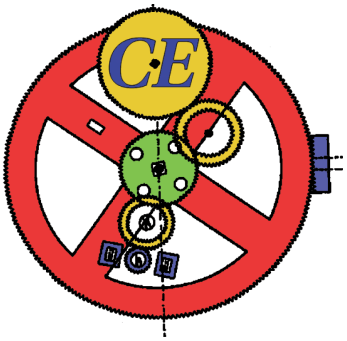

Computer Engineering
Mekelweg 4,
2628 CD Delft
The Netherlands
http://ce.et.tudelft.nl/

2017

MSc THESIS

Design and implementation of a LoRa based Gateway

Shubhankar Dixit

Abstract

The project focuses on the hardware-software co design of a LoRaWAN based industrial IoT gateway used for proprietary applications. Long Range Wide Area Network, abbreviated as LoRaWAN is a network and data layer running over the LoRa PHY layer which operates at 868 MHz[29]. The surge in LoRa has led big market players like SemTech to licence devices operating over a free network. Also gateway manufacturers have seized on the opportunity of this growing market and 2 industrial gateways, [15], [27] have captured most of the market. In order to break this monopoly, FactoryLab B.V, an Industrial IoT company from Zwijndrecht, The Netherlands has developed a low cost Linux based gateway which can be used for proprietary applications. The project aims at developing and comparing the gateway with industry standards. Although the hardware for the gateway couldn't be tested in time for the finalization of this report, various tests are performed using an improvised hardware setup which emulates the FactoryLab hardware and the results approximated and compared to the industrial gateways. The Range to Cost ratio for the test setup was calculated to be 4.8 meters/euro and when pitched against the other gateways, showed a maximum increase of 26.6%.

CE-MS-2017-10

Keywords: LoRa, LoRaWAN, Linux, Single Board Computing system, Range, Cost, Range to Cost ratio, Kerlink, Multitech, KiCad, PCB design, fabrication

Faculty of Electrical Engineering, Mathematics and Computer Science

Design and implementation of a LoRa based Gateway

THESIS

submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

in

EMBEDDED SYSTEMS

by

Shubhankar Dixit
born in Ajmer, India

Computer Engineering
Department of Electrical Engineering
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology

Design and implementation of a LoRa based Gateway

by Shubhankar Dixit

Abstract

The project focuses on the hardware-software co design of a LoRaWAN based industrial IoT gateway used for proprietary applications. Long Range Wide Area Network, abbreviated as LoRaWAN is a network and data layer running over the LoRa PHY layer which operates at 868 MHz[29]. The surge in LoRa has led big market players like SemTech to licence devices operating over a free network. Also gateway manufacturers have seized on the opportunity of this growing market and 2 industrial gateways, [15], [27] have captured most of the market. In order to break this monopoly, FactoryLab B.V, an Industrial IoT company from Zwijndrecht, The Netherlands has developed a low cost Linux based gateway which can be used for proprietary applications. The project aims at developing and comparing the gateway with industry standards. Although the hardware for the gateway couldn't be tested in time for the finalization of this report, various tests are performed using an improvised hardware setup which emulates the FactoryLab hardware and the results approximated and compared to the industrial gateways. The Range to Cost ratio for the test setup was calculated to be 4.8 meters/euro and when pitched against the other gateways, showed a maximum increase of 26.6%.

Keywords: LoRa, LoRaWAN, Linux, Single Board Computing system, Range, Cost, Range to Cost ratio, Kerlink, Multitech, KiCad, PCB design, fabrication

Laboratory : Computer Engineering
Codenumber : CE-MS-2017-10

Committee Members :

Advisor:	Dr. Ir. A.J.van Genderen, CE, TU Delft
Chairperson:	Dr. Ir. S. Cotofana, CE, TU Delft
Member:	Dr. Ir. M.A. Zuniga, ES, TU Delft
Member:	G. Strietman, FactoryLab B.V.
Member:	H.W. Snippe, FactoryLab B.V.

"The art of progress is to preserve change amid order, and order amid change."

Dedicated to my parents, my Mother for showing me everything I am and my Father for showing me everything I can be.

Contents

List of Figures

List of Tables

List of Acronyms

ABP Activation By Personalisation

AES Advanced Encryption Standard

ALOHA Advocates of Linux Open-source Hawaii Association

AppEUI Application Unique Identifier

AppSKey Application Session Key

BGA Ball Grid Array

C/R Ratio Cost to Range Ratio

CSS Chirp Spread Spectrum

CTR Cost to Range

DDR2 Dual Data Rate 2

DRAM Dynamic Random Access Memory

DevAddr Device Address

DevEUI Device Unique Identifier

GPRS General Packet Radio Service

ISM frequency band Industrial, Scientific and Medical frequency band

IIoT Industrial Internet of Things

IoT Internet of Things

LRPWAN Long Range Personal Wide Area Network

LoRa Long Range

LoRaWAN Long Range Wide Area Network

MAC Media Access Control

MCU Micro Controller Unit

MHDR

MIC Message Integrity Code

MTM Machine To Machine

NVM Non Volatile Memory

NwkSKey Network Session Key

OAT On Air Time

OTAA Over The Air Activation

PCB Printed Circuit Board

PLL Phase Locked Loop

RADAR Radio Detection and Ranging

RFU Radio Frequency Unit

ROM Read Only Memory

RSSi Received Signal Strength Intensity

RootFS Root File System

SF Spreading Factor

SPI Serial Peripheral Interface

TSCH Time slotted channel hopping

UART Universal Asynchronous Reception and Transition

UNB Ultra Narrow Band

Acknowledgements

In the final memoir of your journey over the past 3 years, starting by thanking people who helped you finish it, seems fitting.

I want to begin this note by conveying my heartfelt gratitude to my supervisor, Prof. Arjan van Genderen. After a rocky few months prior to the start of this thesis, I was in need for some support and assurance and you supplied it in abundance without saying much. Your patience gave me the confidence to work to my hearts content and I could never appreciate that enough. Also I would like to thank the team at FactoryLab, along with my supervisor and colleague Gertjan Strietman for making me feel comfortable in a new environment and giving me the freedom to do my thing. I'm glad I did this and I;m excited about the future. Finally, I would like to thank the University for giving me and thousands of dreamy eyed students the perfect platform to grow and also a crisp reality check that nothing worth achieving is easy. "Let's see you get out as fast as you got in." Took me 3 years, but it was all worth it.

There is nothing in this world more important than family and if I haven't done it enough, I want to thank my parents, Deepali and Atul Dixit, who for 25 years, have made unending sacrifices without breaking a smile. Thank you for everything that you are and for always having faith. "It's you!", my mom would say, "Of course you can do it!". My brother Shantanu, who has eternally held me higher in his graces than I deserve, you will always be my first and best friend. Continuing with family, I want to thank Apurva for playing such an important part in helping me finish this. I don't need to tell how much you mean to me, because I tell you everyday. Rounding up I want to thank Gautham, Ekta, Sunny and Dimitris. You guys are my family, you know that. My every achievement has your stamp on it.
I want to thank Siddharth, Nupur, Siva and Saurav for being the best people a guy could ever live with. My twin Monica, Leoncio, Joaquin, Aniruddha for never letting me feel lonely. I want to thank all the Kaenebaays, you guys are my oldest friends and a continuous support. Also you guys are my most reliable source for sports news, which is cool. Finally I would like to thank my friends, my mates from my band, Hirad, Kostas, Michael, Thomas and Nick, thank you guys for bearing with my absences.

Also, anybody who has ever prayed for my success and also ever wished for my failure, thanks a lot, here I am.

Shubhankar Dixit
Delft, The Netherlands
August 25, 2017

1 Introduction

The internet of things is a fast growing, multi-faceted area of the hi-tech development community and with good reason. The 'smart' era has forced obsolete standards, like single function devices, high power consumption and 'wired' communication out and made way for low power, wireless and multi action devices. As seen from the Gartner Hype Cycle [14] in Figure 1.1 which is a graph like cycle for emerging technologies, going through various stages, has put the IoT platform at the peak of the expectations for 2014 and 2015 along with the most highly anticipated technology standards, and platform has dropped back to the high development phase indicating further development and rise in the arena. The importance of IoT in a growing market, commercial and industrial, is rising meteorically and has since drawn comparisons to the rise of the internet itself. The nomenclature of the two can be confusing as IoT is a sub domain of the internet of everything with the key differentiating idea being that the requirement for the availability of resources for efficient internet of things operation is very low. The 'things' mentioned here are physical (generally) wireless devices, which means that every aspect of the communication and deployment vector, which includes power, energy, memory and processing time, needs to be reduced. Eventually the information collected and processed by the devices can be communicated to the internet or the web for human interaction and feedback.

Considering a scenario where there are multiple devices available which gather information about many parameters which could benefit a function, there are a few components involved. The First component is the device itself, which could be a sensor node, a cell phone etc. These devices are one of the many 'things' in an IoT network and are constantly consuming power and collecting information which needs to be relayed to the user. The other major component is the server or storage and processing device with computational ability far greater than the end devices. This stores and processes the data sent by the devices for human intervention. As important as these two things are, there is a third and equally important component in this wireless network which is a gateway. This gateway is a device that is placed between a node and the server and gathers data from multiple nodes, verifies the integrity, packs and transmits to the server. In a WiFi and smart phone scenario, the wireless router is the gateway. This project focuses on the development and design of a LoRa [4] based gateway which is to be deployed for proprietary applications.

This project was carried out at FactoryLab B.V, an IoT company based in Zwijndrecht. The company provided all the specifications of the project and also all the resources required for implementation. In this chapter, 'Introduction', Section 1.1 details the motivation behind the project and what led to the conception of the idea. Section 1.2 formally defines the project, Section 1.3 points to the goals of the project

and the Section 1.4 details how the report is structured.

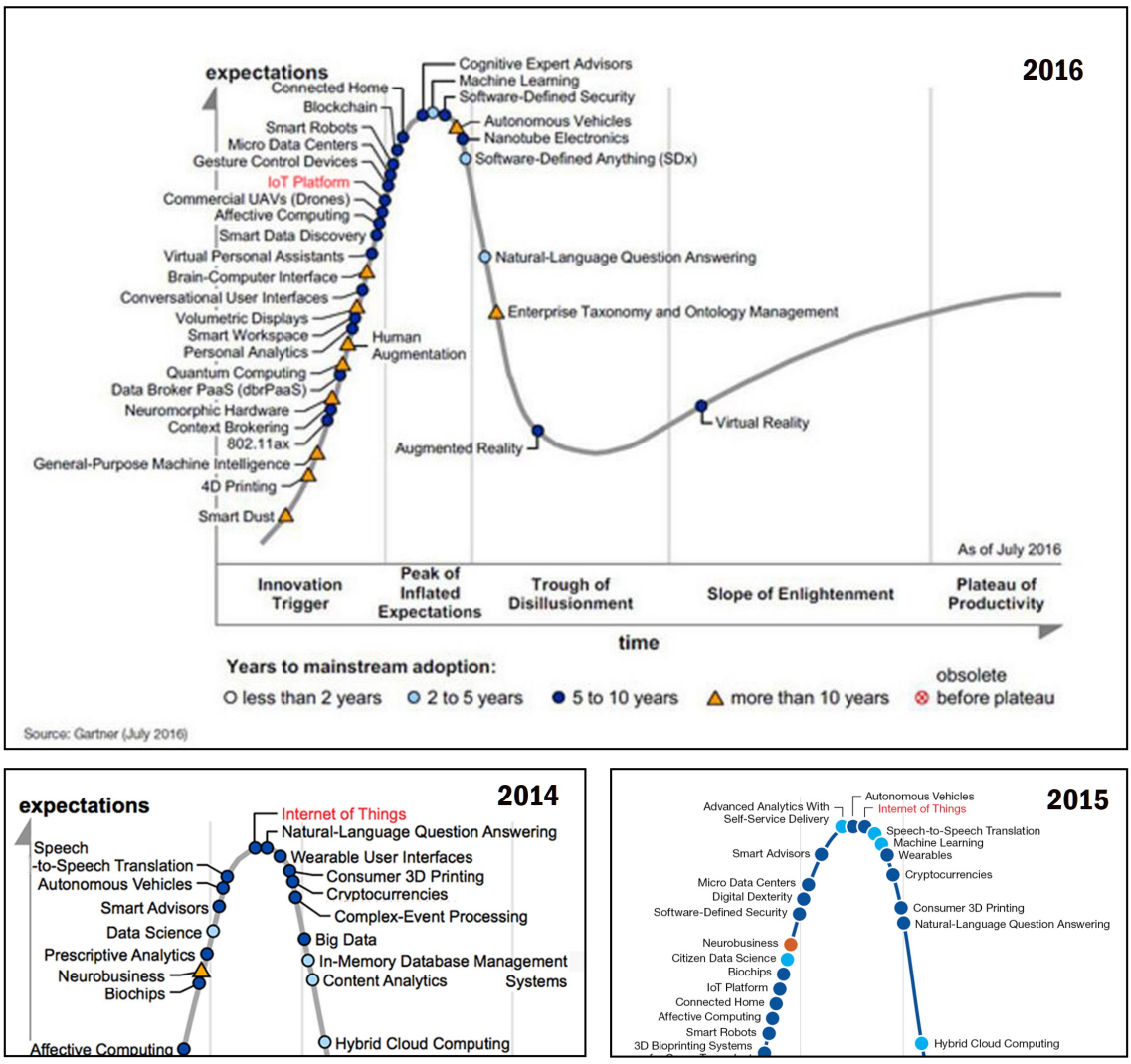

Figure 1.1: The Gartner Hype Cycle for 2014-2016

1.1 Motivation

The advancement in communication technology in the last decade has made wireless communication protocols irreplaceable in the domain of embedded electronics. Current

trends have promoted the use and implementation of many radio based protocols due to the fact that short range radio transmission is inexpensive, secure and also easily available. Short range radio standards like WiFi and Bluetooth have been on the pinnacle of short range communication for a while now and have proven to be reliable. Long range communication on the other hand has taken a back seat to this surge and has been under developed. The surge to perform multiple complex actions with little resources limits the usage of WiFi and Bluetooth due to their range and scalability issues. While GHz range protocols are not effective for long range communication, cellular can be used for long range operations as they are already scaled to a large extent but there is the problem of cost as the licences required to operate at cellular levels are expensive. This is where Long Range Personal Wide Area Networks or LRPWAN's are found to be useful and one such standard is LoRa. LoRa stands for "*Long Range*" and is a communication standard that is used for long range, low power applications.

LoRa was first introduced in 2012 and over the last few years has been extensively used for IoT applications. One of the more important features of using a LoRa based network is that along with being long range and low power, the LoRa band is free use. This means that anybody who has the required hardware which can communicate on the LoRa band, can do so for free. This aspect of the standard has made it very attractive to industries and hobbyists alike and this interest has promoted the growth and development of the network and also the communication framework which is known as LoRaWAN. Although the network is free to use, the hardware required to capture and modulate the frequencies on the network, is not free and is licensed and patented by a few major market holders. While anyone with the resources can develop the hardware radios required for communication, the use of the LoRaWAN protocol for LoRa transmission has to be registered and certified by the LoRa Alliance. More on this can be found in Section 2.1.
In the development cycle of the LoRa network the main focus has been on the development of the end nodes as seen in figure 1.2 and not on the gateways which are used as intermediaries. Hence, this lack of development has led to a monopoly held by a few companies in the development of LoRa gateway which is an essential component of the LoRa network without which the application fails as there is no communication between the node and the user.

This monopoly on the other hand presented an opportunity to develop a LoRa based Linux gateway for proprietary use. The motivation behind using the Linux based system is it's ease of use, its popularity and also the fact that most network relay systems are based in Linux as they are more convenient to develop. This project aims at disrupting this monopoly and introducing a low cost and competitive gateway developed for FactoryLab B.V.

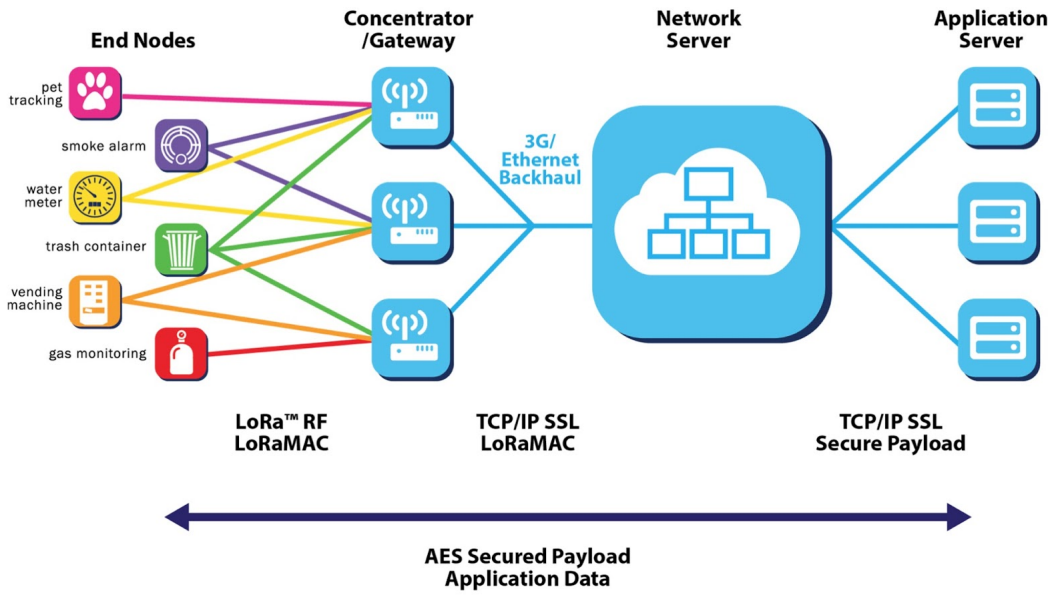

Figure 1.2: The general LoRa network structure

1.2 Definition

The project aims at hardware-software co design of a LoRa Gateway which will be deployed for proprietary use with the FactoryLab nodes for various industrial IoT applications. The basic idea is to create a cost effective Linux based LoRa gateway that can compete with the industry standards in terms of range and operation while reducing costs significantly. The gateway will use a LoRa based Semtech[36] radio, which is the standard radio module currently being used for gateway systems. This radio runs a certified LoRaWAN stack and is coupled with a base band concentrator to create a standard LoRa communication environment. This communication module is coupled with a Linux based computing module which will be developed in house at FactoryLab. There are many restrictions placed on the development of this device which are detailed in the following segment.

The LoRa radio available for receiving and transmitting LoRa packets are standard and there is no working around their choice. Along with being standard, the base-band concentrator [37] which is required for multi channel operations is very expensive and no costs were reduced in that choice, this meant that costs had to be reduced by other choices. Apart from monetary restrictions, for the functioning of the LoRa network, the LoRaWAN stack that is certified for the radio chip-sets has to be used. This is because the LoRaWAN framework and the contents of each packet need to remain standard as per regulations of the LoRa Alliance, this is to promote a truly free network protocol. In the future a new LoRa stack can be created and certified by the LoRa Alliance with more freedom for software development.

1.3 Project Goals

The objective of this project involves the low cost hardware development of the gateway and implementing a Linux based operating system complimenting it. The research question then is to optimize the range to attain an acceptable range to cost ratio. Following are the goals of the project sequenced in the order of their prospective completion.

- Accumulation of sufficient knowledge of LoRa and the LoRaWAN protocol to understand the challenges of the project.
- Creating the model software and hardware architecture which can be used for development.
- Choosing the best hardware and developing the hardware schematic for PCB development using KiCAD[16], which is a free to use hardware development software, currently being developed by CERN[9][10].
- Creating the optimal PCB footprint for high speed transmission using KiCAD.
- Development of on board software and Linux kernel specific to the chosen hardware.
- Testing and optimization of the software for loss less reception of LoRa packets.
- Analyzing and comparison of range with Industry standard gateways in terms of range to cost ratio.

1.4 Report structure

This thesis report starts with an Introduction to the project and its requirement, followed by chapter 2, which specifies the background information on the subject which supports the understanding of the project. This chapter also covers comparison between various LoRa devices. Chapter 3 provides a comprehensive literature overview. Chapter 4 focuses on the methodology used in the development of the gateway for both the software and the hardware segments and the implementation of the gateway with the particular specifications. In Chapter 5, results of the project can be found and finally Chapter 6 presents summary, conclusions and future aspects of the project.

2 Background

Long Range Wide Area Network or LoRaWAN, was released around two years ago by the LoRa Alliance as a wireless communication protocol. LoRa is proprietary radio technology which is owned and patented by the SemTech Corporation. The main aim of the protocol is to provide energy sustainable communication solutions for battery powered devices. The term LoRa is used in many respects in the communications arena, where it could imply the physical layer protocol used in the communication standard, the architecture of the network itself or the MAC layer protocol known as LoRaWAN. LoRa is a long range connectivity standard which utilizes the Chirp Spread spectrum technology at 868-900 MHz for Europe and United States respectively. The MAC protocol for the LoRa standard is defined by the LoRaWAN MAC protocol, this defines the physical layer and the data link layer standards for LoRa. The physical network was initially developed by SemTech industries for low power, high range applications. In a typical LoRa network many end-devices or nodes can connect to a singular middle device which then furthers the information to the user end. This middle device or the Gateway is the prime focus of this thesis.

The LoRa network follows the "star of stars" topology as is depicted in figure 1.2. Following is the basic architecture of a LoRa network: The end devices or nodes are wirelessly communicating with the gateway by the means of a LoRa connection with a low throughput. These gateways then further communicate the received and interpreted packet frames to the network server or back-end with a higher throughput interface like Ethernet. The gateway acts as a bidirectional relaying system and a protocol converter. In this chapter we will look at the background of LoRa and LoRaWAN in detail and also at their differences.

Keywords: Spreading Factor, LoRa, LoRaWAN, On Air time, Class A,B,C end nodes, DevEUI, AppEUI, AppSkey, NwkSkey, Range, cost.

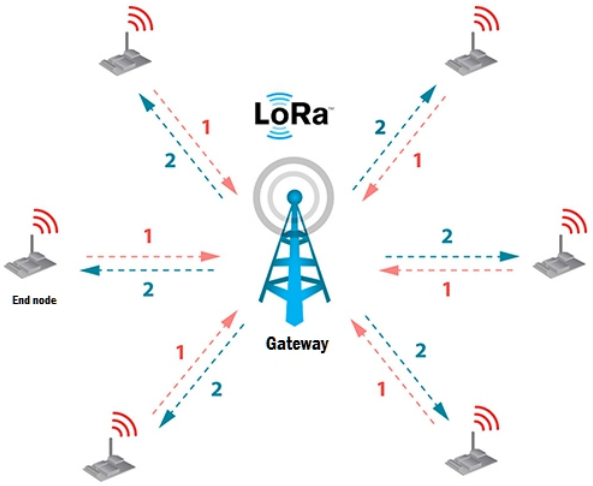

Figure 2.1: Star-topology used in LoRa connection

2.1 LoRa vs LoRaWAN

2.1.1 LoRa

LoRa, as is its name, provides long range and reliable connectivity by using a chirped spread spectrum technique with a large spectrum. The LoRa PHY layer is licensed and patented by the Semtech corp. The LoRa layer utilizes a derivative of the Chirp Spread Spectrum (CSS) which was developed for radio detection and ranging(RADAR) applications within the 1940s. Using CSS, the signal is spread over a larger frequency (ISM) band by creating a linear frequency alteration [24]. CSS employs the use of signals which have a high bandwidth time, generally greater than 1 which helps resist interference. Keeping the chirps in a broader spreading spectrum also reduces multi-path fading and hence improves the performance in urban environments [26]. CSS modulation techniques are notable for low power consumption and hardiness against channel degradation challenges like interference and multi-path attenuation. This modulation technique offers a variety of data speeds for various frequency ranges. In LoRa networks,the data rate is chosen according to the range needs, and there's a trade-off between rate and communication range. This ability of modulating different chirp rates gives LoRa the advantage of being able to decode many signals a the same time. These different chirp rates are known as spreading factors and LoRa numerates six of these. These spreading factors are created by compromising throughput for higher on air time.A higher SF results in higher range of transmission. As seen in [26][24], the payload is a key factor in the On Air Time and the OAT can range from 0.9 to 1.2 seconds on SF12, which is the highest spreading factor. A table with various spreading factors, data and bit rate can be found in table 2.1. The LoRa modulation scheme also adds redundancy by utilizing variable cyclic error correction.

Data Rate	Spreading Factor	Radio bit rate (bytes/sec)
0	12	31
1	11	55
2	10	122
3	9	220
4	8	390
5	7	683

Table 2.1: Spreading factor vs data rate for LoRaWAN

2.1.2 LoRaWAN

LoRaWAN, unlike the LoRa physical layer, is open and is defined and maintained b a group of supporting parties known as LoRa Alliance [22]. The LoRaWAN specification as described by the LoRa alliance [22] defines that LoRa operates in the 868MHz ISM frequency band for Europe and at 900 Mhz in the United states. These frequencies are specified under the ISM (industrial, Scientific and Medical) band. The According to the European telecommunications standards institute, each node device should have a duty cycle of 0.1% and 10% depending on the operational sub bands [22]. The Effective Radiated Power (ERP) emitted by each node is thus regulated [24]. The 3 ISM band regulations are defined as:

- (867-868.6 MHz), (869.7 - 870 MHz) :OAT with 1% duty cycle which amounts to 36 seconds per hour and an ERP cap of 14bBm.
- (868.7 - 869.2 MHz) :OAT with 0.1 % duty cycle which amounts to 3.6 seconds per hour %d an ERP cap of 14bBm. (LoRa)
- (869.4-869.65 MHz): OAT with 10% duty cycle which amounts to 360 seconds per hour and an ERP cap of 27bBm.

LoRaWAN has mainly three kinds of classes of end nodes depending on the needs of the protocol. These are Class A,B and C devices. Class A device implementation is mandatory while the other two are optional.

2.1.2.1 Class A

Class A end devices are **bi directional devices** where each up-link burst is followed by two down-link receive windows. This is an ALOHA type system where the end devices schedules its own transmission time depending on the transmission and communication needs. This also means that down-link communication has to wait for the next upload link. Either of the two receive slots can be used to download data transmitted over LoRa and in case the data is received in the first receive window, the second receive window is skipped [26]. As mentioned before, Class A implementation is mandatory which means

that although Class A can be modified to be run as Class B and C, the the device should still be compatible with class A operations [21]. The class A packet specification can be seen in figure 2.2

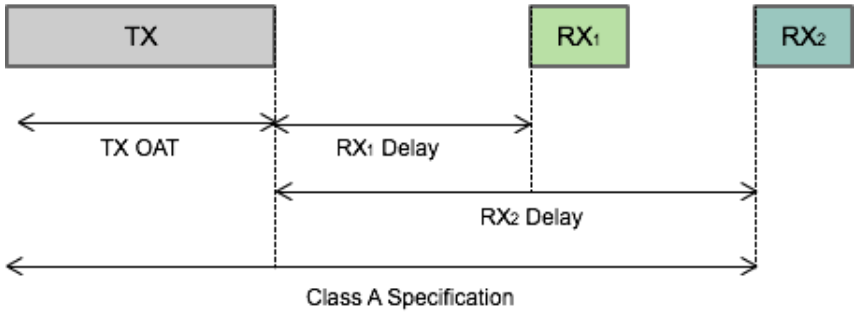

Figure 2.2: Class A packet specification

2.1.2.2 Class B or beacon

Class B end devices are similar to class A devices with the difference that these devices have **Scheduled receive slots**. This means that a class B device can schedule extra receiving slots, additional to the two receive slots defined in class A, as per the requirement. These receive slots are initiated after receiving a time synchronization packet from the gateway in order to alert the server that the device is in receive mode. The gateway also transmits beacons periodically to the nodes to main time synchronization [26].

2.1.2.3 Class C or continuous reception

Class C devices are **Bi-directional with maximal receive slots**, which means that these class of end devices are always listening except when they need to transmit. Class C devices require higher power consumption than the other two classes but this comes with the advantage of reduced latency (minimum) between node and server [21].

The protocol utilizes different keys for device network and application to create tighter security. LoRaWAN incorporates many device identifiers. Each device is assigned a Device unique identifier called DevEUI which is determined by the developers/users. The device address is minimum 4 bytes and up to 8 bytes. A 8 byte application identifier known as the AppEUI is used to identify the application for the node. Further, the security is based on an AES 128 bit model operating in counter mode, or CTR mode [6]. The 128 bit AES key or the AppKey is used to create a 16 byte Network Session key (NwkSKey) and a 16 byte application session key (AppSKey) which are known to both the network server and also the end node. The MIC or message integrity code, which is used to check the correctness of the message is generated using the NwkSKey while the AppSkey is used to encrypt the payload and also decrypt it. The encrypted data is

generated by using the XOR operation with the streamed AppSKey.
The packet structure for a LoRaWAN packet can be found in figure 2.3 and further discussion on the packet structure and join procedures is performed in chapter 4.

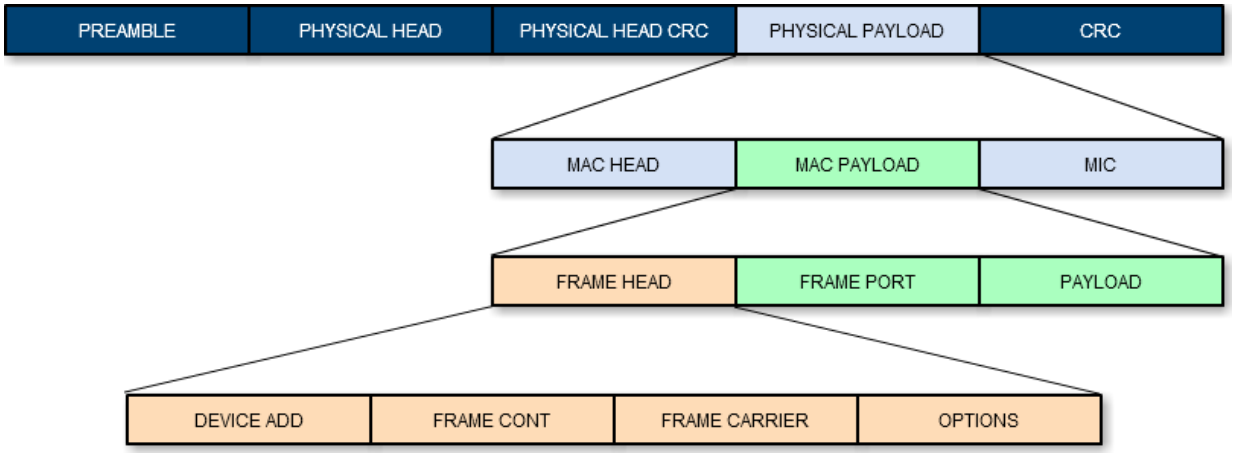

Figure 2.3: LoRa packet format

2.2 Comparison with industrial standards

This section discusses the various commercial LoRa gateways available in the market as of the execution of the report. The aim of the comparison made is to lay the background for the importance of optimizing the range to cost ratio of a LoRaWAN gateway. The Gateways discussed are the Kerlink LoRa IoT station [15]and the MultiTech Conduit[27] with a 3G interface. While there are other gateways available in the market, not all of them are eligible for this comparison for various reasons. The Libelium [12] gateway runs the a simple link protocol created by Libelium and is thus only supported by their own devices and not all LoRaWAN devices certified by the LoRa Alliance [13]. Another gateway that is commonly spoken of is The Things Network gateway [42] which a LoRaWAN based gateway available at a low cost of 300 euros. This gateway, although perfect for the comparison is not available as of the conclusion of this report and will be available by 2018 and would provide a more accurate sight of its abilities. Another gateway called the Link Labs LL BST-8 is also spoken of a lot but is also not qualified for this comparison as it uses a proprietary network and data protocol called Symphony Link [18] which, like LoRaWAN devices, runs on the LoRa PHY layer.

2.2.0.1 Range

The range of a gateway is the most important factor when considering its purpose and although a part of the LoRaWAN component family, the gateway is not required to be highly low power as it generally stationary and can be run by a constant voltage supply. The gateways considered as industry standards vary greatly in achieved range of communication as they differ in hardware design and field of use. There are many

factors that can influence the range of a gateway like the range of the gateway depends on the height at which the gateway antenna is located coupled with the ability to minimize packet loss. Most of the gateways listed in this discussion are based on the same SemTech base-band concentrator coupled with SemTech LoRa chips but the range of each of the gateways varies drastically. Further discussion of the commonality in the LoRa concentrator and the LoRa radio modules is carried out in the Section 4.1.

2.2.0.2 Cost

To achieve maximum range, high quality hardware needs to be deployed in a high quality design environment. This means that the cost is influenced by the design choices of the board itself along with the component choices and features incorporated. As stated earlier and discussed in Section 4.1, the basic working model of all the gateways is similar with the LoRa communication module, the base-band concentrator and the radios being the same in each of the gateways which begs the question of the disparity in the prices of each of the devices. The costs of the device itself can vary with the intended use of the gateway and the quality of the components used. Besides this argument, there is significant possibility for reducing the price of a LoRaWAN gateway. Table 2.2 shows the comparison between the range and measured costs of the aforementioned industry standard gateways[20].
A combination of optimizing the cost of the developed FactoryLab Gateway along with maintaining the or possibly improving the range in comparison to some of the gateways mentioned earlier, forms the core of the projects definition.

Sno.	Name	Max. Advertised Range(Mts)	Cost(Euros)
1	Kerlink IoT Station	$\tilde{9}000$	$\tilde{1}500$
2	Multitech conduit	$\tilde{4}000$	$\tilde{5}00$

Table 2.2: Various Costs and ranges of LoRa Gateways

2.3 Conclusion

In this chapter, there is a semi detailed overview of the background information regarding the definitions of LoRa and LoRaWAN and some differences between them. Table 2.2 refers to the spreading factors used in the LoRaWAN protocol and an overview of the LoRa packet format is followed by the representation of the LoRa format in 2.3. Some light is shed about the various industrial LoRa Gateways which are used for testing and comparison purposes and the Range and cost factors of these gateways are discussed in Section 2.2.

3 Literature Overview

Although the LoRa platform was conceived and patented in 2013, the technology didn't take full flight until the following year. With the highly efficient and readily available LoRaWAN stack available to free use, not a lot has also gone into furthering the technology but more into creating applications and refining the platform. This non exploration into further studying and understanding the LoRa platform implies that not a lot of literature has been published about LoRa but a quite a few articles and projects have been published on various platforms on the developmental aspects. This chapter focuses on the the literature and background survey conducted on understanding the LoRa platform and its applications in an industrial scenario. Section 3.1 focuses on the Software aspects of the project consisting of the LoRa segment along with the Linux operating system research. Section 3.2 presents the conclusions from the review and some major findings derived.

3.1 Research and Findings

Stan, Timnea, and Gheorghiu provide the an initial evaluation of the LoRa standard and major insight into the key concepts involved in understanding the surface of the protocol itself. [38] also provides key understanding of the differences between LoRa and other radio standards and the various possible applications possible. This provided a good starting point into the research process and terminologies involved. Magrin, Centenaro, and Vangelista in [24] present a detailed overview of the LoRa platform in their introduction to their topic. There are details present about the IEEE 802.15.4 standard which is the LoRa base and its use in the IoT environment using the star topology in the unlicensed Sub GHz spectrum. [24] also highlight the effective performance of the Long Range Personal WAN networks and the viability in modern day IoT scenarios. The paper also highlights a 95% success packet transmission rate through a simulated experiment. While [24] highlight the effectiveness and viability of the LoRa platform, Rizzi et al. present [35] which is focused on the usability of the LoRa network in the industrial environment. IIoT or industrial Internet of things follow the the Industry 4.0 paradigm which also includes cyber-security and other high yield fields, which incorporate machine to machine protocols like LoRa for sustainable production.[44, 34, 5, 39, 41, 19] present a more detailed view on industry 4.0 and its aspects, which are unimportant to the project but important in understanding the direction in which the domain is headed. [35] also focuses on prospective modular approach of modifying the LoRaWAN [4] layer with a more TSCH like approach for a more industrially robust platform.

After understanding the implications of using LoRa as an industrial MTM com-

munication protocol, it was imperative to focus on understanding the LoRaWAN protocol itself which included the physical layer and the network layer concepts. Ramachandran et al. in [30] demonstrate the use of the LoRa protocol and one of its most important features, the ability to minimize deployment complexity by where devices and nodes automatically connect to the network and transmit acquired data, hence minimizing the power consumption. Further it explains the importance of CSS [33][43] and its impact on the scalability of the LoRa network, providing multiple data rates over frequency ranges. A Table comparing the data rate, Spreading factor [29][43] and the radio bit rate can be seen in table 2.1. This paper also highlights testing techniques for range which are incorporated in this project.

Testing is the key aspect of this thesis report and there are many factors that are to be considered while executing the tests. Wixted et al., in [43] demonstrate the importance of testing a LoRa device in an urban environment which is ideal for IoT scenarios, i.e. building with glass and concrete. Testing for both the LoRaWAN and LoRa protocols was performed and provided key differentiation between the two. Most importantly reliability tests were performed with the test setup to test the reliability of the network which is the percentage of packet transmission and reception. Although the primary use case of the FactoryLab gateway is in industrial environments, understanding the impact on the range and signal reliability in an urban and indoor environment can provide a benchmark for future development. On the topic of testing the range and reliability of a network, Kim et al. in [17] demonstrate a novel testing approach for range and reliability testing by testing while stationary and moving, which shows that there is not a big difference between the two cases thus creating an easier testing environment. Also tests are performed between LoRa and an UNB(Ultra Narrow Band network) which shows that although LoRa consumes a bit more power, there is a drastic difference in the signal interference and reliability of the network. Ayele et al. also perform analysis on the performance of the a LoRa based radio for indoor applications. [2] presents the implementation and results for impact of distance on various LoRa parameters. The effect of distance between gateway and node on parameters like RSSi, Packet Error Rate, Coding rate and transmission power are demonstrated and the results are helpful in approaching the test of the FactoryLab gateway with a clearer objective and expectation.

Mahalakshmi et al. in [25] helped understand the importance and working of Uboot in an Embedded Linux Environment. Uboot is a bootloader which is universally used in the development of embedded Linux by creating a set of make files for setting up the root file system and essential libraries. Most of the hardware development knowledge though, was referred from [45] which is a book on building embedded Linux systems. The book details the process flow while developing a Linux based environment using many tools and the precautions that need to be observed. Yaghmour, Masters, and Ben-Yossef also detail the hardware requirements and optimal choice of hardware for efficient execution.

3.2 Conclusion

This chapter focuses on reviewing some of the literature that was researched in the development of the FactoryLab gateway and its execution. While [24, 35, 38, 30] provide a clear insight into the LoRa standard and major understanding of the LoRaWAN protocol can be derived from [21] as well, [43, 17, 2] help understand and evaluate the various test scenarios and expected results from the LoRa gateway which is a key aspect of the project. Finally, [25, 45] help understand the importance of the Linux operating system and [45] paints the exact picture needed for developing a modern Linux based embedded platform.

Methodology and implementation

4

The methodology and the implementation of both the software and hardware of the gateway will be described in this chapter. Section 4.1 discusses the hardware implementation and design of the gateway. This section also highlights the hardware choices made and also the design choices. Section 4.2 discusses the techniques used for both the Linux kernel operating on the embedded system and the LoRa software that the OS executes. Section 4.3 presents a conclusion of the chapter.

Keywords: LoRa, LoRa packet, Join procedure, Activation By Personalization, Over The Air activation, Linux, Buildroot, Uboot, DDR2, ARM 32 bit, NAND Flash, KiCad

4.1 Hardware

While the software is the key differentiating factor within the domain of LoRa Gateways supported by the fact that the LoRa Radio modem used in each gateway remains the same, hardware choices which include design and component choices can also significantly vary the costs involved in the LoRa Gateway conception. An example to support this is the comparison between the Kerlink IoT station [15] which is considered the industry standard in LoRa Gateways and the Multitech Conduit [27], also an industry favourite. The Kerlink gateway while using the same combination of base-band concentrator and transceivers as the Multitech, is significantly more expensive(2.5 times) and also achieves almost 2.5 times the range of signal reception. This information when coupled with the fact that both of these gateways are available in Linux based models, makes a strong case of hardware design making a possible difference to the Range to Cost ratio. In this section, the choices for the design of the hardware and the components chosen are highlighted.

4.1.1 Computing module

The gateway hardware consists of two main module, the computing modules which contains the computing systems and all peripherals required for network server operation and the LoRa module which is designated for LoRa communication. This section is regard to the former. The computing module required for Linux operation consists of a processing unit, Memory - which consists of non volatile memory and a Random Access Memory, network interface which is a choice depending on the application, user interface and a power interface. Other peripherals and devices can be used per application [45].

4.1.1.1 CPU

The choice of CPU is important for attaining the required results out of an embedded system and while most embedded systems developed for IoT applications consist of Low power or Ultra Low Power CPU's for longevity of the device, this is not a hard requirement for a gateway as the gateway is primarily used in a stationary host location which is not remote. The only other requirement is that a Linux based operating system requires a memory interface and some bus serial bus interfaces for user and network connections. One of the most common manufacturers in this arena is ARM semiconductors [23]. Currently the Linux operating system supports over 40 different ARM based processing units [45]. This implies that the support and documentation for creating the ARM based Single Board Computer(SBC) is common practice and highly documented. Also FactoryLab uses ATMEL produced CPU's for most of their projects, which narrows the options to a few CPU's.

Two main aspects govern the choice of any CPU, form factor and clock speed. The form factor for most modern CPU's is a multi-hundred pin Ball Grid Array(BGA) in a Surface Mount Package which makes the choice quiet straight forward. The Clock speed on the other hand presents a bigger challenge as the choice is much more varied. While most PC based development platforms require GHz range of clock speed for efficient operation, a smaller more specific purpose embedded system can function effectively in a MHz range. This is due to many factors that separate a PC based environment an a function based embedded system, like no requirement for a display unit and driver which utilize most of the CPU, limited peripherals, etc. For this reason, an ATMEL based CPU in the 300-500 MHz range along with a 217 ball BGA Package was chosen for the gateway. Although the clock speed could have been chosen to be higher, compromises had to be made in order to reduce the cost of the hardware setup.

4.1.1.2 Memory

The computing system for the gateway requires 3 different types of memories to be installed in order to function. Firstly the RAM was chosen as a Dual Data Rate, DDR2 which is spaced at either 512MB or 1 GB clocking at a maximum of 400 MHz. This is a decent enough choice as the MCU has DDR2 support and 512 MB should be enough memory to be partitioned between the LoRa Modem and the network interface which are the two major modules running most of the time. Also trace matching for DDR3 is a bit more complicated and more expensive, this made a 400 MHz 512 MB RAM a good and cost effective choice. The RAM was also only available in a BGA package which is another aspect to consider. Apart from the RAM, a Non Volatile Memory (NVM), which is a mass memory and a data-flash is used. The block addressable NAND Flash memory is chosen as it's affordable and is available in large memory sizes. The NAND Flash is available in a Thin small outline package(TSOP) which makes it less complicated to embed on the board, but compromise on the board pace had to be made, more on that in Section below. The Data Flash is used as a Fail safe memory which means that it's going to be majorly unpopulated as the software is stored on the NAND Flash along with the kernel and the bootloader, but in case there is a memory violation in the NAND

Flash, the bootloader is also going to be placed on the data-flash as a back up. The memory map and layout can be found in Figure 4.1.

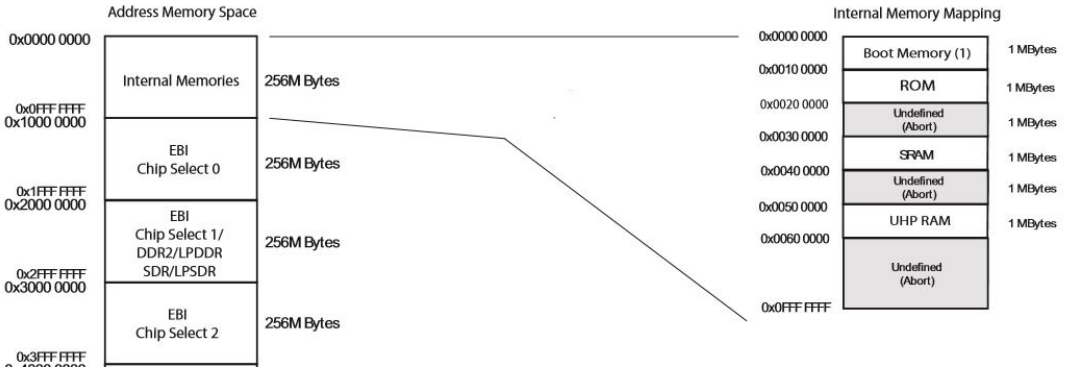

Figure 4.1: Internal memory map for the Atmel CPU

4.1.1.3 Network Interface

The network interface, along with the LoRa communication interface form two of the most important and fundamental blocks of the LoRa Gateway. The LoRa module is used to connect with the end nodes over the LoRa network and the network module on the gateway connects with the network server through which the data can be analyzed, processed and displayed. The LoRa connection module remains quite standard across the various gateways tested as the LoRa PHY layer can only be used while using the Semtech base-band concentrator in combination with two Semtech transceivers. The connection block diagram can be seen in Figure 4.2. The Sx1301 [37] is a multi channel base-band concentrator which is required for LoRa based gateway devices. The Sx1301 consists of a network emulator which can emulate 49 LoRa channels over 8 physical channel interfaces. This ability of multiplexing channels and with different chirp selected frequency, creates a 'base' band for the network and concentrates all incoming signals in one place for demodulation. This means that multiple devices transmitting at multiple spreading factors or chirp rates can communicate with the gateway at the same time. this is an important feature for a network gateway. The Sx1301 contains and executes the LoRa PHY layer on the gateway, which is proprietary technology and also explains why the documentation for this device is not available. The base-band concentrator further connects to two transceivers in the front end. These transceivers are the Sx125x series and have a system defined interface with the concentrator. These transceivers have a choice of full or half duplex operation with analog filtering at both reception and transmission. The combination of these two sets of devices is mandatory for LoRa gateway operation.

The network interface could comprise of a single network protocol module or as in this case, multiple modules for choice and also to provide redundancy. The Gateway utilizes the Ethernet protocol for primary communication. The Ethernet is controlled via the LAN 9514 chipset which is the industry standard and is produced by microchip.

The LAN controller also has a USB upstream port and 4 USB downstream ports which can be used for adding peripherals and features. Currently the gateway contains one USB-A type port as a device port to connect storage media etc. and a micro USB host port which is used as a device port to connect to host for programming and debugging. Apart from the LAN or Ethernet interface a modern application also require network over cellular interface as cellular technology is the current trend for wireless communication. For this reason, the gateway also deploys a 3G/4G GPRS module for network over cellular access. This module serves as a remote wireless network connection option and also as a fail-safe for the Ethernet module depending on the application requirements.

The cellular IoT which is also sometimes known as machine type communication [3] has specific requirements and is characterized by connection to large number of devices with smaller payload structure [40]. This presents a challenge for the software deployment and changes, the hardware deployment is comparatively simple, As simple as a serial connection between the micro controller unit and the GPRS unit. The GPRS module also supports a SPI bus connection but a UART format was chosen as the AT91 MCU supports 2 hardware defined SPI buses, one of which is dedicated to the LoRa Radio modem and the other to the flash memory. The option was to populate the latter with the SPI flash and the GPRS module but was later deemed not required as the GPRS did not require SPI speeds for communicating effectively.

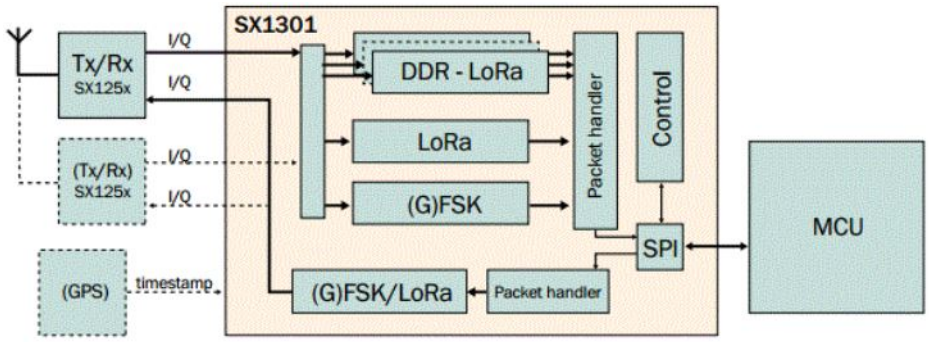

Figure 4.2: Reference connection for the Semtech LoRa module

4.1.1.4 Power Interface

Creating the power interface is perhaps the most important and challenging part of the hardware design as there are many considerations to be made. The challenge was that the MCU core, RAM, NAND Flash and the LoRa Radio all have different power considerations. The table 4.1 shows the various power considerations for all modules on

the gateway, and this means that the power plane of the board is a bit more complicated than perceived. Also the various modules require various start-up currents and while the MCU core, RAM, NAND Flash and LAN chip require a start-up current of less than 1 Amp, the GPRS module and LoRa Radio require a stable start-up current of 1.2 Amps and 1.6 Amps respectively which means that a USB power supply cannot be used to power the gateway as it cannot supply 1.6 Amps of stable current for the LoRa module start-up. Figure 4.3 shows the impact caused by start-up current caused by the different modules.

Hence, the power connector is a 12V barrel jack input and a Non-Isolated DC to DC converter is used to convert the 12V input to 3.33V input which is then further fed into the various power regulators to get 1.8V and 1V supplies. 12V input is also fed to a LM317 which then converts it to 5V for the USB device port. The 3.3V is chosen as a more efficient switching regulator to reduce linear losses caused by the voltage difference between 5V and 3.3V.

Sno.	Part	Op. voltage (V)	Op. Current(Amp)
1	MCU core	1.0	0.35
2	DDR2 RAM	1.8	0.295
3	NAND FLASH	3.3	0.03
4	LAN9514	3.3	0.288
5	GPRS	3.3	1.2
6	LoRa Module	3.3	1.2
7	USB Dev. Port	5	0.5

Table 4.1: Various Power considerations for hardware

4.1.1.5 Creating design and challenges

Creating the schematic for the hardware was a time consuming process as all the modules had to be created from scratch. KiCad [16] which is a free and open source Electronics design tool was used for the design and not all modules are readily available in the libraries. After generating all the modules by hand and creating relevant libraries, which was a significant challenge, the schematic was created, proofed and deemed correct by the in house hardware developer. The PCB design and optimization was a challenge as well as it involved the proper placement of all the components and connectors to suit an industry appropriate hardware design. Creating traces for high speed modules like the DDR2 RAM is a difficult task and requires length matching of all traces on the RAM bus. The SDA10 pin required for the DDR bus is placed further away from all other bus points on the MCU. This meant that the trace for the

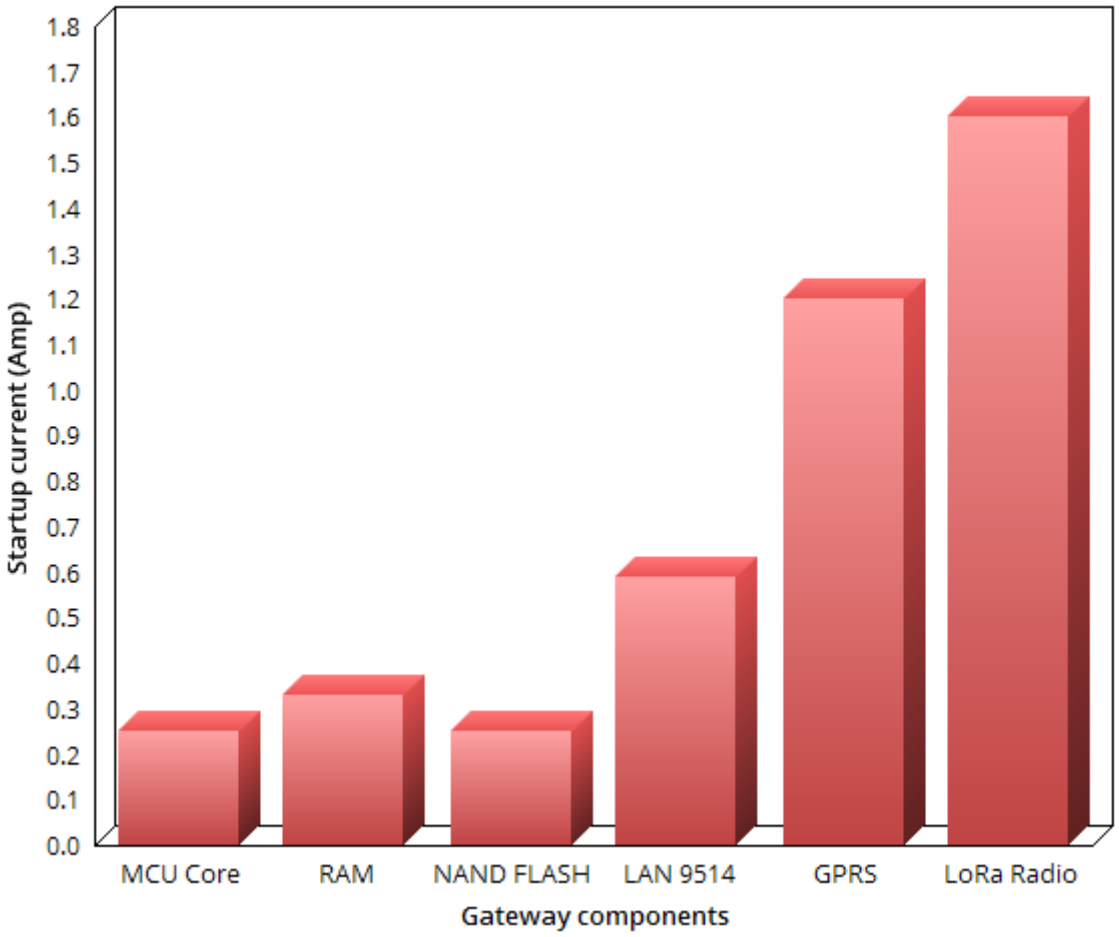

Figure 4.3: Start-up impact based on current consumed during start-up

SDA10 pin was longer and larger than all the other bus pins, which is not ideal as any mismatch in the high speed bus traces would mean that the data would arrive at different time points causing conflicts and RAM errors. Hence all the RAM traces were tuned to the longest SDA10 trace around, 25mm which is long but still less than 10% of the operating wavelength of the RAM signal hence we can still ignore any transmission line effects.

Many challenges and delays arrived during manufacturing of the board as the PCB itself was very complicated and large in size, also the use of two BGA component made the manufacturing difficult and expensive, driving up the Range to cost ratio. The board was conceived to be 6 layers. The many layers of the board are, in the order from top to bottom, Top parts, GND, Signal, Signal, Power and Bottom parts. The ball diameter for the MCU package, which is 217 BGA, is 0.4mm with a adjacent ball distance of 0.8mm, which is quite standard and enough space to fit the vias in between the ball

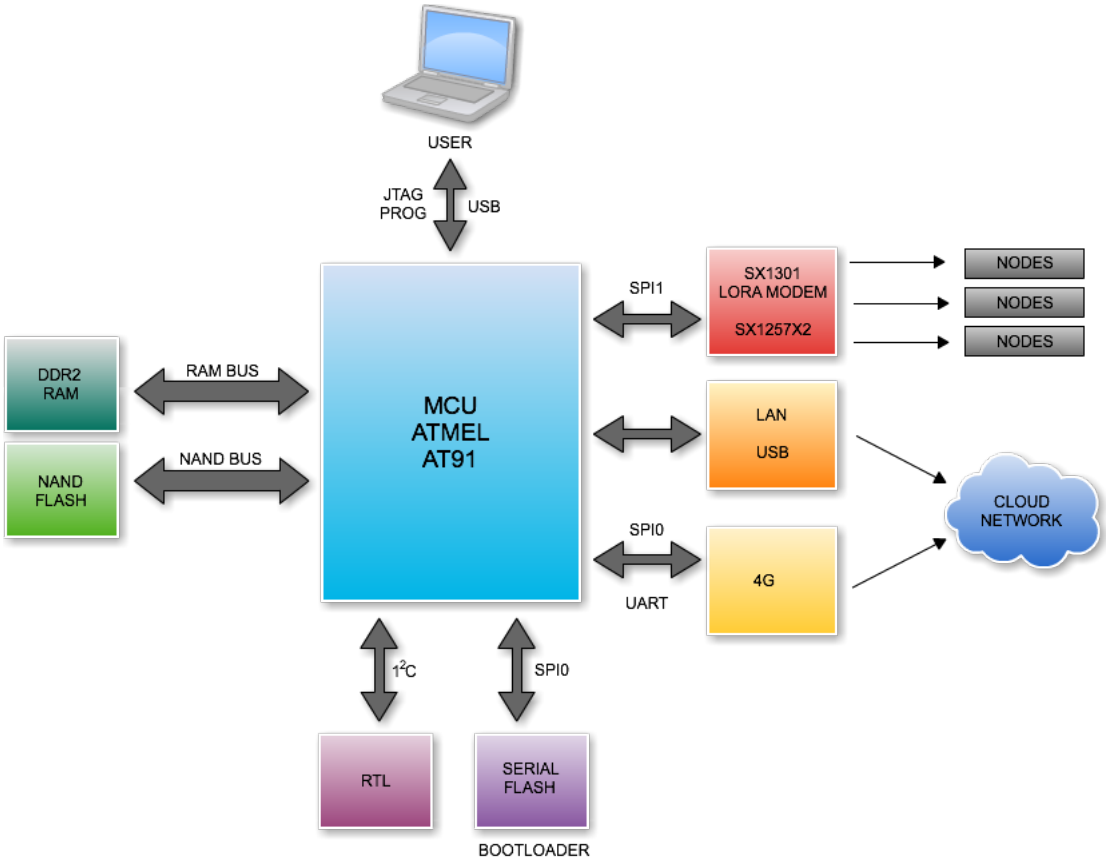

Figure 4.4: Reference hardware block diagram

land patterns but this via diameter also includes the drill diameter which is twice the annular ring diameter around the hole for the via. Finding a manufacturer which can do all these for a decent price was almost impossible and there were many manufacturers that couldn't achieve a possible scenario. Ultimately the costs had to be increased for the prototype board to be able to manufacture the board itself for testing purposes, the Range to cost ratio is going to be evaluated for at least 100 pieces and the cost has to be significantly higher for manufacturing than previously perceived.

4.2 Software

The software for the LoRa gateway is a two faceted model. First a Linux kernel is compiled to operate on the embedded hardware environment and further a LoRa Gateway software is compiled for execution of the LoRa communication effectively. As much as there is freedom in developing and selectively compiling the Linux kernel, many restrictions are imposed on the development of the LoRa software for the gateway. The software is bound to be standard as per the LoRa Alliance and any major change in the LoRa stack warrants a certification from the Alliance. As FactoryLab is a member of the LoRa Alliance, a modified stack can be developed for future versions but the standard LoRa

stack for the Sx1301 and Sx1257 which is defined by Semtech is used for the gateway's LoRa module.

4.2.1 Linux kernel

Linux based operating embedded systems are the most sought after and effective way of operating on a IoT system due to its transparency and ease of manipulation. Linux based embedded operating systems are very appealing for a host of reasons and are sometimes preferred over RTOS as well. One of the most important feature regarding the choice of Linux as the operating system, apart from the ease of use, flexibility and community support, is the cost of Linux capable devices is significantly low as compared to others. This also springs from the fact that the system is extremely popular. This factor weighs in heavily in the development of the gateway as it improves the range to cost ratio which is the core developmental question of the thesis project.

4.2.1.1 Boot Process

The boot process begins with an internal NVM bootloader which is placed on the Read Only Memory that tries to find a valid boot program on the various memories installed. These are SPI flash and the NAND flash except the USB device. If there is a valid boot routine present on either of the memories then the bootloader doesn't look any further. While initializing the gateway software, there is no bootloader available for the ROM boot to encounter, hence an external boot utility called SAM-BA is used, which is an in-system programmer created by Atmel [1] which listens on the serial USB port for a supported device. As the NVM bootloader cannot directly compile and boot into the Linux kernel, a secondary bootloader or bootstrap is required to initialize the clocks and the DRAM before loading the Linux kernel. The bootstrap used here is the AT91 bootstrap provided by Atmel. This bootstrap sets up the DRAM and sets the next bootable file(UBoot) at a memory address in the DRAM. The bootstrap is placed at 0x00 in the SPI data-flash.

UBoot defines and initializes all devices and peripherals connected to the target processor. UBoot also performs the following functions [25].

- ***Initializing the CPU*** - During this stage the UBoot bootloader maps the memory and define the register values. It then acquires the boot code at 0x0000 in the flash memory. The code generated and stored is in assembly file called 'start.s'.
- ***Initializing the on board peripherals*** - UBoot then finds and initializes the PLL, DRAM and also initializes the memory devices by dividing the sectors and determining what large is each sector. UBoot then also starts up and configures the Serial UART.
- ***Initializing the Devices*** - Finally all other devices are initialized and configured. These include the USB ports, Ethernet etc.

The AT91 Bootstrap then configures and loads the Uboot bootloader which is available in the DRAM. The UBoot initializes the USB port, creates the partition tables and then further gives the user access to the Linux kernel available on the USB host.

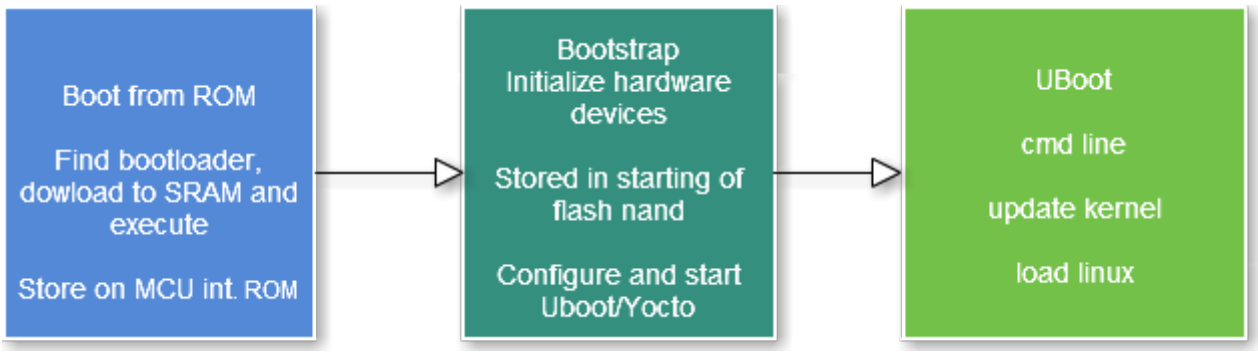

Figure 4.5: Flow of the boot process

4.2.1.2 Linux Root File System

Unlike regular personal computers, installing a Linux kernel on a custom embedded computing platform isn't a straightforward job. There are multiple dependencies to take care of, for example configuring file system support for all devices on the embedded platform and controlling the build of all rootfile images. This process can be achieved manually but for a gateway system with a lot of peripherals and devices, which means a lot of considerations, this task can be extremely difficult and time consuming. For this reason, some automation is preferred for generating the RootFS. This can be done by utilizing an automation tool called BusyBox [8]. BusyBox is a tool that combines UNIX functions like ls, cat et al. in an executable file. Busy box was created by keeping memory restrictions in mind and is decent for embedded processing platforms.
Although BusyBox can be used to great advantage, there are a few limitations. BusyBox combines UNIX functions into an executable but does not populate the Linux RootFS. This means that all the Linux functions that provide the ease and transparency will be installed on an executable file but the environment required to execute this will still need to be created. to avoid the mishaps of human error, BuildRoot is used to create the complete file system. Buildroot [7] is a set of makefiles which can be utilized for building the Linux kernel and the RootFS. Buildroot was created keeping embedded processors in mind and hence it has support packages for many, if not most Linux capable ARM based controllers.

4.2.2 LoRa gateway software

After compiling and testing the Linux kernel for the target environment, the next step is to initialize and compile the LoRa gateway software. The LoRa devices send LoRa packets to the gateway and the gateway forwards these packets to the LoRa network server via the network protocol in place. The network devices could be Ethernet or

GPRS based on the requirements. The software module is divided into many modules and this section covers the various modules used, their function and implementation in the gateway software.
As seen in Section 2.1, the physical layer of the software module functions on the CSS modulation technique. As seen in [33], the LoRaWAN standard utilizes CSS to transmit LoRa packets in the 868 MHz band.As mentioned earlier, in the beginning of Section 4.2, there is a limitation on the software that can be used for the gateway chipset and the gateway software is provided by the Semtech developers for use with the Sx1301 concentrator along with the 1257 chipset. Apart from any major modifications, the software contains the same core for all gateways using the LoRaWAN protocol. In this scenario, this section explains the functions that the software executes at the core level.

4.2.2.1 LoRa MAC Message frame

As seen in the 2.3, the LoRa messages carry a Preamble, a physical header, a CRC and most importantly the PHYPayload which contains the LoRa message. Further the PHYPayload contains the MHDR, the MAC Payload and the Message Integrity Code which is detailed in the section below. The MAC header or MHDR specifies the type of message that is being received/transmitted and the type of specification of the frame format itself. The Message type can be of 6 different types as seen in Table 4.6. The join request and join accept are also detailed in the section 4.2.2.2. In the MAC Payload itself there are many fields as seen in figure 2.3 but the Frame Header or MHDR is a significant one as it contains teh device address and the Frame octet control which includes ADR. ADR or adaptive data rate is a very important feature of the of a LoRa protocol. LoRa gives the end devices the ability to individually to change and utilize any possible data rates available to them. This aspect is then utilized by LoRaWAN to optimize the data rate of a stationary end node to suit the requirements of the network and this process is known as Adaptive Data Rate. If the ADR flag is enabled, the command to change and control the data rate of the end device is handed over to the network to maintain good signal quality and integrity. This also helps in achieving higher battery life of end devices as higher spreading factors may not be required in certain scenarios and manually changing the spreading factor every time is not ideal.

4.2.2.2 Join procedure

As defined in the LoRaWAN specifications [4]. there are two types of joining procedures that are used by the end nodes to connect to the LoRa Gateway, these are called Activation By Personalization(ABP) and Over The Air Activation(OTAA). In OTAA [21], the procedure requires a Device ID or DevEUI, an unique identifier for each device and is based on the EUI64 format, an application key or AppEUI for activation which as a unique application identifier also in EUI64. Apart from the two key there also exists an AppKey which is specific and unique to each end node and is encrypted by the 128AES encryption. This form of activation does not specify any device specific session keys and these session keys (NwkSKey and AppSKey) are derived from the AppKey to secure and verify the messages and all transmissions made at the network level. When a device tries to join a LoRa network, it sends an OTAA join request which contains the AppEUI, a

MType	Description
000	Join Request
001	Join Accept
010	Unconfirmed Data Up
011	Unconfirmed Data Down
100	Confirmed Data Up
101	Confirmed Data Down
110	RFU
111	Proprietary

Figure 4.6: MAC Message types in LoRaWAN format

Devnonce [30], which is a randomly generated value by stringing together RSSI points, which the network server traces to avoid any invalid join requests. The OTAA procedure is the more secure procedure of the two as no security keys are stored on the device and are derived from the over the air transmitted code. The OTAA procedure also facilitates changing networks which can help with roaming and mobile devices. The formula for calculating the Message Integrity Code(MIC) for a Join Request is shown in eq 4.1 and the frame for a join request is demonstrated in figure 4.7

$$cmac = aes128\_cmac(AppKey, MHDR|AppEUI|DevEUI|DevNonce) \tag{4.1}$$

$$MIC = cmac[0..3] \tag{4.2}$$

Join Request	Octets
AppEUI	8
DevEUI	8
DevNonce	2

Figure 4.7: Join Request frame and octets

After the join request is sent, The network or gateway has to allow the end node to joining by accepting the Join Request and issuing a Join Accept message.If the join request is declined then nothing is issued. The join accept message contains 3 fields in which one is the AppNonce which is 6 octets and is unique identifier generated by the LoRa server and is utilized by the end node to extract the Network and Application session keys (NwkSKey and AppSKey). The formula for generating the two keys is seen in eq 4.3 and the frame for a join accept is in figure 4.8. Further the message integrity

check can be calculated as seen in eq. 4.5. Unlike the Join Request message the Join Accept message utilizes the Application key for AES encryption as seen in eq. 5.1

$$NwkSKey = aes128\_encrypt(AppKey, AppNonce|DevNonce|pad_{16}) \tag{4.3}$$
$$AppSKey = aes128\_encrypt(AppKey, DevNonce|AppNonce|pad_{16}) \tag{4.4}$$

$$cmac = aes128\_cmac(AppKey, MHDR|AppNonce|DevAddr|RFU) \tag{4.5}$$
$$MIC = cmac[0..3] \tag{4.6}$$

$$aes128\_decrypt(AppKey, AppNonce|DevAddr|RFU|MIC) \tag{4.7}$$

Join Accept	Octets
DevAddr	4
RFU	2
AppNonce	6

Figure 4.8: Join Accept frame and octets

Activation By Personalization or ABP, is a process that allows the end device to connect with the gateway without the aforementioned join and accept process.The three identifiers required for joining the LoRa network i.e, the Device address, the Network Session Key and the Application Session Key, are stored inside the memory of the node. This procedure is insecure as any compromise in the security of the device can cause all the messages to be compromised. This kind of network joining method is used for low security applications like home networks or testing where security is not he highest concern and no roaming between networks is required. In this method the end device first receives a message beacon from the gateway/network to get network information like channel etc.

4.2.3 Conclusion

This chapter highlights the specifics that were involved in the final hardware/software co-design of the gateway. Section 4.2 thrives in the details of the software for the design of the gateway module, starting with Section 4.2.1.2 which talks about the Linux kernel that is used for the gateway and the boot process that is involved in the start-up of the MCU. Uboot is the bootloader that is used for initializing the kernel and then

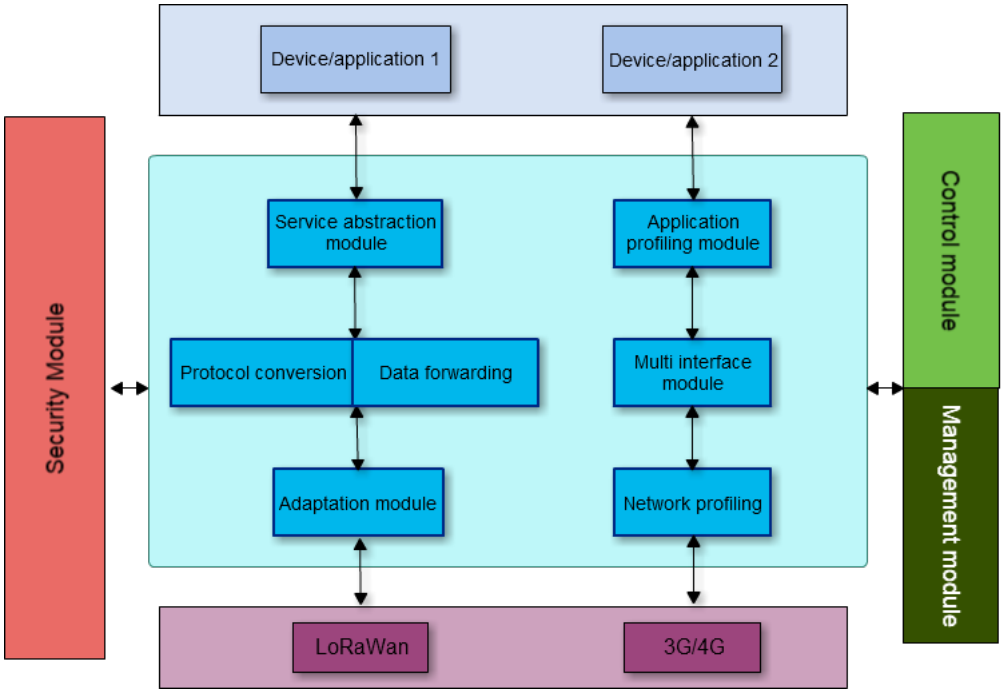

Figure 4.9: Reference software block diagram

initializing the peripherals on the board. The boot flow can be seen in figure 4.5. This is followed by a detailed explanation of the Linux root file system setup and how Buildroot is used to create the kernel root system to avoid any human errors. Following this the 4.2.2 details the LoRa software used for the transmission of LoRa packets from the end nodes and forwarding the data to the network server. The join procedures used in the process are explained in detail and figure 4.9 shows the basic software flow.

Section 4.1 thrives in the details of the hardware design choices and implementation, Section 4.1.1 enlists all the building blocks of the computing system and further sections explain in detail, the choice behind the computing system, memory and power systems and their design challenges. Figure 4.4 shows the model hardware block diagram. The chapter is finally concluded with the design details and the challenges faced.

5 Experiments and Results

This section contains the experiments and tests that were run on the various LoRa Gateways and the tests run on the software created for this project. Unfortunately the hardware for the gateway could not be manufactured in time for testing and accumulating results, as on the date of submission of this report. So to provide an estimate and approximation on the capabilities of the actual hardware, a Raspberry Pi [32] first edition was used along side the LoRa Radio module. The Raspberry Pi contains a 700 MHz ARM 11 processor, 190 Mb of dedicated CPU RAM along with an on board 4GB memory card and an Ethernet and USB port [31]. Although the Clock speed on the Raspberry Pi is higher, the clocking cap can be defined in the software to be 300MHz to emulate the gateway hardware, the SDRAM available on the Gateway is higher that 512 MB which is higher than the Raspberry Pi , which would means that an effect(if any) caused by the SDRAM on the performance would be in the favour of the gateway hardware as there will be possibly no, or lesser delay in caching instructions and executing the LoRa and Network Stack [11].

The LoRa module on he other hand, as discussed in previous chapters, is quite standard for all gateways and uses the SX1301 base-band concentrator [37] coupled with two 868 MHz transceivers, which are the SX1257 [36]. These modules are standard on all industry standard LoRa Gateways certified by the LoRa Alliance and can be used coupled with the LoRa Stack available for the SX1257 transceivers. These modules are found on the MTAC LoRa card created by MultiTech [28]. These are used for testing along with the Raspberry Pi and the interface was created using a USB to SPI converter and tested over the Raspberry Pi's USB port. This protocol conversion delay also works in the favour of the FactoryLab LoRa Gateway as the gateway contains a direct hardware interface with the LoRa concentrator over dedicated SPI.

This chapter starts with the results of hardware design and the with the range measurement over standard conditions and changes in various parameters so create a stark contrast in various gateways and their operational range.

The cost aspect for the FactoryLab gateway can be derived as an approximation by calculating the manufacture prices summed, for the price of 100 pieces including the hardware manufacturing costs and the casing of the gateway. The estimated cost range for 100 pieces is in the range of 250 - 350 euros which would be drastically reduced when put into mass production as done by the other companies. For the sake of the worst case scenario, all the Range to cost ratios will be determined with the higher bound of the cost, which is ***350 euros*** per gateway unit.

Keywords: Kerlink IoT Station, MultiTech Conduit, Raspberry Pi, Multitech MTAC, spreading factor, antenna height, Range to Cost ratio

5.1 Hardware prototype

The hardware design was a tedious and time consuming task. As mentioned in Section 4.1, KiCad was used to create design of the hardware ranging from the schematic for the design to the PCB design and manufacturing scheme. Figure 5.1 shows the side by side view of the top of the manufactured board and the 3D design. Figure 5.2 shows a dynamic side side view and figure 5.3 shows the bottom of the board. The board consists of 6 layers which are enumerated as Top, GND, Signal, Signal, Power and Bottom.

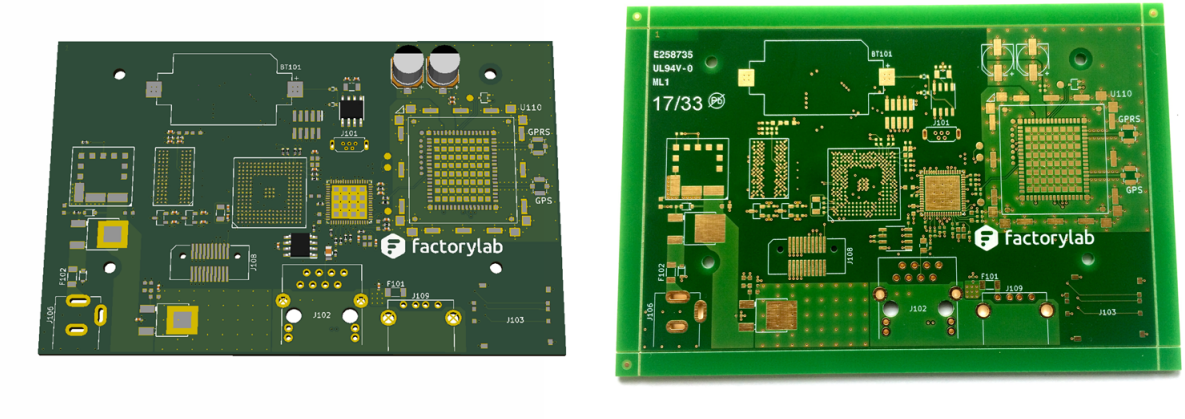

Figure 5.1: 3D and actual top view of gateway hardware

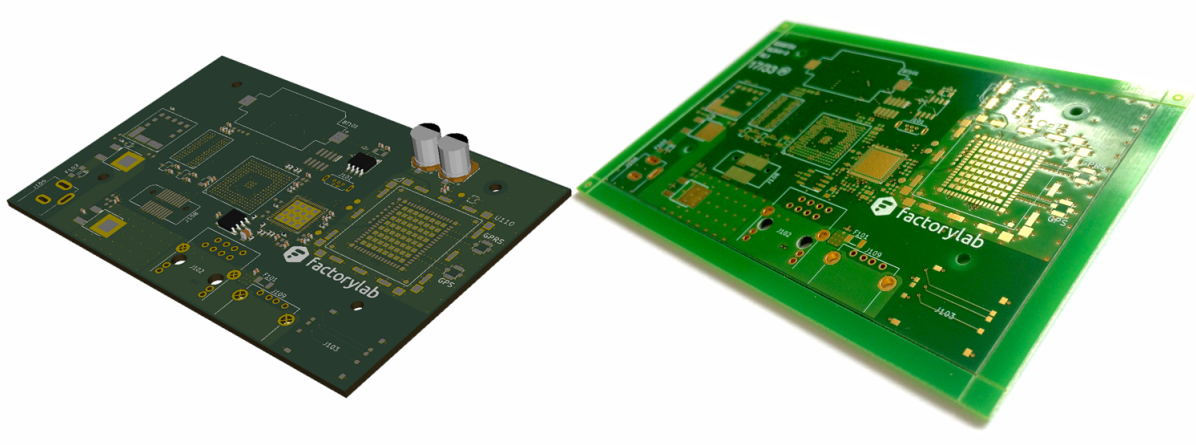

Figure 5.2: 3D and actual side view of gateway hardware

At the time of finalization of the report, the final prototype of the gateway was assembled and can be seen in figures 5.4, 5.5, 5.6. This hardware could not be tested

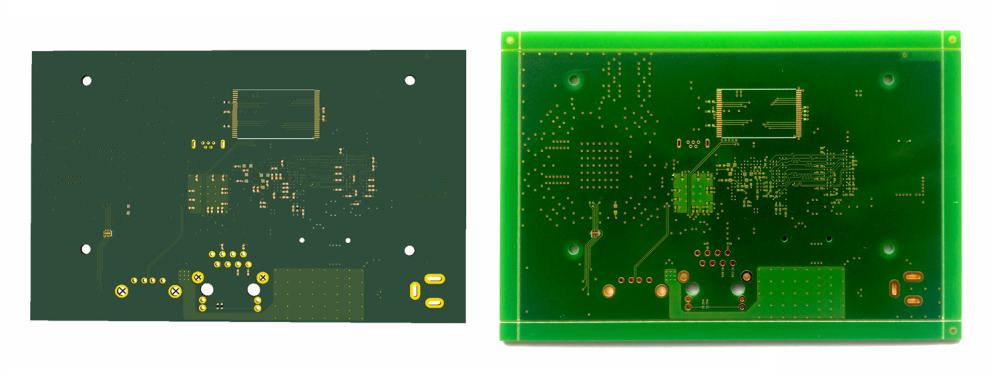

Figure 5.3: 3D and actual bottom view of gateway hardware

per the finalization of this report because it was missing some key power components and the GPRS connectors for cellular connections. The LoRa module to be used for testing is the MTAC LoRa MCard which is interfaced via US for testing purposes and future optimization. After the finalization of the SBC system the LoRa module will be manufactured for proprietary use. This will not hinder the initial testing, because as mentioned earlier, the LoRa module for most gateways is the same and although there will be developmental changes around the LoRa Radios and base-band concentrator, the core functionality remains the same with minor differences due to hardware design.

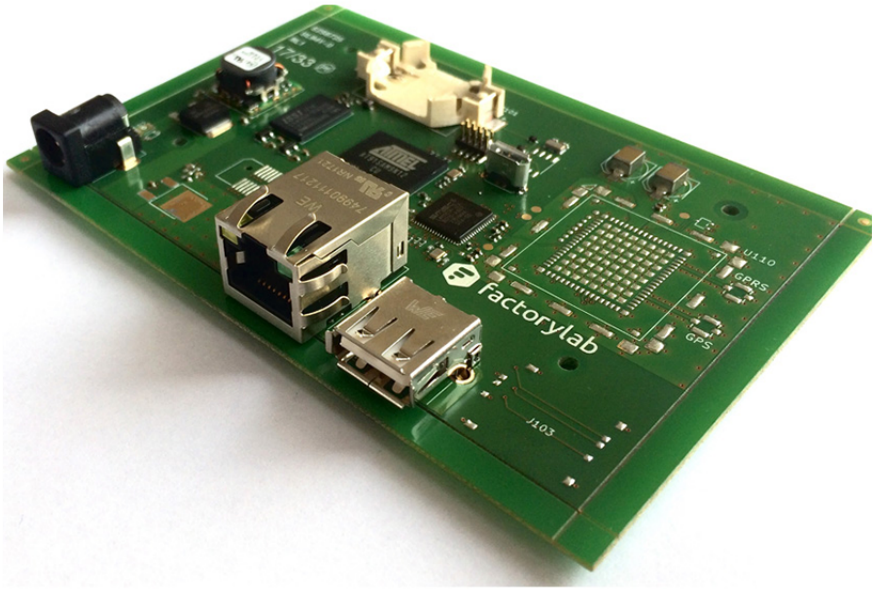

Figure 5.4: Side view of Completed and assembled gateway hardware

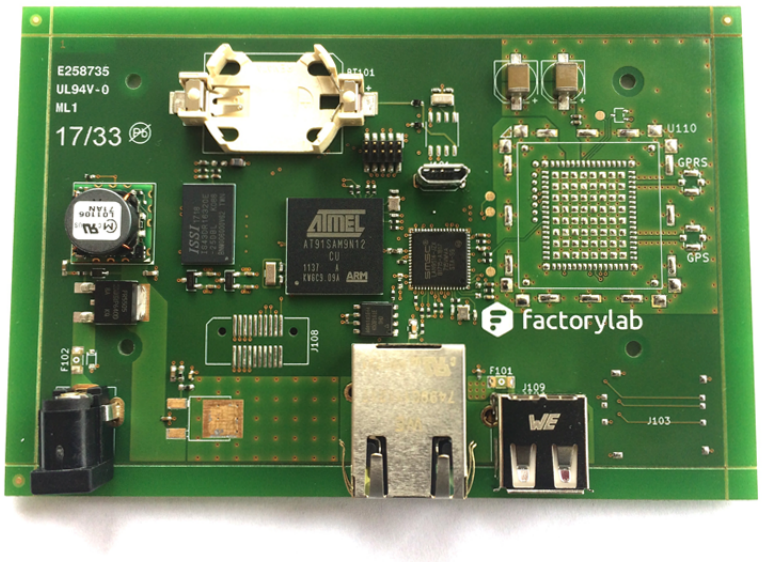

Figure 5.5: Top view of Completed and assembled gateway hardware

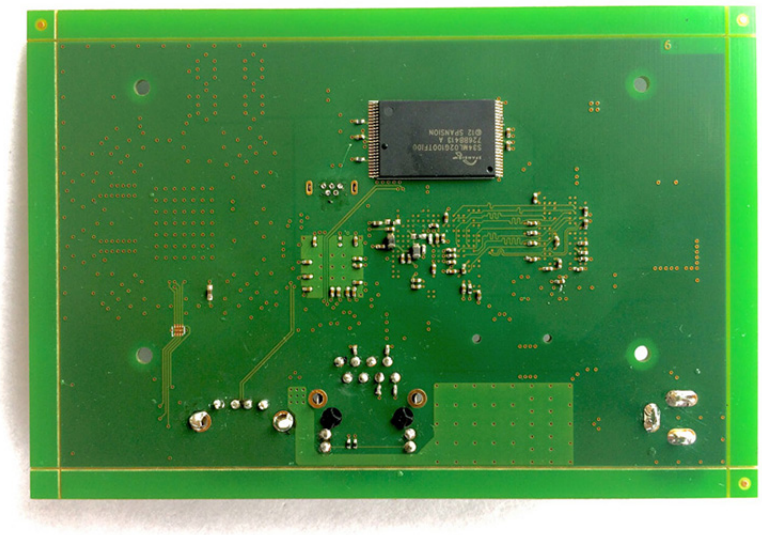

Figure 5.6: Bottom view of Completed and assembled gateway hardware

5.2 Range to Cost Ratio for Kerlink IoT Station

This section covers the results or the range analysis for various gateways. As the Kerlink IoT station was occupied during the testing phase, the range tests done previously for the gateway were used for comparison. The Kerlink gateway was mounted on a platform which was 25 meters higher than relative ground level and the spreading factor was set at 12 for maximum range. The following figure 5.7 shows the various points at which the range was tested and the ranges that was achieved. The maximum range tested with this setup was *6.5 Kilometers or 6500 meters* which is lower than advertised but still quite high considering that many factors affect the range like surrounding environment, density of concrete structure population etc. The range to cost ratio is calculated per 5.1 and is measured in meters/euro. The R/C ratio for the Kerlink IoT station is found in 5.2 to be 4.33 euros.

$$R/C = Maximum\ Range\ Achieved/Commercial\ Selling\ Price \tag{5.1}$$

$$R/C_{Kerlink} = 6500/1500 = 4.33\ Meters/Euro \tag{5.2}$$

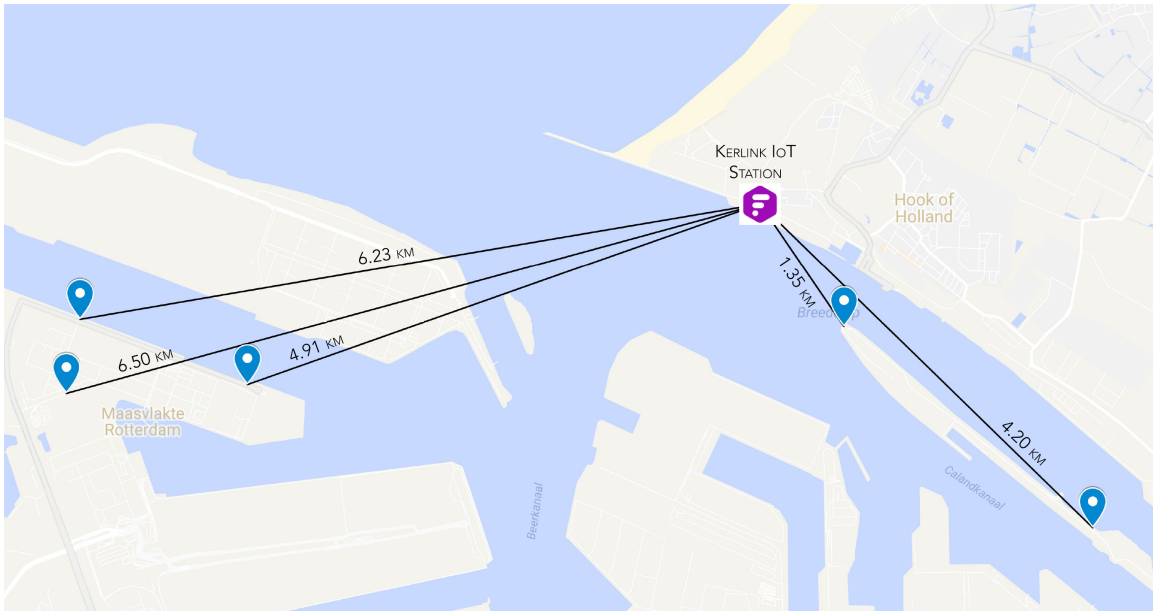

Figure 5.7: Mapped ranges for Kerlink IoT station at Antenna 25m and SF=12

5.3 Medium density urban environment: Antenna height at 2m

5.3.1 Varying Spreading Factor

This test was conducted at standard conditions with the MultiTech Conduit [27] and the FactoryLab test setup as described in he beginning of the chapter. The gateways are placed near a window of a building in Delft at approximately 2m high from relative ground level. The connections were established by Over The Air activation of the node. The following behaviour was observed for the various gateways. The nodes is programmed as Class A. Table 5.1 shows the obtained results for the gateway placed at 2m from ground level. The maximum range at each Spreading factor was obtained by observing the incoming packet flow. Figure 5.8 shows the mapped ranges for the Multitech Gateway and the FactoryLab test setup at varying spreading factors and figure 5.9 shows a graphical representation of the same. Figure **??**, shows a snippet from the terminal taken from the test of the Multitech Conduit. The Snippet shows the mosquito client started and started receiving packets and then was stopped.

S.No.	SF	Distance MTc	Distance FL	Rssi MTc	Rssi FL
1	7	340	270	-97	-95
2	8	480	350	-102	-99
3	9	610	470	-115	-112
4	10	720	550	-120	-117
5	11	800	640	-131	-125
6	12	920	760	-139	-137

Table 5.1: Range measurements for gateways at 2m and varying spreading factors.

5.3.2 RSSI vs Distance with constant SF=12

The gateways are placed near a window of a building in Delft at approximately 2m high from relative ground level. The connections were established by Over The Air activation of the node. The following behaviour was observed for the various gateways. The nodes is programmed as Class A. Table 5.1 shows the obtained results. A graphical representation of change in RSSI with increase in distance is shown in figure 5.10

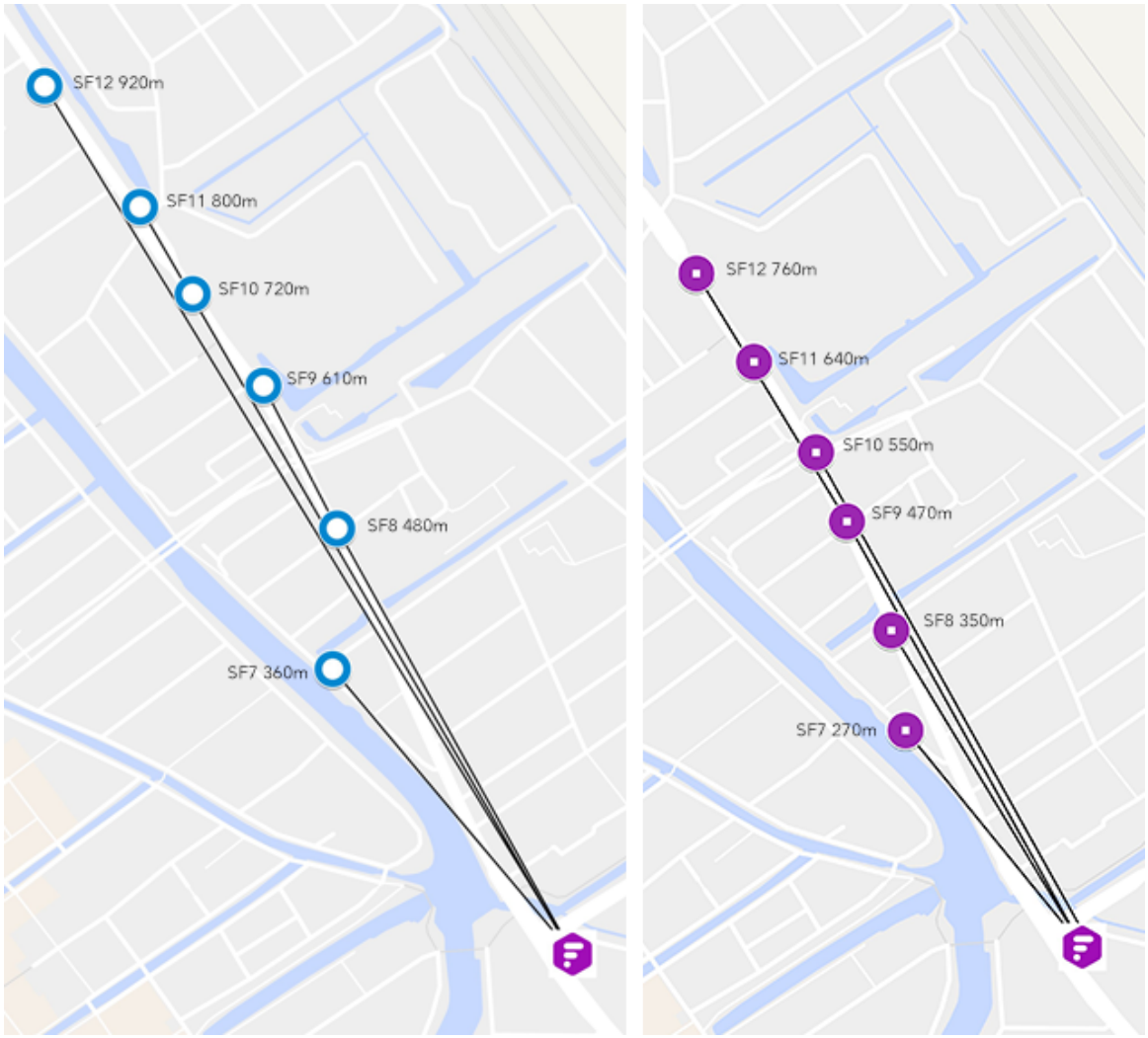

Figure 5.8: Mapped ranges for Multitech Conduit(left) and FactoryLab test setup(right) at Antenna 2m and varying SF

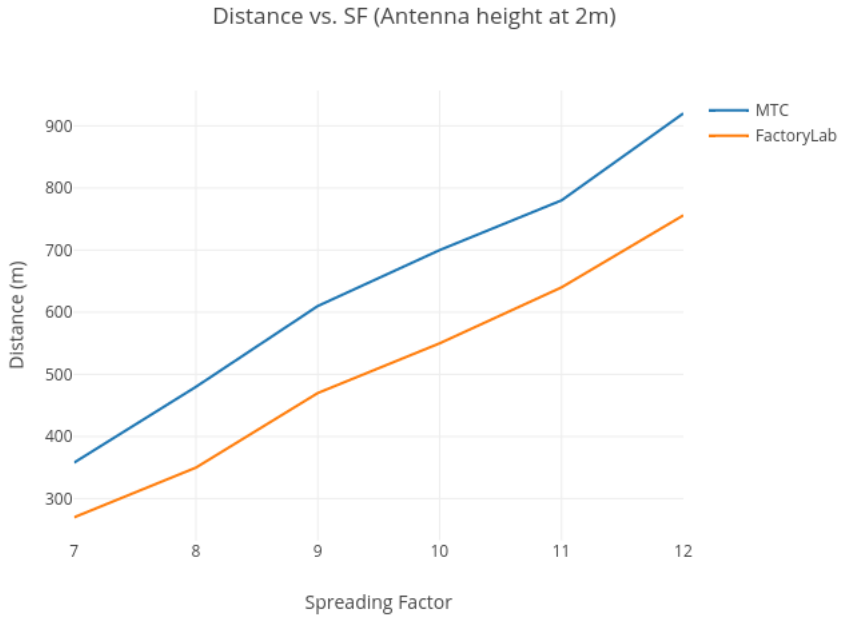

Figure 5.9: Comparison of range for Multitech and FactoryLab setup at varying SF

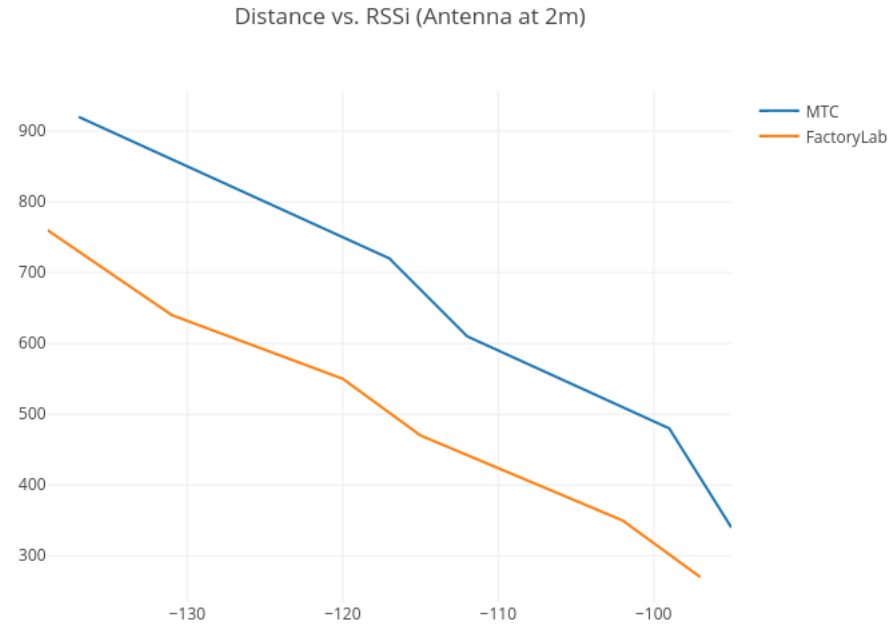

Figure 5.10: Comparison of change in RSSI vs Distance for Multitech and FactoryLab setup at SF=12

```
root@mtcdt:~# mosquitto_sub -t lora/+/+ -v
lora/01-02-03-01-02-03-01-02/packet_recv {"chan":4,"codr":"4/5","data":"QAEAAAYABAABAIjwWlZIUi6O","datr":"SF12BW125",
"freq":867.29999999999995,"lsnr":7,"modu":"LORA","rfch":1,"rssi":-35,"size":18,"stat":1,"time":"2017-07-19T00:39:57.8
31144Z","tmst":2451287620}
lora/01-02-03-01-02-03-01-02/up {"chan":4,"cls":0,"codr":"4/5","data":"U09EQVE=","datr":"SF12BW125","freq":"867.3","l
snr":"7","mhdr":"4001000006000400","modu":"LORA","opts":"","port":1,"rfch":1,"rssi":-35,"seqn":4,"size":8,"timestamp"
:"2017-07-19T00:39:57.831144Z","tmst":2451287620}
lora/01-02-03-01-02-03-01-02/packet_recv {"chan":7,"codr":"4/5","data":"QAEAAAYABgABrV5z5dXtXp9Y","datr":"SF12BW125",
"freq":867.89999999999998,"lsnr":8.1999999999999993,"modu":"LORA","rfch":1,"rssi":-37,"size":18,"stat":1,"time":"2017
-07-19T00:40:11.010372Z","tmst":2464463900}
lora/01-02-03-01-02-03-01-02/up {"chan":7,"cls":0,"codr":"4/5","data":"U09EQVE=","datr":"SF12BW125","freq":"867.9","l
snr":"8.2","mhdr":"4001000006000600","modu":"LORA","opts":"","port":1,"rfch":1,"rssi":-37,"seqn":6,"size":8,"timestam
p":"2017-07-19T00:40:11.010372Z","tmst":2464463900}
lora/01-02-03-01-02-03-01-02/packet_recv {"chan":1,"codr":"4/5","data":"AAMDAwMDAwMDAgEDAgEDAgHDcgpmOgA=","datr":"SF7
BW125","freq":868.29999999999995,"lsnr":10.800000000000001,"modu":"LORA","rfch":0,"rssi":-42,"size":23,"stat":1,"time
":"2017-07-19T00:41:28.147719Z","tmst":2541627427}
lora/01-02-03-01-02-03-01-02/mac_command {"data":"02","id":3}
lora/01-02-03-01-02-03-01-02/joined (null)
lora/01-02-03-01-02-03-01-02/packet_sent {"codr":"4/5","data":"IEUir7AdzEGVFeHsWQN4WrSeIWVUZKQ8so2rCA0Wljfe","datr":"
SF7BW125","freq":868.29999999999995,"ipol":true,"modu":"LORA","ncrc":false,"powe":11,"rfch":0,"size":33,"tmst":254662
7427}
lora/01-02-03-01-02-03-01-02/packet_recv {"chan":3,"codr":"4/5","data":"QAEAAAYAAAABZYJKWdnxBEeR","datr":"SF12BW125",
"freq":867.10000000000002,"lsnr":6.7999999999999998,"modu":"LORA","rfch":1,"rssi":-41,"size":18,"stat":1,"time":"2017
-07-19T00:41:34.722645Z","tmst":2548177012}
lora/01-02-03-01-02-03-01-02/up {"chan":3,"cls":0,"codr":"4/5","data":"U09EQVE=","datr":"SF12BW125","freq":"867.1","l
snr":"6.8","mhdr":"4001000006000000","modu":"LORA","opts":"","port":1,"rfch":1,"rssi":-41,"seqn":0,"size":8,"timestam
p":"2017-07-19T00:41:34.722645Z","tmst":2548177012}
^Z
[8]+  Stopped(SIGTSTP)        mosquitto_sub -t lora/+/+ -v
root@mtcdt:~#
```


Figure 5.11: A terminal Snippet from the multitech output shows a LoRa packet

5.4 Light density urban environment: Antenna height at 10m

5.4.1 Varying Spreading Factor

This test was conducted at standard conditions with the MultiTech Conduit [27] and the FactoryLab test setup as described in he beginning of the chapter. The gateways are placed in the FactoryLab office in Zwijndrecht at approximately 10m high from relative ground level. The connections were established by Over The Air activation of the node. The following behaviour was observed for the various gateways. The nodes is programmed as Class A. Table 5.1 shows the obtained results for the gateway placed at 2m from ground level. The maximum range at each Spreading factor was obtained by observing the incoming packet flow. Figure 5.12 shows the mapped ranges for the Multitech Gateway at varying spreading factors while figure 5.13 shows the same for the FactoryLab gateway setup. Figure 5.14 shows a graphical comparison between the two at varying spreading factors. MTc and FL represent Multitech and FactoryLab respectively.

S.No.	SF	Distance MTc	Distance FL	RSSI MTc	RSSI FL
1	7	521	516	-95	-91
2	8	750	708	-102	-99
3	9	1140	1040	-110	-105
4	10	1320	1210	-121	-118
5	11	1560	1360	-130	-126
6	12	1930	1700	-138	-133

Table 5.2: Range measurements for gateways at 10m and varying spreading factors.

5.4.2 RSSI vs Distance with constant SF=12

The gateways are placed in the FactoryLab office in Zwijndrecht at approximately 10m high from relative ground level. The connections were established by Over The Air activation of the node. The following behaviour was observed for the various gateways. The nodes is programmed as Class A. Table **??** shows the obtained results. A graphical representation of change in RSSI with increase in distance is shown in figure 5.10

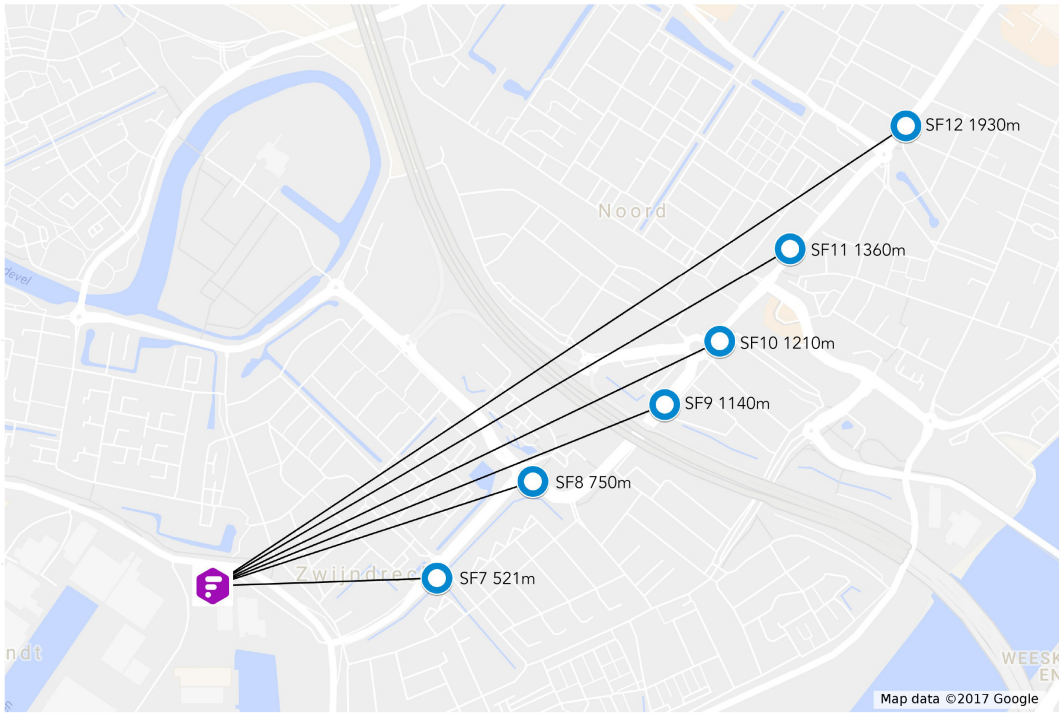

Figure 5.12: Mapped ranges for Multitech Conduit at Antenna 10m and varying SF

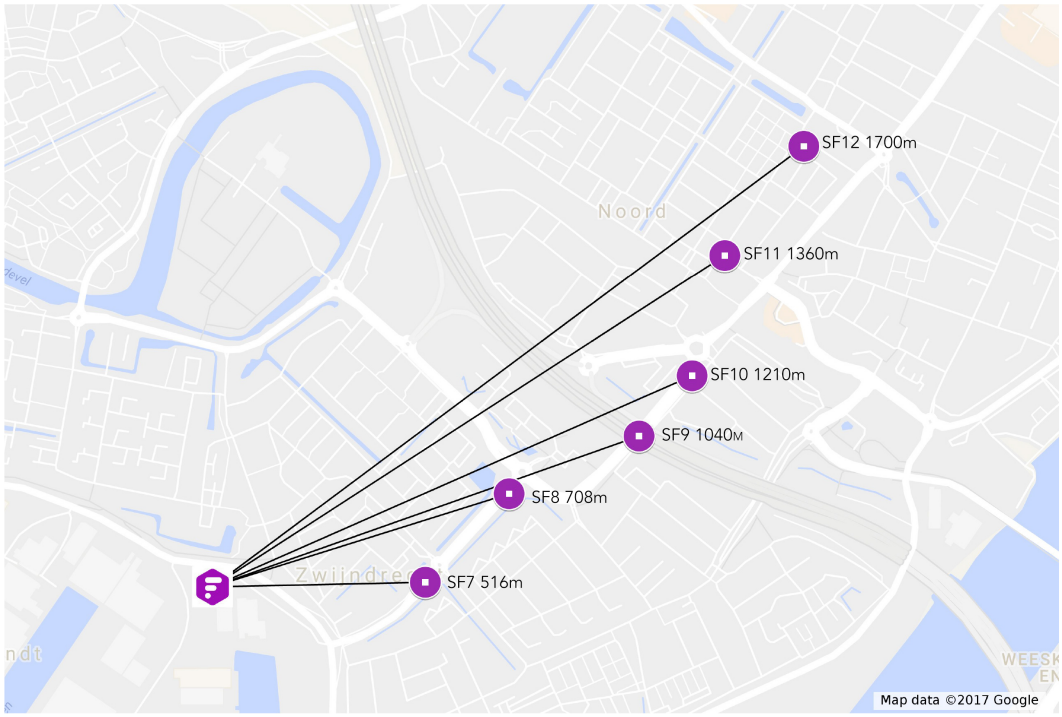

Figure 5.13: Mapped ranges for FactoryLab setup at Antenna 10m and varying SF

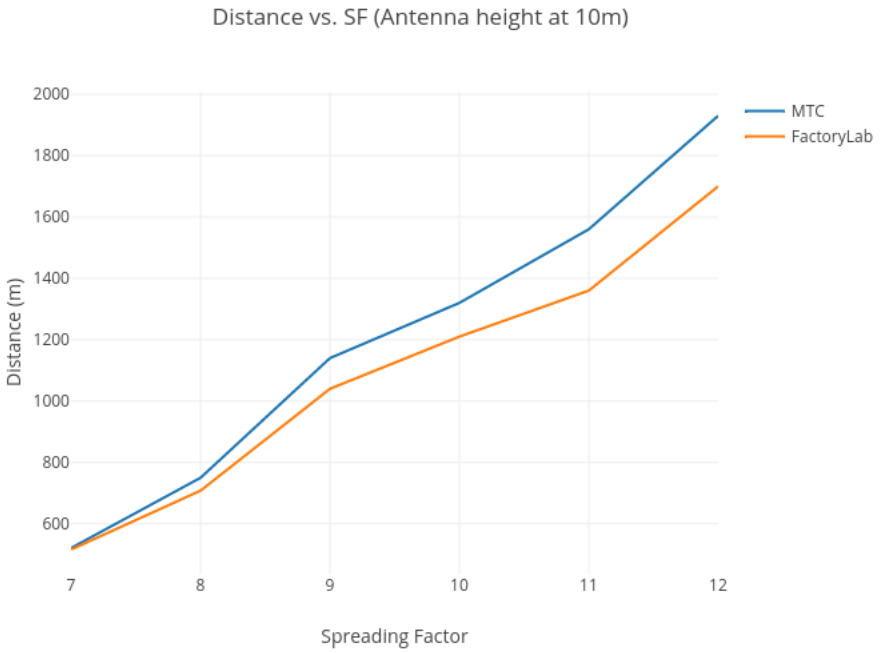

Figure 5.14: Comparison of range for Multitech and FactoryLab setup at varying SF

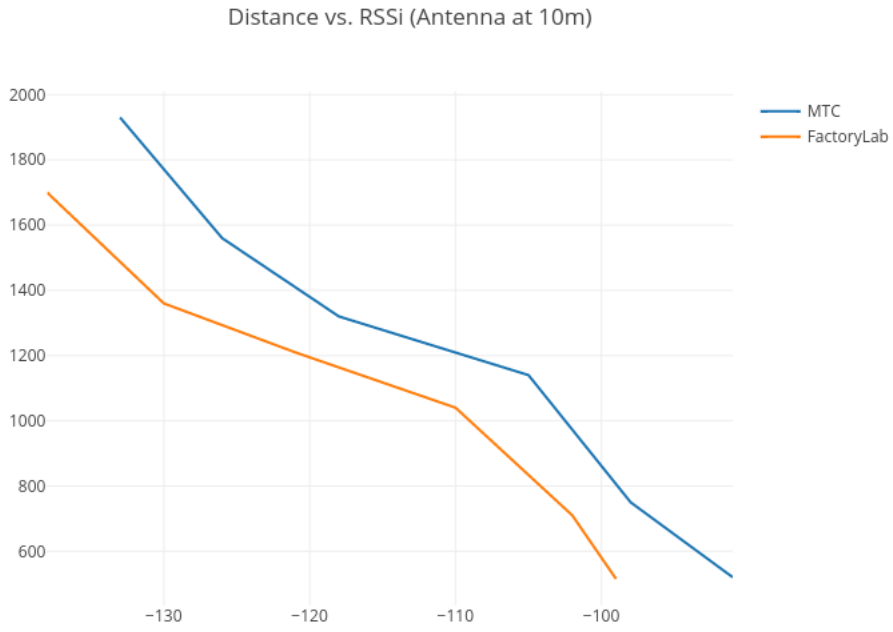

Figure 5.15: RSSI vs Distance for Multitech and FactoryLab setup at SF=12

Range to cost ratio for test setup and observations From the results obtained in 5.3 and 5.4, we can derive the Range to Cost ratios for the Multitech Conduit gateway and the FactoryLab setup. The range considered for both the gateways for the estimation f the ratio is the maximum range achieved at any instant. This turns out to be the range at SF=12 with the antenna height of 10 meters. This is the theoretical expectation and is further proven by the experiments. The cost of the Multitech gateway as per table 2.2 is found to be an all inclusive ***500 euros*** which includes the LoRa module as well the GPRS module. Using eq 5.1 we can calculate the Range to Cost ratio for the Multitech Gateway at 1930 meters for 500 euros as shown in 5.3 as 3.8 meters/euro.

$$R/C_{Multitech} = 1900/500 = 3.8\ Meters/Euro \tag{5.3}$$

Similarly the R/C for the FactoryLab gateway setup can be calculated for the highest attained range at SF=12 with the antenna at 10m. The range was found to be approximately 1700 meters. Earlier in the section an estimate of the price of 100 pieces of the FactoryLab gateway was presented and the upper bound of the price range, at ***350 euros*** was selected as it would present the best worst case Range to Cost estimation. Using 5.1 the calculation can be made for the FactoryLab test setup as shown in 5.4 as

$$R/C_{FactoryLab} = 1700/350 = 4.8\ Meters/Euro \tag{5.4}$$

This estimation of the range to cost ratio for the FactoryLab gateway can be considered the lower bound of the actual ratio which will be measured later. This is because the cost in the ratio is the higher bound of the range of costs for 100 pieces. With mass production like the Multitech and even the Kerlink IoT station, the costs of the gateway will me much lower, also this being said there is also the theoretical probability that the measured range for the gateway would be similar or better for the actual hardware model. This moves to show that the initial assumption that the costs for the gateway can be significantly reduced while maintaining decent range. While the FactoryLab setup did not attain the ranges the Multitech could due to various reasons, further development of the gateway could provide better proof of efficient operation for lower unit cost.

5.5 Conclusion

This chapter highlights the experiments and tests that were carried out. Initially it is mentioned that the actual hardware for the gateway could not be manufactured in time for the test and accumulation of results and an alternate setup of comparable hardware consisting of a Raspberry Pi along with a LoRa concentrator module, soft modified to mimic the FactoryLab gateway hardware. The tests on the Kerlink IoT station were carried out in the field and due to the unavailability of the gateway, the results had to be limited. Further two different tests are carried out with the remaining two gateways. The tests are varying in the sense that the gateways are tested in standard conditions, with varying spreading factors and fixed antenna height for two different gateway placements. One at 2m from ground level in Delft and the other at 10m from

ground level in the FactoryLab offices in Zwijndrecht
The results of the first test can be seen from figures 5.8, 5.10 and table 5.1. The results for the second test with a higher placement of the antenna for both gateways can be found in figures 5.12, 5.13, 5.14 and in table 5.2.

With this information, the range to cost ratio can be determined for the three gateways and is found to be 4.33 for the Kerlink IoT station, 3.8 for the Multitech Conduit and 4.8 for the FactoryLab test setup. All ratios are measured in meters per euro.

6 Conclusions

The final chapter of this thesis report finalizes all the findings and research conducted during the development process. The main objective of this project was to achieve an improved range to cost ratio for a newly developed gateway for LoRa communication by optimizing software for dedicated hardware. While the hardware was designed and developed in house at FactoryLab in Zwijndrecht, the manufacturing process couldn't be completed in time and hence actual results could not be derived. An approximation of the results although, can be derived from a setup using comparable hardware running the software and kernel. This chapter first summarizes the previous chapters in Section 6.1 and highlights the learning extracted, then the main contributions of this project are discussed in Section 6.2. These contributions are compared to the project goals and the extent of the project goal achievement is discussed. Finally the chapter concludes the expectation of future work and a discussion of further steps which will be taken in the further development of the Gateway.

6.1 Summary of the Thesis report

The report begins with an introduction to the topic and project in Chapter 1. The chapter begins with a differentiation between the internet(internet of everything) and the internet of things which says that, the main difference between the internet of everything and the internet of things is that 'things' are physical devices. While the internet of things is a sub domain of IoE, it is specific to data communication between physical devices. This means that another major factor in this consideration is that the consumption and importance of power consumption in said devices. Starting from understanding this difference, the chapter further dives into the main motivation behind the conception of the gateway which is branching off from the monopoly that has been created in the LoRa gateway market by a few big players. After the invention or deployment of the LoRa and LoRaWAN standards, which provide long range and low power communication alternatives for the IoT domain, the apparent and glaring advantage over the current standards like Bluetooth and WiFi became imminent. This posed the technology for large development and this led to attraction and developmental interest from big market players like SemTech and Microchip. Further the development of the gateways, which form an integral part of any LoRa network was undertaken by few companies and with the lack of competition, the prices went very high. Considering the fact that all gateways use the same chipset and transceivers for operation, an opportunity to create a lower cost gateway with similar capabilities presented itself.

Following the motivation for the gateway design, the chapter focuses on the definition or official problem statement and defines the project goals. The project aim

being the hardware-software co-design of a LoRa based Linux gateway for proprietary use for IoT applications and to compare the gateway to industry standards to optimize the range to cost ratio of the device. The chapter is concluded with the project goals and a detailed structure of the report.

Following the introduction, Chapter 2 formally semi-details some background the technology used in development. The chapter begins with definition of the LoRa protocol and the LoRaWAN stack providing some technical details about the protocol. This is followed by the difference between LoRa and the LoRaWAN stack along with the definitions for both. The Section 2.1.2 contains the definitions for various parameters in the LoRaWAN stack and also the definitions for each of the three device classes used in the stack. In summary, Class A is bidirectional devices where each up-link is followed by two default down-link windows. Class B devices where the down-link windows are scheduled and not fixed and the Class C devices where there are maximum receive slots. Various terms used in the connection procedures are also defined and the frame is depicted in figure 2.3. Section 2.2 shows the various industrial gateways in deployment and the range and cost factors for each. These results are summarized in table 2.2.

Chapter 3 thrives in the details of the literature that helped in creating a good foundation for the development of the thesis. This chapter focuses on reviewing some of the literature that was researched in the development of the FactoryLab gateway and its execution. While [24, 35, 38, 30] provide a clear insight into the LoRa standard and major understanding of the LoRaWAN protocol can be derived from [21] as well, [43, 17, 2] help understand and evaluate the various test scenarios and expected results from the LoRa gateway which is a key aspect of the project. Finally, [25, 45] help understand the importance of the Linux operating system and [45] paints the exact picture needed for developing a modern Linux based embedded platform.

Chapter 4 details the methodology and design implementation of the software and hardware for the gateway. Section 4.1.1 highlights the importance of the Linux kernel as the choice of operating system and further explains the boot process that the gateway undergoes when power cycled. In summary first the ROM is booted and the system looks for the bootloader which is downloaded to the RAM and executed. Next, the bootstrap initializes the hardware devices the addresses to these devices are stored in the starting of the NAND Flash, then the UBoot bootloader is executed which initializes the CMD line and loads the Linux kernel on RAM. Further the use of buidroot is explained in Section 4.1.1.2 and the process of generating the kernel is explained. An example of the kernel configuration can be seen in figure []. Further Section 4.1.2 expands on the LoRa software which is used while further explaining the LoRa frame in detail followed by detail explanation of the join procedures, Activation by personalization and Over the air activation are explained. Figure 4.9 shows the reference software block diagram. Section 4.2 details the hardware design and component selection process for the gateway

hardware. This explains the choices made in choosing the components with respect to maintaining a good range to cost ratio. Further sections detail the specific modules of the gateway hardware and the design choices surrounding them. Finally Section 4.2.1.5 enlists the design challenges.

Chapter 5 lists and demonstrates the results attained in the project. First the test setup is explained and the hardware used to emulate the gateway hardware that could not be manufactured before testing is described. Then the various test scenarios are listed and the results from various tests are seen. The range to cost ratio is shown for all test gateways and final conclusions can be made from there.

6.2 Contribution and comparison to project goals

As listed in Section 1.3 the following project goals and the extent of their achievement is highlighted.

- *Accumulation of sufficient knowledge of the LoRaWAN protocol to understand the challenges.* A sufficient but not extensive knowledge of the LoRaWAN structure was attained. This was particularly needed to understand the communication requirements, the packet structure and various factor which affect the range of the network. The knowledge of understanding the difference between LoRa and LoRaWAN and implementing a network structure that incorporates LoRa based devices operating with the LoRaWAN network protocol provides insight and promotes future work into LoRa development.

- *Creating model software and hardware for effective development.* Creating the software model as seen in section 4.1 was imperative in the understanding the flow of the software and creating the eventual software design. The model hardware on the other hand was an important step in understanding the hardware requirements and designing the hardware with minimal errors in order to reduce the manufacturing costs. This helped immensely in reducing the cost of the prototype hardware and eventually improving the range to cost ratio.

- *Choosing hardware and developing the hardware schematic.* The hardware schematic proved to be the most time consuming part of the project as almost all of the modules had to be designed by hand and then further placed/connected to create the most efficient hardware schematic possible for PCB design. Creating a schematic and PCF design proved to be the defining point in the quest to improve the Range to Cost ratio as much as was possible. As the LoRaWAN software stack is defined by the LoRa Alliance and the software for the gateway is provided by the Semtech Corporation for open development, they key in understanding the requirements and executing said requirements within a specified cost frame was the most important factor in Range to Cost ration improvement.

- *Creating PCB footprint for High speed transmissions.* A peculiar challenge as the design of the final prototype board hinged on the design and optimization of the high speed data devices. As there was no formal experience in this field, the hardware developer at FactoryLab helped create an optimal design for the gateway. The PCB was also designed Using KiCad which is a free hardware design tool which is being developed rigorously by CERN for open public.

- *Development of on board Linux Kernel and testing* Developing the Linux kernel with hardware specific requirements was a nice challenge as there was no former experience in developing embedded operating systems before. Lot of research was carried out about the principles of creating embedded Linux systems and considering all parameters required for reducing flaws in the operating system environment.

- *Analyzing range for various gateways and evaluating the Range to cost ratio* Clearly the most important goal in the project. The Ranges for all three LoRa based gateways was analyzed using various tests like measuring the RSSI over increasing distance and measuring the effect of changing the Spreading factor on the range. The antenna height was also changed the tests were performed again to realize the theoretical assumption that, the range increases with increasing spreading factor and using the gateway in a low density environment with the antenna at a good height changes the range drastically.
  The Range to cost ratios for the Kerlink, Multitech and FactoryLab gateway were found to be 4.3, 3.8 and 4.8 meters per euro respectively. Thus the task of achieving similar ranges as the Multitech gateway, an industrial standard while also reducing the cost by almost 50%, was achieved.

6.3 Future Work

In an ongoing product development project there are many challenges faced in the initial stages of the development cycle which also brings the scope for a lot of future work. Especially in a project involving hardware-software co design of a product, there are many areas for improvement in both the faces of the design model. The aspects that can be worked upon and improved in the future are as follows:

- Initially there is scope for development of a new LoRaWAN base-band concentrator apart from the one provided by Semtech. This is an essential step in further hampering the monopoly in the LoRa market and also further reducing the Range to cost ratio of many devices. Creating a LoRa stack for proprietary use with the FactoryLab nodes and filing and achieving certification from the LoRaWAN alliance would go a long way in establishing the product int he market.

- While the FactoryLab Gateway hardware is ready to be tested, there is still the job of completing development of the breakout LoRa module. This module is essential to the future goals of the project as making the LoRa module for the gateway plug and play could further help reduce the costs of maintenance of the gateway as the LoRa module, one of the most expensive parts of the gateway, now becomes

interchangeable. This goes a long way in reducing other costs that are not included in this project.

- After thorough testing of the Gateway, there is a possibility that some hardware changes would be required and deemed essential. These could include changing peripherals and other components and creating a more cohesive gateway module. While most gateways are considered 'dumb' devices, in the sense that they perform only a few functions of receiving, unpacking, checking, repacking and forwarding, the introduction of a Linux based system and the future development of LoRa could provide a solid platform to introduce a little smartness in the gateway, providing it with a multi functional base of operation.

- While the Multitech is an industrial gateway, what gives the Kerlink the distinct edge is that the Kerlink IoT station is a more physically robust and sturdy gateway module. This characteristic can be credited to the casing around the electronics on the gateway. The future of the FactoryLab gateway should include a industrial grade, robust and IP68 waterproof variant which could extend the reach of LoRa into deeper, more specific markets.

- A final developmental aspect of the Gateway could be, adding compatibility for various other communication and HMI standards thus enabling the user to cross match various radio/wireless or wireless standards creating better platform for application based IoT development.

Bibliography

[1] *Atmel SAM-BA In-system Programmer*. URL: http://www.atmel.com/tools/ATMELSAM-BAIN-SYSTEMPROGRAMMER.aspx.

[2] E. D. Ayele et al. "Performance analysis of LoRa radio for an indoor IoT applications". In: *2017 International Conference on Internet of Things for the Global Community (IoTGC)*. July 2017, pp. 1–8. DOI: 10.1109/IoTGC.2017.8008973.

[3] A. Azari and G. Miao. "Fundamental tradeoffs in resource provisioning for IoT services over cellular networks". In: *2017 IEEE International Conference on Communications (ICC)*. May 2017, pp. 1–7. DOI: 10.1109/ICC.2017.7996885.

[4] D. Bankov, E. Khorov, and A. Lyakhov. "On the Limits of LoRaWAN Channel Access". In: *2016 International Conference on Engineering and Telecommunication (EnT)*. Nov. 2016, pp. 10–14. DOI: 10.1109/EnT.2016.011.

[5] P. Bellagente et al. "Enabling PROFINET devices to work in IoT: Characterization and requirements". In: *2016 IEEE International Instrumentation and Measurement Technology Conference Proceedings*. May 2016, pp. 1–6. DOI: 10.1109/I2MTC.2016.7520417.

[6] *Block cipher mode of operation*. July 2017. URL: https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation#CTR.

[7] *Buildroot - Making Embedded Linux Easy*. URL: https://buildroot.org/.

[8] *BusyBox information*. URL: https://www.busybox.net/about.html.

[9] *CERN Accelerating science*. URL: http://home.cern/.

[10] *CERN Accelerating science*. URL: http://home.cern/cern-people/updates/2017/07/kicad-reaches-new-heights.

[11] *Computer Memory UpgradeYour guide to computer memory upgrades, RAM buying guide, and guide to install compatible RAM*. URL: http://computermemoryupgrade.net/memory-influence-on-performance.html.

[12] *Development*. URL: http://www.libelium.com/development/waspmote.

[13] *Development*. URL: http://www.libelium.com/development/waspmote/documentation/lora-gateway-tutorial/.

[14] *Hype Cycle for Emerging Technologies, 2017*. July 2017. URL: https://www.gartner.com/doc/3768572/hype-cycle-emerging-technologies-.

[15] *Kerlink IoT station*. URL: http://www.kerlink.fr/en/products/lora-iot-station-2/wirnet-station-868.

[16] *Kicad*. URL: http://kicad-pcb.org/.

[17] D. H. Kim et al. "Design and implementation of object tracking system based on LoRa". In: *2017 International Conference on Information Networking (ICOIN)*. Jan. 2017, pp. 463–467. DOI: 10.1109/ICOIN.2017.7899535.

[18] Link Labs. *Symphony Link*. URL: https://www.link-labs.com/symphony.

[19] Sujuan Liu, Chuyu Xia, and Zhenzhen Zhao. "A low-power real-time air quality monitoring system using LPWAN based on LoRa". In: *2016 13th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT)*. Oct. 2016, pp. 379–381. DOI: 10.1109/ICSICT.2016.7998927.

[20] *LoRa Gateways and Concentrators at LORIOT.io*. URL: https://www.loriot.io/gateways.html.

[21] *LoRaWAN MAC Specification*. URL: https://src.aw-som.com/git/awsom/loramac-node/raw/6a65548e24d2262b1af647469b239823f740c780/Doc/LoRa%5C%20MAC%5C%20Specification.pdf.

[22] *LoRaWAN specification*. URL: https://www.lora-alliance.org/technology.

[23] Arm Ltd. *Architecting a Smarter World Arm*. URL: http://www.arm.com/.

[24] D. Magrin, M. Centenaro, and L. Vangelista. "Performance evaluation of LoRa networks in a smart city scenario". In: *2017 IEEE International Conference on Communications (ICC)*. May 2017, pp. 1–7. DOI: 10.1109/ICC.2017.7996384.

[25] V. Mahalakshmi et al. "Implementation of an embedded system of UART and LED using UBOOT Linux". In: *2015 International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM)*. May 2015, pp. 477–483. DOI: 10.1109/ICSTM.2015.7225464.

[26] K. Mikhaylov, .. Juha Petaejaejaervi, and T. Haenninen. "Analysis of Capacity and Scalability of the LoRa Low Power Wide Area Network Technology". In: *European Wireless 2016; 22th European Wireless Conference*. May 2016, pp. 1–6.

[27] *MultiConnect ConduitTM*. URL: https://www.multitech.com/brands/multiconnect-conduit.

[28] *MultiConnect mCardTM*. URL: https://www.multitech.com/models/94557302LF.

[29] Prashant Panigrahi. "LoRaWAN Frequency Bands - LoRa Tutorial". In: *3GLTEInfo* (2017). URL: http://www.3glteinfo.com/lora/lorawan-frequency-bands/.

[30] G. S. Ramachandran et al. "x03BC;PnP-WAN: Experiences with LoRa and its deployment in DR Congo". In: *2017 9th International Conference on Communication Systems and Networks (COMSNETS)*. Jan. 2017, pp. 63–70. DOI: 10.1109/COMSNETS.2017.7945359.

[31] *Raspberry Pi*. Aug. 2017. URL: https://en.wikipedia.org/wiki/Raspberry_Pi.

[32] *Raspberry Pi 1 Model B*. URL: https://www.raspberrypi.org/products/raspberry-pi-1-model-b/.

[33] B. Reynders and S. Pollin. "Chirp spread spectrum as a modulation technique for long range communication". In: *2016 Symposium on Communications and Vehicular Technologies (SCVT)*. Nov. 2016, pp. 1–5. DOI: 10.1109/SCVT.2016.7797659.

[34] S. Rinaldi, P. Ferrari, and A. Flammini. "Analysis of IEEE 1588-based Cyber Physical System for micro grid automation". In: *2015 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control, and Communication (ISPCS)*. Oct. 2015, pp. 31–36. DOI: 10.1109/ISPCS.2015.7324676.

[35] M. Rizzi et al. "Using LoRa for industrial wireless networks". In: *2017 IEEE 13th International Workshop on Factory Communication Systems (WFCS)*. May 2017, pp. 1–4. DOI: 10.1109/WFCS.2017.7991972.

[36] *Semtech SX1257*. URL: http://www.semtech.com/wireless-rf/rf-transceivers/sx1257/.

[37] *Semtech SX1301*. URL: http://www.semtech.com/wireless-rf/rf-transceivers/sx1301/.

[38] V. A. Stan, R. S. Timnea, and R. A. Gheorghiu. "Overview of high reliable radio data infrastructures for public automation applications: LoRa networks". In: *2016 8th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*. June 2016, pp. 1–4. DOI: 10.1109/ECAI.2016.7861130.

[39] I. Stojmenovic. "Machine-to-Machine Communications With In-Network Data Aggregation, Processing, and Actuation for Large-Scale Cyber-Physical Systems". In: *IEEE Internet of Things Journal* 1.2 (Apr. 2014), pp. 122–128. ISSN: 2327-4662. DOI: 10.1109/JIOT.2014.2311693.

[40] T. Taleb and A. Kunz. "Machine type communications in 3GPP networks: potential, challenges, and solutions". In: *IEEE Communications Magazine* 50.3 (Mar. 2012), pp. 178–184. ISSN: 0163-6804. DOI: 10.1109/MCOM.2012.6163599.

[41] F. Tao et al. "IoT-Based Intelligent Perception and Access of Manufacturing Resource Toward Cloud Manufacturing". In: *IEEE Transactions on Industrial Informatics* 10.2 (May 2014), pp. 1547–1557. ISSN: 1551-3203. DOI: 10.1109/TII.2014.2306397.

[42] *The Things Gateway*. Aug. 2017. URL: https://www.thethingsnetwork.org/docs/gateways/gateway/.

[43] A. J. Wixted et al. "Evaluation of LoRa and LoRaWAN for wireless sensor networks". In: *2016 IEEE SENSORS*. Oct. 2016, pp. 1–3. DOI: 10.1109/ICSENS.2016.7808712.

[44] L. D. Xu, W. He, and S. Li. "Internet of Things in Industries: A Survey". In: *IEEE Transactions on Industrial Informatics* 10.4 (Nov. 2014), pp. 2233–2243. ISSN: 1551-3203. DOI: 10.1109/TII.2014.2300753.

[45] Karim Yaghmour, Jon Masters, and Gilad Ben-Yossef. *Building Embedded Linux Systems*. OReilly Associates, 2008.