

As central we could have used the sensor node board already developed, but since our

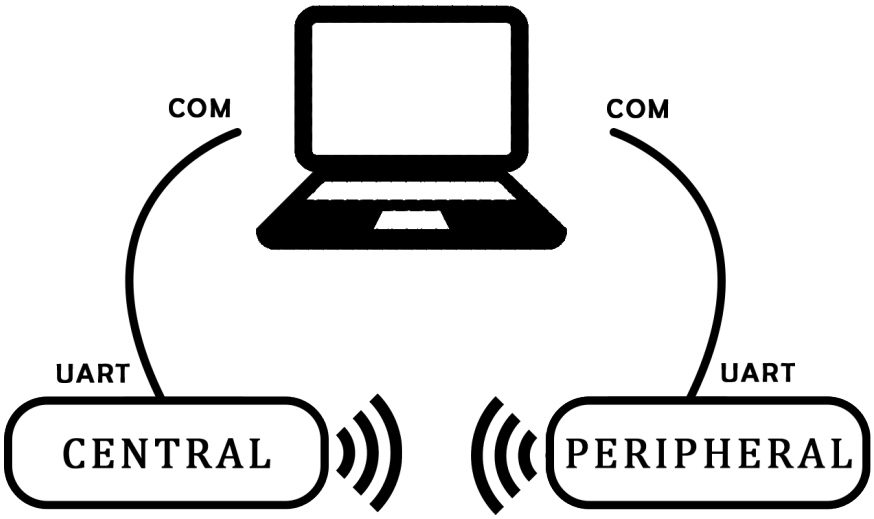


Figure 26: Test Kit Configuration

goal is just to program and test only the BLE communication part we were left with two alternatives: develop a specific test board or use a Development Kit.
With the first option being unnecessarily complicated and the second one way faster we searched for an already built board implementing the the nRF52840 SoC produced by Nordic Semiconductor. Luckily Nordic provide a wide range of hardware and software

development tools.
We specifically picked the development kit nRF52840-DK and programmed it using SEGGER Embedded Studio.

3.1. nRF52840-DK

The nRF52840-DK [6] is the development kit for nRF5280 SoC.
Its key features are:

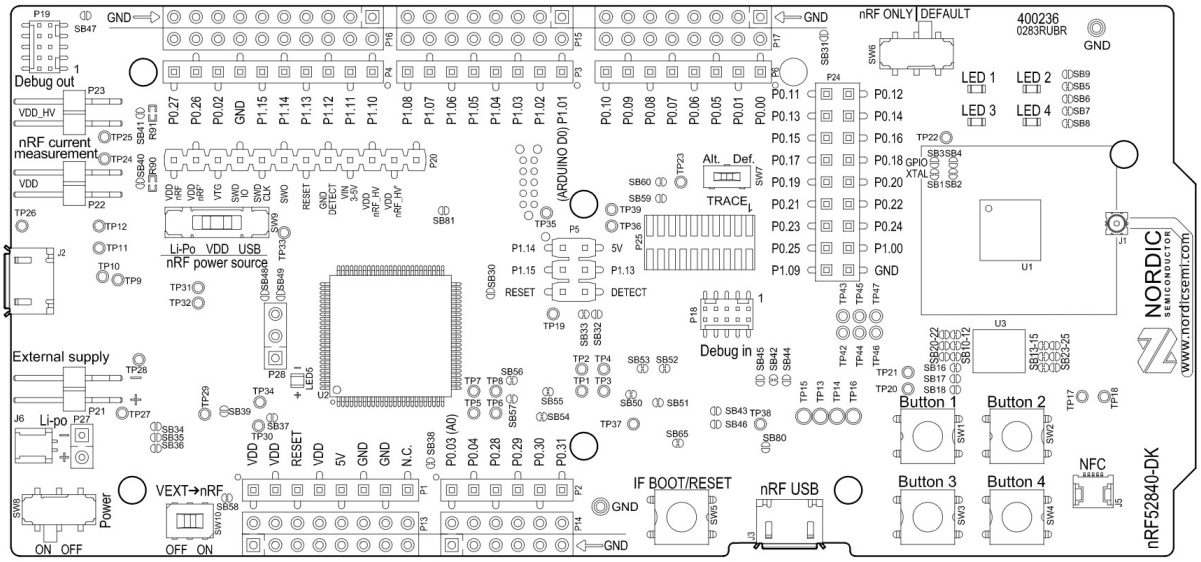


Figure 27: nRF52840 DK top view

- nRF52840 flash-based Bluetooth® Low Energy
- Buttons and LEDs for user interaction
- I/O interface for Arduino form factor plug-in modules
- SEGGER J-Link OB Debugger with debug out functionality
- UART interface through virtual COM port
- USB
- Flash memory

It makes available all features and GPIO of the mounted SoC, implementing an onboard SEGGER J-Link debugger to enable programming and debugging without additional hardware. It includes an NFC antenna and on-board external memory connected to the QSPI peripheral.

There are four buttons and LEDs both active-low connected to dedicated ports of the SoC. The onboard interface also features an UART through virtual COM Port with flexible Baud Rate and Tri-state lines.
The DK is typically powered with USB, but can be also powered BY an external source like a CR2032 battery holder with its Li-Po battery connector.
One of the pros that comes with using a pre-built board is that Nordic makes available a whole lists of software libraries regarding PIN definition, Debug options and precompiled BSP (Board Support Package) for an easier and way faster implementation.

3.2. SoftDevice

SoftDevice is a precompiled software made available by Nordic Semiconductor, it allows us to implement a custom wireless protocol, we just need to include it inside our project. It makes firmware programming and wireless communication management way easier by providing SoftDevice Application Program Interfacse (API) to applications for high-level programming.
There are different versions of SoftDevices depending on which type of device (Central or Peripheral), version of nRF is going to be used and the protocol that it can provide (BLE and/or ANT).
For example to use BLE protocol on nRF52840 for both Central and Peripheral Roles we used the S140 version.
The S140 SoftDevice integrates a Bluetooth 5.1 Low Energy protocol stack and supports running up to twenty connections concurrently, with an additional observer role and broadcaster role. SoftDevice consists of three main parts with all of them implementing their relatives API:

- **SoC Library** implementation and nRF Application Programming Interface (API) for shared hardware resource management
- **SoftDevice Manager** implementation and nRF API for SoftDevice management
- **Bluetooth protocol API** implementation of protocol stack and API

Application Programming Interface

The SoftDevice Application Programming Interface (API) is available to applications as a C programming language interface based on Supervisor Call (SVC) and defined in a set of header files.

All types of SoftDevices with the same versions number share the same API.
Each SoftDevice API calls a SVC interrupt who addresses the correct function allowing applications to compile without the need to include API function address at compile time.
SoftDevice interrupts are used to trigger events for the application which is then in charge of handling them and invoking the relative functions to obtain the relevant data.
The application must respond to events to ensure that the SoftDevice is working properly. Especially in BLE if there is no response to certain protocol stack events the procedures may time out and result in a link disconnection.
There are two types of errors used to confirm whether an API function call was or wasn't successful:

- SoftDevice assertions
- Attempts by the application to perform unallowed memory accesses

3.2.1. Event Handlers

To be able to use *SoftDevice* and its BLE stack functionality in our application we first need to initialize them. After they are configured we still have one important step remaining: Event Handlers management.
Event handlers are used by the SoftDevice to perform proper action based on generated events.
In our case study we are using BLE, UART and FLASH Handlers.
These Handlers catch different events that happen during BLE communication, uart reception and flash readings and sends a proper response concerning that event.

3.2.2. Nordic Uart Service

In our application case devices' job is to be transparent in a communication meaning just relying UART messages over BLE protocol. As we said before one drawback of BLE is that to communicate it is mandatory to choose a service to be used.
Luckily Nordic semiconductor developed the **Nordic UART Service** with the goal to emulate a serial port over BLE. It has two characteristic:

- RX Characteristic (Write property)
- TX Characteristic (Read and Notify property)

Data received from a peer through BLE is passed to the NUS application and then to the module, the same will happen when sending data to peer, data received through UART

first will be passed to the NUS application and then to the peer.
We implemented it on the central side and after a connection has been established the two devices will exchange Serial messages using Characteristic's properties.
Remembering that there are two different devices, one working as a central and one as a peripheral we will talk about their code, remembering that the central is normally represented by a USB dongle with the responsibility to initialize the connection after finding a peripheral in advertising mode.

3.3. Power Profiler Kit II

The PPK2 is a tool produced by Nordic Semiconductor allowing the measurement of the power consumption of an external board [8].
It can operates in two different modes: *Source*, where the device is used as a generator and measuring the provided power; *Ampere-meter*, PPK2 is just working as an amperometer so the external board must have a different power source. Here below we can appreciate PPK2 main features:

- Measuring range of 500nA to 1A
- Resolution down to 200nA
- Power supply voltage ranging from 0.8V to 5.0V
- 100 kHz sampling frequency
- 8 digital pin
- USB port for communication and programming
- Desktop application

In our case scenario I decided to use the PPK2 in the ampere meter configuration so i'm able to power up the DK using the USB port (see Fig.29 for the connection diagram). This way it can still be programmed and debugged also during current measurements without always having to unplug the J-Link and plug the PPK2 every time.
The problem is that the DK doesn't come ready to perform this type of connection but we need to slightly modify it to access nRF SoC power source. As it can be seen in Fig.28 there are 2 pin-outs (P22) directly connected to the power lines but they are shorted by factory through a shorting bridge (SB40) which needs to be cut using something sharp.
To then restore the normal functioning after measurement it is mandatory to solder again the shorting bridge or apply a jumper on P22. [7]

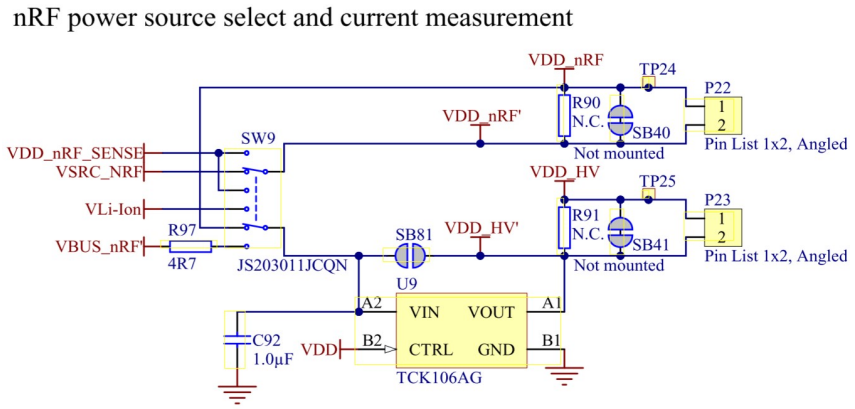


Figure 28: nRF Power Sources selected and current measurements

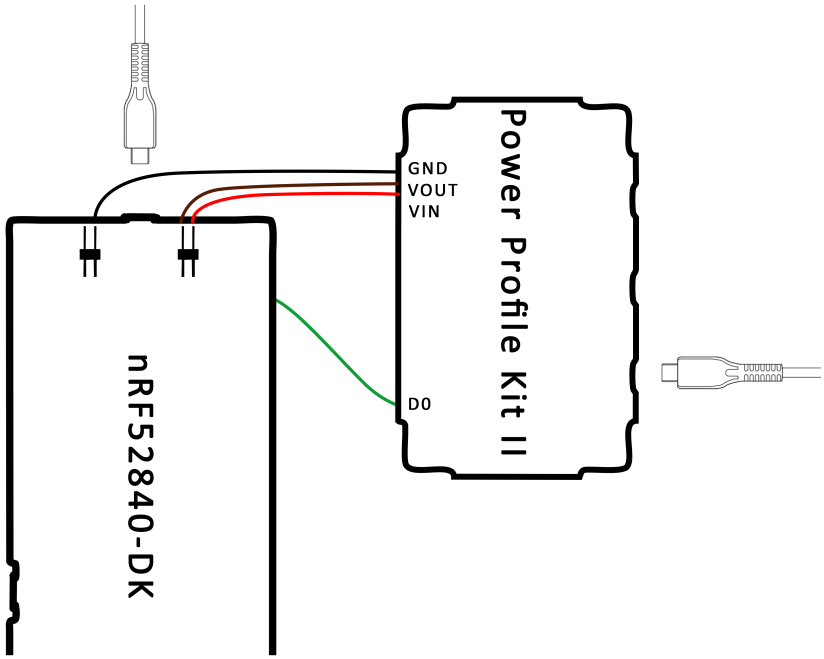


Figure 29: nRF-DK and PPK2 connection diagram

3.4. Wireless InSite

To deeply analyze BLE waves propagation we used a suite developed by RemCom providing 3D Ray-tracing and empirical models for analysis of site-specific radio wave propagation and wireless communication systems coming in different subscription packages. [9] Thanks to its processing capabilities it provides efficient and accurate predictions in Outdoor and Indoor Scenarios with no restriction on geometry shape or transceiver height. This software allows us to build a custom situation importing the needed 3d objects also

with a different material for every single one of them. One of its pros is that based on the chosen material various characteristic are taken into account changing reflection, absorption and diffraction properties leading to a much more accurate simulation.
It theoretically allows us to use an infinite number of transceivers with a wide variety of common Antennas stored in a library or we can even design a custom one using XFdtd, an external tool still provided by Remcom.
Frequencies ranging from 200MHz to 100 GHz can be analyzed obtaining as results received power and path loss, propagation path, reception latency, time and delay of arrival, electric field magnitude and phase and much more.

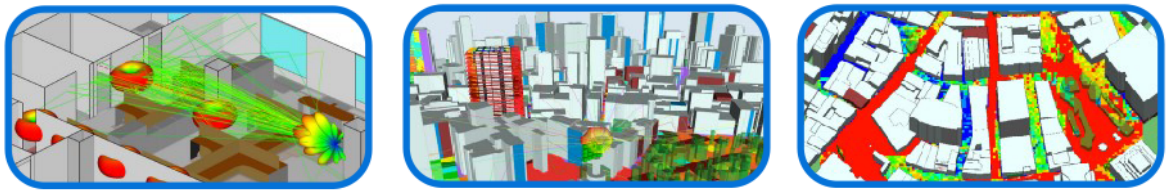


Figure 30: Wireless Insite Indoor and Outdoor coverage predictions examples

4 | Firmware

In this section we will deeply analyze the firmware relative to the BLE modules (Both Central and Peripheral). The framework implemented is a pre-compiled stack produced by Nordic. The only drawback with this method is that, since only high level functions are available, the behavior of the boards is not completely controllable and the performances are strictly correlated with the stack implementation used.
Basically both of them during the boot up phase load the peripheral IP from flash and then initialize the BLE module based on it, after this phase the peripheral device stay in advertising mode waiting until a central is establishing a connection, then, when they are connected their only job is to wait for a Serial message with its character terminator and sending it to the other device.
Some commands have been chosen to be able to program the devices during run time changing their IP, status and functionality.

4.1. Transmitter

The functioning of the peripheral board (sensor node) can be seen as a Finite State Machine (FSM) with basic functional state (Fig. 31):

- **OFF**: the device is OFF and isn't draining current from the battery.
- **ON**: this state is the startup one where the nodes' configuration is performed including BLE stack parameters, power management and external interfaces.
- **ADV**: after the configuration phase is finished the peripheral starts broadcasting advertising packets waiting for a connection request from a central device.
- **CON**: when a response is received a connection is initialized and started and the peripheral uses the Nus service to exchange data with the central until a disconnection happens.After a connection is established the peripheral must subscribe to the notify property of the NUS service in order to be able to be notified when a message is sent by the central. During this phase the device is relying messages received on

the serial interface to the connected central via BLE.

During every working phase the device is continuously listening for serial commands from the UART interface that can change some parameters to perform different tests and the booting address.

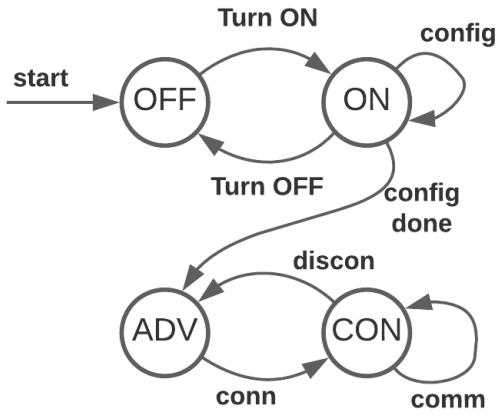


Figure 31: Peripheral Diagram Flow

4.1.1. Configuration

During the configuration phase the operations executed can be seen in Fig.32. During the *Peripherals Initialization* the configured ones are:

1. *UART1*: for PC connection and as the protocol to reply via BLE
2. *TIMER1*: for debugging purpose and required by the SoftDevice
3. *LEDs and BUTTONs*: for debugging purposes
4. *FLASH STORAGE*: it is used to store some parameters used during the configuration phase, in detail connection interval, default address and packet size.
5. *GPIO*: for an easier power measurement
6. *IP*: an IP saved inside the flash is loaded up to be used as BLE default address

In *BLE stack* we allocate the needed resources to save BLE parameters and to reserve memory for its functioning.
Inside the *Power Management* part event handlers are registered.

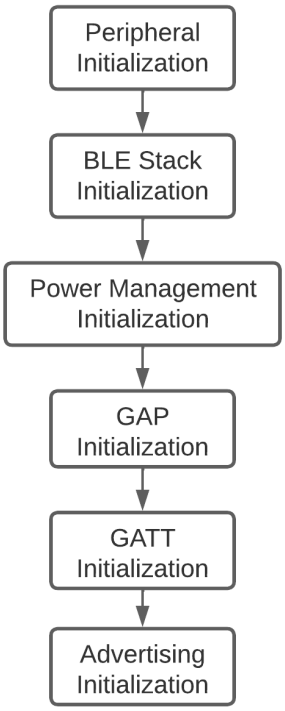


Figure 32: Peripheral configuration workflow

GAP initialization sets connection intervals' values as shown in table 10 that will be exchanged during connection establishment for connection monitoring.

Moving on to the *GATT* phase the module for the events' handling is declared and the

GAP parameters	
Device Name	BLE_UART
Security	None
Minimum Connection Interval	$20ms$
Maximum Connection Interval	$75ms$
Slave Latency	0
Connection Timeout	$4s$

Table 10: GAP parameters

required services are registered, more in detail the two chosen ones are battery (BAS) and Nordic Uart (NUS). Since both of them are SIG-adopted their 128-bit UUID (Universally Unique Identifier) is used.

The last step is the *Advertisement* stage where I set an advertising interval of 40ms and a timeout of 180s to stop if a receiver isn't present. It can be seen the we are not advertising services, this is done to lower advertising duration and relative consumption. In

fact our central device is choosing the device to connect using the IP address saved inside the flash. In figure 33 we show the structure of a real advertising packet. Implemented services aren't needed because the connection criterion used is IP matching. The device is obviously put in scannable and connectable mode to be available to all the centrals in proximity.

```
Bluetooth Low Energy Link Layer
    Access Address: 0x8e89bed6
  Packet Header: 0x1660 (PDU Type: ADV_IND, ChSel: #2, TxAdd: Random)
      .... 0000 = PDU Type: 0x0 ADV_IND
      ...0 .... = Reserved: 0
      ..1. .... = Channel Selection Algorithm: #2
      .1.. .... = Tx Address: Random
      0... .... = Reserved: 0
      Length: 22
    Advertising Address: fe:b2:6e:51:e9:69 (fe:b2:6e:51:e9:69)
  Advertising Data
    Flags
        Length: 2
        Type: Flags (0x01)
        000. .... = Reserved: 0x0
        ...0 .... = Simultaneous LE and BR/EDR to Same Device Capable (Host): false (0x0)
        .... 0... = Simultaneous LE and BR/EDR to Same Device Capable (Controller): false (0x0)
        .... .1.. = BR/EDR Not Supported: true (0x1)
        .... ..0. = LE General Discoverable Mode: false (0x0)
        .... ...1 = LE Limited Discoverable Mode: true (0x1)
    Device Name: Nordic_UART
        Length: 12
        Type: Device Name (0x09)
        Device Name: Nordic_UART
    CRC: 0x0e7abb
```

Figure 33: Advertising packet structure collected using Wireshark software

4.2. Receiver

The purpose of the central board (Gateway) is to search for peripheral in advertising mode to then connect to them and start communicating and exchanging packets. It almost share the same structure such as the peripheral that can be resumed in the finite state machine in figure 34. In this case the four basic steps are:

- **OFF**: the device is OFF.
- **ON**: same as the BLE peripheral the node configure its peripheral, BLE stack and interfaces.
- **SCAN**: after the configuration is done the board starts the scanning phase searching for peripherals with the predefined IP and implementing NUS and Battery service.

- **CON**: when a matching peripheral is found a connection request is sent to start the transmission. After a connection is established the central must subscribe to the notify property of the NUS service in order to be able to be notified when a message is sent by the peripheral. Inside this state the device is relying messages received on the serial interface to the connected peripheral via BLE.

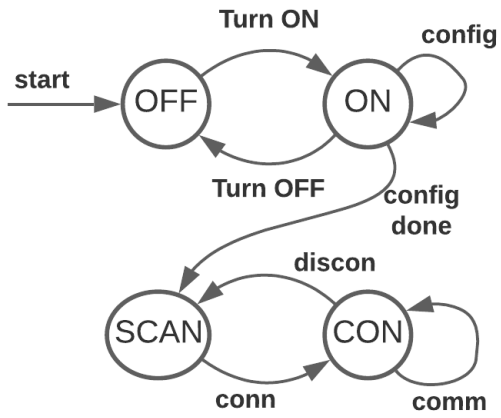


Figure 34: Central Diagram Flow

4.2.1. Configuration

The configuration process is shown in figure 35. During the first phase the following peripheral are configured:

1. *UART1*: for PC connection and as protocol to be replied via BLE
2. *TIMER1*: for debugging purpose and required by the SoftDevice
3. *LEDs and BUTTONs*: for debugging purposes
4. *GPIO*: for an easier power measurement, in detail we used pin 27 as state Pin raising and clearing it to know which state is the micro working in.

During *BLE Stack Initialization* the resources needed for BLE functioning are allocated (RAM and Event Handler).
Inside the*Power Management* part event handlers are registered.
DB discovery module is used by the application to perform discovery of services and characteristics at the peer server. When a service is found this module is in charge of investigating all its characteristics and properties.

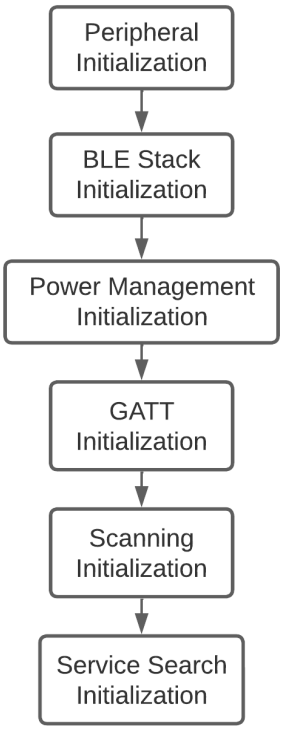


Figure 35: Central configuration workflow

Lastly the *Scan Configuration* sets scan parameters to be used: shown in table 11.

SCAN parameters	
Scan Interval	$160ms$
Scan Window	$80ms$
Scan Timeout	∞
Active Scan	ON
Physical Link	$1Mbps$

Table 11: SCAN parameters

4.3. Command Analysis

To test the board functioning we needed to real time change the BLE stack parameters, normally this would mean reprogram it to apply changes, but to make it faster I defined a list of commands that can be sent through UART and BLE. Using them we can change the boot-up IP, connection interval, ATT MTU, physical rate and Data Length.

The only problem is that the boards is continuously receiving messages to rely them so we needed to differentiate commands from usual messages: this is why I decided to use the character '$' that defines the start of a new command and then check the correct syntax (length and values).
In table 12 the list of the available command is shown.

Inside both serial and BLE string reception event handler we check if a command is

Command	Function
$01;xxxxxxxx;	IP: xxxxxxxx defines the new value
$02;xxxx;	CON INT: xxxx defines the new value
$03;xxx;	ATT MTU: xxx defines the new value
$04;x;	PHY RATE: x defines the new value

Table 12: GAP parameters

received and if it happens a flag is checked. Then, inside the main loop the command is analyzed and implemented with a switch structure.

4.4. Communication Process

Using Wireshark software and an nRF-52840DK we created a BLE sniffer to understand what was happening between our devices.
The communication process, outlined in figure 36 where we can see the first advertising packet that, when received, is answered with a connection request starting the connection procedure highlighted inside the blue box. It starts by sending the default connection parameters (interval, channel map, selection algorithm) and asks for Link Layer feature and MTU size. Going on it reads LL length and BLE available version, the last part of the connection phase reads GATT profiles and services implemented investigating every characteristic and their relative read, write and notify propriety.

When the connection has been established devices keep communicating every connection interval: even if there is no data to be sent the central device (MASTER) sends a packet and the peripheral (SLAVE) responds (Fig. 37). This serve as a handshake to maintain the wireless connection even if no application data needs to be sent for extended durations. To receive live serial data using NuS service the central device must subscribe to its TX characteristic enabling the notify property, this is done by the Master writing the first bit of the Client Characteristic Configurator Descriptor (CCCD), the Slave responds with an ACK and starts sending packets.

Time	Source	Length	Info
7,857	PERIPHERAL	45	ADV_IND
7,858	CENTRAL	60	CONNECT_IND
7,908	MASTER	35	LL_FEATURE_REQ
7,908	SLAVE	33	Rcvd Exchange MTU Request
7,957	SLAVE	35	LL_FEATURE_RSP
7,957	MASTER	33	Sent Exchange MTU Response
8,005	MASTER	33	Sent Exchange MTU Response
8,006	MASTER	32	LL_VERSION_IND
8,054	MASTER	32	LL_VERSION_IND
8,054	SLAVE	35	LL_LENGTH_REQ
8,055	MASTER	37	Sent Read By Group Type Request, GATT Primary Service Declaration
8,103	MASTER	37	Sent Read By Group Type Request, GATT Primary Service Declaration
8,103	MASTER	35	LL_LENGTH_RSP
8,152	MASTER	35	LL_LENGTH_RSP
8,152	SLAVE	32	Control Opcode: LL_VERSION_IND
8,201	SLAVE	44	Rcvd Read By Group Type Response, Attribute List Length: 2, Generic Access Profile, Generic Attribute Profile
8,201	MASTER	38	Control Opcode: LL_CONNECTION_UPDATE_IND
8,249	MASTER	38	Control Opcode: LL_CONNECTION_UPDATE_IND
8,250	MASTER	37	Sent Read By Group Type Request, GATT Primary Service Declaration, Handles: 0x000b..0xffff
8,298	MASTER	37	Sent Read By Group Type Request, GATT Primary Service Declaration, Handles: 0x000b..0xffff
8,347	SLAVE	52	Rcvd Read By Group Type Response, Attribute List Length: 1, Nordic UART Service
8,395	MASTER	37	Sent Read By Type Request, GATT Include Declaration, Handles: 0x0001..0x0009
8,396	SLAVE	35	Rcvd Error Response - Attribute Not Found, Handle: 0x0001 (Generic Access Profile)
8,444	MASTER	37	Sent Read By Type Request, GATT Characteristic Declaration, Handles: 0x0001..0x0009
8,445	SLAVE	60	Rcvd Read By Type Response, Attribute List Length: 4, Device Name, Appearance, Peripheral Preferred Connection Parameters, Central Address Resolution
8,493	MASTER	37	Sent Read By Type Request, GATT Characteristic Declaration, Handles: 0x0009..0x0009
8,494	SLAVE	35	Rcvd Error Response - Attribute Not Found, Handle: 0x0009 (Generic Access Profile: Central Address Resolution)
8,542	MASTER	37	Sent Read By Type Request, GATT Include Declaration, Handles: 0x000b..0xffff
8,542	SLAVE	35	Rcvd Error Response - Attribute Not Found, Handle: 0x000b (Nordic UART Service)
8,597	MASTER	35	Control Opcode: LL_LENGTH_REQ
8,597	MASTER	37	Sent Read By Type Request, GATT Characteristic Declaration, Handles: 0x000b..0xffff
8,604	MASTER	37	Sent Read By Type Request, GATT Characteristic Declaration, Handles: 0x000b..0xffff
8,612	SLAVE	35	Control Opcode: LL_LENGTH_RSP
8,619	SLAVE	53	L2CAP Fragment Start
8,627	SLAVE	47	Rcvd Read By Type Response, Attribute List Length: 2, Nordic UART Tx, Nordic UART Rx
8,634	MASTER	37	Sent Read By Type Request, GATT Characteristic Declaration, Handles: 0x000f..0xffff
8,635	SLAVE	35	Rcvd Error Response - Attribute Not Found, Handle: 0x000f (Nordic UART Service: Nordic UART Rx)
8,642	MASTER	35	Sent Find Information Request, Handles: 0x0010..0xffff
8,642	SLAVE	36	Rcvd Find Information Response, Handle: 0x0010 (Nordic UART Service: Nordic UART Rx: Client Characteristic Configuration)
8,649	MASTER	35	Sent Find Information Request, Handles: 0x0011..0xffff
8,650	SLAVE	35	Rcvd Error Response - Attribute Not Found, Handle: 0x0011 (Nordic UART Service: Nordic UART Rx: Unknown)
8,657	MASTER	38	Control Opcode: LL_CONNECTION_UPDATE_IND

Figure 36: Connection Establishment

8,649	MASTER	26	Empty PDU
8,649	SLAVE	26	Empty PDU
8,656	MASTER	26	Empty PDU
8,657	SLAVE	26	Empty PDU

Figure 37: Typical keep alive messages

4.5. Wireless InSite

One of the major problem with electromagnetic wave they have big limitations when working in metallic environment so we wanted to understand if the connection was achievable and if changing sensor nodes' and gateway's placements made sense relative to the goal of improving power management.

To perform the simulation inside Wireless Insite we first had to build a 3D model of the T3000e. We looked up for this freight wagon's dimensions and they are shown in table 13 [15] and produced an train's unloaded model. However our goal was to characterize and compare transmission results also in case of a loaded wagon: T3000e can carry a

maximum of two maritime containers, so we built a 3D container model with standard ISO dimensions ($12.192m$x$2.438m$x$2.591m$) and nominal load of $27600kg$ [14] to be able to build a loaded train version and used both of them during simulation. We can see the models in figure 38.

Dimension	Value
Length	$34,2m$
Loading length	2 x $16,185m$
Height of loading plane	$1,135 \div 1,155m$
Boogie distance	$28,4m$

Table 13: T3000e dimensions

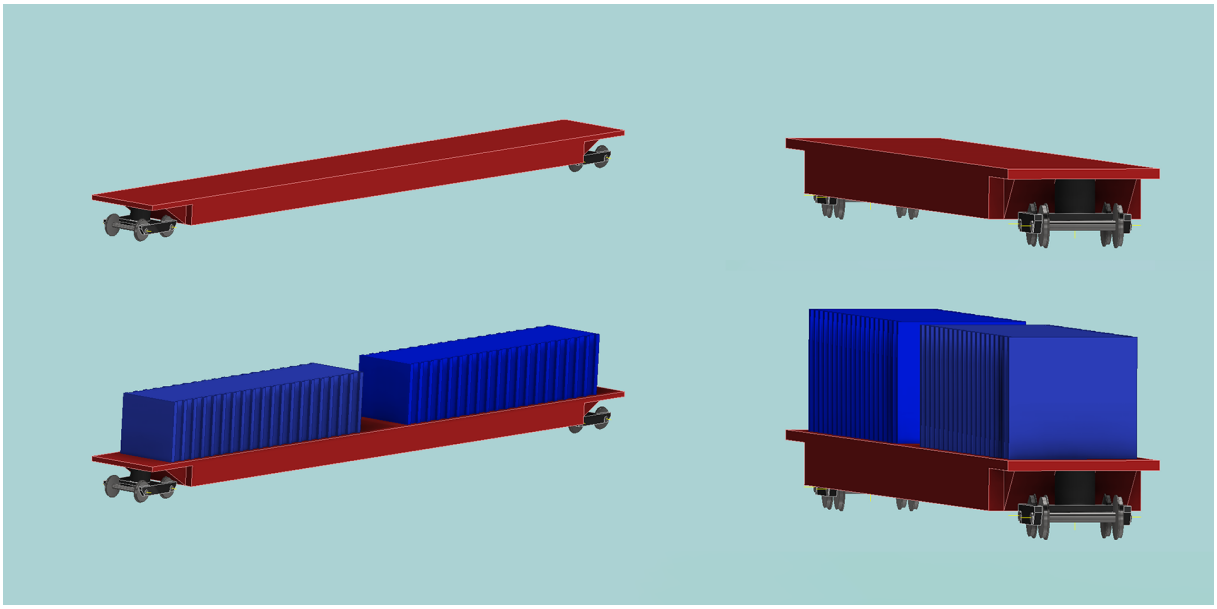


Figure 38: 3-D train models used

Going on I exported whole system's .stl file and moved on; to produce a simulation I opened Wireless Insite and followed these steps:

1. Imported train model to use it for the unloaded simulation and base above which i placed two containers files for the loaded situation. For all the models the composing material chosen was metal.

2. In urban and outdoor environments it is mandatory to import or generate a *Terrain*. I didn't have a prebuilt terrain so I created a rectangular one with 34x3m dimensions and used 2.4 GHz concrete as characteristic material.

3. Since propagation paths and properties depends on the transmitted wave length Wireless Insite requires base *Waveform* parameters definition. So I defined a sinusoidal waveform with frequency of 2.4GHz that is the carrier wave used in Bluetooth protocol.

4. We also had to define which type of *Antenna* were transceivers equipped with. The used modules are the BT840F and they come with a PCB dipole one. Its omnidirectional gain radiation pattern is shown in figure 39.

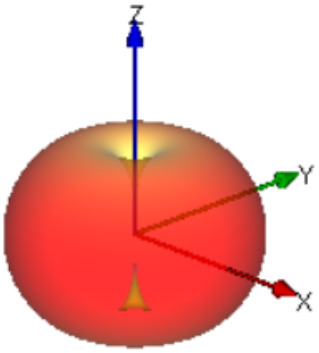


Figure 39: 3D omnidirectional gain radiation pattern

5. All the *Transceivers* used must be now defined and placed. In our configuration there were 2 transmitters (sensor node) and 1 receiver (gateway) with all of them featuring the same previously defined antenna.

6. Inside Wireless Insite I generated a *Study Area* defining the simulation area, simulation model and simulation parameters (number of reflections, diffractions, transmissions and path lower threshold). The propagation model used is X3D. It provides a highly accurate, full 3D propagation model capable of running on a graphics processing unit (GPU) and using multi-threading for fast runtimes. X3D does not have any restrictions on geometry shape or transceiver height.

7. Last step is to select the requested results (path loss/gain, propagation path, received power), start the *Simulation* and wait for its completion to analyze them.

Until now the situations analyzed were done in open field with no obstacles other than the train apart its cargo, however each car also spends part of its life stationary in a warehouse, through tunnels, or with objects around. This is the reason why I also built two more trains placed beside the base one and a tunnel to simulate both remaining

situations.
In figure 40 we show Wireless Insite interface seen at the end of the configuration process, we can see the loaded train version, the concrete terrain, two sensor nodes (BC and MBP) and the receiver (Gateway) and the simulation area used, i.e., the area in the shape of a volume with highlighted green edges.

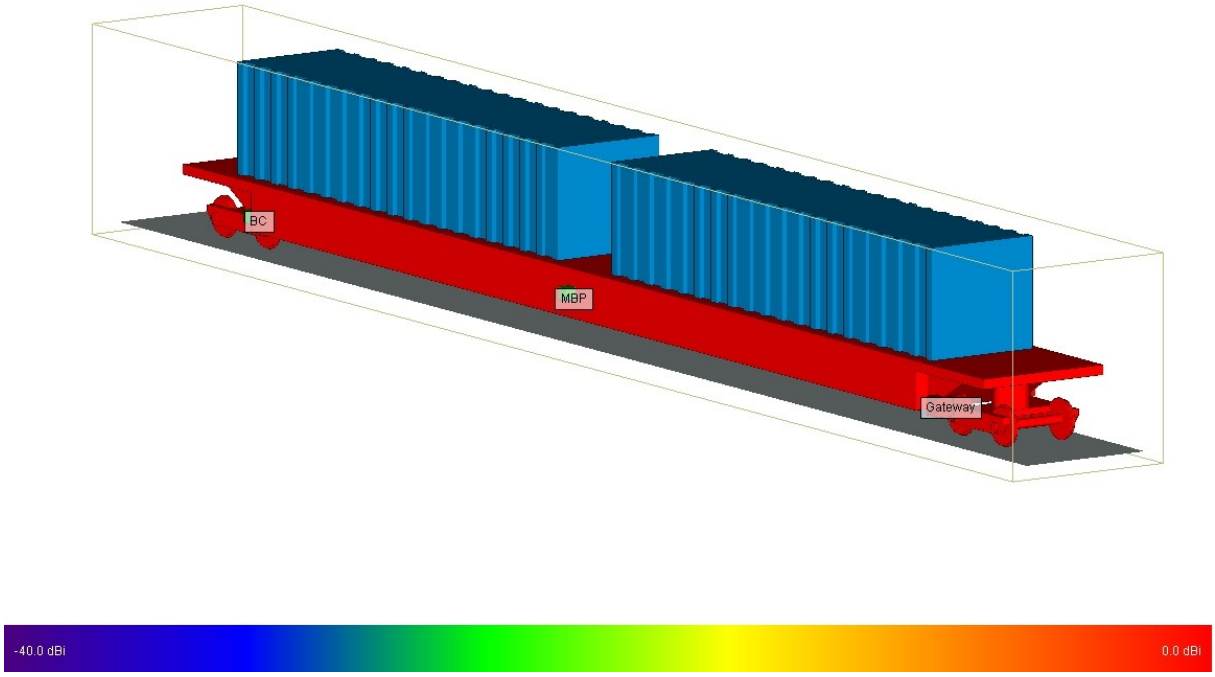


Figure 40: Wireless Insite final interface

5 | Results

Inside this chapter we will first see and then analyze the results obtained regarding BLE performances in matter of throughput and energy consumption related to different chosen parameters and compare them with the theoretically obtained ones. Later on the focus will move onto the BLE wave propagation simulation part using Wireless Insite software.

5.1. Power Consumption and Throughput

During testing we mainly focused on the transmitters (peripheral) since they are the standalone sensors, not connected to a wired power source (see 1.3) and so their software and whole implementation gravitates around power consumption and optimization.
We proceeded dividing BLE communication in its most important phases:

- **Advertisement**
- **Connection**
- **Transmission**

And analyzed which parameters were influencing energy consumption and data rate.
Inside the first two following sections we will talk about how advertising and connection are related to energy consumption, while in the last section we will analyze both energy and throughput performances during transmission phase.

5.1.1. Advertisement

As said the transmitter works in different state, the first one is the advertising phase where it is advising its presence to the reachable gateway.
We first measured the current consumption during an advertising phase and deeply analyzed what was happening and which operation was the micro processing.
In figure 41 we can see the data gathered during a process and its instantaneous power consumption with a time resolution of $10\mu s$.

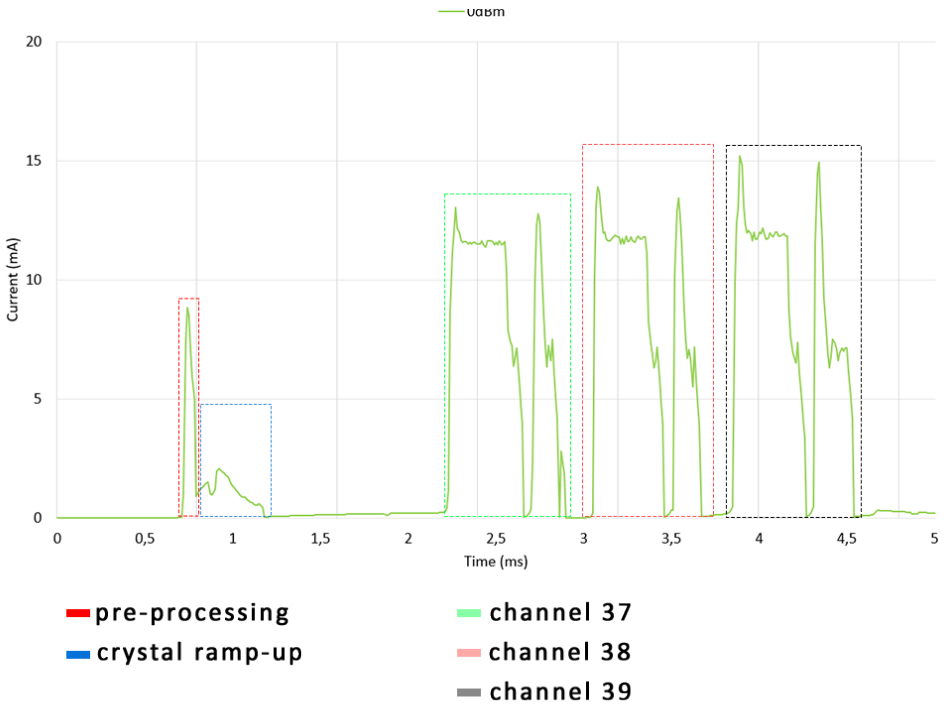


Figure 41: Current consumption during one advertising process

Analyzing it we can appreciate 4 main phases: pre-processing, where the microprocessor is loading up all the parameters needed and the 3 channels used for advertising (37, 38, 39). Every single advertising event where the radio is working can be decomposed in 3 phases shown in figure 42: transmission , switching between reception configuration and in the end the receiving part.

During the transmission phase the device is advertising its presence sending the configured data, then, since this type of advertisement is **scannable**, it switches to the reception configuration to listen if a central sends a scan request. Differently to a non connectable advertisement mode (Beacon Example) the device will not switch to the reception part during advertising due to the fact that it isn't excepting a scan request (shown in figure 43). This behaviour lead to shorter advertisement operations and consequently reduced current consumption, unfortunately in our case it was mandatory to use the connectable configuration.

We implemented different **transmitting powers** (-40dBm to 8 dBm) and carried out various current measurements (Fig 44), compared them to understand if lowering it and losing range is worth relatively to the energy saving gained.

In fact lower transmission power in application requiring high distances with disturbance

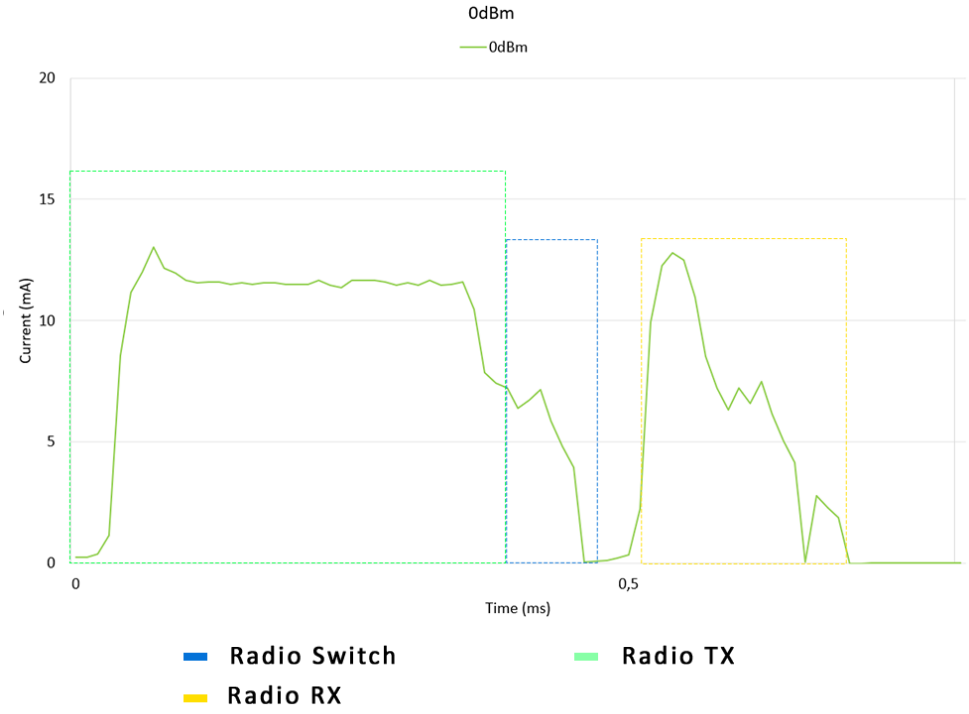


Figure 42: Current consumption during advertising focused on 1 particular channel

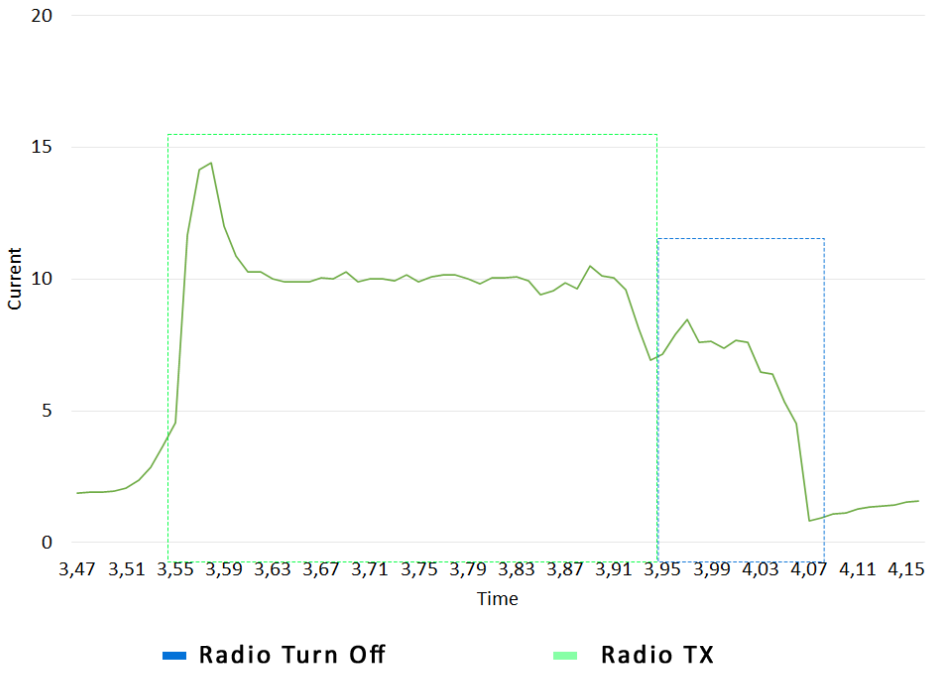


Figure 43: Current consumption during non connectable advertising focused on 1 particular channel

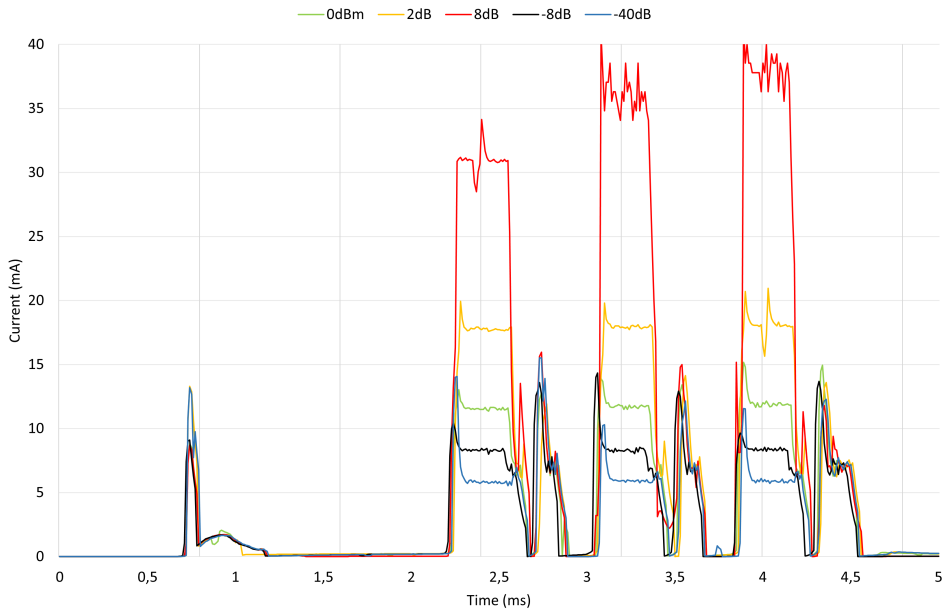


Figure 44: Advertising current consumption with different Tx powers

presence may cause more frequent errors and re-transmission leading to lower throughput and an energy consumption increment. However we will talk about later in the propagation simulation chapter about the required transmitting power calculated, here we only deal with current consumption.
Using the collected data, the average current and the battery charged drained during one advertising phase have been calculated and are shown in figure 46. It is obvious that increasing the output power leads to an increase of the energy consumption and the battery charge drained.
All the calculation done are based on a single advertising event, apart from TX power level we can change advertising interval and duration with the goal to decrease the power consumption. It is adjustable from $20ms$ to $10.24s$ (in non connectable function the minimum is 100 ms). In detail increasing advertising interval from $100ms$ to $1s$ will decrease the average current consumption by 93%.
After all the measurement we wanted to double check them. Luckily Nordic Semiconductor makes available an Online Power Profile [5] able to simulate BLE energy consumption depending on the chosen SoC and protocol parameters so we used this tool to compare our results with those declared (figure 45) appreciating their great similarity.

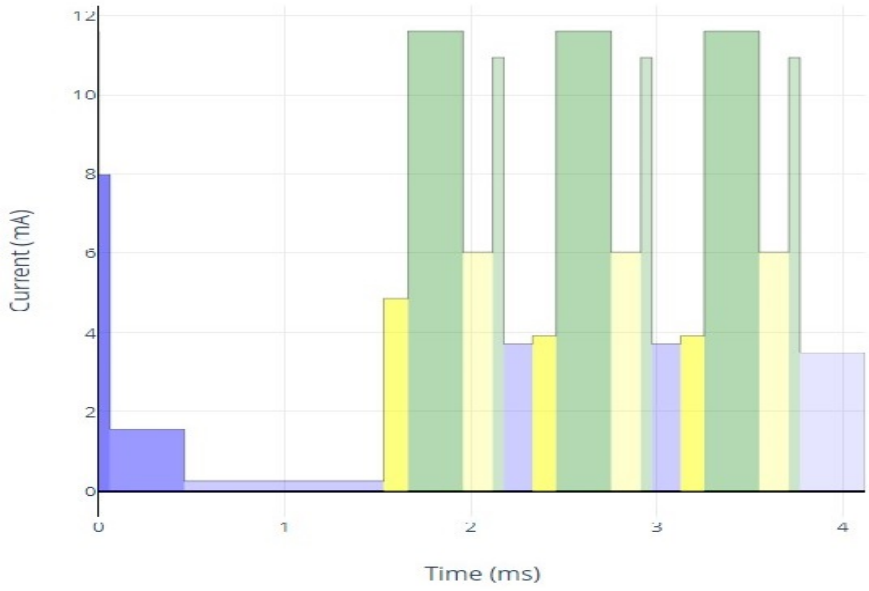
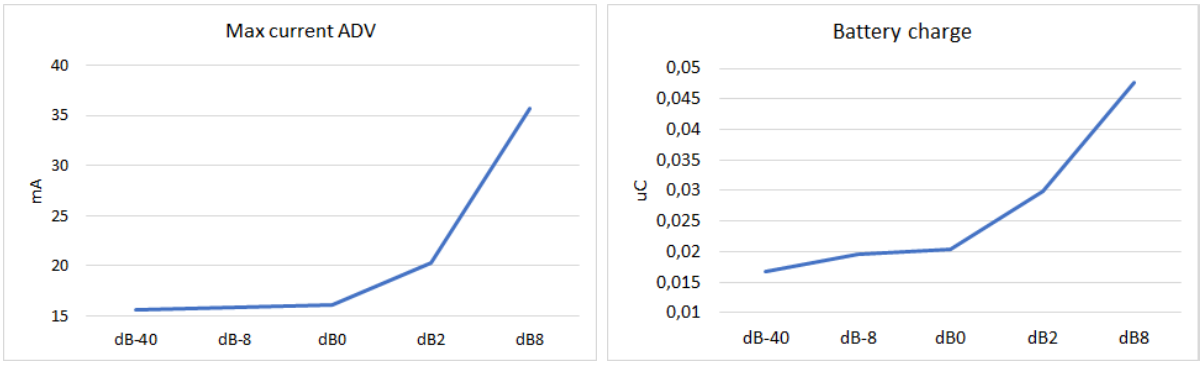


Figure 45: Simulated current consumption during one advertising process



(a) (b)

Figure 46: Average current **(a)** and Battery charge drain **(b)** during advertising

5.1.2. Connection

Testing the connection phase we obtained the results shown in figure 47. Connection is only possible when connectable advertising packets are sent and we can see that after pre-processing and advertising on channel 37 phase a gateway's response is received and answered with the needed parameters to successfully establish a connection.

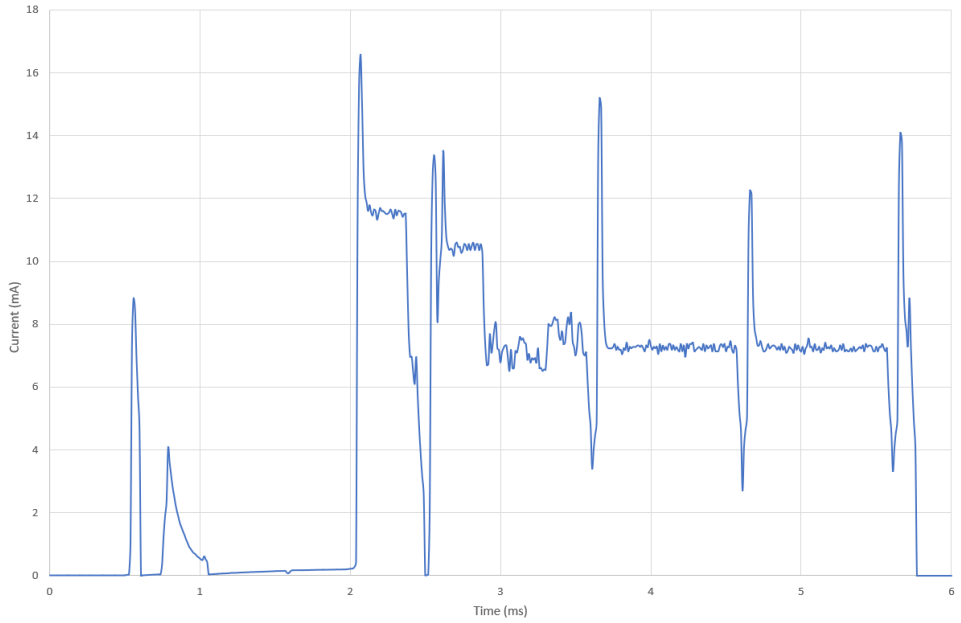


Figure 47: Current consumption during one connection sequence

5.1.3. Transmission

In terms of data rate the most important phase is the transmission one. We tested it using different parameters to characterize maximum data throughput and Energy consumption. We worked with 3 different focus: ATT MTU, Connection Interval and Physical Rate.
The problem is that the data rate is strongly variable when instantaneous measurements are performed due to disturbances and transmission errors, so i chose to use a relatively long interval to perform measurements and calculate its average. In particular I decided to build a vector of 1024 Kbytes, wait for a transmission flag to start sending it from the peripheral device and, when the transmission is done, the central side uses the elapsed time to calculate the average throughput and energy consumption.
We also calculated the theoretically maximum achievable throughput and used it as benchmark to check our results' correctness.

Calculation

We analyzed the transmission phase and its on-air traffic. In figure 48 we can see a complete period of an effective 1 packet communication. It starts with a request (R) coming from the central (1 every connection interval) to which the peripheral decide to reply or not depending on slave latency parameter. Then there is a pause (IFS should be 150μs) before the peripheral responds with the requested data. Lastly another IFS

Figure 48: BLE 5.0 transmission sequence

happens.
R packets sent by the master are empty PDU so their length is determined by: Preamble, Access Header, Header and CRC and easily calculated:

$$PDU = 1 + 4 + 2 + 3 = 10byte \tag{5.1}$$

Packet duration is related to physical speed, below we shown an example with a PHY layer of 2Mbps (fastest).

$$R_D = \frac{Length}{PhysicalSpeed} = \frac{10 * 8}{2M} = 40\mu s \tag{5.2}$$

Instead T packets can also be empty if there isn't new data to be sent, but we are going to analyze their maximum possible length. In this case it share the same structure as the empty PDU case but it has a Payload and MIC before CRC. Now the length and duration are:

$$PDU = 1 + 4 + 2 + 251 + 4 + 3 = 265byte \tag{5.3}$$

$$T_D = \frac{Length}{PhysicalSpeed} = \frac{265 * 8}{2M} = 1060\mu s \tag{5.4}$$

One complete transmission period is then composed of 2 IFS, 1 R and 1 T packet:

$$Duration = R + IFS + T + IFS = 40 + 150 + 1060 + 150 = 1400\mu s \tag{5.5}$$

and the throughput:

$$Throughput = \frac{Payload}{Duration} = \frac{251bytes}{1040\mu s} = 1.4Mbps \tag{5.6}$$

Data Length

In this case i used 8 different Packet Size values changed on run time, calculated and measured each data rate and energy consumption.
We can see that twice the Packet size doesn't mean double throughput because more

packets still need longer time to be sent:

$$Duration_{27} = R + IFS + \frac{(27+14)*8}{2Mbps} + IFS = 40 + 150 + 164 + 150 = 504\mu s \quad (5.7)$$

$$Duration_{256} = R + IFS + \frac{(251+14)*8}{2Mbps} + IFS = 40 + 150 + 1060 + 150 = 1400\mu s \quad (5.8)$$

However the larger is the packet size the smaller the ratio with the overhead thus increasing data throughput.

$$Throughput_{27} = \frac{27*8}{504\mu s} = 428,571Kbps < Throughput_{251} = \frac{251*8}{1400\mu s} = 1434,285Kbps \quad (5.9)$$

Results obtained keeping the same physical rate and GAP event length (see table 14) are shown in figure 49.

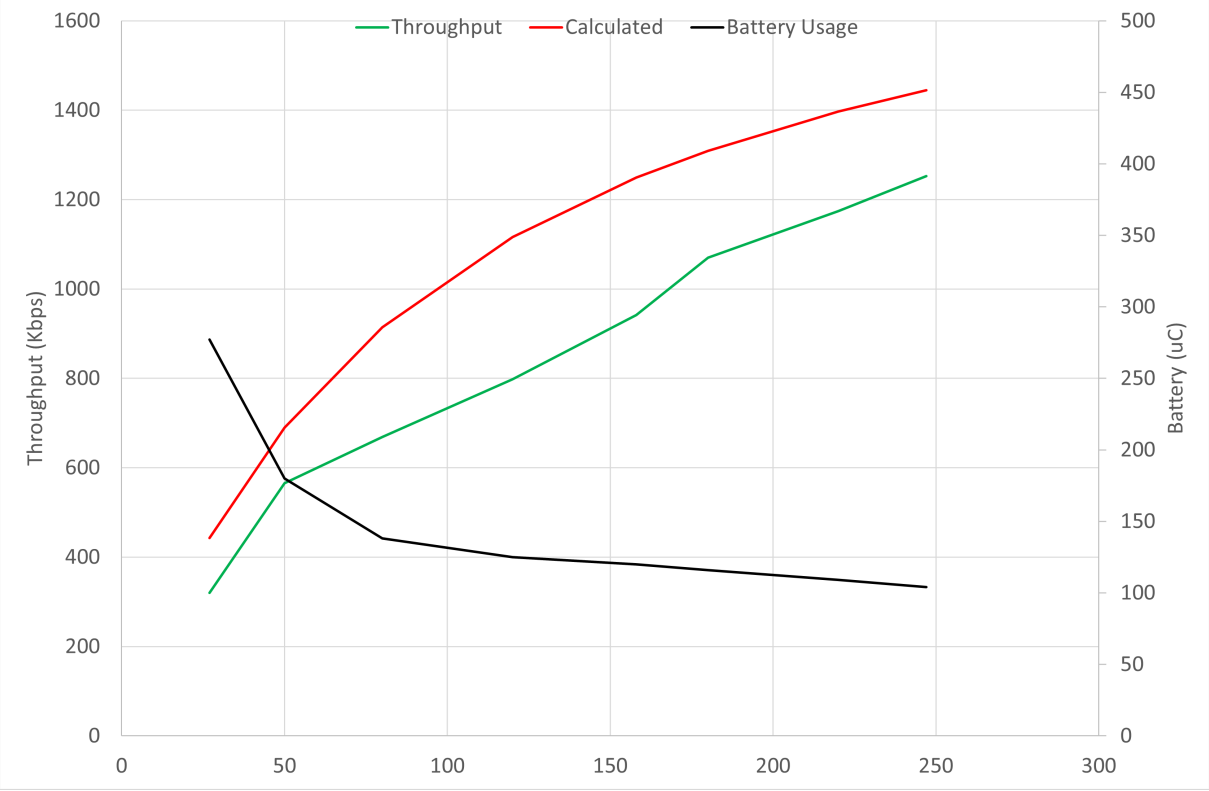


Figure 49: Data throughput and current consumption during transmission with various Packet Size

Parameters	Values
Packet Size	$27 \div 247 byte$
Physical Rate	$2 Mbps$
Connection Interval	$400 ms$
GAP Event Length	$400 ms$

Table 14: Parameters used varying Packet Size

Physical Rate

Regarding physical rate we tried the 3 available ones (2Mbps, 1Mbps, Coded) and we observed that when the physical velocity changes from 1M to 2M the data rate nearly doubles. It is not exactly twice due to the constant inter-frame spacing ($IFS = 150\mu s$ for them all). Below we show the calculation done and in figure 50 how they compare with the real life measurements.

$$D_{2M} = \frac{10 * 8}{2Mbps} + IFS + \frac{(251 + 14) * 8}{2Mbps} + IFS = 40 + 150 + 1060 + 150 = 1400\mu s \tag{5.10}$$

$$D_{1M} = \frac{10 * 8}{1Mbps} + IFS + \frac{(251 + 14) * 8}{1Mbps} + IFS = 80 + 150 + 2120 + 150 = 2500\mu s \tag{5.11}$$

$$D_{125K} = \frac{10 * 8}{125Kbps} + IFS + \frac{(251 + 14) * 8}{125Kbps} + IFS = 640 + 150 + 16960 + 150 = 17900\mu s \tag{5.12}$$

$$Thr_{2M} = \frac{251 * 8}{D_{2M}} = 1434,285Kbps \tag{5.13}$$

$$Thr_{1M} = \frac{251 * 8}{D_{1M}} = 803.2Kbps \tag{5.14}$$

$$Thr_{125K} = \frac{251 * 8}{D_{125Kbps}} = 112.18Kbps \tag{5.15}$$

Parameters	Values
Packet Size	$247byte$
Physical Rate	$2Mbps$, $1Mbps$, Coded ($125Kbps$)
Connection Interval	$400ms$
GAP Event Length	$400ms$

Table 15: Parameters used varying physical rate

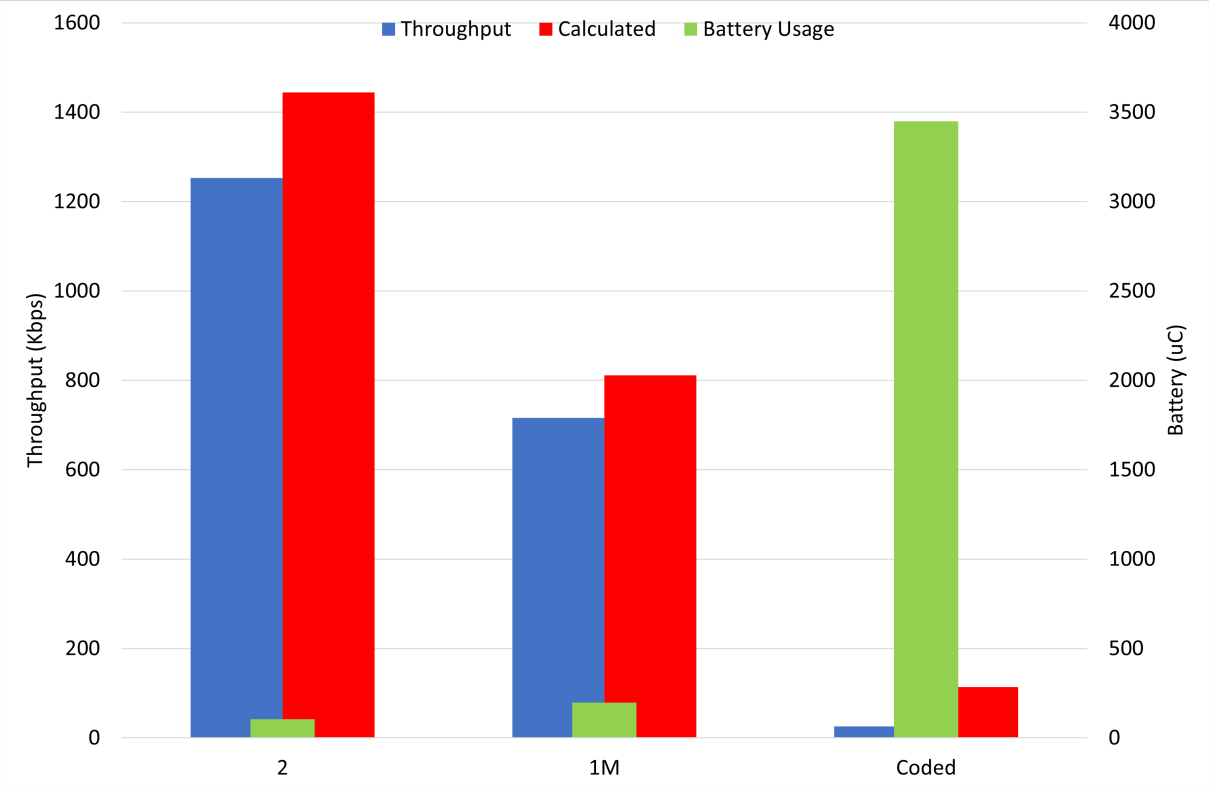


Figure 50: Data throughput and current consumption during transmission with various Physical Rate

Connection Interval

During this phase we tested 5 different connection interval values ranging from 7.5 to 400ms with other parameters as shown in table 16 and results can be seen in figure 53.

We used Unacknowledged data transfer, where, differently from Acknowledge data transfer, connection interval doesn't influence that much the throughput [12]. In fact when using the second one the sender sends data and has to wait till the next connection interval begins to receive central's response with an acknowledgement packet; while in our configuration any number of packets can be sent within one connection interval. The only

Parameters	Values
ATT MTU	$247byte$
Data Length	$251byte$
Physical Rate	$2Mbps$
Connection Interval	$7.5 \div 400ms$
GAP Event Length	$400ms$

Table 16: Parameters used varying connection interval

problem is that data packets must be aligned so that the end of the last packet coincides with the one of the connection interval.
In figure 51 we show an example where they aren't aligned and it is possible to send a maximum of 4 packets in one interval, the 5th packet does not fit and has to wait for the next connection interval leading to a throughput reduction.

To increase throughput we must avoid skipping packets due to the non synchronization

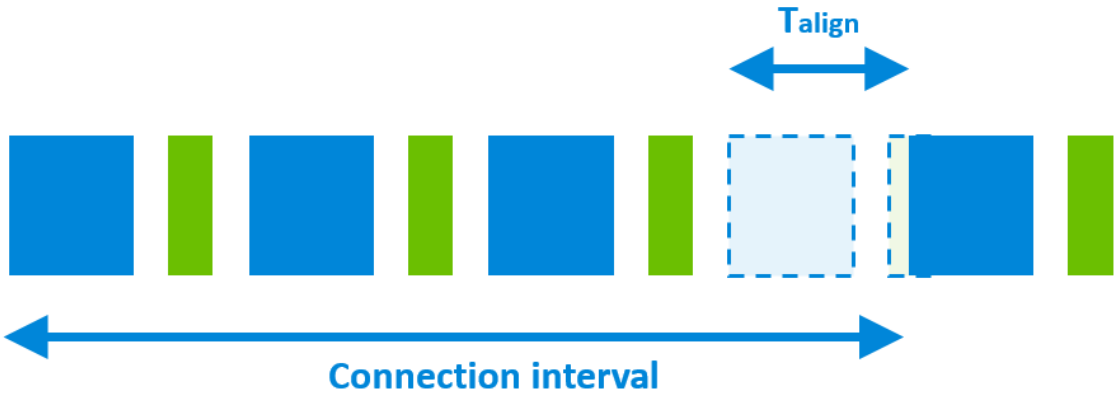


Figure 51: BLE Unacknowledged data transfer example

between connection interval and packet's duration so that the remaining time (T_{align}) becomes minimal.
We calculated the maximum number of packets that can be sent inside a connection event and the relative T_{align}:

$$PK_{1M} = \frac{ConnInter}{T_{1M}} = \frac{12000\mu s}{2500\mu s} = \lfloor 4.8 \rfloor = 4packets \tag{5.16}$$

$$T_{align} = ConnInter - PK * T_{1M} = 12000\mu s - 4 * 2500\mu s = 2000\mu s \tag{5.17}$$

Results are shown in figure 52. As we can see its tendency is described by a saw-tooth reaching its apex when the packet duration is an exact divider of the interval value: with

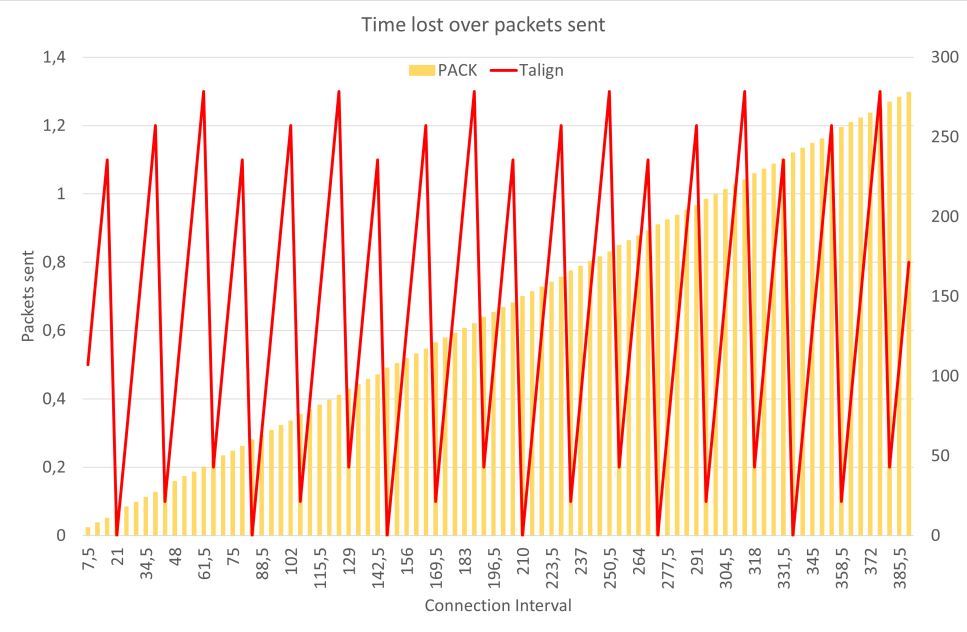


Figure 52: Packet sent and T_{align} with different connection intervals

a 21ms connection interval it can send a maximum of 15 packets,84 ms 60 packets, 147ms 105 packets and so on...
Since lowering T_{align} and reducing the ratio with the packet sent means an higher throughput it would seems obvious to choose a longer connection interval however the following applies:

- Too many packets may be queued while waiting for the next connection interval and you can run out of memory.
- You may have to wait much more to re-synchronize if the connection breaks because of consecutive CRC errors. Long connection interval is hence not recommended in noisy environments where CRC errors are expected.

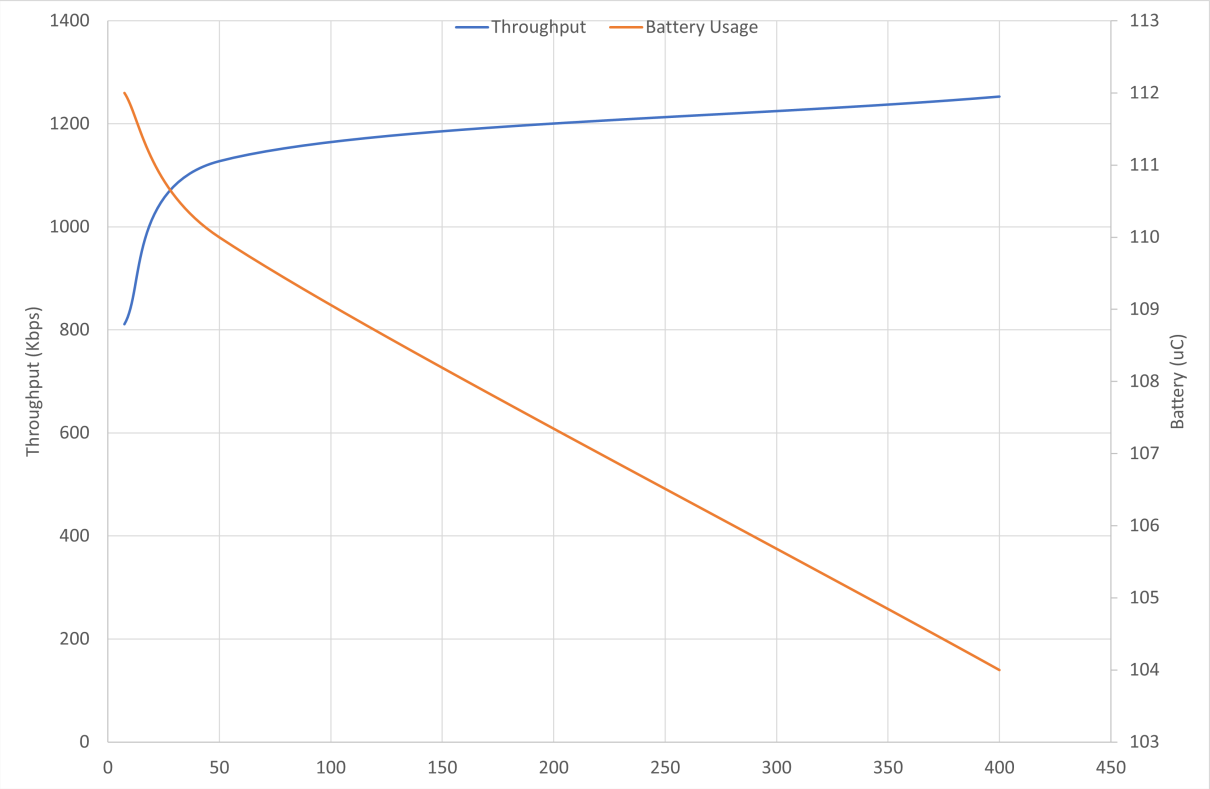


Figure 53: Data throughput and current consumption during transmission with various Connection Interval

5.2. Propagation Simulation

We simulated BLE waves propagation from the two placed sensor nodes (MBP and BC) to the Gateway in both loaded and unloaded case studies while gathering transmission loss, propagation paths and total received power.

Unloaded train

First we are going to see the unloaded scenario where the only wagon is present: we gathered all the feasible propagation paths and their path loss; the best three ones are shown in table 17.

	MBP	BC
Path #1	$-71.03dBm$	$-107.7dBm$
Path #2	$-82.37dBm$	$-110.1dBm$
Path #3	$-85.38dBm$	$-121.9dBm$

Table 17: Path Loss in outdoor unloaded scenario

Analyzing them we can see a drastic difference in best path's received power between MBP and BC sensor nodes, this is firstly because BC's distance to the gateway is doubled with respect to MBP one; then BC is almost completely shielded in its position and its propagation path has to perform a greater number of reflection before reaching the gateway.
Figure 54 shows the three best routes described in the table above from a viewpoint below the train. it is easy to see that the paths coming from the BC exploit more diffraction phenomena than those from the MBP, this causes greater path losses that added to the bigger distance of the first sensor drastically reduce the power of the received signal.

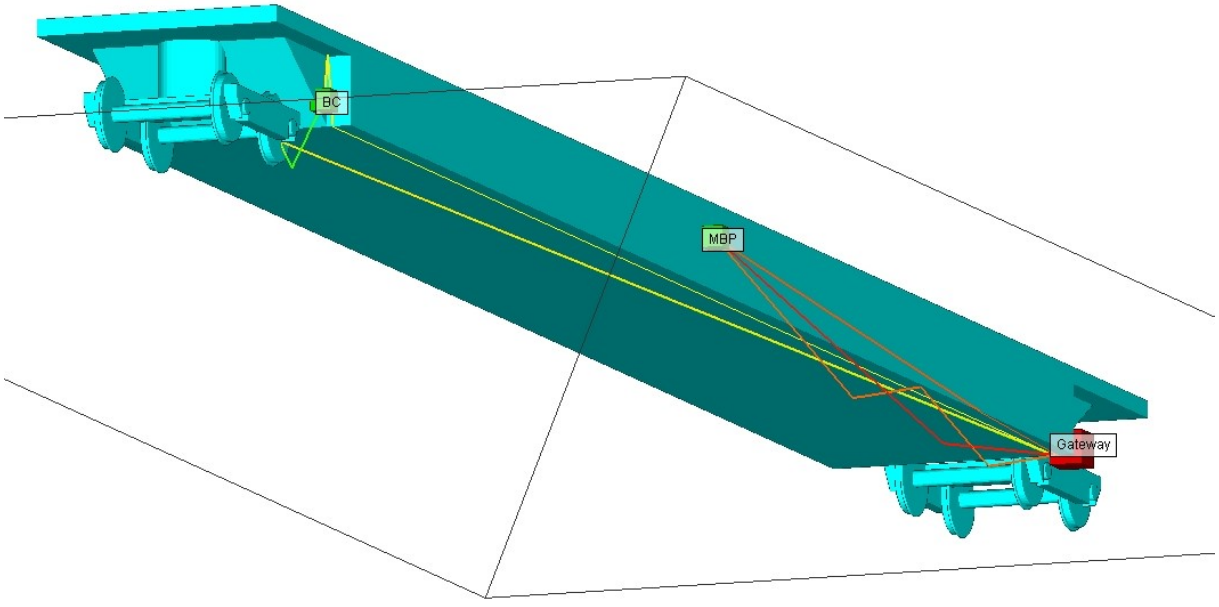


Figure 54: BLE wave propagation simulation in Wireless Insite for the unloaded scenario

Loaded train

Moving on we analyzed the same parameters when the train is loaded with two maritime containers.
As it can be seen we obtained exactly the same results as the unloaded case: this happens because any kind of load doesn't act as a wall for the waves since it can't be placed in propagation paths' way and therefore does not create additional electromagnetic wave propagation phenomena affecting transmission's quality.
In figures 55 and 56 we can see the three best propagation paths in the loaded case, respectively from the braking cylinder and main brake pipe.

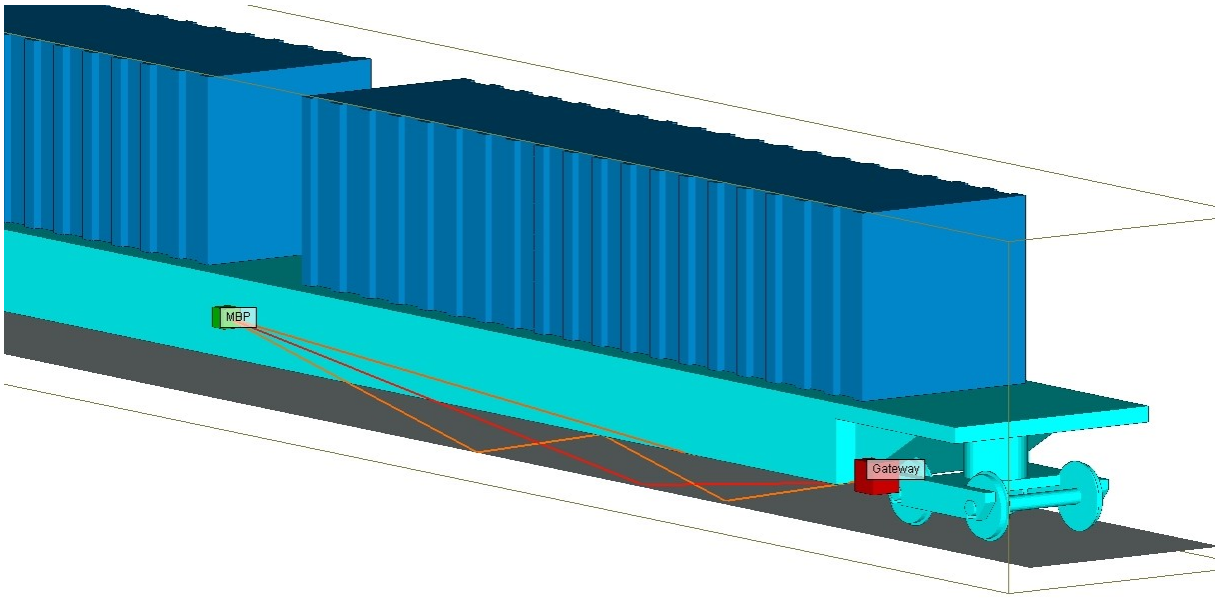


Figure 55: BLE wave propagation simulation in Wireless Insite starting from MBP transmitter to Gateway

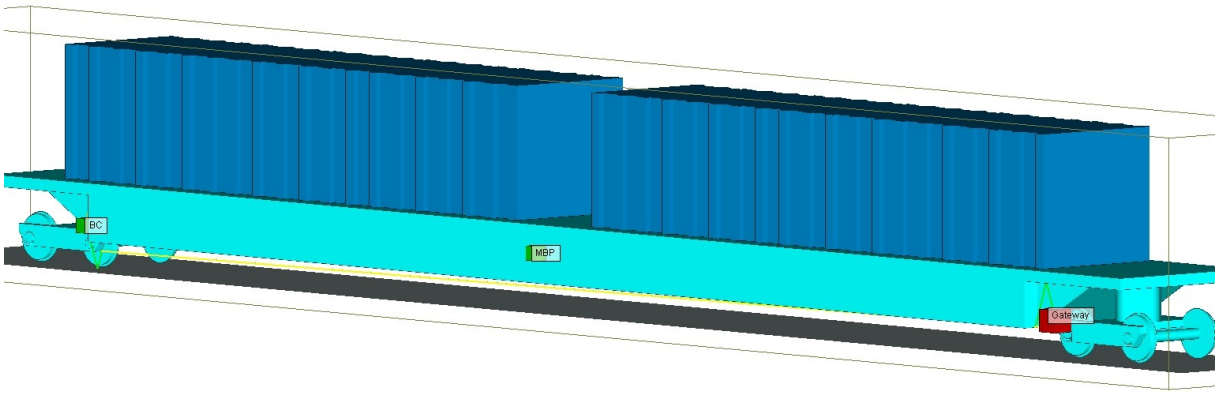


Figure 56: BLE wave propagation simulation in Wireless Insite starting from BC transmitter to Gateway

	MBP	BC
Path #1	$-71.03dBm$	$-107.7dBm$
Path #2	$-82.37dBm$	$-110.1dBm$
Path #3	$-85.38dBm$	$-121.9dBm$

Table 18: Path Loss in outdoor unloaded scenario

Obstacle situation

For the last simulation phase, we wanted to know whether or not the presence of obstacles in the environment around the train would affect the effectiveness of the transmission and, if so, how. This is this reason why I studied the system in case the car is parked in a warehouse between other trains and when passing through a tunnel. Will talk about results after showing both simulations.

First situation is when the wagon is stored in a warehouse: we supposed that it is parked

	MBP	BC
Path #1	$-59.86dBm$	$-67.5dBm$
Path #2	$-59.95dBm$	$-78.84dBm$
Path #3	$-61.61dBm$	$-84.3dBm$

Table 19: Path Loss when parked in warehouse

between two other trains as shown in figure 57. In this case there is a major improvement

in both sensors' efficiency as it can be seen in figure 19 with the three best feasible paths in terms of losses.

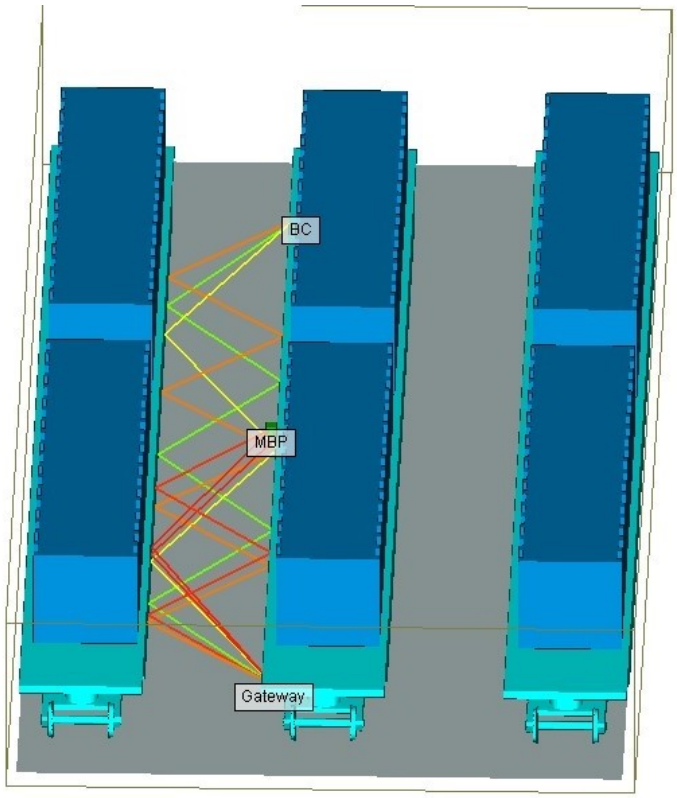


Figure 57: BLE wave propagation during train warehouse storage

The remaining situation is when the wagon is passing through a tunnel. It is represented in figure 58. All objects are rendered in a wireframe effect to allow us to see through them otherwise we would not be able to appreciate the communication pathway.
Table 20 tells us that:

- the first two best BC path are the same as the outdoor scenario, only the remaining ones are reflecting on the tunnel structure and consequently carrying a reduction of path loss due to the shorter distance covered.
- MBP has way shorter communication trajectories with all of them reflecting on the tunnel sides thus leading to a greater efficiency improvement in each path and in whole communication performance.

	MBP	BC
Path #1	$-61.81dBm$	$-107.7dBm$
Path #2	$-61.99dBm$	$-110.1dBm$
Path #3	$-71.03dBm$	$-113.6dBm$

Table 20: Path Loss in tunnel scenario

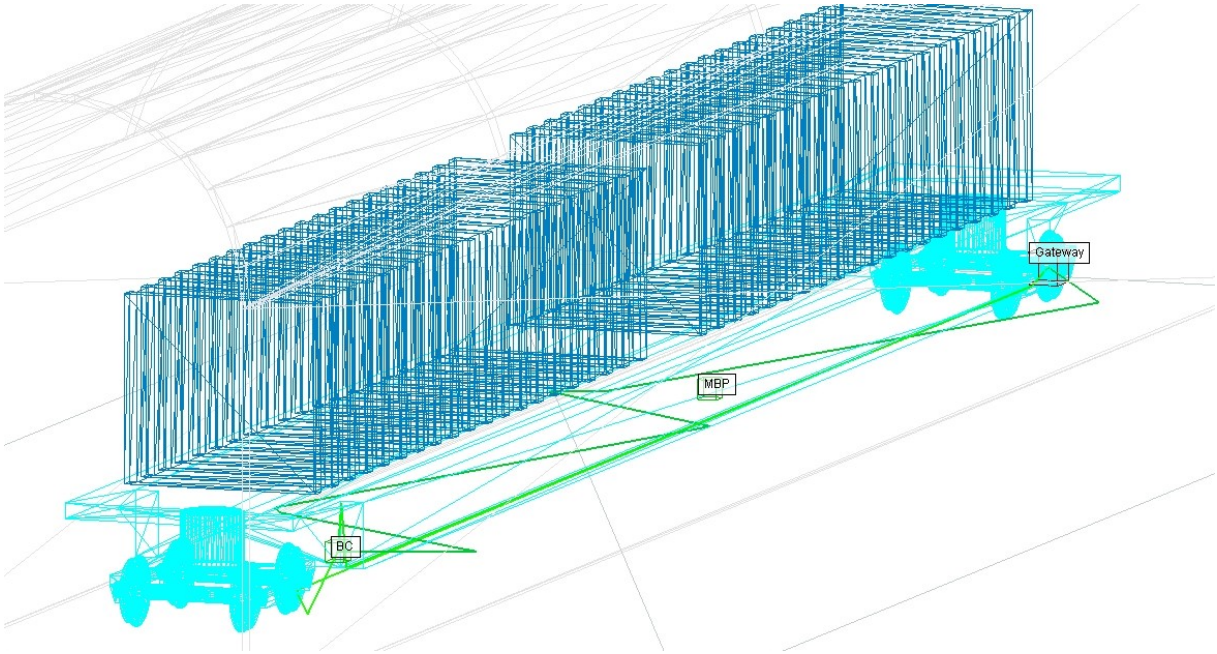


Figure 58: BLE wave propagation simulation from both sensor nodes during train passage through tunnels

Now Knowing how received power reflects to RSSI inside nRF modules we calculated the expected Received Signal Strength Indicators merging all paths' received power, results are shown in table 21, these values are computed with a transmitting power set to $0dBm$ ($1mW$).

These results show us that the transmission is affected only in a positive way when there are objects around the train. This happens because, instead of obstructing the transmission, the near obstacles work as objects on which BLE waves can reflect improving transmission's efficiency.

RSSI values tell us that to establish a secure communication and let our gateway receive the signal (sensitivity of $-99dBm$) on the MBP side we can us a transmission power of $0dBm$, while the BC one must be set to a value higher than $5dBm$. These are the minimum required value, they can be chosen with the optic of minimum power consumption,

OUTDOOR		WAREHOUSE		TUNNEL	
node	**RSSI**	**node**	**RSSI**	**node**	**RSSI**
MBP	−68	MBP	−57	MBP	−54
BC	−104	BC	−68	BC	−100

Table 21: expected RSSI in different scenarios

however we recommended to use a slightly higher value to not encounter communication errors and have to resend multiple packets and consequently lower transmission efficiency.

6 | Conclusion and future developments

The activity presented in this thesis had the aim of analysing the performances and possibly pointing out the available improvements, in terms of energy consumption, that the actual system developed presents and can implement.
The major phase during BLE connection is clearly the transmission one.
We first saw how **Data Length** can influence efficiency: clearly the best situation is when during every data exchange the central device is sending the greater number of Bytes possible (*251 Bytes*). For this scenario I would suggest to build a memory map long multiple of 251 Bytes and continuously collect measurement saving them inside to send them when it is filled. However, in some situations, it could also not be possible to wait for whole package collection due to the lost of real-time control, that's where I suggest to just collect the longer data possible and send it.
Regarding **Physical Rate** we analysed the three possible values and, in our situation, we could clearly see that $2Mbps$ is the best solution having almost half the consumption compared to $1Mbps$ and not even comparable to the coded solution specifically presented for long-range communications.
While with respect to **Connection Interval** I cannot directly indicate the best value because, as said before, it is related to how noisy the environment is. However I can still tell that, in our case, the best choice is a multiple of $35ms$ so that T_{align} is perfectly zero guaranteeing no time lost.
For the **Wave Propagation** simulation Expected RSSI values are shown in table 21 and we can notice that transmission is feasible in every analysed situation and is curiously affected only in a positive way when there are objects around the train. In detail with the actual sensor nodes' placements received powers tell us that, to establish a secure communication and let our gateway (sensitivity of $-99dBm$) receive the signal, on the MBP side we must set a transmission power of at least $0dBm$, while the BC one must be set to a value higher than $5dBm$. Remember that these are the minimum required value to establish a connection, however I recommended to use slightly higher power to

not encounter communication errors and having to resend multiple packets with the effect of lowering transmission's efficiency.
Moreover talking about **Future Developments** I would suggest to build a much more accurate train model and deeply study the waves propagation using Wireless Insite software to find the best possible sensor board's placements and actually test them in real case scenario. Studying the framework of a chosen train model can lead to find structural element that could work as reflector or antenna increasing transmission range.
Sensor placement and solar energy blackout could also be used to understand when a train is passing through a tunnel or is parked in a warehouse so it could be possible to lower transmission power due to better propagation paths.
From a protocol optimization point of view I think we already reached the best energy efficiency situation and also lower it is, nowadays, not so relevant. Only with better positioning and better connection scheduling we could ensure lower transmission power and receiver sensitivity guaranteeing even lower power consumption.

Bibliography

[1] S. Cii, G. Tomasini, M. L. Bacci, and D. Tarsitano. Solar wireless sensor nodes for condition monitoring of freight trains. *IEEE Transactions on Intelligent Transportation Systems*, 23(5):3995–4007, 2022. doi: 10.1109/TITS.2020.3038319.

[2] S. Cii, G. Tomasini, M. L. Bacci, and D. Tarsitano. Solar wireless sensor nodes for condition monitoring of freight trains. *IEEE Transactions on Intelligent Transportation Systems*, 23(5):3995–4007, 2022.

[3] Eurostat. Railway freight transport statistics. `https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Railway_freight_transport_statistics`.

[4] L. M.P. Shm for informed management of civil structures and infrastructure. *Journal of Civil Structural Health Monitoring*, 10, 2020. ISSN 2190-5479. doi: \url{https://doi.org/10.1007/s13349-020-00439-8}.

[5] Nordic. Online power profiler. `https://devzone.nordicsemi.com/power/w/opp/2/online-power-profiler-for-bluetooth-le`.

[6] Nordic. nrf52840-dk. `https://www.nordicsemi.com/Software-and-Tools/Development-Kits/nRF52840-DK`, 2008.

[7] Nordic. nrf52840-dk measuring current. `https://infocenter.nordicsemi.com/index.jsp?topic=%2Fstruct_nrf52%2Fstruct%2Fnrf52.html&cp=5`, 2008.

[8] Nordic. Power profile kit ii. `https://www.nordicsemi.com/Products/Development-hardware/Power-Profiler-Kit-2`, 2020.

[9] Remcom. Wireless insite. `https://www.remcom.com/electromagnetic-applications`.

[10] H. Sharma, A. Haque, and Z. A. Jaffery. Maximization of wireless sensor network lifetime using solar energy harvesting for smart agriculture monitoring. *Ad Hoc Networks*, 94:101966, 2019. ISSN 1570-8705.

[11] B. SIG. Ble speed. https://www.bluetooth.com/.

[12] SILabs. Ble throughput. https://docs.silabs.com/bluetooth/2.13/general/system-and-performance/throughput-with-bluetooth-low-energy-technology#connection-interval.

[13] S. M. Tosi J, Taffoni F. Performance evaluation of bluetooth low energy: A systematic review. *Sensors*, 22, 2017.

[14] G. Transport. Standard containers' dimension. http://www.globetransport.it/containers/.

[15] Txlogistik. T3000e characteristics. https://www.txlogistik.eu/wp-content/uploads/T3000e_technical_specification_30.pdf.

[16] J. Vetter, P. Novák, M. Wagner, C. Veit, K.-C. Möller, J. Besenhard, M. Winter, M. Wohlfahrt-Mehrens, C. Vogler, and A. Hammouche. Ageing mechanisms in lithium-ion batteries. *Journal of Power Sources*, 147(1):269–281, 2005. ISSN 0378-7753.

[17] F. Zanelli, F. Castelli-Dezza, D. Tarsitano, M. Mauri, M. L. Bacci, and G. Diana. Design and field validation of a low power wireless sensor node for structural health monitoring. *Sensors*, 21(4), 2021. ISSN 1424-8220. doi: 10.3390/s21041050. URL https://www.mdpi.com/1424-8220/21/4/1050.

[18] F. Zanelli, M. Mauri, F. Castelli-Dezza, E. Sabbioni, D. Tarsitano, and N. Debattisti. Energy autonomous wireless sensor nodes for freight train braking systems monitoring. *Sensors*, 22(5), 2022. ISSN 1424-8220.

[19] F. Zanelli, E. Sabbioni, M. Carnevale, M. Mauri, D. Tarsitano, F. Castelli-Dezza, and N. Debattisti. Wireless sensor nodes for freight trains condition monitoring based on geo-localized vibration measurements. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, 237(2):193–204, 2023. doi: 10.1177/09544097221100676.

List of Figures

1 Condition Based Maintenance Workflow 1
2 Diffraction **(a)**, reflection and transmission **(b)** example 5
3 Scheme of wireless connection layout fot freight train 7
4 Gateway architecture 8
5 External **(a)** and internal **(b)** view of the sensor node 9
6 Hardware Architecture 10
7 BT840e Block Diagram 12
8 Software architecture. We can notice the BT840F part highlighted in blue and the STM32 part in red 15
9 GPS Data recorded during field experiments 17
10 Position of sensor nodes on Breaking Valve (BC), Braking Main Pipe (BMP) and Weighting Valve (WV) 18
11 Pressure data during travel **(a)** and example of breaking event **(b)** 19
12 Example of collected pressure values 20
13 Battery and Temperatures Data 21
14 Battery and Temperatures Data 22
15 Pressure Trends along the whole track section **(a)** and during a breaking event **(b)** 22
16 Different Network Types 26
17 BLE Protocol Stack 27
18 BLE Timing Slots 28
19 Example of BL channels reliability map 28
20 Structure of ATT packet 31
21 BLE packet construction 32
22 Advertising and Scanning time intervals example with a advertising interval of $10ms$, scan window $12.5ms$ and scan interval $6.5ms$ 34
23 Connection time intervals example with a connection interval of $7.5ms$, IFS $150\mu s$, R $40\mu s$ and Tx packets $30byte$ using a $2Mbps$ physical speed 35
24 The Fresnel zone is the grey one, wider in the middle between the two devices 36

25 Data Exchange between master and slave 37

26 Test Kit Configuration 39
27 nRF52840 DK top view 40
28 nRF Power Sources selected and current measurements 44
29 nRF-DK and PPK2 connection diagram 44
30 Wireless Insite Indoor and Outdoor coverage predictions examples 45

31 Peripheral Diagram Flow 48
32 Peripheral configuration workflow 49
33 Advertising packet structure collected using Wireshark software 50
34 Central Diagram Flow 51
35 Central configuration workflow 52
36 Connection Establishment 54
37 Typical keep alive messages 54
38 3-D train models used 55
39 3D omnidirectional gain radiation pattern 56
40 Wireless Insite final interface 57

41 Current consumption during one advertising process 60
42 Current consumption during advertising focused on 1 particular channel . . 61
43 Current consumption during non connectable advertising focused on 1 particular channel 61
44 Advertising current consumption with different Tx powers 62
45 Simulated current consumption during one advertising process 63
46 Average current **(a)** and Battery charge drain **(b)** during advertising . . . 63
47 Current consumption during one connection sequence 64
48 BLE 5.0 transmission sequence 65
49 Data throughput and current consumption during transmission with various Packet Size 66
50 Data throughput and current consumption during transmission with various Physical Rate 68
51 BLE Unacknowledged data transfer example 69
52 Packet sent and T_{align} with different connection intervals 70
53 Data throughput and current consumption during transmission with various Connection Interval 71
54 BLE wave propagation simulation in Wireless Insite for the unloaded scenario 72

55 BLE wave propagation simulation in Wireless Insite starting from MBP transmitter to Gateway 73
56 BLE wave propagation simulation in Wireless Insite starting from BC transmitter to Gateway 74
57 BLE wave propagation during train warehouse storage 75
58 BLE wave propagation simulation from both sensor nodes during train passage through tunnels 76

List of Tables

1 Pressuremeter main feature . . . 11
2 BT840e main feature . . . 11
3 LTC3331 main feature . . . 13
4 Photovoltaic Panel characteristics . . . 13
5 Battery characteristics . . . 14
6 Node Current Consumption . . . 16
7 T3000e Travels Distances . . . 17

8 Bluetooth Classes . . . 24
9 Bluetooth Physical Rate Speeds [11] . . . 25

10 GAP parameters . . . 49
11 SCAN parameters . . . 52
12 GAP parameters . . . 53
13 T3000e dimensions . . . 55

14 Parameters used varying Packet Size . . . 67
15 Parameters used varying physical rate . . . 68
16 Parameters used varying connection interval . . . 69
17 Path Loss in outdoor unloaded scenario . . . 72
18 Path Loss in outdoor unloaded scenario . . . 74
19 Path Loss when parked in warehouse . . . 74
20 Path Loss in tunnel scenario . . . 76
21 expected RSSI in different scenarios . . . 77

Ringraziamenti

Al termine di questo elaborato, speranzoso di non avervi logorato in inutili digressioni, ritengo doveroso riservare un piccolo spazio per ringraziare tutte le persone che, volenti o nolenti, con il loro supporto mi hanno affiancato in questo meraviglioso e impegnativo percorso di approfondimento e sviluppo delle mie conoscenze acquisite durante gli anni universitari.

Innanzitutto non posso non ringraziare le due persone che hanno avuto più influenza nel mio percorso educativo: Grazie Mamma e Papà.

Per avermi insegnato tutto quello che sapete ed instillato un'innata curiosità e voglia di sapere. Per avermi sempre sostenuto, anche a vostra insaputa, nei momenti più pesanti. Per avermi fatto adoperare e sporcare le mani non limitando la mia esperienza a quella puramente bibliotecaria. Per avermi insegnato il rispetto e l'altruismo e creduto in me supportando ogni mio obiettivo.

Grazie a mio fratello.

Anche se spesso litighiamo sulle cose più futili grazie per sostenermi silenziosamente, per avermi mostrato l'incredibile tenacia e le capacità di cui sei dotato, per non aver mai mollato e anche da fratello minore mostrato la via da seguire. Ricorda solo di credere sempre in te stesso prima degli altri.

Grazie ai nonni.

Perché mi avete sempre visto in grado di conseguire qualsiasi sogno.

Grazie allo zio Sandro.

Senza il quale avrei difficilmente risolto molte delle difficoltà e problematiche che ho incontrato sul cammino e esortarmi a fomentare la mia passione in questo ambito.

Grazie Alessandra.

Per avermi aiutato e sostenuto attraverso le difficoltà quotidiane, per essermi sempre stata vicina e per il tempo che mi hai riservato.

Grazie a tutti gli amici, motociclistici e non, per ultimi non per importanza. Grazie per avermi fatto vivere spensieratamente e per tutti i momenti di diletto vissuti assieme, per le avventure affrontate e per tutte quelle a venire.

Grazie a chiunque ci ha tenuto e si sente parte di questo traguardo.

Infine grazie a te mio lettore.