Thesis Title

"LoRa protocol analysis and performance evaluation using Raspberry Pi equipment "

Margarita Bitzi

Department of Digital Systems
Master Degree in Digital Communications and Networks
University of Piraeus
Greece

Acknowledgement

I would first like to thank my thesis advisor Prof. Angelki Alexiou of the Departement of Digital Systems at University of Piraeus for her important role to my Master Thesis and Dr. Antonis Gotsis for his guidance, patience and aid .

As this master thesis was held in collaboration with Ms. Maria Kouvatsou, I would like to thank her for the cooperation, the patience and support. Without her the accomplishment of this difficult goal, wouldn't be achieved.

I would also like to acknowledge my family for their steadily support .

Contents

Abstract

The purpose of this Master thesis is to present a thorough assessment of Lora Protocol, either for its Theoritical Backround or its Practical Applications to experiments that were held.

Firstly, we tried to place the Lora Technology to the IoT scenery as it is formed today, and compare it with its peer technologies. Afterwards, we examined the LoRa's technology technical anatomy. And finally we presented a set of experiments, which assessed the LoRa technology coverage.

Chapter 1

Introduction

1.1 Introduction

The fourth industrial revolution, as it is provisioned by the experts, bears many technological breakthroughs such as cyber-physical systems, robotics, AI implementations, biotechnology, 3D printing and fifth-generation wireless technologies (5G) . Among all these impressive innovations, we can not neglect the future form of Internet, the so called Internet of Things (IoT) which ,in the near future, is expected to be Internet of Everything (IoE).

If Internet of things(IoT) demands every object to be connected, then Internet of Everything (IoE) expects that objects will be connected even with human beings and other living species. Without doubt, this is a Technological evolution that revolutionizes not only the conventional Telecommunications System but brings a new era in our everyday life. Thus, IoT/IoE implementatios are expected to change the world.

Internet of Everything is a concept that describes a network of Internet of thing and Internet of Humans (IoH). (IoE) appears as a concept that contains both the IoT and the Internet of Humans (IoH), including the capability to share data between each other (IoT and IoH) or among themselves using machine to machine (M2M) or machine to human (M2H) communications. Following a similar approach, we could shape the IoT definition to include two different concepts: industrial IoT (iIoT) and consumer IoT (cIoT), exhibiting a new scenario that will dominate the world's communications in the near future, at least in terms of number of participating devices.

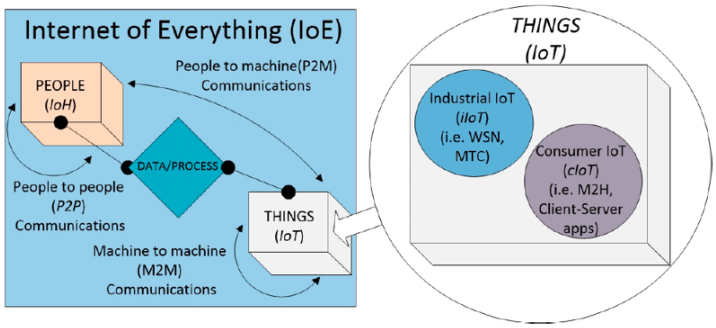


Figure 1. Internet of Everything concept.

Connection IoT and IoE [5]

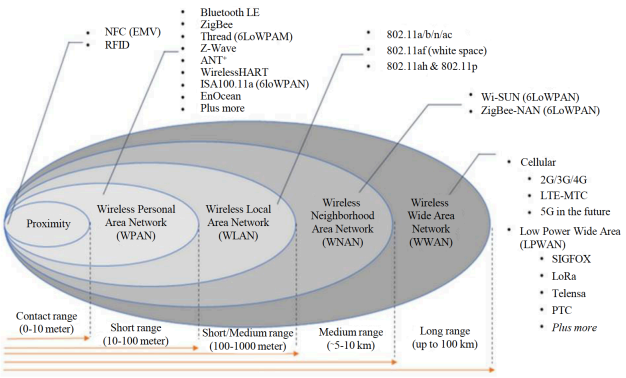
Thus, we can imagine the contribution of this upcoming technology to the already existing Iot Technology, as Smart cities, Smart Hospital , Smart Irrigation ,Smart Agriculture etc. A world fully connected it can't be but a safer world.

1.2 IoT Technologies

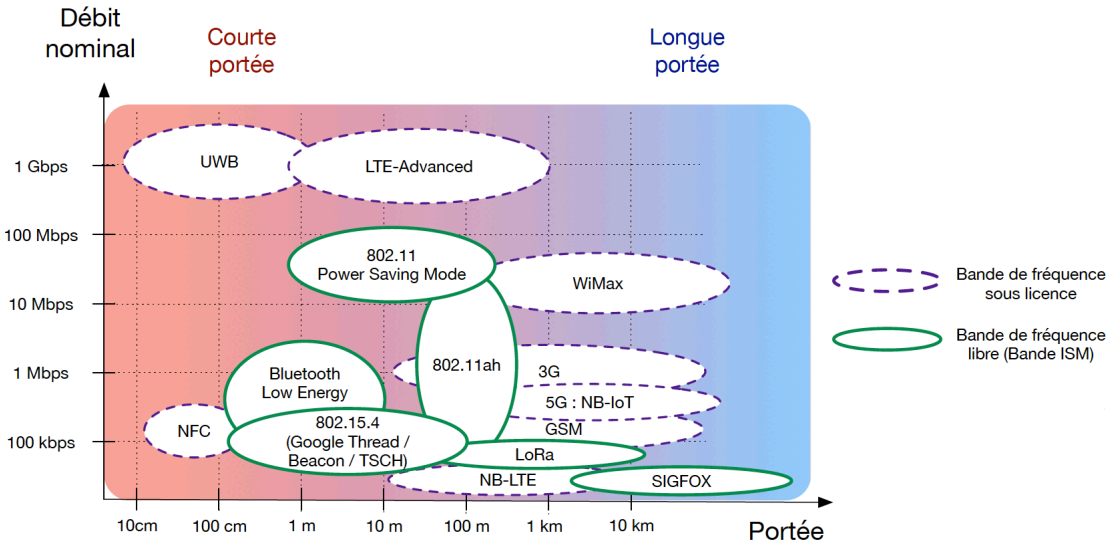
1.2.1 The Big Picture

The essential difference between conventional “Internet” and “Internet of Things” is that in the IoT, we need “less of everything” available in a given device or network device: less memory, less processing power, less bandwidth, less available energy. This is either because “things” are battery driven and maximizing lifetime is a priority or because their number is expected to be massive.[9]

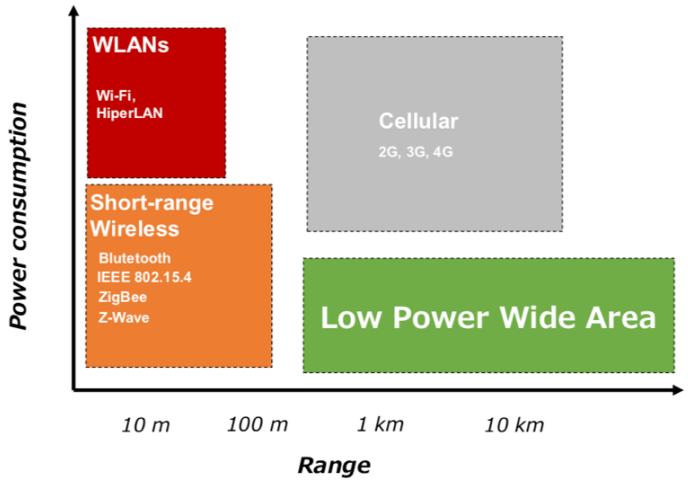
To visualise the scenery of Iot Technologies it would be useful for the reader to have a look to an IoT Technology map, which presents the differences between the existing technologies, regarding their Range-Coverage(Km), their Data Rates(Bps) and of course their Energy Consumptions:



A presentation of all the Iot technologies regarding their Coverage.[6]



A presentation of all the Iot technologies regarding their Data Rate.



A presentation of all the Iot technologies regarding their Power Consumption.[7]

So to the next section, we are going to examine the most popular technologies categorized by their Standardization Organization.

1.2.2 Most Popular IoT Technologies

There are many IoT Technologies proposed from different parties and we can distinguish three big groups:

1. 3GPP-based technology approach for IoT.
2. IEEE-based technology approach for IoT.
3. Protocols from Independant parties.

1. **3GPP-based technology approach for IoT.**

   3GPP stands for 3rd Generation Partnership Project and is a standards organization which develops protocols for mobile telephony. Its best known work is the developement and maintenance of GSM, UMTS and related 4G standards, including HSPA, LTE and related 4G standards (LTE Advanced and LTE Andvance Pro), 5G NR and relted 5G standards.

   3GPP had a determinant role regarding the establishment of IoT protocols and standards that are cellular based.

   Traditional cellular options such as 4G and LTE networks consume too much power and don't fit well with applications where only a small amount of data is transmitted infrequently (e.g. meters for reading water levels, gas consumption, or electricity use). Cellular IoT is meant to meet the requirements of low-power, long-range applications that Iot technology demands.

   3GPP has proposed 3 cellular based standards :

   (a) MTC LTE Cat-M (0,1).
   (b) EC-GSM-IoT.
   (c) NB-IoT.

   (a) **MTC LTE Cat-M (0,1).**

   Machine Type Communications is the term used in 3GPP to refer to Machine-to-Machine (M2M) communications, that is, machine devices talking to each other through mobile networks or locally.

   LTE-M (LTE-MTC [Machine Type Communication]), is a type of low power wide area network (LPWAN) radio technology standard developed by 3GPP to enable a wide range of cellular devices

and services (specifically, for machine-to-machine and Internet of Things applications).[1][2] The specification for eMTC (LTE Cat-M1) was frozen in 3GPP Release 13 (LTE Advanced Pro), in June 2016

Cat-1 , Category 1 was included in the LTE specifications already in the beginning, Release 8. With a Cat-1 UE, it is possible to achieve 10 Mbps downlink and 5 Mbps uplink channel data rates.

Cat-0 , Category 0 is one of the newest standardized categories from Release 12. Cat-0 UEs are intended for IoT use cases, and provide 1 Mbps data rates for both up- and downlink.

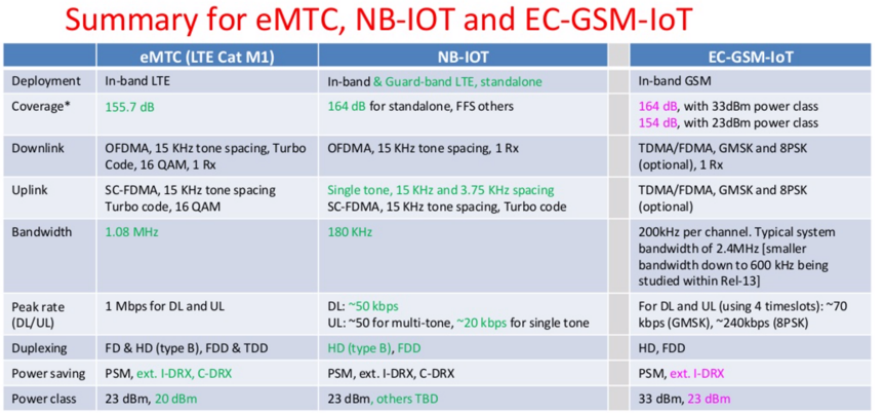
Cat-M1, Category M1 (which has informally also been referred to as Category M), refers to Release 13, where further complexity reduction techniques on top of the ones for Cat-0 are standardized.

(b) **EC-GSM-IoT.**

EC-GSM-IoT, earlier referred to as EC-EGPRS, stands for Extended Coverage GSM for IoT. It includes the latest enhancements to the GSM and EGPRS standards to support better coverage and other IoT enhancements. EC-GSM-IoT supports 20 dB coverage improvements and can be deployed in the existing GSM networks.

(c) **NB-IoT.**

NB-IoT stands for Narrowband IoT and is a new narrowband radio technology being standardized in 3GPP. It covers all the components sought after: low complexity, low power consumption and long range. Some key characteristics include 180 kHz bandwidth and uplink and downlink data rates of about 200 kbps with half-duplex operation. Although this is a new radio interface, NB-IoT deployments can be made ”inband”, so that existing resource blocks in the LTE carrier are used. The term is not to be confused with LTE-M, which refers to more direct use of LTE evolution for MTC and IoT use cases. NB-IoT is subject to a lot of standardization activities at the moment. The complexity reduction compared to Cat-1 is up to 90 percent.



Summary for eMTC, NB-IOT and EC-GSM-IoT

	eMTC (LTE Cat M1)	NB-IOT	EC-GSM-IoT
Deployment	In-band LTE	In-band & Guard-band LTE, standalone	In-band GSM
Coverage*	155.7 dB	164 dB for standalone, FFS others	164 dB, with 33dBm power class 154 dB, with 23dBm power class
Downlink	OFDMA, 15 KHz tone spacing, Turbo Code, 16 QAM, 1 Rx	OFDMA, 15 KHz tone spacing, 1 Rx	TDMA/FDMA, GMSK and 8PSK (optional), 1 Rx
Uplink	SC-FDMA, 15 KHz tone spacing Turbo code, 16 QAM	Single tone, 15 KHz and 3.75 KHz spacing SC-FDMA, 15 KHz tone spacing, Turbo code	TDMA/FDMA, GMSK and 8PSK (optional)
Bandwidth	1.08 MHz	180 KHz	200kHz per channel. Typical system bandwidth of 2.4MHz [smaller bandwidth down to 600 kHz being studied within Rel-13]
Peak rate (DL/UL)	1 Mbps for DL and UL	DL: ~50 kbps UL: ~50 for multi-tone, ~20 kbps for single tone	For DL and UL (using 4 timeslots): ~70 kbps (GMSK), ~240kbps (8PSK)
Duplexing	FD & HD (type B), FDD & TDD	HD (type B), FDD	HD, FDD
Power saving	PSM, ext. I-DRX, C-DRX	PSM, ext. I-DRX, C-DRX	PSM, ext. I-DRX
Power class	23 dBm, 20 dBm	23 dBm, others TBD	33 dBm, 23 dBm

Comparison of 3GPPP standards. [4]

2. Protocols standardized from IEEE.

IEEE stands for "Institute of Electrical and Electronics Engineers" and is a professional association of electrical engineers . It was formed in 1963 from the amalgamation of the American Institute of Electrical Engineers and the Institute of Radio Engineers. IEEE has a significant role in scientific research regarding technological issues and had determinant presence in the standardisation of many well known technologies such as Wi-Fi.

Thus, IEEE , as a pioneer institute ,was one of the organisations that paved the way for IoT standardisation protocols and their suggestions are the followings:

IEEE 802.11ah (Wi-FI)

IEEE 802.11ah is a protocol released by IEEE in 2017 as a new release of WiFi, named, Wi-Fi HaLow (pronounced "HEY-Low") that targets to describe IoT communications. It uses 900 MHz license exempt bands to provide extended range Wi-Fi networks, compared to conventional Wi-Fi networks operating in the 2.4 GHz and 5 GHz bands. It also benefits from lower energy consumption, allowing the creation of large groups of stations or sensors that cooperate to share signals, supporting the concept of the Internet of Things (IoT).The protocol's low power consumption competes with Bluetooth and has the added benefit of higher data rates and wider coverage range. It supports two data rates that are adjusts to the two different bandwidths the standads uses : For BW=1Mhz:0.15-4Mbps and For BW=2: 0.65-7.8Mbps. The Typical range is 100-1000m. The transmission technique is OFDM and the modulation technique can vary among BPSK,QPSK,16-QAM, 64-QAM, 256-QAM.

IEEE 802.15.4

IEEE 802.15.4 is a standard that defines low rate wireless personal area networks (LR-WPANs). Created by IEEE 802.15 TG4 in 2003 when the goup was chartered to investigate a low data rate solution with multi-month to multi-year battery life and very low complexity. It is operating in an unlicensed, international frequency band. Potential applications would be sensors, interactive toys, smart badges, remote controls, and home automation.

It is the basis for the Zigbee, ISA100.11a, WirelessHART, MiWi, 6LoWPAN, Thread and SNAP specifications, each of which further extends the standard by developing the upper layers which are not defined in IEEE 802.15.4. In particular, 6LoWPAN defines a binding for the IPv6 version of the Internet Protocol (IP) over WPANs, and is itself used by upper layers like Thread.

Technically it supports a transfer rate of 250 kbit/s, real-time suitability by reservation of Guaranteed Time Slots (GTS), collision avoidance through CSMA/CA and integrated support for secure communications. Devices also include power management functions such as link quality and energy detection.

IEEE 802.15.1 (Bluetooth)

Bluetooth technology was invented in 1994 by engineers at Ericsson. In 1998, a group of companies agreed to work together using Bluetooth technology as a way to connect their products. These companies formed the Bluetooth Special Interest Group (SIG), an organization devoted to maintaining the technology. This means that no single company "owns" Bluetooth technology, but that many members of the Bluetooth SIG work together to develop Bluetooth technology. Bluetooth SIG developed Bluetooth specification. Afterwards this specification became a part of IEEE 802.15.1 standard.

In 2011, the Bluetooth SIG announced the Bluetooth Smart logo so as to clarify compatibility between the new low energy devices and other Bluetooth devices. In contrast with previous Bluetooth flavors, BLE has been designed as a low-power solution for control and monitoring applications. BLE is the distinctive feature of the Bluetooth 4.0 specification. BLE operates in the 2.4 GHz Industrial Scientific Medical (ISM) band and defines 40 Radio Frequency (RF) channels with 2 MHz channel spacing.

	Zigbee	Bluetooth	IEEE 802.11ah
Standard	IEEE 802.15.4	IEEE 802.15.1	IEEE 802.11ah
Frequency band	EU: 868 MHz NA: 915 MHz Global: 2.4 GHz	2.4 GHz	Sub-1GHz
Data rate	868 MHz band: 20 kbps 915 MHz band: 40 kbps 2.4 GHz band: 250 kbps	1 Mbps	If BW = 1 MHz: 0.15-4 Mbps If BW = 2 MHz: 0.65-7.8 Mbps
Typical range	2.4 GHz band: 10-100 m.	10-30 m.	100-1000 m.
TX power	1-100 mW	1-10 mW	<10 mW - <1 W (depending on the country's regulations)
Bandwidth per channel	868 MHz band: 0.3 Mhz 915 MHz band: 0.6 MHz 2.4 GHz band: 2 MHz	1 MHz	1, 2, 4, 8 or 16 MHz
Modulation	BPSK (+ASK), OQPSK	GFSK	BPSK, QPSK, 16-QAM, 64-QAM, 256-QAM
Transmission technique	DSSS	FHSS	OFDM
Topology	Multihop	Star	Single-hop
Battery operation	From months to years	From days to weeks	From months to years
Power saving mechanisms	Only in ZigBee RF4CE	Only in Bluetooth Low Energy (BLE)	Native
Packet length	≈ 100 bytes	From kbytes to Mbytes	≈ 100 bytes
Typical scenarios	Multihop networks with few nodes	Multimedia data exchange between nearby nodes	One-hop networks with many nodes

Comparison of Ieee standards. [3]

3. **Protocols from other parties.**

Sigfox

Sigfox is a standard created by a French Enterprise. It uses the ISM band and an Ultra-Narrow Band (UNB) modulation with Differential Binary Phase-Shift Keying (DBPSK). It operates in the 200 KHz of the publicly available band to exchange radio messages over the air. Each message is 100 Hz wide and transferred at 100 or 600 bits per second, depending on the region. As a result, long distances can be achieved while being very robust against noise.

WeightLess

Weightless is a set of LPWAN open wireless technology standards for exchanging data between a base station and thousands of machines around it. These technologies allow developers to build Low-Power Wide-Area Networks.Originally, there was three published Weightless connectivity standards Weightless-P, Weightless-N and Weightless-W. Weightless-N was an uplink only LPWAN technology. Weightless W was designed to operate in the TV whitespace. Weightless (Weightless-P) was the true winner with its true bi-directional, narrowband technology designed to be operated in global licensed and unlicensed ISM frequencies.

LoRa

LoRa (short for long range) is a spread spectrum modulation technique derived from chirp spread spectrum (CSS) technology. Semtech?s LoRa devices and wireless radio frequency technology (LoRa Technology) is a long range, low power wireless platform that has become the de facto technology for Internet of Things (IoT) networks worldwide. LoRa Technology and the open LoRaWAN.

The LoRaWAN open specification is a low power, wide area networking (LPWAN) protocol based on LoRa Technology. Designed to wirelessly connect battery operated things to the Internet in regional, national or global networks, the LoRaWAN protocol leverages the unlicensed radio spectrum in the Industrial, Scientific and Medical (ISM) band. The specification defines the device-to-infrastructure of LoRa physical layer parameters and the LoRaWAN protocol, and provides seamless interoperability between devices. While Semtech provides the radio chips featuring LoRa Technology, the LoRa Alliance®, a non-profit association and the fastest growing technology alliance, drives

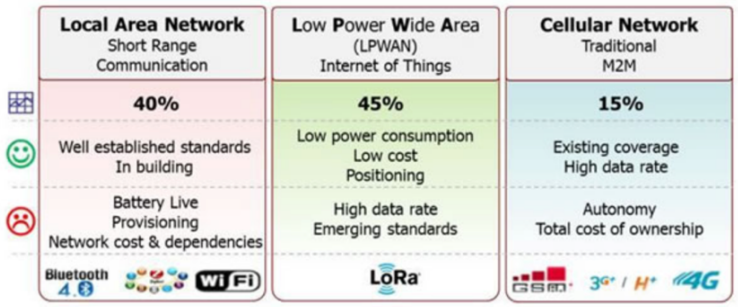
the standardization and global harmonization of the LoRaWAN protocol.

	Sigfox	Lora	Weightless-P
Total Bandwidth(kHz)	200	1000	100
Typical Data Rate(bps)	100	2466	3200
Simultaneous Demod	0.016	0.08	1
Number of Channels	2000	8	8
PHY Throughput(bps)	3200	1536	25600
Repetition Rate	1	1	1
Up Dn Ratio	1	1	2
Protocol Overhead	2	2	2
Multi-Cell Interference	1.14	1.5	1.3

A sigfox-lora-weightless Comparison.[**?**]

1.2.3 Why LoRa

Lora is an upcoming IoT Protocol that gains ground in comparison to its peers standards because of its low power consumption and low cost. Despite, its lower data rate , for certain application (as Agricultural applications) can be the best choice , and this is because some IoT applications, demand a simple connection that data rate is not a critical factor.



In this Thesis we are going to conduct a set of experiments that utilize LoRa PHY in order to transmit basic messages.

Under examination, is considered to be LoRa Radio Coverage related to different Spreading Factors and and different Transmit Powers.

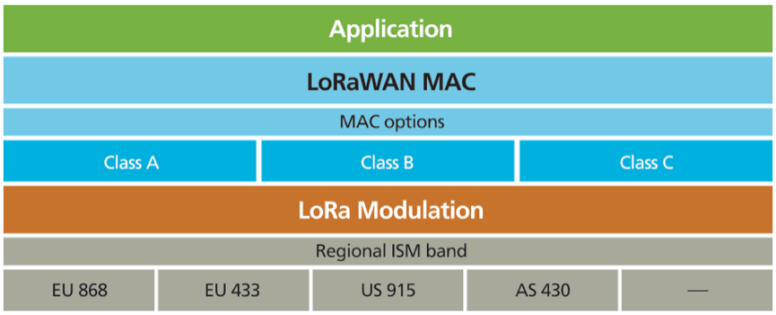
Chapter 2

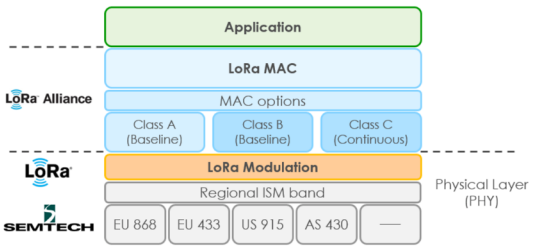
Lora's Technical Anatomy

LoRa stands for "Long Range" and is a long-range wireless communications system, promoted by the LoRa Alliance. This system aims at being usable in long-lived battery-powered devices, where the energy consumption is very importance. LoRa consists of two layers :

1. A physical layer using the Chirp Spread Spectrum (CSS) radio modulation techniqueand

2. A MAC layer protocol (LoRaWAN), although the LoRa communications system also implies a specific access network architecture





The radio and modulation part of the LPWAN technology is specified and patented by the company Semtech. The LoRa Alliance is in charge of standardizing the LoRaWAN part of the stack(Mac Layer).

2.1 Lora Physical Layer-Modulation

The radio wave modulation technology behind LoRa was developed by engineers at French company Cycleo which was founded in 2009 and acquired by Semtech, a supplier of analog and mixed-signal semiconductors, in 2012 to "complement" the latter's long-range low power radio frequency technology portfolio.

The LoRa physical layer, developed by Semtech, allows for long-range, low-power and low-throughput communications. It operates on the 433-, 868- or 915-MHz ISM bands, depending on the region in which it is deployed. The payload of each transmission can range from 2–255 octets, and the data rate can reach up to 50 Kbps when channel aggregation is employed. The modulation technique is a proprietary technology from Semtech.[8].

	Europe	North America	China	Korea	Japan	India
Frequency band	867-869MHz	902-928MHz	470-510MHz	920-925MHz	920-925MHz	865-867MHz
Channels	10	64 + 8 +8	In definition by Technical Committee	In definition by Technical Committee	In definition by Technical Committee	In definition by Technical Committee
Channel BW Up	125/250kHz	125/500kHz				
Channel BW Dn	125kHz	500kHz				
TX Power Up	+14dBm	+20dBm typ (+30dBm allowed)				
TX Power Dn	+14dBm	+27dBm				
SF Up	7-12	7-10				
Data rate	250bps- 50kbps	980bps-21.9kpbs				
Link Budget Up	155dB	154dB				
Link Budget Dn	155dB	157dB				

Chirp Spread Spectrum (CSS)

Spread spectrum technique, uses wideband, noise-like signals that are hard to detect, intercept, or demodulate. Additionally, spread-spectrum signals are harder to jam (interfere with) than narrow band signals. These low probability of intercept (LPI) and anti-jam (AJ) features are why the military has used spread spectrum for so many years. Spread-spectrum signals

are intentionally made to be a much wider band than the information they are carrying to make them more noise-like. There many spread spectrum techniques and Lora modulation uses The Chirp Spread Spectrum.

In digital communications, chirp spread spectrum (CSS) is a spread spectrum technique that uses wideband linear frequency modulated chirp pulses to encode information.

CHIRP stands for Compressed High Intensity Radar Pulse and

A chirp waveform is a Sinusoidal waveform whose frequency varies in time either linearly , or geometrically :

In a linear-frequency chirp or simply linear chirp, the instantaneous frequency f(t) varies exactly linearly with time. The waveform can be written as [17]

$$s(t) = a(t)cos[\Theta(t)]$$

where $\Theta(t)$ is the phase, and $a(t)$ is the envelope of the chirp signal which is zero outside a time interval of lenght T. The instantanous frequency is defined as :

$$f_M(t) = \frac{1}{2\pi}\frac{d\Theta}{dt}.$$

The chirp rate is defined by

$$\mu(t) = \frac{df_M}{dt} = \frac{1}{2\pi}\frac{d^2\Theta}{dt^2}$$

and represents the rate of change of the instantanous frequency.
Waveforms with $\mu(t) > 0$ are the up-chirps. and those with $\mu(t) < 0$ are the down-chirps.

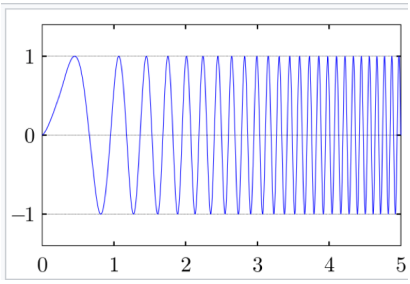
for a linear chirp $\mu(t)$ is constant, and hence $f_M(t)$ is a linear function of t, and $\Theta(t)$ is a quadratic function. If we take the waveform to be centered at t=0 it can written as:

$$s(t) = a(t)cos[2\pi f_c t + \pi\mu t^2 + \varphi_0]$$

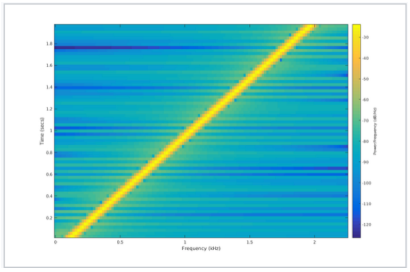
where f_c is the center frequency and a(t)=0 for $|t| > \frac{T}{2}$. It is convenient to defi

ne the bandwidth B as the range of the instantanous frequency, so that:

$$B = |\mu|T.$$

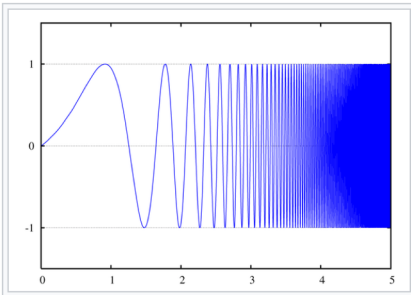


A linear chirp waveform; a sinusoidal wave that increases in frequency linearly over time

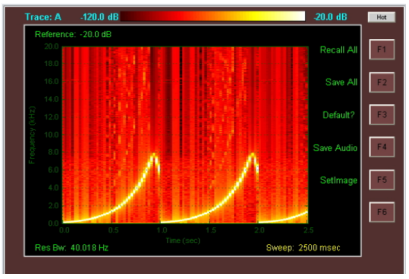


Spectrogram of a linear chirp. The spectrogram plot demonstrates the linear rate of change in frequency as a function of time, in this case from 0 to 7 kHz, repeating every 2.3 seconds. The intensity of the plot is proportional to the energy content in the signal at the indicated frequency and time.

In a geometric chirp, also called an exponential chirp, the frequency of the signal varies with a geometric (exponential)relationship over time.

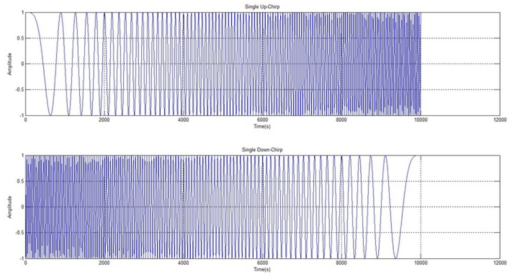


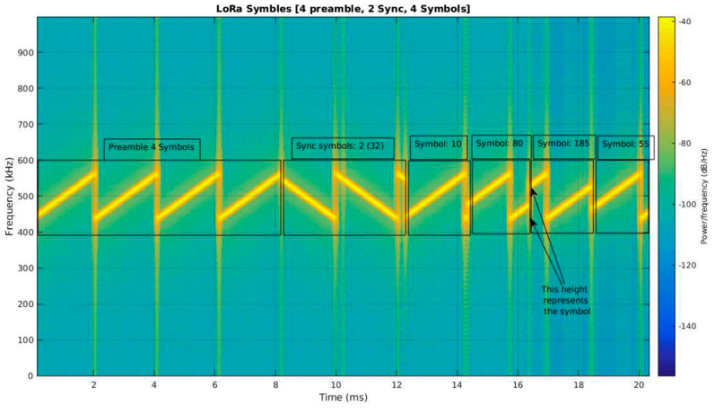
An exponential chirp waveform; a sinusoidal wave that increases in frequency exponentially over time



Spectrogram of an exponential chirp.

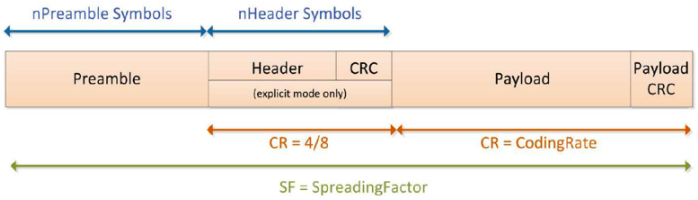
LoRa Symbols consists from upchirps and downchirps. Upchirps : Increases frequency in time. Downchirps: Decreases frequency in time.



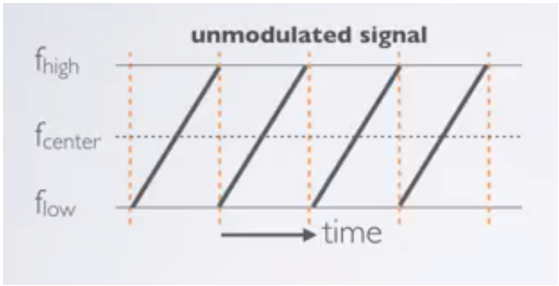


Lora's packet frame[14]:

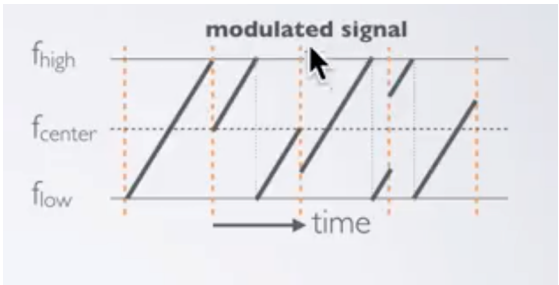
- Physical Frame (explicit mode)
  - In the following picture, the preamble includes the synchronization word.
  - Explicit mode includes the explicit header + CRC



Chirp signals are used as carrier signals where message is encoded on. Chirps are cyclically-sifted in time and it is the frequency jumps that determines how the data is encoded onto the chirps. To an unmodulated signal corresponds the below spectrogram:



To a modulated signal corresponds the following spectrogram:



Frequency Hopping with LoRa

Frequency hopping spread spectrum (FHSS) is a wireless technology that spreads a signal over rapidly changing frequencies. The frequency hopping

mode of the LoRa modem can be enabled by setting FreqHoppingPeriod to a non-zero value in register RegHopPeriod of the semtech's chip.

The principle behind the FHSS scheme is that a portion of each LoRa® packet is transmitted on each hopping channel from a look up table of frequencies managed by the host microcontroller(semtech's chip). After a predetermined hopping period the transmitter and receiver change to the next channel in a predefined list of hopping frequencies to continue transmission and reception of the next portion of the packet. The time which the transmission will dwell in any given channel is determined by FreqHoppingPeriod which is an integer multiple of symbol periods:

$$HoppingPeriod = T_s x FreqHoppingPeriod[21]$$

2.1.1 Other Modulations supported by LoRa

There are also others modulation supported by LoRa. More specifically, Lora modulator supports FSK(Frequency Sift Keying) modulation and OOk modulation.

Frequency-shift keying (FSK) is a frequency modulation scheme in which digital information is transmitted through discrete frequency changes of a carrier signal(the chirp).

On-Off Keying (OOK) is an amplitude-shift keying (ASK) modulation that represents digital data at the presence or absence of a carrier wave and it is simply deploed by switching on and off the power amplifier.

LoRa's demodulator from the other hand, can also demodulate FSK, GFSK, MSK and GMSK modulated signals.

GFSK (Gaussian frequency-shift keying modulation) filters the data pulses with a Gaussian filter to make the transitions smoother. This filter has the advantage of reducing sideband power, reducing interference with neighboring channels, at the cost of increasing intersymbol interference.

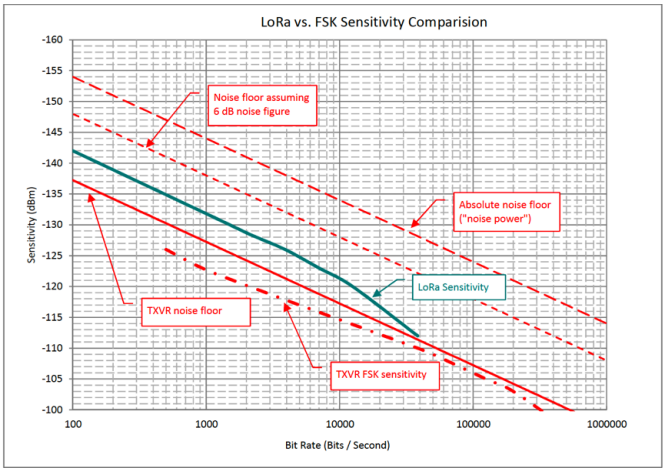
MSK (Minimum frequency-shift keying or minimum-shift keying) is a particular spectrally efficient form of coherent FSK. In MSK, the difference between the higher and lower frequency is identical to half the bit rate. Consequently, the waveforms that represent a 0 and a 1 bit differ by

exactly half a carrier period.

GMSK (Gaussian Minimum Shift Keying) is a form of modulation based on frequency shift keying that has no phase discontinuities and provides efficient use of spectrum as well as enabling high efficiency radio power amplifiers.

2.1.2 Lora's Sensitivity :

$NoiseFloor = 10 * log_{10}(k * T * B * 1000)$ (dBm)
Where:
Noise Floor = equivalent noise power (dBm)
K = Boltzmann's Constant (1.38 *10-23)
T = 293 kelvin ("room temperature")
B = channel bandwidth (Hz)
1000 = scaling factor from Watts to milli-Watts
This can be simplified as:
$NoiseFloor = -174 + 10 * log_{10}(B)$ (dBm)
Where:
-174 = 10 * log10 (k * T * 1000) as defined above
B = channel bandwidth (Hz) as before
Lora Protocol Descrption[12]



2.1.3 Lora's Link Budget :

The link budget of a wireless system or network is a measure of all the gains and losses from the transmitter, through the propagation channel, to the target receiver. These gains and losses include system gains and losses associated with the antenna, matching networks, etc. as well as losses associated propagation channel itself (either though modelling or measured data). Typically randomly varying channel mechanisms such as multipath and Doppler fading are taken into account by factoring additional margin depending on the anticipated severity. The link budget of a network wireless link can be expressed as:

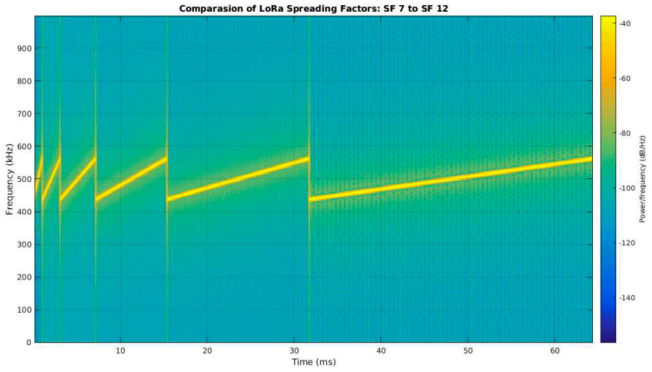
$$P_{RX}(dBm) = P_{TX}(dBm)+G_{SYSTEM}(dB)-L_{SYSTEM}(dB)+l_{CHANNEL}(dB)+M(dB)$$

Where:
P_{RX}= the expected power incident at the receiver
P_{TX} = the transmitted power
G_{SYSTEM} = system gains such as those associated with directional antennas, etc.
L_{SYSTEM} = losses associated with the system such as feed-lines, antennas (in the case of electrical short antennas associated with many remote devices), etc.
$l_{CHANNEL}$ = losses due to the propagation channel, either calculated via a wide range of channel models or from empirical data.
M = fading margin, again either calculated.

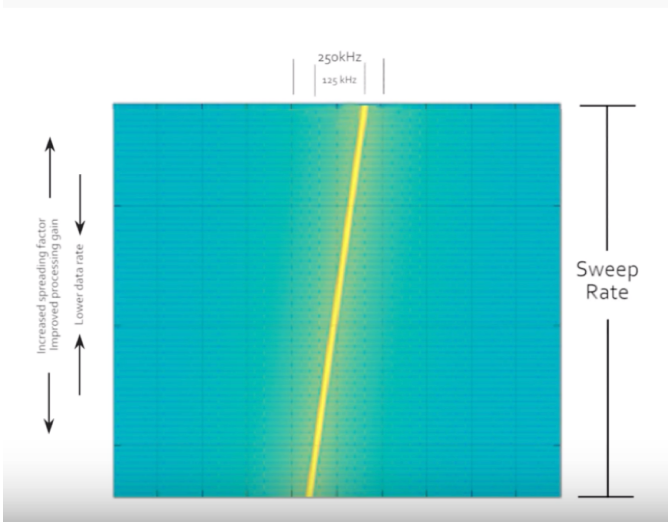
2.1.4 LoRa Wireless Transmission Principals.

As mentioned before, LoRa is a spread Spectrum technology, which use chirps to transmit data. The spreading factor expresses the number of bits that is carried by a symbol. As LoRa is basically an FSK modulation, the bits to be transmitted are expressed by the jumping of the signal between two frequencies, and the chirp pulse plays the role of the carrier signal. The spreading factor of LoRa protocol is, essentially, the sweep rate between the frequencies. Six different SF are supported by the LoRa modulation.

LoRa uses SF7 to SF12 spreading factors.



Bandwidth expresses the range of frequencies via the signal can be transmitted. As shown to the below spectrogram.



As The SF increases, the the symbol duration increases, the packet and consequently the Time on air of the packet (ToA), the message transmission

(a) Transmitter (b) Receiver

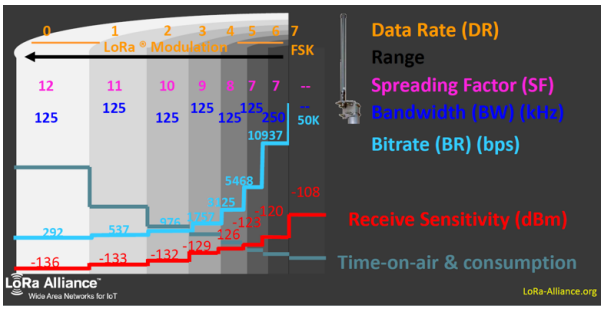
Figure 2.1: LoRa Transmission

time in other words.

In the contrary the as the Spreading Factor increases the data rate decreases, because as the SF increases, the same data bits are transferred with more chips.

$$DataRate(bits/sec) = SF * \frac{BW}{2^{SF}} * \frac{4}{4+CR}$$

CR:Coding Rate



DataRate	Configuration	Indicative physical bit rate [bit/s]
0	LoRa: SF12 / 125 kHz	250
1	LoRa: SF11 / 125 kHz	440
2	LoRa: SF10 / 125 kHz	980
3	LoRa: SF9 / 125 kHz	1760
4	LoRa: SF8 / 125 kHz	3125
5	LoRa: SF7 / 125 kHz	5470
6	LoRa: SF7 / 250 kHz	11000
7	FSK: 50 kbps	50000
8..15	RFU	

TXPower	Configuration
0	20 dBm (if supported)
1	14 dBm
2	11 dBm
3	8 dBm
4	5 dBm
5	2 dBm
6..15	RFU

LoRa Spreading Factors (125kHz bw)

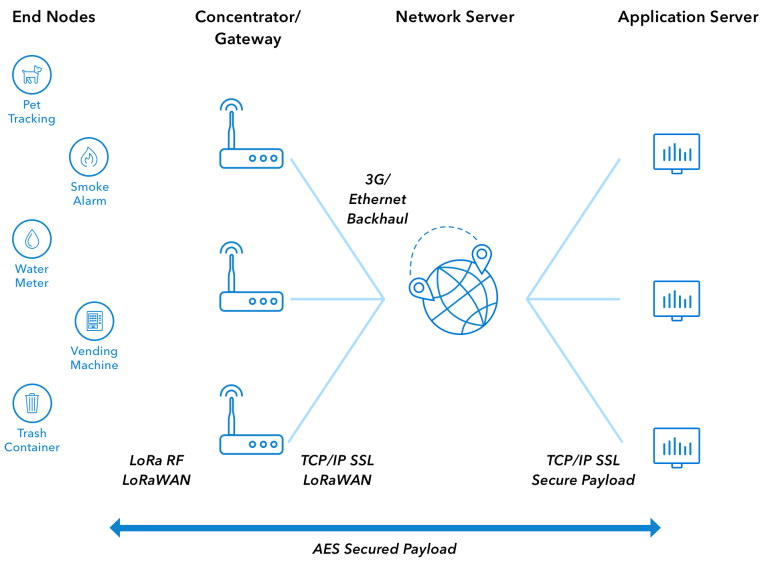
Spreading Factor	Chips/symbol	SNR limit	Time-on-air (10 byte packet)	Bitrate
7	128	-7.5	56 ms	5469 bps
8	256	-10	103 ms	3125 bps
9	512	-12.5	205 ms	1758 bps
10	1024	-15	371 ms	977 bps
11	2048	-17.5	741 ms	537 bps
12	4096	-20	1483 ms	293 bps

2.2 Lora Mac Layer-LoRaWAN

LoRaWAN is a media access control (MAC) protocol for wide area networks. It is designed to allow low-powered devices to communicate with Internet-connected applications over long range wireless connections. LoRaWAN can be mapped to the second and third layer of the OSI model. It describes star-shaped networks and it is implemented on top of LoRa or FSK modulation in industrial, scientific and medical (ISM) radio bands. The LoRaWAN protocols are defined by the LoRa Alliance and formalized in the LoRaWAN Specification which can be downloaded on the LoRa Alliance website[10]. Unlike Lora Physical Layer which is proprietary and is owned by semtech, LoRaWAN is an open source protocol.

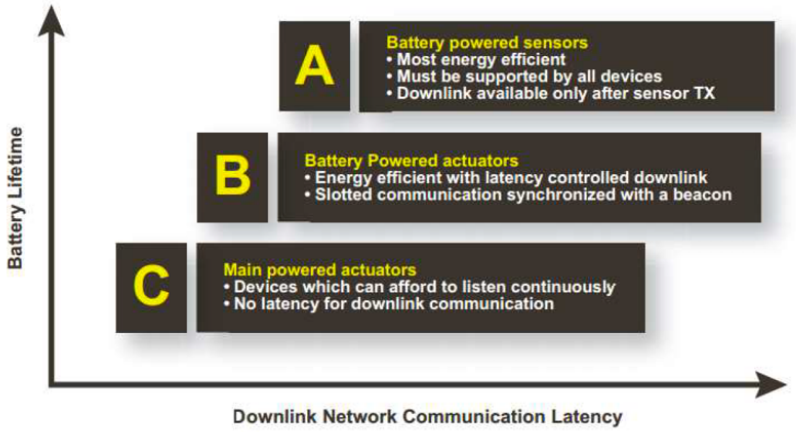
The LPWAN typically has star topology and consists of BSs relaying data messages between the End Devices and an Application Server. The Network Server implements the MAC layer and network management functions.The BSs can be connected to the central server via backbone internet protocol (IP) based link, and the wireless communication based on LoRa or GFSK modulation is used to move the data between EDs and the BSs.

LoRaWAN Architecture:



LoRaWAN Architecture.[10]

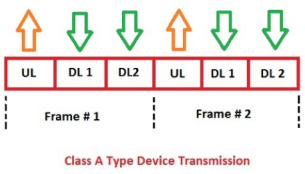
LoRaWAN end Devices:

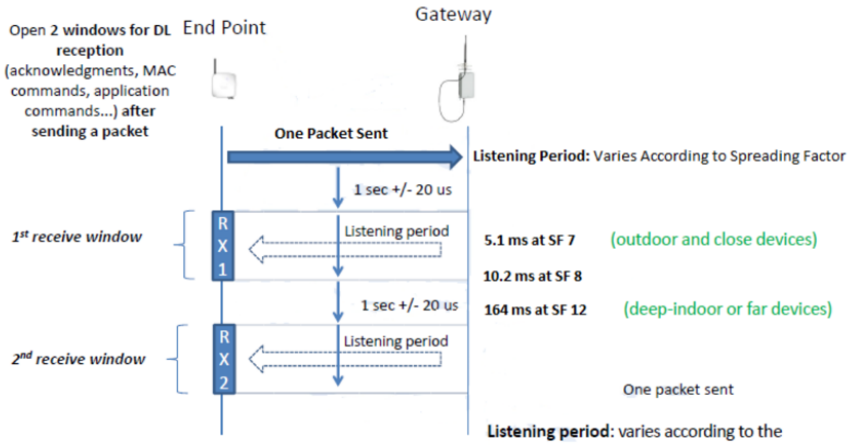


End Devices Classification .[14]

End Devices Classification:

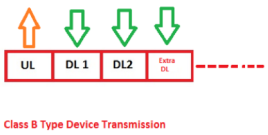
Bi-directional end-devices (Class A): End-devices of Class A allow for bi-directional communications whereby each end-device's uplink transmission is followed by two short downlink receive windows. The transmission slot scheduled by the end-device is based on its own communication needs with a small variation based on a random time basis (ALOHA-type of protocol). This Class A operation is the lowest power end-device system for applications that only require downlink communication from the server shortly after the end-device has sent an uplink transmission. Downlink communications from the server at any other time will have to wait until the next scheduled uplink.[14]

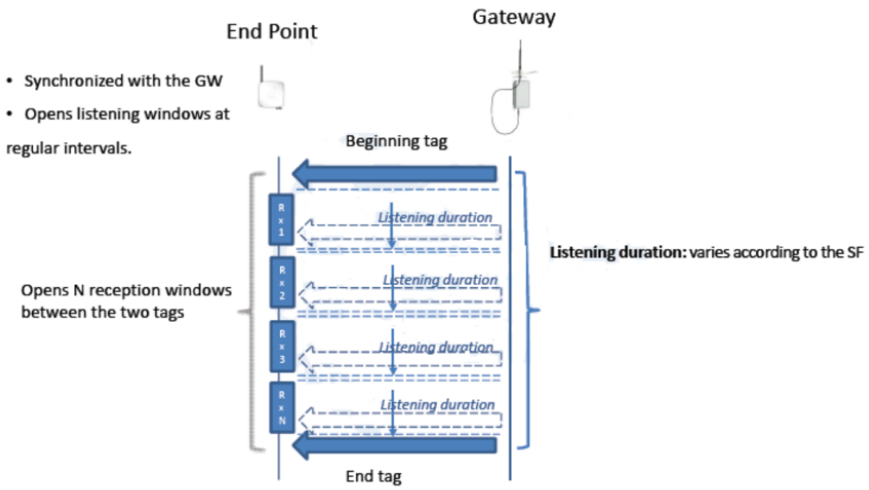




Class A Device Communication Description.[15]

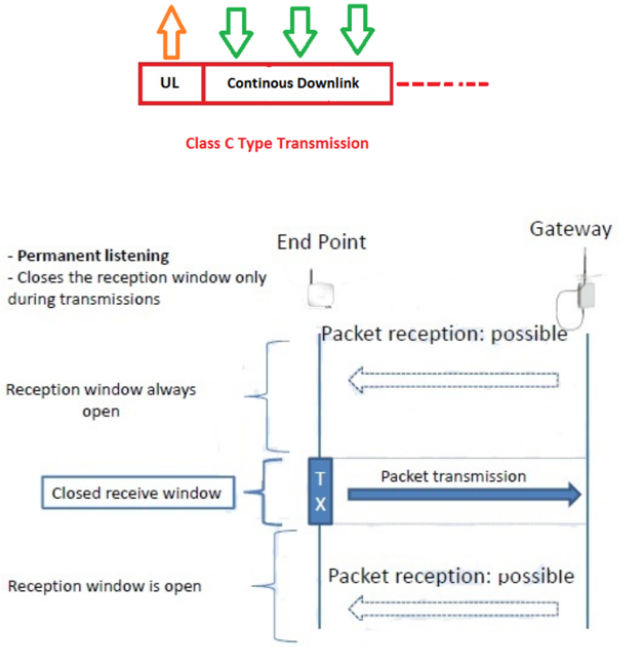
Bi-directional end-devices with scheduled receive slots (Class B) - Beacon: In addition to the Class A random receive windows, Class B devices open extra receive windows at scheduled times. In order for the end-device to open its receive window at the scheduled time, it receives a time-synchronized beacon from the gateway. This allows the server to know when the end-device is listening.[14]





Class B Device Communication Description.[15]

Bi-directional end-devices with maximal receive slots (Class C): End-devices of Class C have almost continuously open receive windows, only closed when transmitting.[14]



Class C Device Communication Description. [15]

LoRa **Device Classes Comparision**

Class Type A	**Class Type B**	**Class type C**
Battery Powered	Low Latency	No Latency
Bidirectional with 1 UL+ 2DL Slot	Bidirectional with scheduled Downlink slots	Bidirectional Most of the time listening mode
Unicast messages	Unicast and Multicast messages	Unicast and Multicast messages
• Small Payloads • Long Intervals	• Small Payloads • long Intervals • Periodic Beacon from gateway	• Small payloads
End-device initiates communication (uplink)	Extra receive window	Server can initiate transmission at any time
Server communicates with end-device (downlink) during predetermined response windows	Server can initiate transmission at fixed intervals	End-device is constantly receiving

End Devices Comparison. [15]

2.3 Difference between Lora and LoRaWan

There are many misunderstandings when referring to LoRa and LoRaWan, fact that stress the necessity to shed some light on this problem. LoRa refers to a wireless modulation that enables communication with very low power consumption. LoRaWAN refers to a star network protocol with LoRa chips for communication. It is based on the base station, which can monitor 8 frequencies with several spreading factors and almost 43 channels.

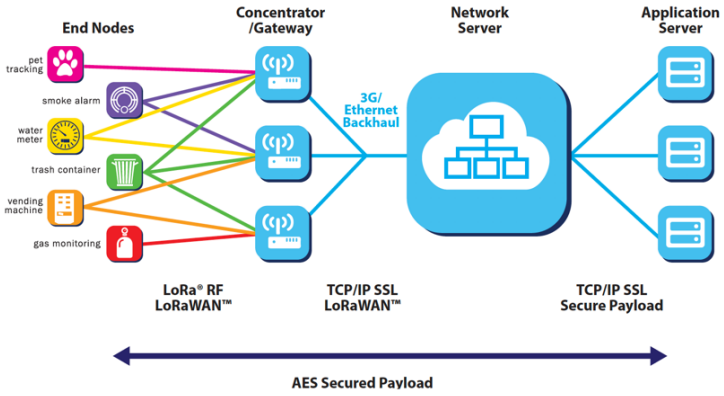
It is possible to use the LoRa modulation as point-to-point or as a star network without LoRaWAN. It might also be possible to use LoRaWAN as a network with other radio links, but this wouldn't be really practical. And even to write your own MAC protocol and use Lora's Radio modulation.

In other words, LoRa is the physical layer: the chip. LoRaWAN is the MAC layer: the software that is placed on the chip to enable networking.

Chapter 3

Lora's Market Applications

LoRa Technology according to semtech[11], has registered over 600 known uses cases for smart cities, smart homes and buildings, smart agriculture, smart metering, smart supply chain and logistics, across the world. With 97 million devices connected to networks in 100 countries and growing, LoRa Technology is the DNA of IoT, creating a Smarter Planet.



A few implementations of Lora standard to left side.[12]

Lora's Use Case's concerns, mainly, the following sectors:[13]

1. **Smart Agriculture**

   Smart Agriculture aims in measuring environmental conditions that influences agricultural production, farmind, cattle ranching, autonomous irrigations in farms.

2. **Smart Cities**

   Smart Cities projects aims in connecting city services such as lighting, parking, waste removal, and more, that enables the time and money economy.

3. **Smart Environment**

   Smart environment projects encompasses projects that garantees citizen's safety from environ mental dangers. Smart flood sensors, Natural Disaster Communication, Water System Monitoring Residential Community Networks are some of the Lora project of this category.

4. **Smart Healthcare**

   LoRa Technology's low power, low cost and reliable performance make it suitable for critical smart healthcare applications. IoT solutions comprised of LoRa-based sensors and gateways can monitor high-risk patients or systems around the clock, ensuring health and medical safety are never overlooked.

5. **Smart Homes and Buildings**

   LoRa Technology's low power qualities and ability to penetrate dense building materials make it an ideal platform for IoT-connected smart home and building devices. In addition, the long range capabilities make it possible for LoRa-enabled sensors and the LoRaWAN® protocol to track assets that stray from home.

6. **Smart Industrial Control**

   Industrial operations can benefit from the deployment of IoT-connected sensors for various always-on monitoring functions. Due to the long range, low power, and long battery life of LoRa®-based devices, sensors in manufacturing plants or mobile industries can relay critical data to a LoRaWAN network where it can be analyzed and businesses operations can be optimized.

7. **Smart Metering**

   Traditional utility operations are labor intensive and utilize subjective measurement by field personnel. Additionally, meters are often located in dense urban environments, indoors or even underground, which can be difficult or impossible to reach by many wireless technologies. By implementing a smart utilities infrastructure comprised of sensors and gateways utilizing LoRa devices and the LoRaWAN protocol, utility

and metering companies can collect data remotely and use personnel more efficiently to streamline operations.

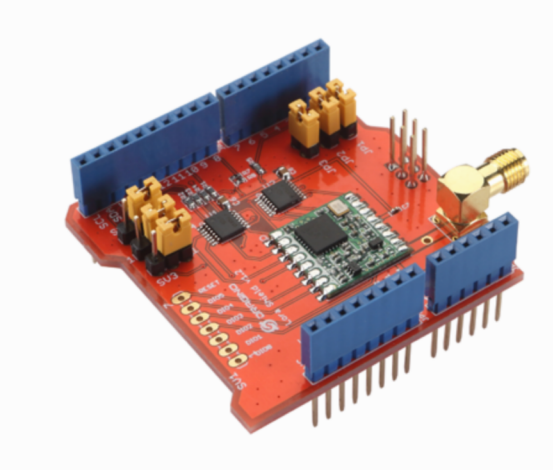
8. **Smart Supply Chain and Logistics**

   LoRa Technology facilitates smart supply chain and logistics to track highly valued assets that are in transit. Due to LoRa Technology's long range and low power consumption qualities and GPS-free geolocation abilities, cargo, vehicles and other assets can be easily monitored over large geographic regions and within harsh environments.

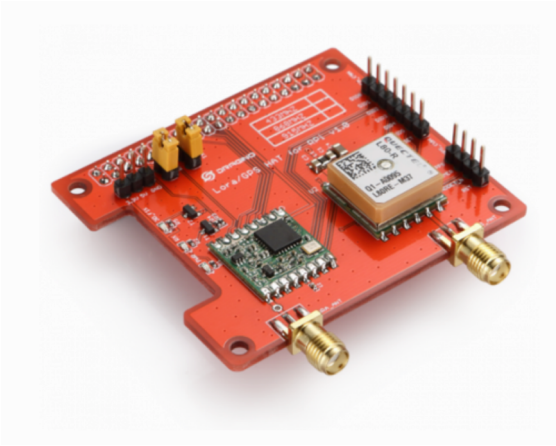
Chapter 4

Commercial Hardware for LoRa Protocol

As it was expected ,the advent of Lora technology, attracted the interest of many hardware manufacturers. At the moment, we can find many modules , that offers communication under Lora's rules. The most popular are modules that can be connected to Arduino and Raspberry pi, and convert them into nodes. Also, well known are the Lopy-pycom nodes. Some of them are shown below:



Arduino Lora shield from Dragino[18]



Raspberry pi Lora shield from Dragino[19]

Pycom module Pycom[20]

These are the most popular Lora modules, there are also other hardware manufacturers that create similar modules. But, all modules, regardless the manufacturer, shares one thing in common, and this is the Semtech's Lora chip.

4.1 Semtech's LoRa Chip

As LoRa modulation is a proprietary protocol, it is expected that all the hardware regarding lora would wear semetch's chips. This chip,in fact, is a transceiver.

More precisely, The SX1276/77/78/79 transceivers feature the LoRalong range modem that provides ultra-long range spread spectrum communication and high interference immunity whilst minimising current consumption.

For the sake of this thesis, SX1276 transceiver was used for Raspberry's Lora Hat .

Table 1 SX1276/77/78/79 Device Variants and Key Parameters

Part Number	Frequency Range	Spreading Factor	Bandwidth	Effective Bitrate	Est. Sensitivity
SX1276	137 - 1020 MHz	6 - 12	7.8 - 500 kHz	.018 - 37.5 kbps	-111 to -148 dBm
SX1277	137 - 1020 MHz	6 - 9	7.8 - 500 kHz	0.11 - 37.5 kbps	-111 to -139 dBm
SX1278	137 - 525 MHz	6- 12	7.8 - 500 kHz	.018 - 37.5 kbps	-111 to -148 dBm
SX1279	137 - 960MHz	6- 12	7.8 - 500 kHz	.018 - 37.5 kbps	-111 to -148 dBm

[21]

For our experiments we used the SX1276 transceiver.

1.1. Simplified Block Diagram

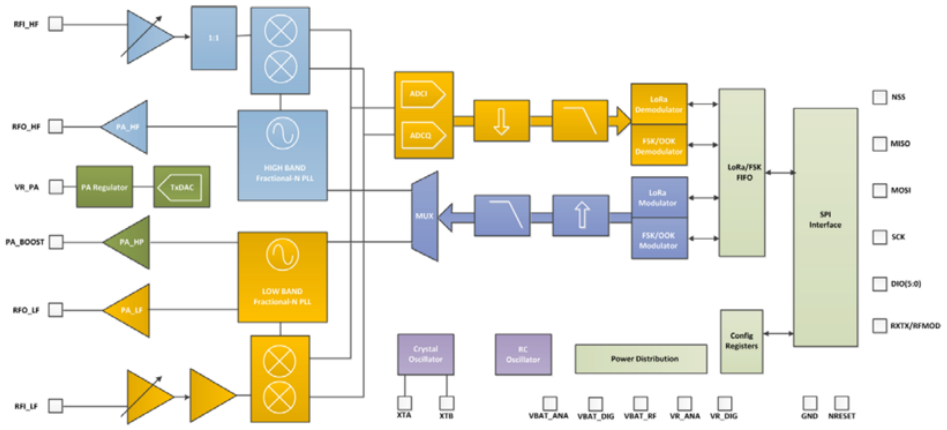


Figure 1. Block Diagram

Simplified Block Diagram of the chip[21]

In fact, SX1276/77/78/79 is a half-duplex, low-IF transceiver.

LNA : LNA is an electronic amplifier that amplifies a very low-power signal without significantly degrading its signal-to-noise ratio. A typical amplifier increases the power of both the signal and the noise present at its input, whereas LNAs are designed to amplify a signal while minimizing additional noise. So at first the received RF signal is first amplified by the

LNA. The LNA inputs of owr tranceivers are single ended to minimize the external BoM and for ease of design.

Mixer Stage : Mixer is responsible the conversion to differential is made to improve the second order linearity and harmonic rejection. The signal is then down-converted to in-phase and quadrature components at the intermediate frequency (IF) .

ADCs : A pair of sigma delta ADCs perform data conversion, with all subsequent signal processing and demodulation performed in the digital domain. The digital state machine also controls the automatic frequency correction (AFC), received signal strength indicator (RSSI) and automatic gain control (AGC). It also features the higher-level packet and protocol level functionality of the top level sequencer (TLS), only available with traditional FSK and OOK modulation schemes.

Frequency synthesizers: generate the local oscillator (LO) frequency for both receiver and transmitter, one covering the lower UHF bands (up to 525 MHz), and the other one covering the upper UHF bands (from 779 MHz). The PLLs are optimized for user-transparent low lock time and fast autocalibrating operation. In transmission, frequency modulation is performed digitally within the PLL bandwidth. The PLL also features optional prefiltering of the bit stream to improve spectral purity.

Modems : The SX1276/77/78/79 are equipped with both standard FSK and long range spread spectrum (LoRa®) modems. Depending upon the mode selected either conventional OOK or FSK modulation may be employed or the LoRa® spread spectrum modem.

SX1276/77/78/79 also include two timing references, an RC oscillator and a 32 MHz crystal oscillator.

SPI interface : All major parameters of the RF front end and digital state machine are fully configurable via an SPI interface which gives access to SX1276/77/78/79's configuration registers. This includes a mode auto sequencer that oversees the transition and calibration of the SX1276/77/78/79 between intermediate modes of operation in the fastest time possible.

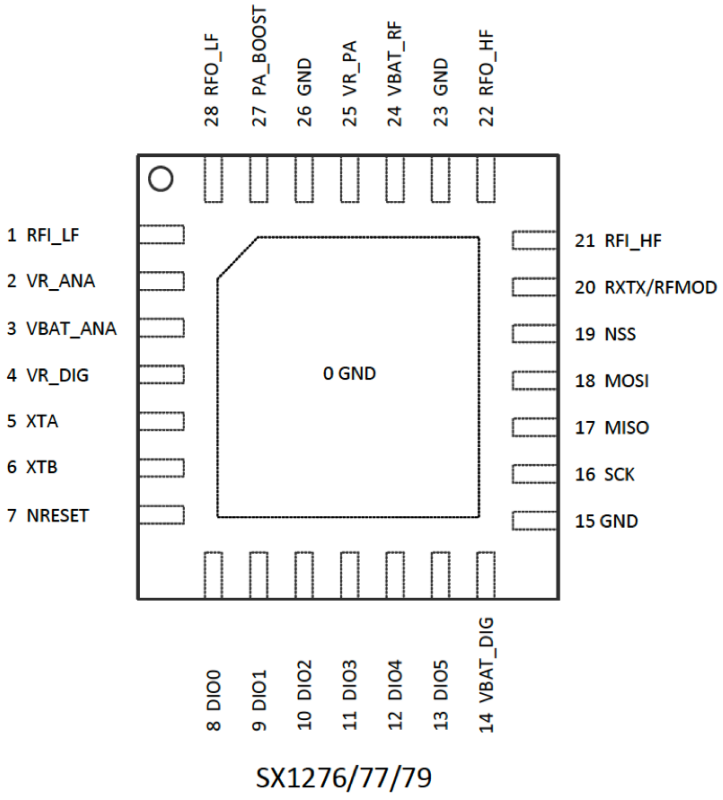
FIFO data buffer : A 256 byte RAM data buffer which is uniquely accessible in LoRa mode. This RAM area, herein referred to as the FIFO Data buffer, is fully customizable by the user and allows access to the received, or

to be transmitted, data. All access to the LoRa® FIFO data buffer is done via the SPI interface.

Pins (General purpose IO)

General purpose IO pins are available used in LoRa® mode. Their mapping is shown below and depends upon the configuration of registers RegDioMapping1 and RegDioMapping2.

The Pin equivalent of the chip:



[21]

Pins description:

1.4. Pin Description

Table 2 Pin Description

Number	Name	Type	Description
	SX1276/77/79/(78)	SX1276/77/79/(78)	SX1276/77/79/(78)
0	GROUND	-	Exposed ground pad
1	RFI_LF	I	RF input for bands 2&3
2	VR_ANA	-	Regulated supply voltage for analogue circuitry
3	VBAT_ANA	-	Supply voltage for analogue circuitry
4	VR_DIG	-	Regulated supply voltage for digital blocks
5	XTA	I/O	XTAL connection or TCXO input
6	XTB	I/O	XTAL connection
7	NRESET	I/O	Reset trigger input
8	DIO0	I/O	Digital I/O, software configured
9	DIO1/DCLK	I/O	Digital I/O, software configured
10	DIO2/DATA	I/O	Digital I/O, software configured
11	DIO3	I/O	Digital I/O, software configured
12	DIO4	I/O	Digital I/O, software configured
13	DIO5	I/O	Digital I/O, software configured
14	VBAT_DIG	-	Supply voltage for digital blocks
15	GND	-	Ground
16	SCK	I	SPI Clock input
17	MISO	O	SPI Data output
18	MOSI	I	SPI Data input
19	NSS	I	SPI Chip select input
20	RXTX/RF_MOD	O	Rx/Tx switch control: high in Tx
21	RFI_HF (GND)	I (-)	RF input for band 1 (Ground)
22	RFO_HF (GND)	O (-)	RF output for band 1 (Ground)
23	GND	-	Ground
24	VBAT_RF	-	Supply voltage for RF blocks
25	VR_PA	-	Regulated supply for the PA
26	GND	-	Ground
27	PA_BOOST	O	Optional high-power PA output, all frequency bands
28	RFO_LF	O	RF output for bands 2&3

[21]

Chapter 5

Experiments

In this chapter we will present a set of a experiments that took place in several places, indoor and outdoor, with purpose to examine the protocol coverage, in different places and ambiances. We raspberry Pis, in order to achieve this goal.

The signal strength indicators that we used were RSSI and SNR.

Received Signal Strength Indicator (RSSI)

RSSI or this signal value is measured in decibels from 0 (zero) to -120 (minus 120). The closer the value to 0 (zero), the stronger the signal will be.

$P_{RX} = P_{TX}xG_{TX}xG_{RX}\lambda 4$

Friis's free space transimission equation.

$$RSSI = 10xlogP_{RX}P_{Ref}$$

Signal to Noise Ratio (SNR)

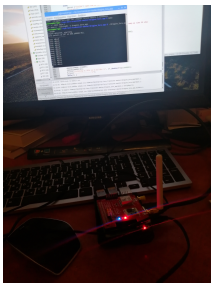
Signal-to-noise ratio is defined as the ratio of the power of a signal (meaningful information) to the power of background noise (unwanted signal):

$$SNR = P_{signal}P_{noise}$$

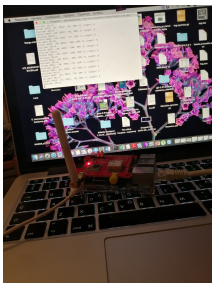
5.1 Raspberry Pi

5.1.1 Indoor Experiment with Raspberry Pi

With this equipment, 2 experiments were performed, one indoor and one outdoor. The coverage was assessed with three different Spreading Factors SF=7, SF=10, SF=12 and two different Transmit Powers TXPWR=17(max) and TXPWR=2(min) in 6 different floors.



(a) Transmitter



(b) Receiver

Figure 5.1: LoRa Transmission

In 6 storey apartment building the Indoor assessment was held with the Transmitter placed in the 6th floor. The receiver was moved thoughtout the floors and presented the following results:

SF=7, TXPW=17:

Floors	RSSI	SNR
6	-95dBm	9
5	-99 dBm	9
4	-97 dBm	8
3	-98 dBm	7
2	-99 dBm	5
1	-96dBm	6
0	-98dBm	3
-1	-99dBm	-4

SF=10, TXPW=17:

Floors	RSSI	SNR
6	-100dBm	8
5	-97 dBm	8
4	-97 dBm	8
3	-96 dBm	8
2	-97 dBm	7
1	-96dBm	3
0	-97dBm	0
-1	-96dBm	-2

SF=12, TXPW=17:

Floors	RSSI	SNR
6	-98dBm	9
5	-97 dBm	8
4	-98 dBm	8
3	-99 dBm	7
2	-101 dBm	7
1	-102dBm	6
0	-102dBm	4
-1	-100dBm	2

SF=7, TXPW=2:

Floors	RSSI	SNR
6	-99dBm	8
5	-96 dBm	7
4	-101 dBm	5
3	-98 dBm	-3
2	-96 dBm	-1
1	-100dBm	-5
0	-103dBm	-6
-1	-100dBm	-9

SF=10, TXPW=2:

Floors	RSSI	SNR
6	-94dBm	9
5	-98dBm	8
4	-97 dBm	6
3	-99 dBm	3
2	-100 dBm	-2
1	-100dBm	-5
0	-100dBm	-7
-1	-96dBm	-9

SF=12, TXPW=2:

Floors	RSSI	SNR
6	-100dBm	8
5	-101dBm	8
4	-98dBm	2
3	-99dBm	6
2	-98dBm	-3
1	-99dBm	-6
0	-100dBm	-5
-1	-99dBm	-9

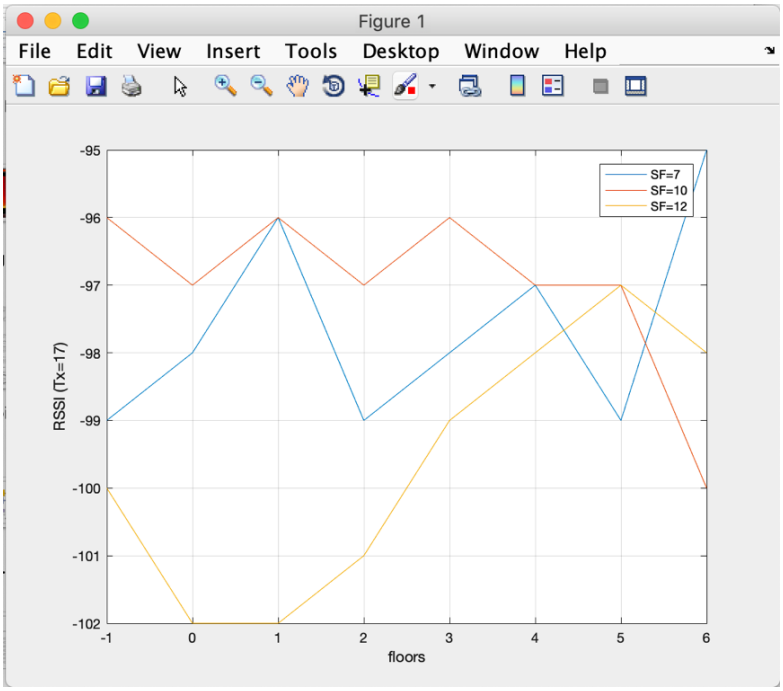


Figure 1
File Edit View Insert Tools Desktop Window Help
SF=7
SF=10
SF=12
RSSI (Tx=17)
floors
-95
-96
-97
-98
-99
-100
-101
-102
-1
0
1
2
3
4
5
6

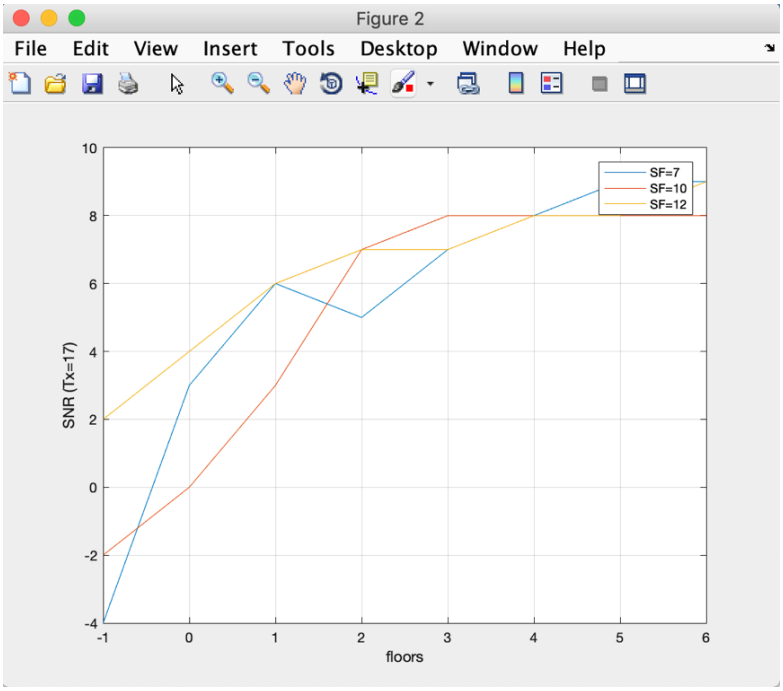
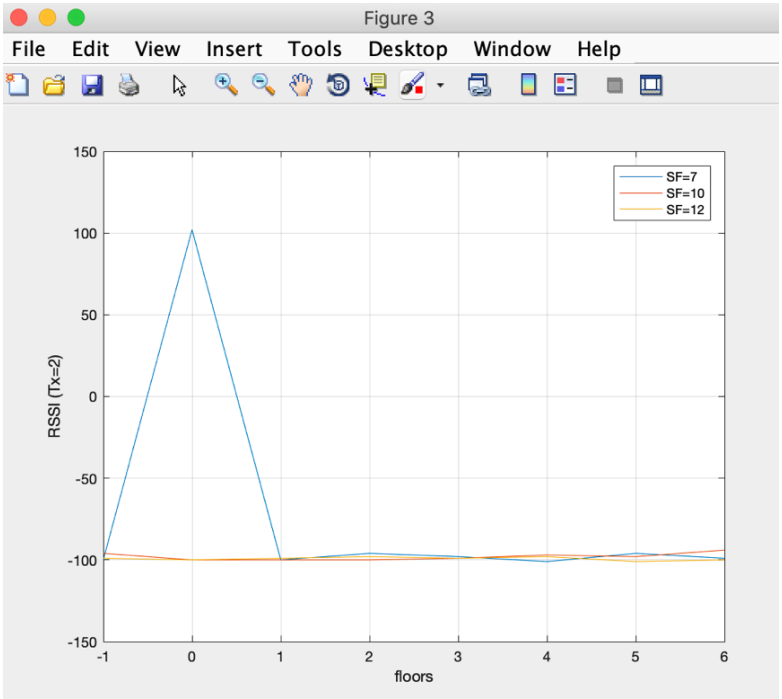
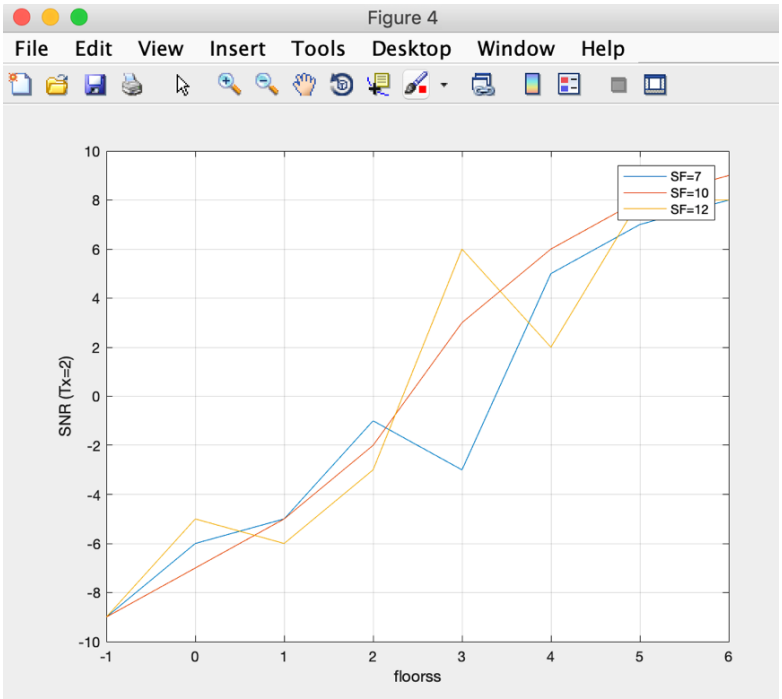


Figure 2
File Edit View Insert Tools Desktop Window Help
SF=7
SF=10
SF=12
SNR (Tx=17)
floors
10
8
6
4
2
0
-2
-4
-1
0
1
2
3
4
5
6





Comments:

As most effective was proved to be the txpw = 17 as a factor, sf seemed not to play an important role, as we had a satisfying coverage with all sf .

With txpw = 2 we also achieved coverage on all floors but not as good snr. If power consumption from 17(max) to 2(min) is not a critical factor for any potential application, as a good combination is proved to be the txpw = 17, sf = 7 .

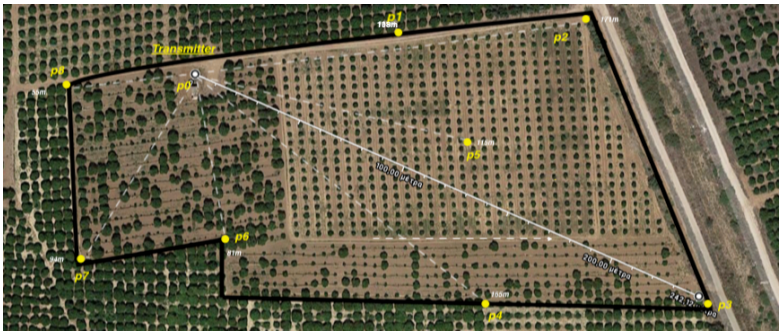
(a) Transmitter

(b) Receiver

Figure 5.2: LoRa Transmission

5.1.2 Outdoor Experiment with Raspberry Pi

The experiment was held in a field of 30000 m^2 with Orange trees. The coverage was assessed with three different Spreading Factors SF=7, SF=10, SF=12 and two different Transmit Powers TXPWR=17 and TXPWR=2 in 8 different.



With the following results:

SF=7, TXPW=17:

Points	RSSI	SNR
P0 (0m)	-98dBm	9
P1 (108m)	-96 dBm	1
P2(171m)	-99 dBm	-7
P3(243m)	-97 dBm	-7
P4(155m)	-99 dBm	0
P5(115m)	-99dBm	8
P6(81m)	-99dBm	8
P7(94m)	-99dBm	-1
P8(55m)	-99dBm	7

SF=10, TXPW=17:

Points	RSSI	SNR
P0	-99dBm	8
P1	-99 dBm	3
P2	-99 dBm	-6
P3	-104 dBm	-7
P4	-98 dBm	3
P5	-98dBm	2
P6	-100dBm	9
P7	-101dBm	8
P8	-94dBm	8

SF=12, TXPW=17:

Points	RSSI	SNR
P0	-102dBm	8
P1	-96 dBm	6
P2	-98 dBm	1
P3	-100 dBm	-8
P4	-98 dBm	4
P5	-98dBm	3
P6	-100dBm	7
P7	-98dBm	8
P8	-96dBm	8

SF=7, TXPW=2:

Points	RSSI	SNR
P0	-99dBm	8
P1	-100 dBm	0
P2	-100 dBm	-3
P3	-	-
P4	-99 dBm	6
P5	-98dBm	-6
P6	-100dBm	3
P7	-102dBm	6
P8	-103dBm	7

SF=10, TXPW=2:

Points	RSSI	SNR
P0	-99dBm	8
P1	-98 dBm	2
P2	-98 dBm	-2
P3	-	-
P4	-99 dBm	0
P5	-100dBm	-6
P6	-103dBm	6
P7	-100dBm	7
P8	-98dBm	8

SF=10, TXPW=2:

Points	RSSI	SNR
P0	-98dBm	8
P1	-97 dBm	0
P2	-100 dBm	-4
P3	-	-
P4	-100 dBm	-3
P5	-97dBm	-6
P6	-98dBm	1
P7	-97dBm	7
P8	-98dBm	8

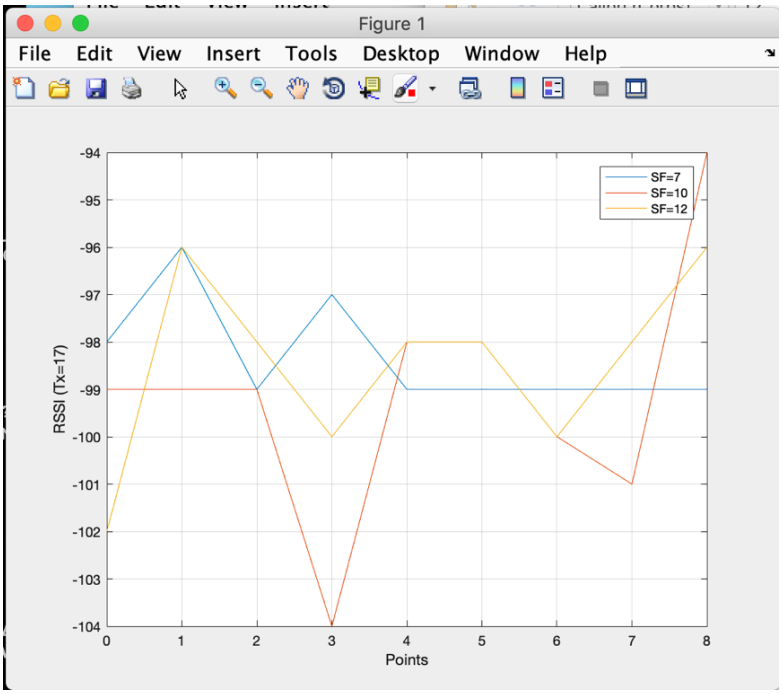


Figure 1
File Edit View Insert Tools Desktop Window Help
SF=7
SF=10
SF=12
RSSI (Tx=17)
Points

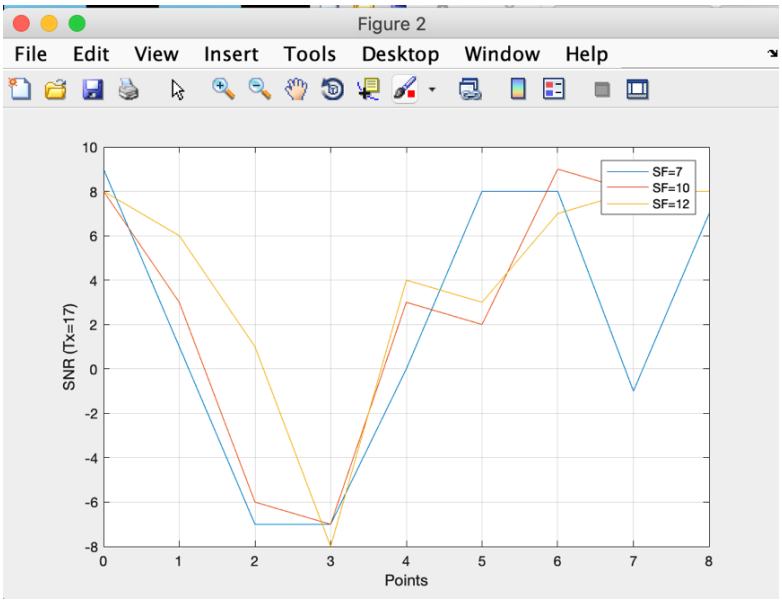
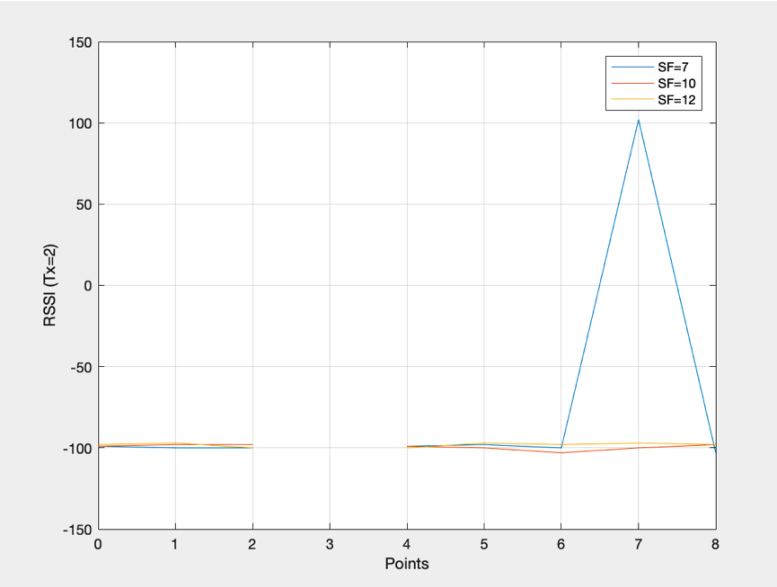
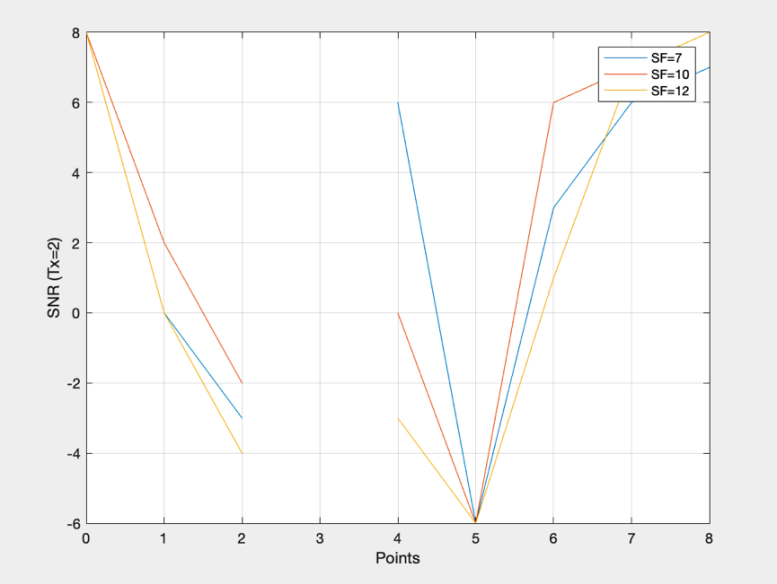


Figure 2
File Edit View Insert Tools Desktop Window Help
SF=7
SF=10
SF=12
SNR (Tx=17)
Points





Comments

During the outdoor experiment, With fixed txpw = 17 with all SF we had a good coverage everywhere, difficult points were considered to be the p3 (about 250m from the transmitter) and p5 which is in the middle of the field between trees (dense vegetation). With fixed txpw = 2 it was observed that none SF was able to reach coveragr to points p3 (it is 250m from the transmitter), however, only SF = 12 seemed to be the most effective.

5.1.3 Code used for Raspberries

The code below has the mission to detect the suitable chip that is used be the LoRa Hat and consequently set the parameters nerded(pins etc.). By changing the SF with the sf_tsf variable and the txpower by the return value of the configPower(23); function we can achieve the transmission parameters needed.

```
/*******************************************************
 *
 * Copyright (c) 2018 Dragino
 *
 * http://www.dragino.com
 *
 *******************************************************

#include <string>
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <string.h>
#include <sys/time.h>
#include <signal.h>
#include <stdlib.h>

#include <sys/ioctl.h>

#include <wiringPi.h>
#include <wiringPiSPI.h>


// ##########################################
// ##########################################

#define REG_FIFO                    0x00
#define REG_OPMODE                  0x01
#define REG_FIFO_ADDR_PTR           0x0D
#define REG_FIFO_TX_BASE_AD         0x0E
#define REG_FIFO_RX_BASE_AD         0x0F
#define REG_RX_NB_BYTES             0x13
#define REG_FIFO_RX_CURRENT_ADDR    0x10
#define REG_IRQ_FLAGS               0x12
```

```
#define REG_DIO_MAPPING_1                0x40
#define REG_DIO_MAPPING_2                0x41
#define REG_MODEM_CONFIG                 0x1D
#define REG_MODEM_CONFIG2                0x1E
#define REG_MODEM_CONFIG3                0x26
#define REG_SYMB_TIMEOUT_LSB                 0x1F
#define REG_PKT_SNR_VALUE                            0x19
#define REG_PAYLOAD_LENGTH               0x22
#define REG_IRQ_FLAGS_MASK               0x11
#define REG_MAX_PAYLOAD_LENGTH               0x23
#define REG_HOP_PERIOD                   0x24
#define REG_SYNC_WORD                            0x39
#define REG_VERSION                                      0x42

#define PAYLOAD_LENGTH                   0x40

// LOW NOISE AMPLIFIER
#define REG_LNA                          0x0C
#define LNA_MAX_GAIN                     0x23
#define LNA_OFF_GAIN                     0x00
#define LNA_LOW_GAIN                         0x20

#define RegDioMapping1                                0x40 // common
#define RegDioMapping2                                0x41 // common

#define RegPaConfig                                   0x09 // common
#define RegPaRamp                                     0x0A // common
#define RegPaDac                                      0x5A // common

#define SX72_MC2_FSK                     0x00
#define SX72_MC2_SF7                     0x70
#define SX72_MC2_SF8                     0x80
#define SX72_MC2_SF9                     0x90
#define SX72_MC2_SF10                    0xA0
#define SX72_MC2_SF11                    0xB0
#define SX72_MC2_SF12                    0xC0

#define SX72_MC1_LOW_DATA_RATE_OPTIMIZE  0x01 //
mandated for SF11 and SF12

// sx1276 RegModemConfig1
#define SX1276_MC1_BW_125                    0x70
#define SX1276_MC1_BW_250                    0x80
```

```
#define SX1276_MC1_BW_500                0x90
#define SX1276_MC1_CR_4_5            0x02
#define SX1276_MC1_CR_4_6            0x04
#define SX1276_MC1_CR_4_7            0x06
#define SX1276_MC1_CR_4_8            0x08

#define SX1276_MC1_IMPLICIT_HEADER_MODE_ON    0x01

// sx1276 RegModemConfig2
#define SX1276_MC2_RX_PAYLOAD_CRCON        0x04

// sx1276 RegModemConfig3
#define SX1276_MC3_LOW_DATA_RATE_OPTIMIZE  0x08
#define SX1276_MC3_AGCAUTO                 0x04

// preamble for lora networks (nibbles swapped)
#define LORA_MAC_PREAMBLE                  0x34

#define RXLORA_RXMODE_RSSI_REG_MODEM_CONFIG1 0x0A
#ifdef LMIC_SX1276
#define RXLORA_RXMODE_RSSI_REG_MODEM_CONFIG2 0x70
#elif LMIC_SX1272
#define RXLORA_RXMODE_RSSI_REG_MODEM_CONFIG2 0x74
#endif

// FRF
#define         REG_FRF_MSB              0x06
#define         REG_FRF_MID              0x07
#define         REG_FRF_LSB              0x08

#define         FRF_MSB                  0xD9 // 868.1 Mhz
#define         FRF_MID                  0x06
#define         FRF_LSB                  0x66

// ----------------------------------------
// Constants for radio registers
#define OPMODE_LORA      0x80
#define OPMODE_MASK      0x07
#define OPMODE_SLEEP     0x00
#define OPMODE_STANDBY   0x01
#define OPMODE_FSTX      0x02
#define OPMODE_TX        0x03
#define OPMODE_FSRX      0x04
```

```
#define OPMODE_RX          0x05
#define OPMODE_RX_SINGLE   0x06
#define OPMODE_CAD         0x07

// ----------------------------------------
// Bits masking the corresponding IRQs from the radio
#define IRQ_LORA_RXTOUT_MASK 0x80
#define IRQ_LORA_RXDONE_MASK 0x40
#define IRQ_LORA_CRCERR_MASK 0x20
#define IRQ_LORA_HEADER_MASK 0x10
#define IRQ_LORA_TXDONE_MASK 0x08
#define IRQ_LORA_CDDONE_MASK 0x04
#define IRQ_LORA_FHSSCH_MASK 0x02
#define IRQ_LORA_CDDETD_MASK 0x01

// DIO function mappings                D0D1D2D3
#define MAP_DIO0_LORA_RXDONE   0x00  // 00------
#define MAP_DIO0_LORA_TXDONE   0x40  // 01------
#define MAP_DIO1_LORA_RXTOUT   0x00  // --00----
#define MAP_DIO1_LORA_NOP      0x30  // --11----
#define MAP_DIO2_LORA_NOP      0xC0  // ----11--

// ########################################
// ########################################
//
typedef bool boolean;
typedef unsigned char byte;

static const int CHANNEL = 0;

char message[256];

bool sx1272 = true;

byte receivedbytes;

enum sf_t { SF7=7, SF8, SF9, SF10, SF11, SF12 };

/*******************************************************
 *
 * Configure these values!
 *
 *******************************************************
```

```
// SX1272 - Raspberry connections
int ssPin = 6;
int dio0  = 7;
int RST   = 0;

// Set spreading factor (SF7 - SF12)
sf_t sf = SF7;

// Set center frequency
uint32_t  freq = 868100000; // in Mhz! (868.1)

byte hello[32] = "HELLO";

void die(const char *s)
{
    perror(s);
    exit(1);
}

void selectreceiver()
{
    digitalWrite(ssPin, LOW);
}

void unselectreceiver()
{
    digitalWrite(ssPin, HIGH);
}

byte readReg(byte addr)
{
    unsigned char spibuf[2];

    selectreceiver();
    spibuf[0] = addr & 0x7F;
    spibuf[1] = 0x00;
    wiringPiSPIDataRW(CHANNEL, spibuf, 2);
    unselectreceiver();

    return spibuf[1];
}
```

```
void writeReg(byte addr, byte value)
{
    unsigned char spibuf[2];

    spibuf[0] = addr | 0x80;
    spibuf[1] = value;
    selectreceiver();
    wiringPiSPIDataRW(CHANNEL, spibuf, 2);

    unselectreceiver();
}

static void opmode (uint8_t mode) {
    writeReg(REG_OPMODE,
    (readReg(REG_OPMODE) & ~OPMODE_MASK) | mode);
}

static void opmodeLora() {
    uint8_t u = OPMODE_LORA;
    if (sx1272 == false)
        u |= 0x8;   // TBD: sx1276 high freq
    writeReg(REG_OPMODE, u);
}


void SetupLoRa()
{

    digitalWrite(RST, HIGH);
    delay(100);
    digitalWrite(RST, LOW);
    delay(100);

    byte version = readReg(REG_VERSION);

    if (version == 0x22) {
        // sx1272
        printf("SX1272 detected, starting.\n");
        sx1272 = true;
    } else {
        // sx1276?
        digitalWrite(RST, LOW);
        delay(100);
```

```
        digitalWrite(RST, HIGH);
        delay(100);
        version = readReg(REG_VERSION);
        if (version == 0x12) {
            // sx1276
            printf("SX1276 detected, starting.\n");
            sx1272 = false;
        } else {
            printf("Unrecognized transceiver.\n");
            //printf("Version: 0x%x\n",version);
            exit(1);
        }
    }

    opmode(OPMODE_SLEEP);

    // set frequency
    uint64_t frf = ((uint64_t)freq << 19) / 32000000;
    writeReg(REG_FRF_MSB, (uint8_t)(frf>>16) );
    writeReg(REG_FRF_MID, (uint8_t)(frf>> 8) );
    writeReg(REG_FRF_LSB, (uint8_t)(frf>> 0) );

    writeReg(REG_SYNC_WORD, 0x34);
    // LoRaWAN public sync word

    if (sx1272) {
        if (sf == SF11 || sf == SF12) {
            writeReg(REG_MODEM_CONFIG,0x0B);
        } else {
            writeReg(REG_MODEM_CONFIG,0x0A);
        }
        writeReg(REG_MODEM_CONFIG2,(sf<<4) | 0x04);
    } else {
        if (sf == SF11 || sf == SF12) {
            writeReg(REG_MODEM_CONFIG3,0x0C);
        } else {
            writeReg(REG_MODEM_CONFIG3,0x04);
        }
        writeReg(REG_MODEM_CONFIG,0x72);
        writeReg(REG_MODEM_CONFIG2,(sf<<4) | 0x04);
    }

    if (sf == SF10 || sf == SF11 || sf == SF12) {
```

```
            writeReg(REG_SYMB_TIMEOUT_LSB,0x05);
        } else {
            writeReg(REG_SYMB_TIMEOUT_LSB,0x08);
        }
        writeReg(REG_MAX_PAYLOAD_LENGTH,0x80);
        writeReg(REG_PAYLOAD_LENGTH,PAYLOAD_LENGTH);
        writeReg(REG_HOP_PERIOD,0xFF);
        writeReg(REG_FIFO_ADDR_PTR, readReg(REG_FIFO_RX_BASE_AD));

        writeReg(REG_LNA, LNA_MAX_GAIN);

}

boolean receive(char *payload) {
        // clear rxDone
        writeReg(REG_IRQ_FLAGS, 0x40);

        int irqflags = readReg(REG_IRQ_FLAGS);

        //  payload crc: 0x20
        if((irqflags & 0x20) == 0x20)
        {
            printf("CRC error\n");
            writeReg(REG_IRQ_FLAGS, 0x20);
            return false;
        } else {

            byte currentAddr = readReg(REG_FIFO_RX_CURRENT_ADDR);
            byte receivedCount = readReg(REG_RX_NB_BYTES);
            receivedbytes = receivedCount;

            writeReg(REG_FIFO_ADDR_PTR, currentAddr);

            for(int i = 0; i < receivedCount; i++)
            {
                payload[i] = (char)readReg(REG_FIFO);
            }
        }
        return true;
}

void receivepacket() {
```

```
    long int SNR;
    int rssicorr;

    if(digitalRead(dio0) == 1)
    {
  if(receive(message)) {
    byte value = readReg(REG_PKT_SNR_VALUE);
    if( value & 0x80 ) // The SNR sign bit is 1
  {
   // Invert and divide by 4
   value = ( ( ~value + 1 ) & 0xFF ) >> 2;
  SNR = -value;
   }
  else
   {
  // Divide by 4
 SNR = ( value & 0xFF ) >> 2;
  }

 if (sx1272) {
   rssicorr = 139;
 } else {
  rssicorr = 157;
  }

 printf("Packet RSSI: %d, ", readReg(0x1A)-rssicorr);
 printf("RSSI: %d, ", readReg(0x1B)-rssicorr);
 printf("SNR: %li , ", SNR);
 printf("Length: %i", (int)receivedbytes);
 printf("\n");
 printf("Payload: %s\n", message);

 } // received a message

} // dio0=1
}

static void configPower (int8_t pw) {
if (sx1272 == false) {
  // no boost used for now
  if(pw >= 17) {
 pw = 15;
 } else if(pw < 2) {
```

```
 pw = 2;
}


  // check board type for BOOST pin
 writeReg(RegPaConfig, (uint8_t)(0x80|(pw&0xf)));
  writeReg(RegPaDac, readReg(RegPaDac)|0x4);

    } else {
  // set PA config (2-17 dBm using PA_BOOST)
  if(pw > 17) {
 pw = 17;
  } else if(pw < 2) {
 pw = 2;
  }
 writeReg(RegPaConfig, (uint8_t)(0x80|(pw-2)));
  }
}


static void writeBuf(byte addr, byte *value, byte len) {
 unsigned char spibuf[256];
 spibuf[0] = addr | 0x80;
 for (int i = 0; i < len; i++) {
 spibuf[i + 1] = value[i];
  }
  selectreceiver();
  wiringPiSPIDataRW(CHANNEL, spibuf, len + 1);
  unselectreceiver();
}

void txlora(byte *frame, byte datalen) {

 // set the IRQ mapping DIO0=TxDone DIO1=NOP DIO2=NOP
 writeReg(RegDioMapping1,
MAP_DIO0_LORA_TXDONE|MAP_DIO1_LORA_NOP|MAP_DIO2_LORA_NOP);
 // clear all radio IRQ flags
 writeReg(REG_IRQ_FLAGS, 0xFF);
 // mask all IRQs but TxDone
  writeReg(REG_IRQ_FLAGS_MASK, ~IRQ_LORA_TXDONE_MASK);

    // initialize the payload size and address pointers
    writeReg(REG_FIFO_TX_BASE_AD, 0x00);
```

```
    writeReg(REG_FIFO_ADDR_PTR, 0x00);
    writeReg(REG_PAYLOAD_LENGTH, datalen);

    // download buffer to the radio FIFO
    writeBuf(REG_FIFO, frame, datalen);
    // now we actually start the transmission
    opmode(OPMODE_TX);

    printf("send: %s\n", frame);
}

int main (int argc, char *argv[]) {

 if (argc < 2) {
 printf ("Usage: argv[0] sender|rec [message]\n");
 exit(1);
}

wiringPiSetup () ;
pinMode(ssPin, OUTPUT);
 pinMode(dio0, INPUT);
 pinMode(RST, OUTPUT);

 wiringPiSPISetup(CHANNEL, 500000);

 SetupLoRa();

 if (!strcmp("sender", argv[1])) {
   opmodeLora();
    // enter standby mode (required for FIFO loading))
    opmode(OPMODE_STANDBY);

    writeReg(RegPaRamp, (readReg(RegPaRamp) & 0xF0) | 0x08);
    // set PA ramp-up time 50 uSec

    configPower(23);

  printf("Send packets at SF%i on %.6lf Mhz.\n",
  sf,(double)freq/1000000);
  printf("------------------\n");

 if (argc > 2)
    strncpy((char *)hello, argv[2], sizeof(hello));
```

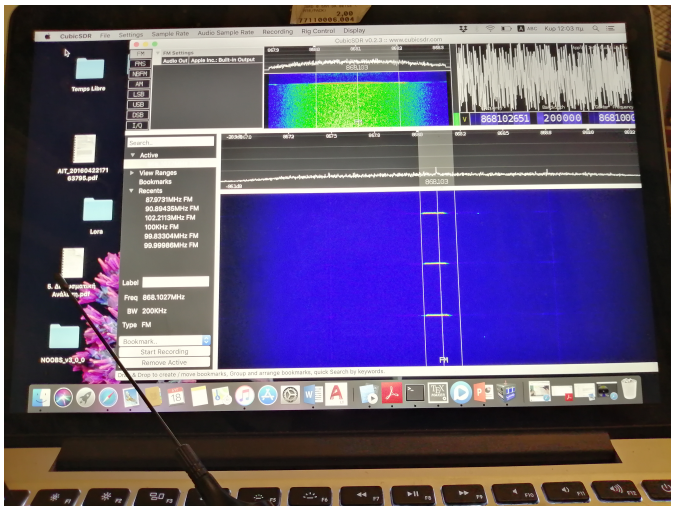
```
  while(1) {
   txlora(hello, strlen((char *)hello));
    delay(5000);
        }
    } else {

 // radio init
  opmodeLora();
  opmode(OPMODE_STANDBY);
  opmode(OPMODE_RX);
  printf("Listening at SF%i on %.6lf Mhz.\n", sf,(double)freq/1000000);
  printf("-------------------\n");
   while(1) {
            receivepacket();
            delay(1);
        }

    }

    return (0);
}
```

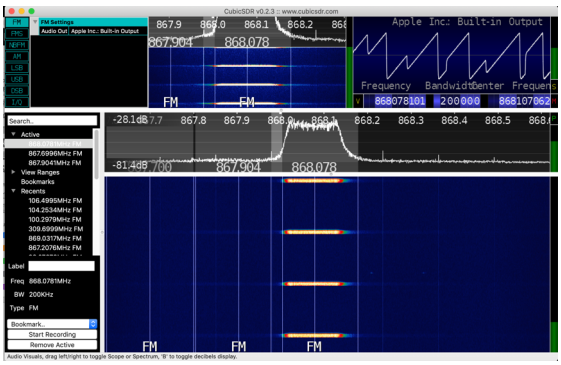
5.2 SDR

Software-defined radio (SDR) is a technique for turning a computer into a radio. But not just an AM/FM radio - by using the computing power on your desktop you can listen and decode a wide variety of broadcasts. SDR can turn your computer into a weather-band receiver, a police/fire report scanner, a music listening station, and more! Instead of manually tuning inductors, its all done in software by chips fast enough to pick up and decode radio waves on the fly.[22]

For this experiment, the equipment that was used was an adafruit sdr the RTL-SDR and the CubicSDR software.

Then as seen in CubicSDR is the appearance of the packet at the appropriate Frequency. So, in 868 MHz we can see a lora packet, in an abstract form. In the black box at right side of the console we can see the chirps form.





Bibliography

[1] *"LoRa protocol analysis and performance evaluation using PyCom equipment"*, Maria Kouvatsou

[2] *"5G Wireless Technologies, Angeliki Alexiou "*, Angeliki Alexiou

[3] T. Adame, A. Bel, B. Bellalta, J. Barcelo, M. Oliver NeTS Research Group, *IEEE 802.11ah: The Wi-Fi Approach for M2M Communications*, Universitat Pompeu Fabra, Barcelona.

[4] Dino Flore, *3GPP Standards for the Internet-of-Things*, Qualcomm Technologies Inc.

[5] Victor Baños-Gonzalez , M. Shahwaiz Afaqui , Elena Lopez-Aguilera and Eduard Garcia-Villegas , *IEEE 802.11ah: A Technology to Face the IoT Challenge*, Sensors

[6] Mahmoud Shuker Mahmoud, Auday A. H. Mohamad, *A Study of Efficient Power Consumption Wireless Communication Techniques/ Modules for Internet of Things (IoT) Applications*, Computer Technology Engineering Department, Al-Mansour University College, Baghdad, Iraq

[7] Usman Raza, Parag Kulkarni, and Mahesh Sooriyabandara, *Low Power Wide Area Networks: A Survey*

[8] Semtech , *AN1200.22 LoRa™ Modulation Basics*, Application Note

[9] Aloÿs Augustin, Jiazi Yi , Thomas Clausen and William Mark Townsley , *A Study of LoRa: Long Range and Low Power Networks for the Internet of Things*, Sensors

[10] *lora-alliance.org*

[11] *https://www.semtech.com*

[12] Technical Marketing Workgroup 1.0 , *A technical overview of LoRa® and LoRaWAN™*, Lora Aliance

[13] *https://www.semtech.com/lora/lora-applications*

[14] Prof. António grilo , *LoRaWAN: An Introduction*, Technico Lisboa

[15] *https://www.techplayon.com*

[16] *https://www.exm.gr*

[17] A. Springer, W. Gugler, M. Huemer, L. Reindl, C.C.W. Ruppel, R. Weigel *Spread Spectrum Communications Using Chirp Signals*, University of Linz, Institute for Communications and Information Engineering, Austria

[18] *https://www.iot-daily.com, LORAWAN VS. SIGFOX VS. WEIGHTLESS-P: SIMULATION RESULTS IN THE "REAL WORLD"*

[19] *https://www.dragino.com*

[20] *https://www.pycom.io*

[21] SEMTECH , *WIRELESS AND SENSING PRODUCTS DATASHEET SX1276/77/78/79 - 137 MHz to 1020 MHz Low Power Long Range Transceiver*

[22] adafruit , *Getting Started with RTL-SDR and SDR-Sharp and CubicSDR*