Comparison Study and Product Development using Wireless Narrowband Low-power Wide-area Network Technologies

ENRIQUE ORTIZ PASAMONTES
Abstract

Nowadays it is more clear that the Internet of things (IoT) is not a transient trend but a completely new industry. The internet of things has the capability to enhance current industries (Industry 4.0), as well as to help protecting the environment and people. The latter is the case with the system developed and described in this thesis.

The possibilities that IoT brings are due to the interconnection of heterogeneous embedded devices to the internet. This thesis focus on LPWANs (Low Power Wide Area Networks), which is a new set of technologies specifically design for the needs of IoT devices. Due to the recent deploy of NB-IoT (Narrow Band IoT) networks it has become more difficult to know what LPWAN is best for a certain application. Thus, the first half of this thesis involves the comparative study of NB-IoT and LoRaWAN LPWANs. This comparison required an in depth study of each technology, specially on the physical and datalink layers. The comparison briefly displays the main characteristics of each technology and explain the main conclusions in a concise manner.

The second part of the thesis describes the development of a GNSS tracker. This tracker will be used on train wagons carrying goods that are dangerous for people and the environment. This thesis report describes the different steps taken, from the requirement specification to the partial development of the software.

Keywords - Internet of Things, Low Power Wide Area Networks, LoRa, NB-IoT, GNSS, PCB Design, Software Design.
Contents

1 Introduction 1
   1.1 Background ................................................. 1
   1.2 Motivation ................................................. 1
   1.3 Purpose ................................................... 1
   1.4 Scope ..................................................... 2
   1.5 Thesis Outline ............................................ 2

I NB-IoT and LoRa Comparison 3

2 Physical Layer 4
   2.1 Spectrum .................................................. 4
   2.2 Channel Bandwidth ......................................... 4
   2.3 Duplexing method .......................................... 5
   2.4 Modulation ................................................ 6
   2.5 Throughput ................................................ 7
   2.6 Spectrum efficiency ...................................... 8
   2.7 Coverage .................................................. 8
   2.8 Energy Consumption ....................................... 10
   2.9 Localization ............................................. 10
   2.10 Summary ................................................. 11

3 Datalink Layer 12
   3.1 Multiplexing method ..................................... 12
   3.2 Capacity and Scalability .................................. 13
      3.2.1 Capacity Comparison ................................ 14
      3.2.2 Scalability Comparison .............................. 15
   3.3 Frame structure ........................................... 16
   3.4 Message Acknowledgement ................................ 17
   3.5 Security .................................................. 18
      3.5.1 Authentication and authorization ................ 19
      3.5.2 Integrity ............................................. 19
## CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5.3 Confidentiality</td>
<td>19</td>
</tr>
<tr>
<td>3.6 Mobility</td>
<td>20</td>
</tr>
<tr>
<td>3.7 Real time suitability</td>
<td>21</td>
</tr>
<tr>
<td>3.8 Report Delay</td>
<td>22</td>
</tr>
<tr>
<td>3.9 Summary</td>
<td>22</td>
</tr>
<tr>
<td>4 Business Considerations</td>
<td>24</td>
</tr>
<tr>
<td>4.1 Availability (Netherlands)</td>
<td>24</td>
</tr>
<tr>
<td>4.1.1 End device Availability</td>
<td>24</td>
</tr>
<tr>
<td>4.1.2 Service Availability</td>
<td>25</td>
</tr>
<tr>
<td>4.2 Costs</td>
<td>25</td>
</tr>
<tr>
<td>4.3 Use Cases</td>
<td>26</td>
</tr>
<tr>
<td>4.3.1 Use Case 1</td>
<td>27</td>
</tr>
<tr>
<td>4.3.2 Use Case 2</td>
<td>27</td>
</tr>
<tr>
<td>4.3.3 Use Case 3</td>
<td>27</td>
</tr>
<tr>
<td>5 Conclusion</td>
<td>29</td>
</tr>
<tr>
<td>II NB-IoT GNSS Tracker development</td>
<td>31</td>
</tr>
<tr>
<td>6 Introduction</td>
<td>32</td>
</tr>
<tr>
<td>6.1 Product Purpose</td>
<td>32</td>
</tr>
<tr>
<td>7 Functional requirements</td>
<td>33</td>
</tr>
<tr>
<td>7.1 General Requirements</td>
<td>33</td>
</tr>
<tr>
<td>7.2 PCB Requirements</td>
<td>33</td>
</tr>
<tr>
<td>7.3 Mechanical Requirements</td>
<td>34</td>
</tr>
<tr>
<td>7.4 Norms and Regulation Requirements</td>
<td>34</td>
</tr>
<tr>
<td>7.5 Embedded Software Requirements</td>
<td>34</td>
</tr>
<tr>
<td>8 Implementation</td>
<td>36</td>
</tr>
<tr>
<td>8.1 System Architecture</td>
<td>36</td>
</tr>
<tr>
<td>8.2 Component Selection</td>
<td>38</td>
</tr>
<tr>
<td>8.3 Power Estimation</td>
<td>39</td>
</tr>
<tr>
<td>8.4 Schematics Design</td>
<td>40</td>
</tr>
<tr>
<td>8.5 PCB Design</td>
<td>45</td>
</tr>
<tr>
<td>8.6 Software Design</td>
<td>46</td>
</tr>
<tr>
<td>8.7 Product pivoting</td>
<td>48</td>
</tr>
<tr>
<td>9 Testing</td>
<td>50</td>
</tr>
<tr>
<td>10 Conclusions and future Work</td>
<td>55</td>
</tr>
<tr>
<td>Bibliography</td>
<td>56</td>
</tr>
</tbody>
</table>
A Energy Consumption calculation
A.1 NB-IoT 59
A.2 LoRa 60
1. Introduction

During this year the Internet of Things (IoT) is still at the peak of expectations on the Gartner hype cycle of emerging technologies [1]. The introduction of embedded systems connected to the Internet is helping to improve current manufacturing processes (Industry 4.0).

1.1 Background

A big technical issue for the application of the Internet of Things is how to provide a connection to the Internet for every device. There is a subset of IoT devices that are deployed at a long distance from traditional power and communication grids. For this kind of battery powered devices a new subset of wireless technologies was developed that allows a wide area coverage while keeping a low power consumption. These are the so called Low Power Wide Area Networks (LPWANs)[2].

Low Power Wide Area Networks are designed to offer long range connectivity along with a low energy consumption. This two advantages come at the cost of a reduced bit rate. Some examples of these technologies are LoRa and NB-IoT, which are compared in this thesis, but also technologies such as Sigfox[3] and Weightless[4] conform the new family of LPWAN technologies.

NB-IoT standard was developed by the 3GPP as part of the release 13 (Rel-13). It was created to address the Internet of things market. The compatibility of NB-IoT with previous LTE infrastructure is causing a fast deployment of this technology.

On the other side, LoRa is an established technology also meant to be used in the scope of IoT. LoRa is the physical specification of the technology and it is a proprietary modulation scheme. LoRaWAN is the protocol on top of LoRa. It was developed by the LoRa Alliance and it is publicly accessible.

1.2 Motivation

Due to the recent deployment of NB-IoT (Narrow Band Internet of Things) networks, it has become more difficult to know which LPWAN is better suited for a specific application. Both technologies have many technical similarities, since at a first glance both technologies are meant to be used by the same type of devices.

1.3 Purpose

The purpose for this comparison is to ease the selection process between LoRa and NB-IoT. This work also pretends to research if the differences between the technologies lead to a difference on the applications at which they can be used. It is
expected that different end applications will require the selection of one technology or the other.

The ultimate purpose of this comparison is to decide which of the two technologies should be used for the second part of this thesis, the development of a GNSS tracker.

1.4 Scope

The scope of this comparison includes the review of LoRa and NB-IoT technologies. These two technologies were chosen because they are the most popular ones at the moment of doing this thesis. The reason to not compare other LPWAN technologies is due to the lack of time required to perform an in-depth study of each of them.

1.5 Thesis Outline

The thesis is divided into two main parts. The first part involves the comparison study of NB-IoT and LoRa technologies. This first part of the thesis consists of three main elements: A comparison between the properties of each technology (Chapters 2 and 3), a compilation of case studies arguments which technology is better suited (Chapter 4) and a conclusion section where the main concepts are wrapped up (Chapter 5). Reading the conclusions chapter is meant to be sufficient to ease the selection process between the two technologies.

On the other hand the second part of this thesis consist on an introduction to the system developed (Chapter 6), the functional requirements for the GNSS module (Chapter 7) and the implementation of both the hardware and software of this device (Chapter 8).

Further chapters describe the testing procedure for both hardware and software (Chapter 9), while the conclusions extracted from this second part of the project, plus the identified future work to perform, is explained in chapter 10.
Part I

NB-IoT and LoRa Comparison
2. Physical Layer

The physical layer is the main responsible for LPWAN properties since this is the layer where modulation, the conversion from data to physical signals, takes place. But besides modulating the physical layer defines many other parameters. Examples are the access method to the medium, the spectrum to use or how much bandwidth to employ per physical channel.

2.1 Spectrum

NB-IoT and LoRa make a different use of the spectrum. The range of frequencies they use and the way they share the medium affects the overall performance of these technologies.

Both technologies were designed to operate in the sub-gigahertz spectrum. The reason is to take advantage of the better propagation these frequencies have. Nonetheless, there is a difference in the spectrum region where these technologies operate. NB-IoT utilizes the licensed spectrum, while LoRa is meant to be used at the unlicensed spectrum.

Using unlicensed spectrum has two main consequences. The first one is the coexistence with other technologies. The devices must be able to communicate and be resilient to the interference caused by other technologies sharing the spectrum. The second consequence is the transmission limitation imposed by the regulation authorities. In Europe the transmission duty cycle can not exceed $1\%$ for almost all the bands at which LoRa operates. This limitation has a great impact on the LoraWAN standard definition, which is LoRa’s data link layer.

Current deployments of NB-IoT are using the same frequencies where LTE is deployed, e.g. 900 MHz for T-Mobile (Netherlands). LoRa on the other hand makes use of the unlicensed 433MHz and 868MHz bands. The 433MHz band is 1 MHz wide, extending to 434MHz. The 868MHz band comprises a 7MHz range from 863MHz to 870MHz.

The following is an abstract from LoRa Documentation that resumes very well the consequences of choosing one spectrum type over the other:

“In practice, operating in license-exempt spectrum provides for no quality of service guarantee (as opposed to licensed operation, whereby the network operator has paid a fee for exclusive access to the spectrum being used) and a device operating in license-exempt spectrum is likely to be interference rather than link-budget limited.”

2.2 Channel Bandwidth

NB-IoT implements a channel bandwidth of 180 KHz. The reason is to keep the compatibility with LTE, since LTE’s physical resource blocks also require 180 KHz
CHAPTER 2. PHYSICAL LAYER

Figure 2.1. TDD (up) and FDD (bottom) representation over frequency and time.

of bandwidth. Some sources also describe NB-IoT bandwidth as 200 KHz. This value is obtained considering the guard bands. Guard bands are reserved space kept unused to avoid interference with adjacent transmissions.

LoRa can use three different bandwidths, the narrowest band is 125 KHz. 250 KHz and a 500 KHz bands can also handled by LoRa. The separation between channels is also 200 KHz (for 125 KHz bandwidth). Doubling the channel bandwidth in LoRa directly doubles the data rate.

These differences in channel bandwidth are not significant enough to cause a performance difference between the two technologies.

2.3 Duplexing method

Both technologies use a half-duplex transmission scheme. The main reason is to keep the hardware complexity, and therefore the price, as low as possible. The half-duplex method allows for the transmission and reception of information, although not at the same time, when the device is receiving information it can not transmit at the same time and vice versa. Nonetheless each technology implements two-way communication in a different manner.

NB-IoT utilize Frequency Division Duplex (FDD) as its duplexing method. This method derives directly from LTE. Although the latter also implements the Time Division Duplex (TDD) method, NB-IoT is currently limited to FDD only. It is expected for 3GPP Rel-14 to include the TDD specification for NB-IoT.

Frequency division duplexing consist on using different frequencies for the transmission and the reception of data. Figure 2.1 (bottom) shows how an NB-IoT device is either receiving or transmitting at different frequencies and never do both at the same time.

The chosen method for LoRa is time division duplex, as the downlink transmission is scheduled at different times than uplink transmission. As with NB-IoT, and
LoRa devices are not capable to transmit and receive at the same time. Figure 2.1 (top) displays the TDD behavior of LoRa devices.

2.4 Modulation

NB-IoT implements a subset of legacy LTE modulations. QPSK is used for down-link, while for uplink either QPSK or BPSK transmissions can be used. The reason for using this subset of modulations is for both reducing the complexity of end devices and reducing the power consumption, since these are low peak-to-average power ratio modulations.

LoRa takes a completely different approach at signal modulation. LoRa is a spread spectrum modulation scheme, thus it requires a large bandwidth in comparison with the amount of data that is transmitted. The spreading is achieved by representing each symbol as a chirp. A chirp is a signal that increases in frequency over time. Figure 2.2 displays a chirp over the time domain. Chirp modulation does not require a carrier signal, since it is the chirp itself what gets modulated to represent the information. Chirp modulation also offers great resistance against Doppler effect and, as a broadband signal, it is robust against multipath and fading.

The modulation performed is a phase shift over the chirp. Each symbol value has a different starting frequency. When the chirp reaches the upper bound frequency limit, it rolls over and continues from the lowest frequency. This behavior is represented at the top section of Figure 2.3.

The number of possible symbol values is $2^{SF}$, where SF is the spreading factor that can have values between 7 and 12. Therefore, the number of bit carried per symbol can be set between the mentioned values. The spreading factor is used to modify the symbol duration with the purpose of improving coverage. The symbol duration depends on the spreading factor by $2^{SF}$, meaning that each unitary increase in SF doubles the symbol duration. The top of Figure 2.3 shows how each symbol is being modulated in phase over the available bandwidth, while the bottom section displays the difference in symbol duration between different spread signals, one superposed to the other.
2.5 Throughput

We can also find great differences with respect to the data rate of each technology. The values described here are the theoretical maximum achievable at the physical layer.

NB-IoT achieves a sustained data rate of 26.2 Kbps for downlink and 62.5 Kbps for uplink transmissions (in-band mode). The peak data rates are also interesting to mention, as if the data to transfer does not exceed the maximum transport block size (1000 bits for uplink and 680 bits for downlink), it is possible to achieve data rates of 250 kbps for uplink and 170 kbps for downlink.

At the application layer, and in the case of a link at the maximum permissible coupling loss, the transmission speed is significantly reduced to 440 bps in downlink and 310 kbps in uplink [7].

LoRa has different data rates depending on the spreading factor used. The LoRa Alliance present a physical data rate from 250 bps for SF12 to 5470 bps for SF7, with a bandwidth of 125 KHz [8]. In Europe it is also possible to use a bandwidth of 250 KHz with SF7. This combination provides a data rate of 11000 bps.

Thus, NB-IoT can transmit at least twice as fast as LoRa. This can be advantageous for applications requiring higher data rates. The reasons for this difference between the two are due to the different modulation schemes each technology employs.
### 2.6 Spectrum efficiency

Spectrum efficiency is denoted as the rate of information that a technology can transmit over a certain bandwidth. The following results are based on the data rates and bandwidths shown in the previous sections.

LoRa is a spread spectrum technology, and thus it is expected to make a less optimal utilization of the spectrum. From the information displayed in Table 2.1 we can see that the spreading factor determines the spectrum efficiency of LoRa. Also it is worth to mention that the bandwidth does not have any effect on the spectrum efficiency, as shown for the SF7 cases.

From this table it is also noticeable the difference of almost an order of magnitude between both technologies. The reason for LoRa to use a spread spectrum modulation is to cope with the interference frequently found at the ISM bands, since this spectrum is shared among other technologies.

### 2.7 Coverage

The coverage in a particular scenario can be obtained by performing a link budget calculation. A link budget calculation takes into consideration all the gains and losses produced from the transmitter to the receiver. Examples of gains are the transmission power, the transmitter and receiver antenna gains or the receiver sensitivity. Examples of losses are the distance between receiver and transmitter or the shadowing losses (Terrain undulations and vegetation).

Different propagation models can be used for calculating the path loss (Friis free space model [9], Okumura-Hata empirical model [10]) and multiple antenna gains can be used when designing a network. Because of these reasons, this work provides the link budget for each technology considering only the sum of the transmission power and the device sensitivity. This way the coverage can be calculated by using the preferred path loss model.

The transmission power is considered because LoRa and NB-IoT have different power regulations in Europe, directly related with the spectrum in which they operate. The challenge to compare the two technologies is located at the device sensitivity, since it follows the equation 1:

\[
S = 10 \log_{10}(kTB) + \frac{C}{N} + NF
\]  

(2.1)
CHAPTER 2. PHYSICAL LAYER

<table>
<thead>
<tr>
<th></th>
<th>NB-IoT (Downlink)</th>
<th>NB-IoT (Uplink)</th>
<th>LoRa</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) Transmission power</td>
<td>35 dBm</td>
<td>23 dBm</td>
<td>14 dBm</td>
</tr>
<tr>
<td>(2) Device sensitivity = (1)-(2)</td>
<td>-127.6 dBm</td>
<td>-141 dBm</td>
<td>-138.08 dBm</td>
</tr>
<tr>
<td>Link Budget</td>
<td>162.6 dB</td>
<td>164 dB</td>
<td>152.08 dB</td>
</tr>
</tbody>
</table>

Table 2.2. Link budget for NB-IoT and LoRa.

The equation shows that the device sensitivity is related with three elements, the required signal to noise ratio (C/N) determined by the transmission scheme, e.g. modulation. The receiver noise factor (NF), which is the noise generated by the receiver and it depends on the hardware implementation, and the thermal noise for a certain bandwidth \((10 \log(kTB))\), which depends on the device temperature and the signal bandwidth.

This equation means that sensitivity results must have the same noise factor and temperature to compare the two in terms of the signal to noise ratio and bandwidth, which are precisely the values dependent on the technology. Table 2.2 offer comparable values since them have been normalized to use the same temperature, 293 °K (19.85 °C), the same receiver noise factor (5 dBm) and the required signal to noise ratio to achieve a 1% error rate.

LoRa sensitivity value is achieved by using a spreading factor of 12. Values for NB-IoT have been obtained from tables VI and VIII and figure 3 at [11]. The downlink link budget corresponds with a transmission repeated 128 times, while the uplink transmission is repeated 16 times. Repetitions are used in NB-IoT to adapt to the required the coverage, while LoRa adapts by modifying the spreading factor. In both cases the transmission time increases and the data rate decreases when improving the coverage.

Also the values shown at the previous table corresponds with the downlink and uplink transmission in in-band mode. Because of the asymmetrical nature of NB-IoT, different transmission power and sensitivity values are used at the base station and at the end devices.

On the other side LoRa claims a maximum link budget of 158 dB [6, 12], although this value was obtained using a transmission power of 20dBm. It is better suited to calculate the link budget using a transmission power of 14 dBm because, as explained at [13, Section 2.3], in Europe the maximum radiated power for the bands at which LoRa operates is restricted. Thus, a more realistic value will be 14 dBm of radiated power and -138 dBm of sensitivity, resulting in a link budget of 152 dB.

It was not possible to find empirical results for NB-IoT coverage. It is a very new technology that is still being deployed. LoRa on the other side has been into market for a while, and studies such as [13] show a measured sensitivity of up to 134.50 dB.
2.8 Energy Consumption

LPWANs focus on reducing the energy consumption to enable the design of cost effective, battery powered devices. Energy consumption is usually a difficult restriction to meet during the design phase of battery powered products. Therefore, energy consumption differences between these technologies can be determinant into choosing one or the other.

Two different attributes were analyzed to compare the energy consumption: The energy required in mW/h per uplink message and the power consumed while in sleep mode, given in mW.

The power calculations consider the transmission of a 20 byte user payload message and the following reception of a data link layer acknowledgement. The transmission procedure for NB-IoT has been extracted from [14, Section 7.3.6.3.1] considering a maximum coupling loss of 164 dB and a 99% confidence of successful transmission [14, Figure 7.3.6.3.1-10]. Acknowledgement timing is the same as the uplink timings assignment, since both elements are transmitted through the PDCCH, as stated in [14, Section 7.3.6.3.1.7].

The transmission procedure for LoRa has been extracted from [8, 6], using a spreading factor of 12 to compare both technologies against their maximum coupling loss and 99% confidence of successful transmission.

The power consumption data has been extracted from datasheets of actual modules. The reason to do so is because the theoretical consumptions are significantly lower than the values shown at the datasheets. We assume the datasheets to have values much closer to reality. We have chosen the power characteristics from the SX1272[12] LoRa module and the Sara N2 [15] module for NB-IoT.

The power calculations are kept in Annex A. The obtained consumption per message is 0.7337 mW/h for NB-IoT and 0.05714 mW/h for LoRa. LoRa spends 12 times less power than NB-IoT for sending a single message. There are two causes for this great difference. The first one is that an NB-IoT module consumes 5 times more power when transmitting. The second one is that NB-IoT has a preliminary synchronization and resource allocation phase. This preliminary phase conforms a great proportion of the overall transmission, specially for the small payload used at this calculation. For greater payload sizes the gap between the two technologies is reduced.

2.9 Localization

In this work we understand the localization capabilities of LPWANs as the effectiveness of geopositioning an end device without using other hardware than the required for the LPWAN connectivity.

In this context LPWANs are not suitable for geopositioning an end device by measuring the time-on-air of a message. A physical limit appears in regard to finding the direct path of a signal[16]:

...
The issue has to do with differentiating the direct path signal from the rest of reflected signals. The minimum difference in path length to distinguish between two paths is directly related with the bandwidth of the signal:

\[ \text{MinPathDifference} = \frac{c}{\text{Bandwidth}} \]  

(2.2)

Where \( c \) is the speed of light in meters per second. Thus, for LoRa and NB-IoT the minimum path length has to be 2400 and 1666 meters respectively. This means that every reflected path that differs less than MinPathDifference from the direct path will sum up and add an error to the final position calculation.

The conclusion is that the technologies compared in this document are not suitable for geolocation when finer resolution than in the kilometer scale is required.

### 2.10 Summary

Table 2.3 displays the main figures of this chapter. Both technologies are similar in aspects such as the bandwidth, the duplexing method and the localization suitability. But nonetheless there is a very significant difference that influences other figures shown: This is the modulation technique. NB-IoT uses a modulation subset from LTE, which allows for a great spectrum efficiency. On the other side LoRa is a spread spectrum modulation, so the throughput and the spectrum efficiency is significantly lower than with NB-IoT. Other important differences are the use of licensed (NB-IoT) and unlicensed (LoRa) spectrum, as well as the power consumption. LoRa benefits in this regard due to its simpler protocol scheme, which does not require as much steps to initialize a transmission as with NB-IoT.

<table>
<thead>
<tr>
<th></th>
<th>NB-IoT</th>
<th>LoRa</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spectrum</td>
<td>Licensed spectrum</td>
<td>Unlicensed spectrum</td>
</tr>
<tr>
<td>Channel Bandwidth</td>
<td>180 KHz</td>
<td>125 KHz &amp; 250 KHz</td>
</tr>
<tr>
<td>Duplexing method</td>
<td>Half duplex frequency division</td>
<td>Half duplex time division</td>
</tr>
<tr>
<td>Modulation</td>
<td>BPSK &amp; QPSK</td>
<td>LoRa</td>
</tr>
<tr>
<td>Max. Throughput</td>
<td>26.2 Kbps DL 62.5 Kbps UL</td>
<td>5.47 Kbps DL 5.47 Kbps UL</td>
</tr>
<tr>
<td>Spectrum efficiency</td>
<td>0.347 bps/Hz</td>
<td>0.044 bps/Hz (SF7)</td>
</tr>
<tr>
<td>Coverage</td>
<td>162.6 dB</td>
<td>152.08 dB</td>
</tr>
<tr>
<td>Power consumption per uplink report</td>
<td>0.7337 mW/h</td>
<td>0.05714 mW/h</td>
</tr>
<tr>
<td>Localization</td>
<td>Not suitable</td>
<td>Not suitable</td>
</tr>
</tbody>
</table>

*Table 2.3. Physical properties for NB-IoT and LoRa.*
3. **Datalink Layer**

The Datalink layer is the second level of the OSI model, right on top of the physical layer. It defines the procedure that the devices in the network should follow in order to communicate between each other. This procedure includes steps such as partitioning the information into smaller segments and reordering them into frames as efficiently as possible to maximize the utilization of the physical medium.

The Datalink layer also has the ability to schedule and prioritize the information flow based on the end user or on the information type. In some technologies the Datalink layer can correct the received frames or automatically ask for the retransmission of the wrong fragments without the intervention of upper layers.

### 3.1 Multiplexing method

It is in the specification of the multiplex method where we begin to see major differences. The multiplex method, also known as multiple access method, defines how and what set of technologies should be used to share the medium between multiple users.

NB-IoT use different access methods for downlink and uplink, OFDMA and SC-FDMA respectively. For downlink transmission the OFDMA method is used. This method leads to a division of the channel bandwidth into subcarriers, as shown on the left hand side of Figure 3.1. It has the advantages of not requiring an equalization process on the receiver size (lower complexity on end devices) and reducing the inter-symbol interference, since each of the symbols are transmitted at a lower speed (long symbol time). However OFDMA has a disadvantage that prevents its use for uplink, which is the high peak-to-average power ratio characteristic of this method.

That is the reason for the use SC-FDMA for uplink transmissions. SC-FDMA is used to improve the power efficiency of uplink transmissions. The only drawback comes from the increased receiver complexity needed at the base station (equalization process). SC-FDMA characterizes for using a wider bandwidth to transmit data. Several subcarriers now bear the same symbol at the same time. The symbol duration is smaller than in OFDMA. SC-FDMA is represented on the right hand side of Figure 3.1.

The medium is shared between end devices without producing collisions, because each device is granted with different resources by the base station. The resource allocation is dynamic and it adapts to the necessities of each device at a given time.

LoRa makes the access to medium much simpler by defining an asynchronous access method. An end device initiates an uplink transmission as soon as it has data to send. This type of behavior is usually described as Aloha like [17]. In this case it is possible to produce a collision if two devices transmit at the same time. This is the case represented at Figure 3.2.

The difference between these two access methods will have a noticeable effect
on its total capacity, as explained in the following section.

### 3.2 Capacity and Scalability

Another aspect to consider is how many devices can be serviced over an area. This question is specially important in densely populated areas, since the quantity of crammed devices can affect the performance of communications.

In terms of future proofness we will also take a look in this section on how these technologies scale. The easiness and cost of increasing the capacity over an area is important because, as stated in [18], the number of IoT devices is expected to increase annually by a 23% growth rate. 16 billion IoT devices are expected to be connected by 2021.
3.2.1 Capacity Comparison

The capacity metric is defined in [14] as the spectral efficiency in number of reports/200 kHz/hour. The bandwidth needed for NB-IoT and LoRa is 180 KHz and 125KHz respectively. Thus, the capacity comparison assumes a single carrier is used for each of these technologies.

The capacity evaluation was done by performing simulations. The simulations use the traffic models and the assumptions in [14, 19]. The traffic model specifies a transmission period per device of 0.47 packets/hour, so the capacity can be expressed as the number of devices supported per site.

NB-IoT simulation results are already available at [19]. LoRa simulation was taken out using the same assumptions that in NB-IoT for consistency purposes. However there is a difference between both technologies that has led to some disparities at the simulation planning.

The main difference between both simulation scenarios is that in LoRa the transmission power is limited. We assumed that the devices will use the maximum transmission power allowed (14 dB). For this purpose LoRa base stations should do an omnidirectional transmission, otherwise if we direct the beam we will exceed the allowed power limit. Also both the base stations and the end devices have the same transmission power limit, and having a directional antenna with some gain at the base station would lead to asymmetrical links where downlink transmissions may not be reachable by the end device.

The outcome is that NB-IoT simulations use three sectors per site, while LoRa sites has only one antenna to cover the same area. Figure 3.3 better exemplifies this difference. The capacity then was compared as the number of devices per site that each technology can handle while keeping a packet transmission error rate below 10%.

In [19] the amount of devices supported by NB-IoT is 71000 devices per cell, which by three cells per site makes a total of 213000 devices per site. LoRa results
are shown in Figure 3.4. In this diagram it is represented the rate of successful transmissions (Y axis) for a varying number of end devices. From this results we can extract that LoRa is capable to handle 21000 devices per site while maintaining a successful transmission rate of 90%.

These values need to be multiplied by the number of channels available in order to get the maximum amount of devices that a site can serve. LoRa can operate in the 433MHz and 868 MHz bands. ETSI regulation [20] specifies the usable spectrum of these bands, also known as the ISM (industrial, scientific, and medical) bands.

It is possible to allocate 8 different 125KHz channels at the 433 MHz band. The 868 MHz band is more complex, since it is divided into 5 different sub-bands. Each sub band has its own power, duty cycle and bandwidth characteristics. Table 3.1, extracted from a LoRa module compliance document [21], illustrate the available channels for each sub band.

Counting with the channels available at both the 433MHz and 868MHz bands we obtain a theoretical total number of 46 channels. The number of channels (physical resource blocks) available at NB-IoT can not be calculated because it depends on the amount of spectrum that the network operator has available (has purchased).

3.2.2 Scalability Comparison

We define scalability as the ability to increase the amount of connected devices over an area. In the capacity section we saw how many devices each technology can serve over a single carrier. Thus, the easier way to increase the devices supported within a site is to increase the number of carriers used.

NB-IoT does not have a specific limit, since it depends on the spectrum that a network operator has available. So far network operators are implementing NB-IoT in in-band mode. This means that NB-IoT signal is shared among the current legacy
 LTE traffic. It is up to the network operator to modify the ratio of spectrum used by each technology, and increase the number of PRBs for NB-IoT at the expense of legacy LTE spectrum. However the bandwidth available for LTE is generally greater than in LoRa, as it is possible to find carriers with up to 20 MHz in bandwidth.

The other alternative for scaling is to increase the site density. A greater number of sites over the same area makes each of the cells to be reduced in size, and thus decreases the number of devices to support for each cell. Both technologies are capable of adapting the transmission power to the channel coupling loss, effectively allowing to increase the site density and reduce the area where interference is produced.

It is much easier to increase the number of sites with LoRa than with NB-IoT. A private network using LoRa can be deployed independently from the network operators, since LoRa runs over unlicensed spectrum. However LoRa is still susceptible to interference from other LoRa devices belonging to different networks. The scalability of LoRa given by increasing the number of sites will not prevent the interference caused by other devices connected to a different network, as they use the same shared spectrum.

In conclusion we can affirm that with NB-IoT it is easier to scale an existing site. NB-IoT in its in-band mode allows a great deal of flexibility while LoRa is limited to the spectrum available by the ETSI regulation. On the other hand it is possible to increase the number of sites with LoRa, while for NB-IoT this procedure is costly and the decision is up to the network operator.

### 3.3 Frame structure

The frame composition for each technology is very different. LoRa has a rather simple frame structure. Three different message payloads can be transported: A join request, a join response and a data frame. The data frame is the most common type of message, since the other two are only used during the device activation procedure. For a detailed view into the header structure and signaling refer to Attachment B.

On the other side NB-IoT has a wide variety of frame structures. This depend of variable length headers due to a more complex network stack and algorithms.

<table>
<thead>
<tr>
<th>Band</th>
<th>Duty Cycle</th>
<th>Power</th>
<th>125 KHz</th>
<th>250 KHz</th>
<th>500 KHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>1%</td>
<td>14 dBm</td>
<td>15</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>G1</td>
<td>1%</td>
<td>14 dBm</td>
<td>3</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>G2</td>
<td>0.1%</td>
<td>20 dBm</td>
<td>2</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>G3</td>
<td>10%</td>
<td>7 dBm</td>
<td>1</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>G4</td>
<td>1%</td>
<td>14 dBm</td>
<td>1</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 3.1. Available channels for European sub bands.
Table 3.2. Maximum application level payload size for a given spreading factor.

<table>
<thead>
<tr>
<th>Max. application payload size</th>
<th>SF 7</th>
<th>SF 8</th>
<th>SF 9</th>
<th>SF 10</th>
<th>SF 11</th>
<th>SF 12</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>222</td>
<td>222</td>
<td>115</td>
<td>51</td>
<td>51</td>
<td>51</td>
</tr>
</tbody>
</table>

The reason for having a rather complex network stack in a resource limited device is the heritage that NB-IoT has on LTE technology. A very detailed view of the NB-IoT header structure can be found at Attachment A.

The biggest application layer payload that LoRa can transmit depends on the used spreading factor. The reason underneath is the duty cycle limitation imposed over the bands at which LoRa operates. The higher the spreading factor is, the longer the transmission time gets for the same amount of application layer data, and transmission times are regulation limited. Uplink and downlink transmissions share the same application payload limit. Table 3.2 present the maximum application payload size for each spreading factor configuration using 125KHz bandwidth, as described at LoRaWAN documentation [8, Table 7].

The application payload also varies at NB-IoT, although the reason is the variability on the headers size mentioned before. Nonetheless NB-IoT has a fixed transport block size (TBS) that differs between the downlink and the uplink transmissions. A maximum TBS of 1000 bits is available at uplink transmission, while 680 bits is the maximum TBS at downlink[14, 22].

Given these values, we can consider the application data as the transport layer PDU. This is all the data coming from upper layers to the transport layer. The reason to choose this level is that NB-IoT defines specific procedures for compressing ip headers [23]. Using the same assumptions as in [14, Section 7.3.6.3.1], the protocol overhead under IP layer and the compressed TCP/IP size given in [23] we obtain a protocol overhead of 31 bytes. With these assumptions the obtained maximum payload size at the application layer is 94 bytes for uplink and 54 bytes for downlink transmission.

### 3.4 Message Acknowledgement

In many scenarios it is imperative to know that a message has successfully reached its destination. The methodology used for checking this is to send as an answer an acknowledgement message (ack) from the receiver back to the transmitter. This functionality can be implemented at different layers of the OSI model, as for example in the application layer. Both NB-IoT and LoRa implement such a functionality in the Datalink layer, although they have important differences that we will analyze in this section.

NB-IoT was derived from LTE, and because of this reason it includes the same message acknowledgement functionality than in LTE. The process is known as Hybrid Automatic Repeat Request (HARQ). It consists on the application of an error correction code on the datagrams before sending them. This coding allows to re-
cover the original message (if the corruption of data is not too severe) and also to
detect the transmission error. If the message was retrieved successfully, the receiver
automatically sends back an acknowledgement message to let know the sender which
datagram in particular was successfully sent.

Figure 3.5 describes HARQ operation in NB-IoT. After sending the scheduling
information (DCI), if the datagram presents irrecoverable errors then the receiver
sends an non-acknowledgement message (NACK) to let know to the sender that the
datagram needs to be sent again. An interesting aspect about NB-IoT is that this
process occur automatically and without the developer needing to configure it. It
is a default process in NB-IoT.

In LoRa the process of message acknowledgement is similar. The receiver sends
ACKs to the sender to let it know that the transmission went fine. However there
are two main differences when compared with NB-IoT; The first one is that the
message acknowledgement needs to be individually toggled for each message by
setting a bit into the LoRa header (it is not a default behavior of LoRa). The
send and most important difference against NB-IoT is that LoRa, by design, can
not handle as much downlink messages as uplink ones[5]. This means that only a
reduced set of uplink messages can be acknowledged by the LoRa gateway. The
utilization of LoRa may not be recommendable depending on the application of the
end device. If a device needs to inform when a frequent event happen, it will not be
possible to make sure for every event of the correct transmission of the notification.

3.5 Security

The Data Link layer can also provide security to the transmission. Three different
aspects are considered in this section: Authentication, integrity and confidentiality.
In this section we will state the differences between NB-IoT and LoRa for ensuring
a safe transmission.
3.5.1 Authentication and authorization

Authentication is the process of ensuring a user identity, while authorization is the procedure to grant or deny a user the access to a certain resource. Authentication is performed in both technologies during the connection procedure. An NB-IoT device has first to initiate the connection procedure with a base station. Afterwards the Mobility Management Entity (MME), which is a part of the LTE network (Evolved Packet Core), authenticates the end device. The end device also authenticates the network (mutual authentication).

LoRa end devices also offer mutual authentication during the connection procedure. A shared key (AppKey) is used for authentication. If both sides know the key, both sides identity is confirmed.

3.5.2 Integrity

To verify the integrity of a message is to ensure that the data is authentic, accurate and safeguarded from unauthorized user modification. NB-IoT implements data integrity through the Packet Data Convergence Protocol (PDCP). The PDCP discern between Control messages (C-plane) and user messages (U-plane), and only provides data integrity to control messages.

LoRa similarly implements integrity by adding the Message Integrity Code (MIC) at the end of the MAC PDU. The MIC is computed using the MAC header and its payload, which contains the already encrypted user data. LoRa provides integrity to all kind of messages.

3.5.3 Confidentiality

Data confidentiality is achieved by encryption. In NB-IoT, the PDCP encrypts the SDU right after adding the integrity protection (if applicable). Encryption is performed in both the control plane and the user plane. In the control plane both the data part and the MAC-I of the PDCP PDU are encrypted. For the user plane only the data part of the PDU is encrypted.

The necessary information to perform this functionality is given by the Radio Resource Control (RRC). This information is the ciphering algorithm to use and the key. The encryption function can also be activated, suspended and resumed by the RRC.

The encryption algorithms used in NB-IoT are the same ones available at legacy LTE. This algorithms are named together as EEA (EPS Encryption Algorithms)[24]. There are 4 different options for encryption:

- 128-EEA0: No encryption.
- 128-EEA1: SNOW 3G encryption algorithm.
- 128-EEA2: AES encryption algorithm.
Lora defines two confidentiality levels. The first one is between the end device and the network server. This level uses the Network Session Key to encrypt the data, so that the information is not disclosed between the end device and the network server. The Network Session Key is used for encryption only when the payload of the message consists of MAC commands (no application level data). Figure 3.6 shows the encrypted information flow through a LoRa network.

The second level ensures confidentiality between the end device and the application server. This is achieved by encrypting the frame payload with a different key, the Application Session Key. The network server does not know this second key, and thus, it cannot extract the user payload. AES is used among each key to implement confidentiality in the two levels.

3.6 Mobility

It is in the nature of low power wide area networks to offer rather limited data rates. Both LoRa and NB-IoT were designed for short messages with a long time span between each transmission. In this context it is assumed that a transmission can take place without needing to change the site (base station or gateway) from which it is served. Thus, it is not necessary to implement a hand-over (in NB-IoT case) during the connected state.

Even so, both technologies conceive the circumstance of initiating a new transmission after the end device has changed its location. If an NB-IoT end device requires a cell change, it should transition to RRC_IDLE state (disconnect) before starting a new connection procedure. It is during the connection procedure when an end device chooses the cell to connect. The decision is made upon the power and quality values of the narrow band reference signal (NRS).

When a LoRa device connects to the network it does not need to establish a connection with a particular gateway. A LoRa transmission is heard and processed
by every gateway within range. This transmission is then forwarded to the network server, where further processing is done to recognize the same packets coming from each gateway as an unique transmission.

The transmission from the LoRa device will make its way through as long as the end device is in range of a LoRa gateway, so the mobility of a device is seamless to both the end device and the gateways. Figure 3.7 describes a case where a transmission from a LoRa end device is received by three different gateways. These three gateways transmit the message to the network server. The network server then process the repeated messages and sends a single message to the application server. The network server also chooses from the previous gateways which one is better suited to send a downlink transmission to the end device.

3.7 Real time suitability

Different definitions have been made to describe what real-time operation is. In this work we use the definition from Hermann Kopetz[25]:

“A real-time computer system is a computer system in which the correctness of the system behavior depends not only on the logical results of the computation, but also on the physical instant at which these results are produced.”

In other words, in a real time system the demanded results have to obey a deadline. NB-IoT does not support many of the legacy LTE features, including quality of service. The network operator can not guarantee a time limit at which a transmission should be made. Therefore, NB-IoT can not be used for delay sensitive data packets.

LoRa is in the same situation as NB-IoT. No quality of service is defined, so it is not guaranteed that a transmission will be sent correctly before a defined time limit.
3.8 Report Delay

Although NB-IoT targets latency insensitive applications, it was designed to have a report time of less than 10 seconds. The latency report in [14, Section 7.3.6.3.1] concludes that a maximum latency of 6284 ms is produced when transmitting 20 bytes of application layer data, achieving a 99% confidence in the successful delivery of the report. The acknowledgement time is not considered in the delay calculation because the packet can already be transmitted to the application server. Table 3.3 compares the latency between NB-IoT and LoRa for the previous conditions and a coupling loss of 144 dB and 152 dB. Delay can not be compared for a 164 dB coupling loss, since LoRa can not cope with that level of path loss.

The conclusion to extract is that LoRa has a reduced latency in comparison with NB-IoT. This is due to only taking the transmission time for the LoRa latency calculation, as it does not require to perform a connection procedure with the gateway. The connection procedure in NB-IoT involves the steps of synchronizing with the base station, retrieving the MIB and performing the random access procedure. These additional steps take up to 2414 ms for a 164 dB coupling loss, corresponding with the 38% of the overall report time.

Note that NB-IoT specification includes an optimized method to reduce the delay on connecting to the base station. This method is called Control Plane Optimization, and works by allowing a small payload to be transmitted through the control plane of the LTE network (Evolved Packet Core). This optimization has not been implemented by either module manufacturers or network operators by the time this report was submitted for review.

3.9 Summary

The main properties of the physical layer have been seen in this chapter. The technologies compared do not show significant differences in regard with security aspects, mobility or real time, as shown in table 3.4. The first difference to point out at the datalink level is the difference at multiplexing (sharing the medium between several users). LoRa use a multiplexing method similar to Aloha, which reduces the site capacity per channel in comparison with NB-IoT. The latter use the same multiplexing method as in LTE, where each user is assigned to a specific resource for each transmission to avoid collisions between different entities.

This chapter also explained how message acknowledgement works for each tech-
CHAPTER 3. DATALINK LAYER

<table>
<thead>
<tr>
<th>Multiplexing method</th>
<th>NB-IoT</th>
<th>LoRa</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication</td>
<td>OFDMA</td>
<td>Asynchronous</td>
</tr>
<tr>
<td>Integrity</td>
<td>Control plane only</td>
<td>All messages</td>
</tr>
<tr>
<td>Confidentiality</td>
<td>EEA encryption suite</td>
<td>AES encryption</td>
</tr>
<tr>
<td>Site capacity for 200 KHz bandwidth</td>
<td>213000 devices</td>
<td>21000 devices</td>
</tr>
<tr>
<td>Mobility</td>
<td>Cell reconnection (Nomadic)</td>
<td>Seamless (truly mobile)</td>
</tr>
<tr>
<td>Real time</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Report delay at 152 dB coupling loss</td>
<td>4749 ms</td>
<td>1187.84 ms</td>
</tr>
</tbody>
</table>

Table 3.4. Data link properties for NB-IoT and LoRa.

This is one of the main differences pointed out in this document. The limitations of LoRa for message acknowledgements makes this technology not suitable for applications where message acknowledgement is required.
4. Business Considerations

This technology comparison would be incomplete without analyzing the business implications of the reviewed technical properties. This section focuses on the technology availability, both in service and device wise, the costs, including module and operational costs and the licensing models for each technology.

Several use cases examples are given at the end of the section to help during the decision making process. Also the following table 4.1 resumes the business considerations and it is based on [2, Table 3].

4.1 Availability (Netherlands)

The possibility to use these technologies depend on their availability. Both the end devices and the network infrastructure are necessary for the utilization of these technologies. The reality is that with NB-IoT both the end devices and the network infrastructure are not fully developed.

4.1.1 End device Availability

Both LoRa end devices and gateways are available at the current moment. The main modem manufacturer at the moment is Semtech. It is the owner of LoRa intellectual property and it offers the first available modems, the SX127X family.

<table>
<thead>
<tr>
<th></th>
<th>LoRa</th>
<th>NB-IoT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deployment model</td>
<td>Private or Nationwide Networks.</td>
<td>Nationwide Networks</td>
</tr>
<tr>
<td>Device availability</td>
<td>Gateways and mobile devices currently available.</td>
<td>First mobile devices expected to be available on second half 2017</td>
</tr>
<tr>
<td>Service availability</td>
<td>Nationwide deployments in several European countries: Netherlands with KPN, Switzerland with Swisscom</td>
<td>Commercially available in Spain with Vodafone and in test stage in Netherlands (T-mobile)</td>
</tr>
<tr>
<td>Licensing model</td>
<td>Licensing on hardware. Paid by hardware vendors</td>
<td>Standardized technology</td>
</tr>
<tr>
<td>Module cost</td>
<td>Cheapest module found for €8</td>
<td>Not yet revealed</td>
</tr>
<tr>
<td>Operational cost</td>
<td>Not publicly available</td>
<td>Not publicly available</td>
</tr>
</tbody>
</table>

Table 4.1. Business considerations for LoRa and NB-IoT.
although other manufacturers like ST are offering their own products after paying royalties to Semtech.

NB-IoT was released very recently and thus there are no available devices yet. Nonetheless product development is at an advanced stage, as preproduction devices have been manufactured for testing purposes. First devices are expected to be available during second half of 2017.

4.1.2 Service Availability

Lora networks have been deployed at a nationwide level by several mobile network providers. Examples are Swisscom in Switzerland, KPN in Netherlands or Proximus in Belgium. NB-IoT is currently being deployed by Vodafone in countries like Spain, where the technology is commercially available through several cities. In the Netherlands T-Mobile is the network operator working to provide NB-IoT coverage. Commercial service is expected by 2018.

4.2 Costs

During the decision making process, the cost is one of the most significant factors to determine the technology to use. Two main costs have been identified: The module cost and the operational cost, which is the data plan fee paid to a network operator for providing connectivity to the end device. These two costs will be compared for NB-IoT and LoRa. This section also provides a cost estimate of LoRa gateways, since its price and usage of free spectrum makes feasible to install a private gateway for certain applications.

At this moment device prices are only available for LoRa, since NB-IoT modules are not commercially available. After a research on several providers, LoRa modems were found to have a value ranging from €7.95 (antratek) up to €23.45 (RSComponents).

NB-IoT module costs were not available at the moment of submitting of this report.

Operational costs for both technologies are not clear yet. Mobile operators do not disclose their pricing criteria, as they are still probing the market to know how to optimize their income.

LoRa rates are disclosed for the Netherlands by one single company: Simpoint. There are one-off costs for key activation that can suppose up to €5 per device, although the pricing decreases when activating a larger number of devices. Also there is an up to €5 monthly fee per bill. The cost per transmitted uplink message ranges from €0.4 to €0.00013, depending on the chosen bundle.
4.3 Use Cases

A set of use cases have been defined to help in the decision making process. The reasons behind the election of one technology or the other are explained for each use case. But first an enumeration of the factors involved in the decision making is given to help in their categorization.

Device type: A distinction is made between a sensor and an actuator. This difference is important to determine if the message flow is mainly going to be up or downstream.

Application type: We consider a message to be critical when it is not acceptable to lose it. The implication of sending a critical message is that an acknowledgement from the receiver is mandatory.

Transmission periodicity: A technology can be better suited than another depending on the transmission frequency that the system requires.

Area density: The interference of other devices is an issue to have in mind at dense areas.

Power consumption: The modem power consumption can be a constraint factor depending on the technical characteristics of a device.

The device type and information kind factors are used in figure 4.1 as part of a decision diagram to choose between the two technologies. These two factors are the ones considered in the diagram because these are the two main considerations to have in mind during the selection process. The characterization of the use case based on these two factors can refuse the utilization of a technology without considering the rest of factors.
4.3.1 Use Case 1

The first use case that we contemplate is of a sensor transmitting non critical information. An use case example is soil moisture metering or mailbox monitoring. For this case scenario we consider that a soil moisture sensor do not have to receive acknowledgements back from the base station (read section 3.4) since the loss of information is not critical for the system operation. Thus we can follow the diagram and see how both technologies are still an option for this use case.

Secondly we have to define if the device performs any function as an actuator. Since we have defined the device as a moisture sensor and it only requires to communicate when data is being uploaded, we can follow into the diagram the 'sensor' branch. This brand let us choose between LoRa and NB-IoT.

For this use case both technologies are suitable, so we can make our decision based on the rest of mentioned factors. These are the power consumption (where LoRa has an advantage), the transmission periodicity (NB-IoT can send information more frequently), and the area density (NB-IoT is better suited for densely populated areas with high levels of interference).

4.3.2 Use Case 2

The second case contemplates both a sensor transmitting critical data or an actuator. If we follow the decision diagram we can see how our choice gets limited to NB-IoT. For the critical data application the device needs message acknowledgement, whereas for the actuator application the device needs to receive instructions through the downlink channel and later on to acknowledge them by sending (uplink) and acknowledgement message.

Both cases are common because they need a bidirectional flow of information. Downlink transmission is very limited when using LoRa, that is why the recommendation is to use NB-IoT for this case.

Examples of such use case is an actuator over a railway junction or an alarm. For the railway junction the system that gave the order needs to know that the order was received and correctly executed. This means that the communication is symmetrical. For each downlink message (order) there is an uplink message (response).

4.3.3 Use Case 3

The last scenario contemplates the case where there is no connectivity at the location we need to place our devices. This situation is also described although it is not contemplated in the decision diagram (No choice can be made if either technology is not available).

Nevertheless for such situations there is one available option, to self deploy a network (figure 4.2). NB-IoT base stations can only be deployed by large network operators. The size and costs of these base stations is out of hand for the large
majority of organizations. At the moment of submission of this report no NB-IOT personal base stations (femtocells) are available.

On the other side LoRa base stations are very affordable. The price range is quite wide, but them can be found for €300 (The things network gateway [26]) and for even less for more simpler ones.

In conclusion, LoRa is a perfect choice for providing coverage at a locations with no other mean of connectivity.
5. Conclusion

Although this report is focused on the comparison of both technologies at a high level, a deeper view is available at the attachments A and B. Many different characteristics were compared through this report. In this section we focus only on the findings that are significant to decide between NB-IoT and LoRa.

These two technologies have many similarities, since both were conceived to be enablers for the IoT. Both technologies are categorized as LPWANs, and thus they are capable of providing long range communication at a low power and low data rates. This characteristics makes them ideal to be paired with power restricted embedded systems requiring the periodic transmission of small data payloads.

The most significant difference between NB-IoT and LoRa is the operation over licensed or unlicensed spectrum. A licensed spectrum is a range of frequencies that can exclusively be used by the licensee, while the unlicensed spectrum is open to the general public. LoRa operates over unlicensed spectrum (433 MHz and 868 MHz bands in Europe), while NB-IoT is meant to be deployed over licensed spectrum. This decision is important because European regulation impose a 1% duty cycle over the 868 MHz band.

The duty cycle limitation has a negative impact on the downlink capabilities of LoRa. A LoRa gateway has to distribute its transmission time among all the end devices that it serves. NB-IoT is not limited because it uses a licensed spectrum. Thus, NB-IoT is better suited for applications where messages need to be acknowledged.

LoRa modulation is very resilient against interference from other technologies, but it can not avoid collisions against other LoRa transmissions. This fact, along with the Aloha-like access to medium method, lead to a reduced number of devices that a LoRa site can handle, when compared against NB-IoT.

LoRa scalability is a concern at densely populated areas (higher amount of interference). All LoRa devices share the same unlicensed spectrum. This spectrum can not be expanded, thus a previous deployment that used to work fine may experience problems when a third party deploys its own LoRa devices. NB-IoT on the other side can be extended to use more spectrum, effectively doubling the capacity when doubling the bandwidth allocation. The use of NB-IoT is recommended in places where LoRa capacity is expected to be exceeded in the future.

An advantage of LoRa against NB-IoT is its reduced energy consumption. This aspect will be of importance in very power limited scenarios, although NB-IoT energy consumption is acceptable.

Both technologies offer similar data integrity and confidentiality properties. Both technologies can use AES encryption and can perform two-side authentication. The difference between the two is that NB-IoT handles keys with the use of SIM cards, while in LoRa the keys have to be handled by the developers. The LoRa Alliance has a set of recommendations for keeping a device secured, but no design safeguards impedes the misapplication or disregard of these recommendations.
For the interest of the system designer, LoRa offers a finely grained configuration of the communication. Many characteristics that are transparent to the developer with NB-IoT can be tweaked at LoRa. This properties can be either beneficial or disadvantageous, as a more detailed software implementation is less cost effective and more prompt to development errors.
Part II

NB-IoT GNSS Tracker development
6. Introduction

The second part of this graduation project involves the design of a new product using one of the LPWAN technologies previously studied. All the steps described in the product design were performed by the student, unless otherwise mentioned. This project consisted on a variation over a similar product that also was under development at that time. The similarities of the two projects led to the fusion of both into a single upgradeable product.

The developed system is a GNSS wagon tracker. GNSS stands for Global Navigation Satellite System, which means that the device needs to use GPS or GLONASS to gather its location and be able to track the position of a train wagon, to which the system will be attach to. In section 2.9 it was discussed if any of the studied technologies were capable to supply an accurate location. The findings shows that this is not possible for the amount of precision required, and thus it was necessary to use a GNSS system for this purpose.

6.1 Product Purpose

In the context of railway safety the need for a wagon tracker system was identified. Train wagons carrying dangerous substances comprise a big risk for the environment and the population. In case of an accident it becomes a priority for the emergency services to locate where this dangerous substances are within the train.

The purpose of this device is to accurately situate the location of the wagon at which it is attached to. This way the dangerous goods can be quickly located in the event of an accident of loss.
7. Functional requirements

In first place a meeting with the client was held to better define what the tracker should do. After this meeting a functional requirements document was written and discussed with the client. The process of defining the functional requirements required of several iterations. The most relevant functionalities are the following:

7.1 General Requirements

In first place it was defined that the device should have an active operating time of 5 years. The operating time depends on how long the batteries of the device will last. The operating time was set to 5 years due to the fact that a longer operating time will increase the size and cost of the tracker (battery), and more importantly, the value perceived by the end user diminishes after this many years because companies do not usually make plans more than five years ahead.

Another general requirement is that the device position accuracy should be better than 2.5 meters. Such a high precision is required because this is the main function of the device. This amount of precision was possible by using GNSS chips that combines not only GPS position data but also GLONASS and Galileo satellite systems.

7.2 PCB Requirements

After the research performed on LPWAN technologies, it was decided to use NB-IoT as the communication mean of the tracker. There are several reasons for using NB-IoT instead of LoRa. The first one is that NB-IoT handles at the datalink layer the message acknowledgement problem. With NB-IoT we can be sure that the data is going to be successfully transmitted when it is physically possible, and that there are not going to be any packet collisions or duty cycle limitations. The second advantage of NB-IoT over LoRa is that NB-IoT has enough bandwidth to perform an over-the-air update of the firmware. LoRa on the other hand is not capable to send several megabytes of information in a short time span.

It was decided to include into the PCB the footprints for both a SIM card holder and an eSIM chip. This decision will make the design more future proof for the incoming eSIM technology.

There are specific requirements for power monitoring (battery level), the GNSS technology to use (Chip with at least GPS and GLONASS) and the means to detect when the train wagon is moving. For the latter an accelerometer was included into the PCB design. The accelerometer chosen is low power and allows to switch on the device by moving it rather than by an external switch. This design decision help to provide a better insulation of the device since no holes for user interaction are needed.
CHAPTER 7. FUNCTIONAL REQUIREMENTS

7.3 Mechanical Requirements

The mechanical requirements are out of scope of this graduation project because the student did not perform the mechanical design. The main mechanical issues to discuss where the waterproofness of the device, the vibrations caused by the train and the way to mount the device into the train wagon.

For water protection it was required to achieve an IP65 waterproof level. In order to withstand the vibrations the railway Norm EN50155[27] was proposed. This standard points to the tests and typical values of Norm EN61373.

The way to mount the device into the train wagon is by using magnets. There are strong enough magnets that proof capable of withstanding the expected weight of the device and the vibrations caused by the train.

7.4 Norms and Regulation Requirements

Norm EN50155 specifies the different grades of temperature resilience a device should have. For this project the defined temperature range is from -40°C to 50°C.

The EN50155 Norm also defines the required resiliency against air humidity. The device must support a relative humidity average of 75% and a 90% humidity average for 30 consecutive days per year.

Electromagnetic interference resiliency and emission requirements are defined in Norm EN50121-4[28]. However section 5 of this norm states that emission measurements should be made over DC and AC ports. The device will not have such ports, so the electromagnetic emission compliance will be easy to meet. Also in section 1 of the same document it is said that ports meant to be used for radio communication should not comply with this norm. Anyways the device antenna will be placed inside the device, so this norm will not suppose any problem.

There is a specific requirement for complying with the CE marking. The reason is that the target market for the product is Europe. Because of the CE marking the device should also comply with RoHs 3[29] and WEEE[30] directives. The functional requirements document also contains specific requirements for this mandates.

7.5 Embedded Software Requirements

It is in this section where most of the functionality of the device is defined. The most relevant requirements are the following:

The device should immediately advertise when the train starts and stops. It is assumed that there will be two events per day (one to start and one to stop). The device should also transmit its position with a period of 30 minutes when the wagon is in movement. This is the main purpose of the device and it is assumed that the train is moving for 12 hours a day.

Another important requirement is the capability to change the operation parameters over the air. This way the device will meet the specific need of the end
As previously mentioned, the device should be capable to update its firmware over the air. This is one of the requirements that turned the balance in favor of NB-IoT rather than LoRa.

An important requirement is that the device should be able to monitor and restart itself when there is an error. This requirement is important since the device will be deployed far from any technician. Also for diagnose purposes it is required for the device to send its status information along with every position message. It was discovered during the technology comparison part of this project that and NB-IoT device is capable of sending 1000 bytes as one big chunk of information. Thus there is enough space left after the position coordinates to send the status information without affecting the battery life.

Finally the device should immediately inform when the position of the device changes its position on the vertical axis. This position change can either mean that the train wagon has derailed or that the device has fallen from the wagon. In either case it is important to provide the information for this event.
8. Implementation

It is in this chapter where a thorough description of the device design is provided. The project implementation, specially the system architecture, is strictly bonded to the requirements specification. Thus both elements jointly changed several times during the implementation phase. Ideally the system requirements are immovable, but we had the liberty to make changes during this phase because this project was internal to the company and was also the first for this kind of product.

In this chapter it is also explained how the whole product took a different direction to better optimize the company’s resources. The product pivoting occurred after the hardware design was finished and almost at the end of the internship period. This is the reason for describing the design process of the tracker in first place and the product pivoting in second place.

8.1 System Architecture

The final product architecture before the product pivoting is described in this section. Figure 8.1 is a block diagram of the different components that conform the GNSS tracker.

The main characteristic of this design is the employment of two different power

---

Figure 8.1. GNSS Tracker block diagram.
nets. This division was necessary because the batteries used to power the device provide a higher voltage than what several components can withstand. This batteries were chosen to comply with the power requirements of the NB-IoT module, which need high current rates during operation.

To handle all the required peripherals the STM32L051C6T microcontroller was chosen. This piece of hardware provides two SPI ports, that are used to connect to the flash chip and the accelerometer. The STM microcontroller was also connected to the GNSS and NB-IoT module through independent UART lines. This allows for an easy method for receiving the GNSS location and transmitting it through the NB-IoT module.

Besides the usual in-out ports required to connect the microcontroller to the peripherals, it is worth to mention the wakeup interrupt signal toggled by the accelerometer to wake up the microcontroller. This signal is represented in yellow color at figure 8.1. This architecture allows the microcontroller to switch to a very low power state and save the energy from the non-replaceable battery. Before reaching this low power state, the microcontroller instructs the accelerometer to toggle the wakeup pin either when movement is sensed or when movement is no longer detected. When this happens the microcontroller wakes up and evaluates the situation to further send position data or get back to the low power state.

GPIO ports were also used as enable signals. This signals are represented in green color at figure 8.1. The purpose for this signals is to cut off the power of the modules from the battery. This is achieved by toggling a MOSFET between the battery and the power pins of the modules. The reason this method is useful to reduce power consumption is that the modules used have a quiescent current even at low power modes. Since the MOSFETS have a much lower quiescent current than the modules, the overall power consumption can be reduced.

For power saving reasons there is also an enable signal connected to the battery monitor. The signal toggles a MOSFET and lets current to flow through the voltage divider of the battery monitor. The voltage obtained at the voltage divider is then connected to an ADC input of the microcontroller. If the MOSFET and enable signals where not used, a constant current will flow through the voltage divider and the batteries will not last the required time.

The battery monitor consists of a voltage divider connected to the ADC port of the microcontroller. The values for the voltage divider were calculated by a colleague having in mind the unusual voltage of the batteries. This way The microcontroller will be save from the battery voltage while being able to precisely measure its voltage. The voltage drop caused by the MOSFET was also taken in consideration during the calculations.

Finally, and in case of malfunction of the GNSS or NB-IoT modules, two GPIO ports were used to support the reset functionality that these modules are capable of. Also it is not described into the block diagram, but the spare GPIO pins were used at a resistor array for the purpose of versioning the board by hardware. This way the same firmware can be written on different board versions and the firmware can later modify its behavior based on the lecture from this resistor array.
8.2 Component Selection

The following are the actual components used for the GNSS tracker design. Passive components such as resistors, capacitors and connectors are not described. A rationale for the selection of each component is also provided in this section.

Microcontroller

It was decided to use an STM32L0 microcontroller due to the low power consumption required for the tracker and the previous experience of the company with this family of devices. The specific model is an STM32L051C6T. It was chosen because it is the most basic one from the STM low power family that has enough pins for the required application. This hardware includes the 2 SPI lines and 3 UARTs needed to connect to the peripherals and to have a spare debug UART. At the end of the internship it was being considered to use a different microcontroller with greater flash size, since the amount of required software surpassed the capacity of this microcontroller.

External Clock

Although the microcontroller can operate with its internal clocks, it was required to use an external one to make use of the RTC functionality. This functionality is required to allow the microcontroller to periodically wake up and take a location measurement. The clock used resonates at 32 MHz.

Accelerometer

The LIS3DHTR Accelerometer was chosen against its competitors because of two reasons: The first one is that it has a low power mode while measuring. This allows for the accelerometer to be constantly sensing when the train starts moving and when the technician is switching on the device. The second reason is that it provides programmable interruption lines, which are necessary to wake up the microcontroller when needed.

Flash Memory

An external flash memory is needed to store a new firmware version and allow an over the air (OTA) firmware update. The component is an MX25R4035F Flash module. This provides 64KB of storage. The same capacity as the flash memory advertised at the microcontroller.

NB-IoT Module

The BC95 NB-IoT module was selected against its limited set of competitors because it offers different versions capable to use different frequency bands. This is important
because the module can be selected to adapt to the frequency of a particular network operator, without fixing the hardware design to one network operator in particular.

**GNSS module**

The GNSS module has to be capable to comply with the 2.5m accuracy requirement, as well as keeping the power consumption to a minimum. As it is explained in the next section, most of the power requirements for the GNSS tracker are due to the GNSS module. The best module for this project is the Quectel L76-L.

This module has the required precision and has a low power mode while saving the ephemeris data. Saving the ephemeris data allows for a hot start operation the next time the module is powered on, which vastly reduces the required time to fetch a new location.

Another characteristic of this module is that it incorporates an internal signal amplifier, so no extra effort has to be placed on designing the peripheral components that this module requires.

### 8.3 Power Estimation

During the component selection process it was required to perform a power estimation analysis, since the GNSS tracker is a device powered by non-rechargeable and non-replaceable batteries. The figures shown correspond to the final component selection.

**GNSS module**

To perform this calculation it was needed to make some assumptions. This assumptions were based on the requirements specification and are the following:

- Two 3100 mAh batteries in parallel.
- Location gathering and transmission every 30 minutes.
- Train journey duration of 12 hours per day.
- Time to first fix (Hot): 5 seconds (1 second in datasheet).
- Time to first fix (Cold): 35 seconds.
- 3% battery self discharge rate per year.
- 2 years shelf period before device deployment.

For the components mentioned in the previous section and the above assumptions it was calculated that the device will be capable to continuously operate for 8 years. Since the requirement specifies a life expectancy of 5 years, the requirement was fulfilled.
The battery duration of the device vastly depends on the time required by the hot start procedure of the GNSS module. Figure 8.2 shows that more than half of the energy is used during this procedure. On the datasheet it is said that one second is required to get the first fix. However, and based on the calculations results shown in Figure 8.3, it was decided to set the hot-start time to first fix to 5 seconds to allow for a margin of error. For 5 seconds it can be seen that the device is capable to function for 8 years. If the GNSS module truly requires one second to get a fix then the life expectancy of the device will reach 14 years.

8.4 Schematics Design

Once the components were chosen it was time to design the schematics and the PCB. The first step was to import to the designer software (Altium) the new components that were not currently available. This action is necessary for each component before being able to include it into a schematic. The process of adding a component
into the software requires of three steps. The first one is the creation of a symbol to represent the component into a schematic. The second step is to define the footprint of the component, this is to define the physical design of the PCB required to mount the component. For example a through hole component will require to define at the footprint the required holes, as well as the pads and silkscreen. Once the component has a defined symbol and footprint, it is time to create and add a three-dimensional model of the component. The software includes a 3D tool that will be proof to be helpful for testing the fitting of the PCB into the case. These steps have to be performed for every component that needs to be added into the software library.

The addition of new components into the library’s software was done on demand and in parallel with the schematics design. It is important not to spend time on adding components if they are not yet needed, since on the first stages of the design process the component selection may change and the time used to add a new component will be wasted. Figure 8.4 shows the component creation process, including the symbol, footprint and 3D design of the GNSS module component.

The second step was to design the schematics. For this purpose it was decided to split the design into several schematics, starting with a high level schematic that included the connections between the different schematics. Six different schematics were defined apart from the Top level schematic. The design for each of them is explained in the following sections:

**Power Schematic**

The power schematic included three different functionalities. The first one is the design of the voltage regulator circuit for supplying power to the microcontroller,
Figure 8.4. GNSS module (L76) component creation process.
Flash and accelerometer. This voltage regulator consists of an ADP160 integrated circuit and a couple of decoupling capacitors as described in the IC documentation. The second functionality described is the use of cable connectors to connect the batteries and the use of zero Ohm resistors to be able to isolate each battery separately. The third functionality is the battery voltage regulator. This regulator includes the novelty from a company colleague of using two MOSFETS to isolate the microcontroller from the high battery voltage. The microcontroller drives the first MOSFET, which gets in charge of driving the second one. This second MOSFET has its source directly connected to the positive terminal of the battery, whereas the drain is connected to the voltage divider. The values of the voltage divider where calculated having in consideration the voltage drop caused by the MOSFET and the tolerable values supported by the microcontroller ADC.

**Microcontroller schematic**

The design of the microcontroller schematic required to add the microcontroller into the software’s library, which process is time consuming as explained in the previous section. This schematic also includes the decoupling capacitors for the microcontroller. The requirements for the decoupling capacitors are described into the microcontroller’s documentation. In this documentation it is also described how the reset and boot pins should be connected. The clock selection and design into the PCB was obtained from the parallel project being done at the company.

The microcontroller schematic also includes the description of the SWD (Single wire debug) and debug UART connectors. The first one is used to flash the microcontroller, while the second one is used during the software development to output debugging data useful to the developer. The versioning resistor array previously described is also specified into the microcontroller schematic. Each version of this board has a different layout into the resistor array.

**GNSS schematic**

The GNSS module has two interesting design decisions on its schematic apart from the expected decoupling capacitors. The first one is the use of a MOSFET to drive low the reset pin of the GNSS module. The second one is the use of a microcoaxial surface mounted connector for the antenna. As well as the layout of a zero Ohm resistor and to open capacitors footprints into the antenna output of the GNSS module. The purpose of the connector is to be able to connect a patch antenna to the board. This patch antenna will be mounted inside the case of the device, stucked to the cover of the case. On the other side, the purpose of the resistor and capacitors footprints is to allow for a future antenna impedance tuning, in order to maximize the performance and power consumption of the module. Figure 8.5 is a close-up view on the GNSS antenna.
CHAPTER 8. IMPLEMENTATION

Figure 8.5. GNSS antenna circuit with tuning resistor and capacitors.

NB-IoT schematic

For the NB-IoT schematic the guidelines described at the documentation were followed. This includes the selection of a power source capable of running the NB-IoT module, the use of the adequate decoupling capacitors (which were extraordinarily large) and the driving to ground of the reset pin using a MOSFET. The schematic also defines the RC circuit for the antenna tuning and more interestingly the inclusion of a protection circuit for driving the signals directed to either the SIM module or SIM card. One of the requirements for this product was the possibility to choose at the assembly stage whether to use a SIM card or a SIM chip, so both components were included into the design. Also this specific NB-IoT module includes a SWD interface, so a connector was included to allow future connectivity to this interface during the development process.

Accelerometer schematic

The accelerometer schematics only includes 5 components other than the integrated circuit itself. This schematic connects four decoupling capacitors between the input power and ground. These capacitors have different values in order to cope with different noise frequencies at the power input. The design also includes a pull-up resistor into the CS pin. This was decided after carefully reading the documentation for the IC, and the reason it is needed is to keep the communication line inactive instead of floating when the microcontroller goes into low power mode. The in-out pins of the microcontroller are in high impedance state when the microcontroller is in low power mode.

Flash schematic

The flash schematic is the most simple of the full set. The flash chip only needs two decoupling capacitors as well as two pull-up resistors. The first one is used into the reset pin, while the second one is used for the CS input.
8.5 PCB Design

The third step was to define the shape of the PCB. The PCB obeys the shape of the case. For this project the case is same as for a previous project developed by the company, so the previous project PCB was used as a template, although some changes were made into the shape to better suit the GNSS schematics.

Altium designer software was used for the design of schematics, the components footprints and the PCB routing. After defining all the footprints that were not previously available, it was time to begin with the Component placement process. This process requires of several iterations along with the routing process, since many times it will be found that there is not enough space to route a trace and several components will need to be reallocated until everything fits nicely.

The PCB ended up requiring four different layers. The Top one is the main routing layer, the second one is mainly a ground plane, although it contains some traces for the case where no routing could be performed on the first layer. The Third layer contains two power planes, each one working at a different voltage. and the bottom layer was used as a ground plane. This last ground plane was designed to allow for a future revision of PCB containing an SMD antenna for the NB-IoT module. Figure 8.6 shows a 3D view of the design.

To test the fitting of the PCB and its surface mount components into the case a PCB fitting test was performed using the 3D options of Altium designer software. Each component was 3D modeled along with its footprint and the 3D shape of the PCB was also imported into the software. The result can be seen in figure 8.7, where
no collisions are produced either with the case or the battery. The collisions are defined into the software as a minimum distance that should always be kept between the components being tested. In this image the two magenta cylinders represent the batteries, while the green shape represent the space available for mounting the SMD antenna mentioned before.

Unfortunately this hardware design was never built and physically tested due to the product pivoting. But nevertheless a similar version was produced and tested by a different team member. This other version and the work performed by the student is explained in section 8.7, where a photograph of a real PCB can be seen.

8.6 Software Design

The work on the software was limited to the last two weeks of the internship. The main factor for the lack of time was due to the first theoretical part of this thesis plus the characteristic changes on the go that these kind of projects deal with, as well as the build times required to manufacture a batch of prototype PCBs.

Nonetheless I was able as a team member to get involved in the software design process. This includes the use of several tools such as Jenkins and GIT to ensure the development of homogeneous code that seamlessly work between all the members of the software team.

The software written during this period was the GNSS driver module. The
language used was C and the task was to provide an interface to communicate to the module through a serial line (send commands and interpret the data received). The most relevant developed functions were:

- To send a sleep command to reduce power consumption.
- To adapt a lexical analyzer to fetch the altitude and longitude coordinates.
- To fetch the new type of messages by adapting callback procedures.
- To set a minimum dilution of precision to fetch accurate coordinates.
- To define and set the commands needed for the module to work as required.

The software use a state machine to schedule and control the different functionalities performed by the device. Figure 8.8 describes the aforementioned state machine. We can see how it has five different states that can be grouped into three families; The first one is the Init state. This state is visited only once on the life time of the device, and its purpose is to configure and initialize the device as required during manufacturing. After this first stage the device transits into the storage module. During this state the device minimizes its power consumption and only checks if the device is being started by a technician. If this is the case then the device enters into the group of active states. Nonetheless the device can be switched off and go back into the Storage mode.

The defined active states are the idle, measurement and communication states. The Idle state have the purpose to set the device into a low power mode until an event triggers it into the measuring state. These events can be either that the train starts or stops moving or that a timer expired and the device needs to take a new location measurement.
The measuring state is self explanatory. The software initializes the required data structures and then fetches the GNSS coordinates by making a call to the GNSS driver module developed by the student. This driver handles all the communications with the GNSS module through the serial line (UART) and GPIO pins. The GPIO pins are used for reseting the device and for waking it up again when needed. The GNSS driver also implements a parser to correctly gather the required information among all the data that the GNSS module outputs.

Once the location position has been fetched, the device enters the communication state. It is in this state when the device gets ready to transmit the fetched data by enabling the power on the NB-IoT module and configuring it to connect to the base station. Once the connection has been established the data is transmitted.

The main difficulties found during the software development had to do with the parsing of the data coming from the serial line. At first it took more than the expected amount of time to figure out how to use the callback functionality that was already being developed as part of the companies library. This callback functions are necessary to follow the same system architecture as the rest of the software components use at the company.

Once that the callbacks were figured out it was time for the actual parsing development. The microcontroller lacked of flash space, so it was decided to manually implement all the string manipulation functions that were required for the GNSS driver. These functions are for example the conversion from string numbers (a chain of characters from 0 to 9 plus the full-stop symbol '.') to actual integers and floating point numbers.

8.7 Product pivoting

After the PCB design of the GNSS tracer it was decided to merge this design with another product being developed at the same time; a temperature and accelerometer sensor. The new idea is to modify the temperature sensor project to permit the mount of an external PCB containing the GNSS module.

For this product change the student was in charge of the development of the GNSS add-on board, as well as to define the in-out pins required to connect to the main board. A different team member got in charge of the modification of the Temperature sensor board to add the required connector to the add-on board.

Figure 8.9 shows the 3D design of the GNSS add-on board (red PCB). The board was easy to design, since the schematics for the previous board were still valid for this design and the components footprints were already made.

The most difficult part was to decide how and where to mount this add-on board. Not much space was left and both the batteries position and the possible location for the connector were restricted. A solution was achieved by mounting the PCB in a 90 degree angle. The GNSS antenna is expected to be attached to the add-on module and mounted on the inside of the case. This way the product will have no external connectors. Image 8.9 shows the fitting test of the GNSS add-on board.
Figure 8.9. Fitting test of GNSS add-on PCB.

(red PCB) into the case and Temperature sensor PCB.
9. Testing

It was not possible to test the hardware design of the PCB since it was not manufactured. However the design got checked by a different team member (as the company design process specifies) and the design got approved. However the hardware test document was written. This document states all the different tests that should be performed on the board to guarantee the correct design.

The hardware test document consist on 34 different test. For each of them the test procedure is explained, as well as the expected result. Each test entry also has the space to indicate the measured result and help in the investigation of possible design faults. The document is divided into sections corresponding with the schematics. The test defined are the following:

Power

For the power tests the different voltages had to be tested. In first place the voltage has to be measured at the batteries, to make sure that these are working properly and that the rest of the board can be tested. After this it is time to test whether the voltage regulator is providing the correct voltage to the rest of the circuit. For this case the expected voltage at the output of the LDO is 3V.

The second phase of the power testing checks the correct operation of the battery monitor system. First of all the MOSFET operation os tested by reading the voltage at the voltage divider using the microcontroller’s ADC. This way this procedure can be performed using software and thus it can be later used for automatic testing during production. The expected voltage when the MOSFET is open is 0V, while the expected voltage while the MOSFET is conducting is 2.115V. This value considers the voltage drop caused by the MOSFET itself.

The third last stage of power testing is measuring the current consumption of the device while performing different tasks. This includes for example measuring the battery voltage, or while being kept in low power mode. At the manufactured board it was detected an abnormal power consumption during low power mode. The investigation carried out ended up finding that a MOSFET was missing a pull-up resistor, which caused it to be floating when the microcontroller switched to low power mode. The microcontroller has a high impedance on its output pins when is kept in low power mode. This MOSFET ended up conducting when it was not supposed to do so, which leaded to an abnormally high power consumption.

Microcontroller

Several microcontroller pins had to be tested to ensure the appropriate electric conditions are met for its operation. This includes for example to check the voltages of the power supply input, the NRST pin and the BOOT0 pin. The NRST pin should have a constant voltage of 3V. This voltage should be very stable to avoid a reset of the microcontroller. This is the reason for connecting a capacitor close to this pin.
The same check is performed at the BOOT0 pin. In this case the expected voltage result is 0V.

At this stage the capability of the microcontroller to be programmed is also tested. An SWD cable must be connected to the proper connector and the microcontroller should be properly flashed. This operation is expected to be performed along with the testing of the debug UART, since the flashed program can output through this interface a 'Hello world message'. If this message is correctly received then it can be ensured both functionalities.

The last test procedure involved with the microcontroller is the testing of the versioning resistors. For each board version the pins will read different values. For the first version a value of 0 is expected on all version pins.

**Accelerometer**

For the accelerometer the connectivity with the microcontroller is checked. In particular two different test must be performed: The first one checks the correct operation of the SPI interface. For this purpose the device ID is read a thousand times at a baud rate of 4MHz. The expected result for each query is 0X33 (the device ID). If this value changes at any of the queries it may indicate that the communication is not stable and further hardware analysis is required.

The second test checks the correct wiring of the wakeup pin. This pin causes an interrupt at the microcontroller to wake it up from the low power consumption mode. To perform this test the accelerometer must be configured to trigger this signal when movement is detected. The microcontroller also needs to change to the low power mode state and the device must be shaken to see if the microcontroller wakes up.

**Flash**

The testing of the flash module requires three stages. The first one is the testing of the SPI interface in the same manner as with the accelerometer module. The connection between the two components needs to be reliable. The second stage checks that the flash module is capable to switch between the standby and the low power modes. For this test the deep power command must be sent from the microcontroller. The expected result is a drop of 4.9uA in the current consumption. When the flash module is powered back on the expected result is an increase of the same value on the power consumption of the device.

Finally the communication lines must be tested when the microcontroller is in low power mode. As previously mentioned, during this state the microcontroller pins stay at a high impedance state. The CS and RESET pins of the flash module need to be at a logic high state in order to keep the flash in an idle state. This means that the expected result (expected voltage) for this pins is 3V. If other value is measured it will indicate a hardware design error related to the pull-up resistors that this pins need.
NB-IoT

Similar test where planned for the NB-IoT module. The UART and SWD interfaces can be checked by the transmission of data. Also the SIM card connectivity needs to be checked in this case, since it is directly connected to the NB-IoT module.

Test to check the power modes where also defined. The NB-IoT module must be capable to switch to sleep mode. If this is the case the expected result should be a current drop of 5.95mA. Also the module should be able to be kept in this state while the microcontroller switched to the low power mode. The expected result when the microcontroller switches its mode is a current decrease of 3mA. If the current rises instead this will mean that the NB-IoT module wakes up from the sleep mode because of a hardware design issue.

It is also necessary to check the voltages of the RESET pin when the microcontroller is kept in the low power mode. The expected output voltage is 3V for this case. This voltage must also be measured when the reset pin is set to high by the microcontroller. The expected voltage for this test is 0V. The reason is that the output pin of the microcontroller is connected to the gate of a MOSFET, which then shorts the RESET pin of the NB-IoT module to ground.

GNSS

For the GNSS module the UART interface has to be checked the same way as for the other components. The RESET pin of this module also have the previously mentioned tests, where the voltage should be the batteries voltage when the MOSFET is not conducting and 0V when the MOSFET gate is triggered by the microcontroller. This module also have a FORCE-ON pin that needs to be tested. This pin is the only mean for the microcontroller to wake up the GNSS module from deep sleep mode. The expected voltage for this pin is 0V when the microcontroller is also in low power mode. If the voltage is not set to logic low then the device will power up and consume a lot of power when it is not required. The FORCE-ON pin should also have a voltage of 3V when the microcontroller sets this pin to logic high. Otherwise the microcontroller will not be able to wake up the GNSS module.

Firmware

The design of the temperature PCB got the chance to be tested in actual hardware, since a small batch was manufactured. This design has many similarities with the GNSS board, since it uses the same microcontroller, NB-IoT module and power design.

On the other hand it was possible to perform basic tests over the GNSS firmware developed by the student. This was possible by connecting a GNSS developing board directly to the power nets and UART of the manufactured Temperature board. Image 9.1 shows the setup for firmware development and testing. On this image we can see the temperature sensor board (green round PCB) connected to the
Figure 9.1. Workbench for testing and development of GNSS firmware.

bench power supply, a patch antenna for the NB-IoT module, the microcontroller programmer (white device with ST logo) and the GNSS development board.

If we take a closer look we can see how the GNSS board and the main PCB connects using 6 different wires. Two of this wires are the VCC and GND to supply power to the GNSS board directly from the main PCB, the gray wires are the UART connection between the modules, and the yellow and orange wires are connected to the GPIO pins on the microcontroller and the FORCE-ON and RESET pins on the GNSS module.

All this connections were required to test the correct operation of the individual functions that where tested. This functions were possible to be tested in sequence thanks to a main program that performed all the developed functions sequentially. This functions are described below.

In first place the GNSS module was waked up toggling the FORCE-ON pin. The expected result from this action is the activation of the LEDs of the GNSS board and the increase of the current measurement showed at the power supply’s display. This results were as expected and thus the FORCE-ON functionality was tested and guaranteed to work.

The second test is the acquisition of the location through the UART interface. For this test the device was configured to send the location result to the computer and then the computer was configured to show the result in a terminal. The position results where stable and correct in location to the place where the test where being
performed.

The third last step was to check whether the microcontroller is capable to set the GNSS module to deep sleep mode. The expected result was a decrease on the current consumed and the power down of the LEDs at the GNSS board. This test showed the expected results since the GNSS module was successfully set to desired power mode.
10. Conclusions and future Work

The evolution of LTE and more specifically the introduction of NB-IoT enables the creation of a new set of battery powered IoT devices. This thesis work demonstrates this by designing a battery powered embedded device capable to operate for more than 5 years without any maintenance, all this while being able to communicate through long distances (wide area network) due to the incorporation of a NB-IoT module.

This work shows the complete process for a product design. At first several meeting were carried with the product owner to specify the product requirements. Later on the schematics and PCB design took place in order to fulfill the tight requirements.

The power calculation showed in this work is specially useful, since it shows what is the power consumption to be expected from the NB-IoT technology. This results showed that with current lithium-ion battery technology it is possible to power up an LPWAN device for several years.

Also a through set of tests were designed to validate the hardware design. This document proofs to be useful for the upcoming projects, since it typifies several common mistakes that the hardware must always be checked for. Finally the application of the GNSS tracker will improve the safety of people and the environment at the railway industry once its development is completed.

On the other hand and due to the lack of time there are several steps that are required to obtain a final product. The first one is the test of the GNSS add-on board by making a prototype PCB and the fitting test with the temperature sensor board. The time required for this should not be greater than 2 weeks, since the manufacture of the PCB does not take long.

There is still a lot of work to do regarding the software. Many requirements have already being developed by different team members, but some others are still pending to be developed, such as the OTA firmware upgrade.

It is also worth to mention that part of the software can not be written yet. This is the case with the NB-IoT module. The reason is that this technology is so new that both the module manufacturer and the network operator have not yet implemented the full set of functionalities that the NB-IoT specification defines. This is the case with eDRX, a technology that allows an NB-IoT end device to be kept in a low power mode for an extended period of time without loosing the session with the base station. This functionality is required to improve the power consumption of the device.

A future step will be to develop the testbeds required for production. Testbeds are used to efficiently test that the manufactured hardware was correctly assembled. It also provides the mean to flash the firmware to the device and make it ready to use by the end user.
Bibliography


A. Energy Consumption calculation

A.1 NB-IoT

Five phases have been identified for an uplink message transmission and an ulterior acknowledgement from the base station:

Synchronization with the base station: This phase require 960 ms for reception of the NPDSCH followed by 30 ms of NPBCH reception during a 810 ms period. The device is in idle mode when not receiving or transmitting.

Random access procedure: During this phase the device is considered to transmit for 320 ms. The prach period can take up to 664 ms. The device is in idle mode when not transmitting.

Narrowband Physical downlink control channel reception: The device receives uplink time allocation for 430 ms. This phase takes 1090 ms. The device is in idle mode when not receiving.

Data transmission: The device needs 2780 ms to transmit 20 bytes of user level data.

Ack reception: The devices keeps listening for 430 ms to retrieve the acknowledgement from the base station. The acknowledgement period takes a total of 1090 ms. The device is in idle mode when not receiving. We assume a successful transmission and no retransmission is required.

The power consumption for the different modes were extracted from the SARA N2 product summary [15]:

- Idle: 6 mA at 3.6 V = 21.6 mW.
- Rx: 20 mA at 3.6 V = 72 mW.
- Tx: 220 mA at 3.6 V = 792 mW.

From the previous timings and power consumptions we can calculate the total power required for the transmission of a 200 bytes message. The results are displayed at Table A.1:
<table>
<thead>
<tr>
<th>Mode</th>
<th>Power</th>
<th>Time</th>
<th>Energy consumed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transmission</td>
<td>792 mW</td>
<td>3100 ms</td>
<td>0.682 mW/h</td>
</tr>
<tr>
<td>Reception</td>
<td>72 mW</td>
<td>1850 ms</td>
<td>0.037 mW/h</td>
</tr>
<tr>
<td>Idle</td>
<td>21.6 mW</td>
<td>2440 ms</td>
<td>0.0147 mW/h</td>
</tr>
<tr>
<td>Total energy consumed for transmission</td>
<td></td>
<td></td>
<td>0.7337 mW/h</td>
</tr>
</tbody>
</table>

Table A.1. Energy consumption of NB-IoT device SARA-N2.

<table>
<thead>
<tr>
<th>Mode</th>
<th>Power</th>
<th>Time</th>
<th>Energy consumed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transmission</td>
<td>145.2 mW</td>
<td>1187.9 ms</td>
<td>0.04791 mW/h</td>
</tr>
<tr>
<td>Reception</td>
<td>35.64 mW</td>
<td>802.8 ms</td>
<td>0.00795 mW/h</td>
</tr>
<tr>
<td>Idle</td>
<td>4.62 mW</td>
<td>1000 ms</td>
<td>0.00128 mW/h</td>
</tr>
<tr>
<td>Total energy consumed for transmission</td>
<td></td>
<td></td>
<td>0.05714 mW/h</td>
</tr>
</tbody>
</table>

Table A.2. Energy consumption of LoRa device SX1272.

A.2 LoRa

LoRa neither has to synchronize with the gateway either to assign resources previous to the transmission. An uplink transmission comprise two phases: The transmission of the user data, and the reception of the acknowledgement through one of the two available reception windows. We assume the coding rate to be 4/3 and we use the data optimization option to correct for clock drift. The preamble length was set to 8.25 symbols.

The LoRa calculator provided by Semtech was used to calculate the power consumption for SX1272 LoRa module [12]. The transmission of 20 bytes at SF 12 is 5939.2 ms, while the reception time is 802.8 ms. The delay between the end of the transmission and the opening of the receive window (known as RECEIVE_DELAY1) is by default 1 second. Thus, the device will be in idle mode for 1 second during each transmission. The calculations results for LoRa can be found at Table A.2.