

# POLITECNICO DI TORINO

Department of ELECTRONICS AND  
TELECOMMUNICATIONS  
(DET)

**Master of Science  
in Communication And Computer Network Engineering**

Master Degree Thesis

## **LoRa: applications and validations in complex urban environment**



### **Supervisors**

Prof. Matekovits Ladislau  
Dr. Allegretti Marco

**Candidate**  
ZhangYongkang

March 2020

# Acknowledgements

Before beginning, I'd like to thank my supervisor Matekovits Ladislau and Allegretti Marco. They helped me so much and gave me a lot of guidance on my thesis. My supervisors are so patient and kindly that they can respond to my questions even in the middle of the night. It was my supervisors who took me into the door of LoRa and Arduino. And finally, I also want to thank the Politecnico di Torino for teaching me a lot.

## Abstract

*LoRa is one of the long range low power wide area network (LPWAN) technologies; it is an ultra-long-range wireless transmission scheme based on spread-spectrum technology adopted and promoted by the American company Semtech. This solution changes the previous compromise between transmission distance and power consumption, and provides users with a simple system that can achieve long distances, long battery life and large capacity, thereby expanding the sensor network. Although LoRa shows good performance for long distance transmission in the countryside or other spacious environments, its radio signals can be interfered by other signal sources or attenuated by walls, trees, buildings etc. This study focuses on transmission performance of LoRa technology performed by measured data at different places with different parameters in complex urban environments. Meanwhile, we combined the LoRa modules with the device Industruino IND.I/O D21G (an Arduino-compatible equivalent of a programmable logic controller) and Ethernet module to get some specific performances.*

*In the beginning, we did some preliminary experiments to be thoroughly familiar with the Arduino (an open-source electronics platform based on easy-to-use hardware and software) environment. The first one is to connect one button with the D21G board, then we uploaded the corresponding program code to the board via Arduino to implement the specific performance as follows. When we press the button, the liquid crystal display (LCD) will turn on and when we raise hand the LCD will turn off. The second performance that we implemented is when we press the button, the LCD will keep bright until we press the button again. Then we move on to three buttons, with a specific sequence to control the LCD “on” and “off”. Then we tried to observe the serial port values from the webpage by using the Ethernet module. After all the works which are mentioned above were finished, we start to perform the validations by evaluating the performances of LoRa in complex urban environments.*

*More specifically, we set up two LoRa modules as transmitters and another LoRa module as receiver. The first transmitter is placed in the office room while the other one is placed on the balcony, transmitting the same data length each time. Then we went outside and hang out and measured the received packets in 14 different places around the transmitters. Finally, the system’s performance of coverage and packet loss rate are analyzed. Since the current popular channel models can not capture signal attenuation well, we would like to focus on the empirical model of Okumura-Hata (analyzing path-loss characteristics based on a large amount of experimental data), which is so far widely used for LoRa. Its estimates is very close to our measurements. By examining a real dataset with tens of thousands of samples, we found that our measurement approach can verify the expected signal link budget within a error of 14 dB. In the concluding section, measurement results are summarized and critically analyzed.*

# Acronyms

<b>LoRa</b>	<b>Long Range</b>
<b>LCD</b>	<b>Liquid Crystal Display</b>
<b>LPWAN</b>	<b>Low Power Wide Area Network</b>
<b>RF</b>	<b>Radio Frequency</b>
<b>IoT</b>	<b>Internet of Things</b>
<b>FSPL</b>	<b>Free Space Path Loss</b>
<b>CSS</b>	<b>Chirp Spread Spectrum</b>
<b>FSK</b>	<b>Frequency-Shift Keying</b>
<b>OOK</b>	<b>On-Off Keying</b>
<b>GMSK</b>	<b>Gaussian Filtered Minimum Shift Keying</b>
<b>MAC</b>	<b>Media Access Control</b>
<b>TDD</b>	<b>Time Division Duplex</b>
<b>CRC</b>	<b>Cyclical Redundancy Check</b>
<b>USB</b>	<b>Universal Serial Bus</b>
<b>TTL</b>	<b>Transistor-Transistor Logic</b>
<b>PLC</b>	<b>Programmable Logic Controller</b>
<b>MCU</b>	<b>Microprogrammed Control Unit</b>
<b>IDE</b>	<b>Integrated Development Environment</b>
<b>EIRP</b>	<b>Effective Isotropic Radiated Power</b>

# CONTENT

<b>1</b>	<b>Introduction</b>	
1.1	Problem description.....	7
1.2	Structure of the thesis.....	7
1.3	LoRa.....	8
1.3.1	Overviews of LoRa technology.....	8
1.3.2	Modulation .....	9
1.3.3	The networks of LoRa.....	10
1.3.4	The structure and format of LoRa packet.....	11
1.3.5	Why environment is important.....	13
1.3.6	Applications of LoRa.....	14
1.3.7	Limitations.....	16
1.4	Arduino.....	16
1.5	Industruino IND.I/O D21G.....	18
<b>2</b>	<b>Background</b>	
2.1	Empirical propagation models.....	21
2.1.1	Okumura propagation model.....	21
2.1.2	Okumura-Hata model.....	27
2.1.2.1	Okumura-Hata model for urban environment.....	27
2.1.2.2	Okumura-Hata model for suburban and open environment	28
2.1.2.3	Limitations for Okumura-Hata model.....	29
2.1.2.4	Some other main models based on published data.....	30
2.2	The LoRa link budget in theory.....	30
<b>3</b>	<b>Experimental setup</b>	
3.1	Dataset.....	32
3.1.1	Parameters of LoRa.....	32
3.1.2	Annotation.....	34
3.2	Experimental Process.....	35
3.2.1	Initial trial with Industruino hardware.....	35

3.2.2	Initial trial modification.....	36
3.2.3	Button counting trial with Industruino hardware.....	38
3.2.4	Data reading trial.....	41
3.2.5	The combination of LoRa and D21G controller.....	47
3.2.6	Measurement of LoRa in urban environment.....	48
<b>4</b>	<b>Measurement results</b>	
4.1	The first measurement campaign performance.....	55
4.2	The second measurement campaign performance.....	57
4.3	The third measurement campaign performance.....	58
4.4	Comparison among the three campaigns.....	60
<b>5</b>	<b>Evaluation</b>	
5.1	Link budget.....	63
5.2	Okumura-Hata model.....	65
5.3	Analysis of measurement results.....	66
<b>6</b>	<b>Conclusions and outlook.....</b>	<b>69</b>

# Chapter 1

## Introduction

### 1.1 Problem description

In this thesis, our interest is measuring LoRa signal performance in complex urban environment and implementing an interactive interface between the Industruino IND.I/O D21G and other devices. The programs are run and the related data are collected on the platform of Arduino. Arduino is a platform which is popular with electronic makers and offers plenty of flexibility through an open source system. At first, just some easy button experiments have been done to know how the Industruino IND.I/O D21G and the Arduino work. One of the problem is that we have to combine the D21G with LoRa modules; finally the communication between D21G and PC with LoRa modules have been implemented successfully.

In the meanwhile, in order to measure the signal performance of LoRa, another important problem is that we have to use another three UNO board devices. When they are combined with the LoRa modules, two of them are used as transmitters and another one is used as receiver. The problem is to correctly set up parameters of LoRa modules and only in this way they can communicate successfully with each other .

Above all, the thesis is going to solve the two basic core problems, which are combining the LoRa modules with D21G board or UNO boards to implement communications with PC or other devices and finally measure the signal performance.

### 1.2 Structure of the thesis

In the first chapter, we first introduce the problems studied in this thesis and their significance. The problems we study are closely related to LoRa with Arduino platform and Industruino IND.I/O D21G device. So next we outline the development and the issues studied in these areas. We did not discuss in detail, but only let the readers have a simple understanding of the problems studied in these areas.

The second chapter focuses on the theoretical knowledge of the methods used in this work. In order to compute the final link budget, we will study with the Okumura-Hata empirical model. The focus of this chapter is on the process on how to get the final link budget used in this study.

The third chapter describes the experimental process. Since we made our own dataset instead of the public one, we have not only discussed the details of the experiment and parameter settings, but also the process of making the dataset.

In the fourth and fifth chapter, we show and evaluate the experimental results. We start with numerical values and images and evaluate our models from the perspective of average precision. Finally in the last chapter, we summarize the finding of our study.

## 1.3 LoRa

### 1.3.1 Overviews of LoRa Technology

LoRa (which stands for long range) is a low-power wide-area network (LPWAN) technology. It is based on spread spectrum modulation techniques derived from chirp spread spectrum (CSS, a spread spectrum technique that uses wide band linear frequency modulated chirp pulses to encode information) [1]. This modulation technique can use the whole allocated bandwidth to broadcast a signal, which makes LoRa robust to channel noise. It was developed by Cycleo of Grenoble, France and acquired by Semtech the founding member of the LoRa Alliance [2]. Nowadays, a variety of IoT (the extension of the Internet from people to things) applications are changing our life style all over the world, with the development of LPWAN. And it is estimated that the number of Internet of Things devices will exceed 50 billion in 2020 [3]. But the most typical devices related to IoT scenario are used with limited energy, which need wireless links to connect them to the Internet. Under such situation, LPWAN is also developing with the aim of offering low data rate between communication of long range. Among the current communication systems, LoRa is one of the most outstanding technology with good performances. Due to its unique technical features, it is the earliest low cost implementation for commercial use and it is becoming more and more versatile in our daily life.

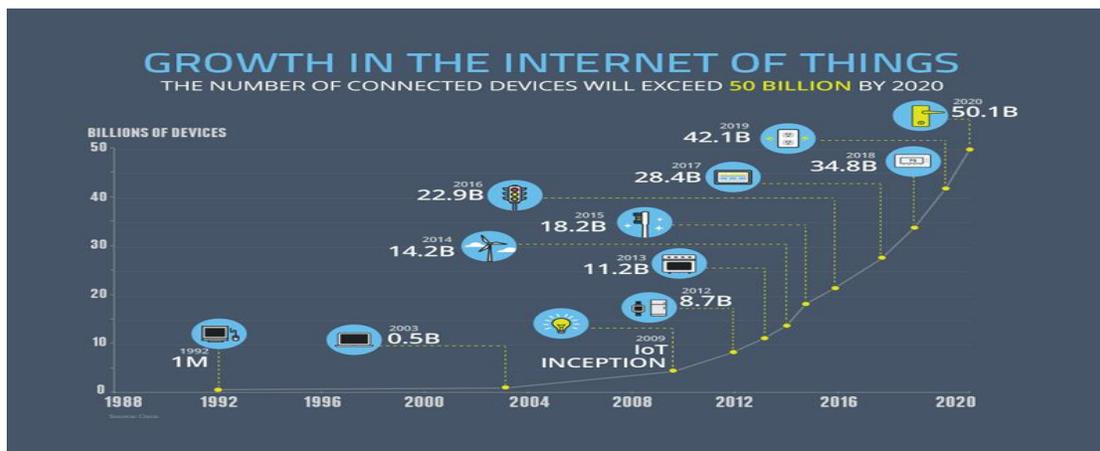


Figure 1.3.1 IoT market growth [3]

LoRa operates in three most commonly unlicensed Scientific and Medical (ISM) bands like 433 (Asia), 868MHz (Europe) and 915MHz (Australia and North America) and allows the long distance communication among low power devices [4].

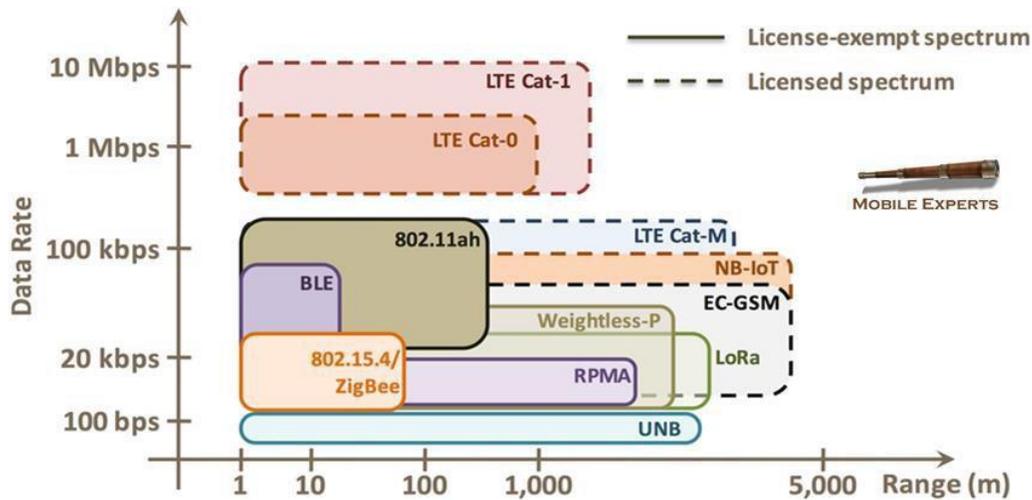


Figure 1.3.2 Range vs. Date rate for IoT connectivity technologies [5]

### 1.3.2 Modulation

LoRa modem uses spread spectrum modulation and forward error correction technology, which combines digital spread spectrum, digital signal processing and forward error correction. The CSS is an old modulation technique developed in 1940 which was originally used for military communication [6].

Compared with traditional FSK (use carrier frequency changes to deliver digital information) and OOK (use unipolar non-return-to-zero code sequence to control the on and off of the sine carrier) modulation technology, LoRa has expanded the coverage of wireless communication links. Another important feature of LoRa modem is that it has stronger anti-interference ability with respect to these traditional modulation technologies. The ability to suppress co-channel GMSK (pre-modulation filtering by a Gauss filter before the data stream is sent to the frequency modulator) interference signals can reach to 20 dB. With such a strong interference resistance ability, LoRa can be also used in hybrid communication networks, in order to expand coverage when the original modulation scheme in the network fails. With the CSS modulation, LoRa uses a high spreading factor to achieve higher signal gain. Generally, the signal-to-noise ratio of FSK needs 8 dB, while LoRa only needs -20 dB.

And with the forward error correction, it adds redundancy to transmission information to effectively resist to multipath fading. Although some transmission efficiency is sacrificed, the transmission reliability is effectively improved. After all, LoRa does not need a high transmission rate.

### 1.3.3 The networks of LoRa

The LoRa network consists of a gateway, a terminal device and a web sever. It's network topology is star structure. Generally speaking, transmission from a terminal device to a gateway is called "uplink", and transmission from a gateway to a terminal device is called "downlink". Based on the MAC layer, there are three types of terminal devices in the LoRa network. These classes are defined as classes A, B and C [7].

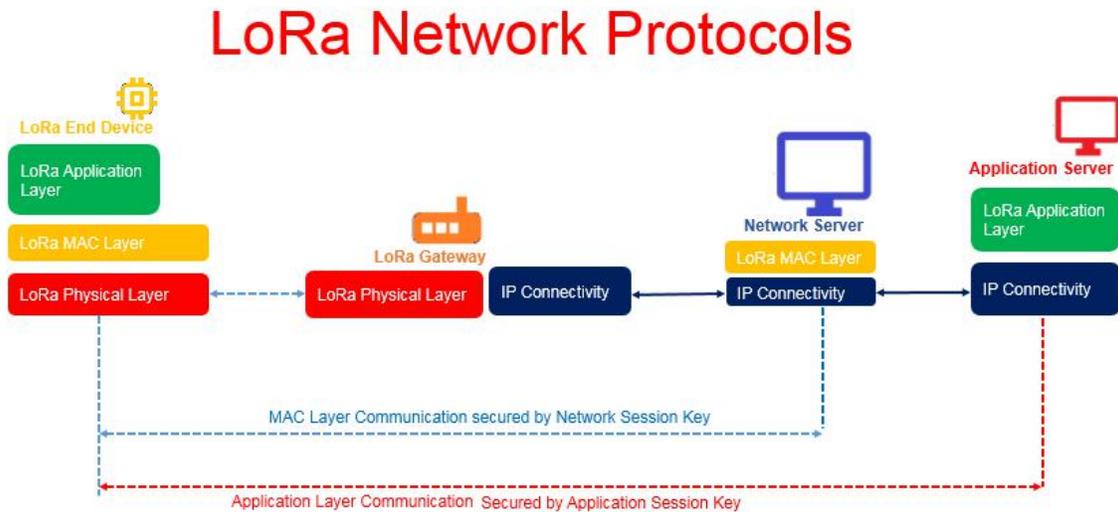


Figure 1.3.3.1 LoRa network protocols [8]

In class A, LoRa frames have one uplink time slot, followed by two downlink time slots. This frame conforms to the TDD (a technique for timely distinguishing wireless channels and continuing uplink operation in downlink operation of a frame period) topology. In general, frames are divided into uplink transmissions and downlink transmissions. The terminal equipment shall be arranged by the terminal equipment according to requirements. It is randomly determined, similar to the ALOHA (random access or contention sending ) protocol [9], and it has the lowest power consumption.

As for class B, beside the two time slots specified in Class A, this type of terminal device uses an additional receive window during the downlink. It will get an additional receive window for a specified duration, and the duration is specified by the gateway using beacon frames. Therefore, in this way, the LoRa system can indicate to the server when the terminal device can listen.

In class C, except for the transmission mode, this terminal equipment can always monitor. Therefore, class C terminal equipment will use more power than the counterparts A and B. Latency is the lowest among all the three types terminal devices which communicate data between the server and the terminal devices.

### 1.3.4 The format of LoRa frame

There are two types of packet formats used in LoRa, the name of which are explicit mode and implicit mode [10]. The header of explicit data packet is short, which mainly contains information such as the number of bytes, the encoding rate, and whether to use CRC or not. The LoRa packet contains Preamble, Header and Payload, which are shown in Fig. 1.3.4.1.

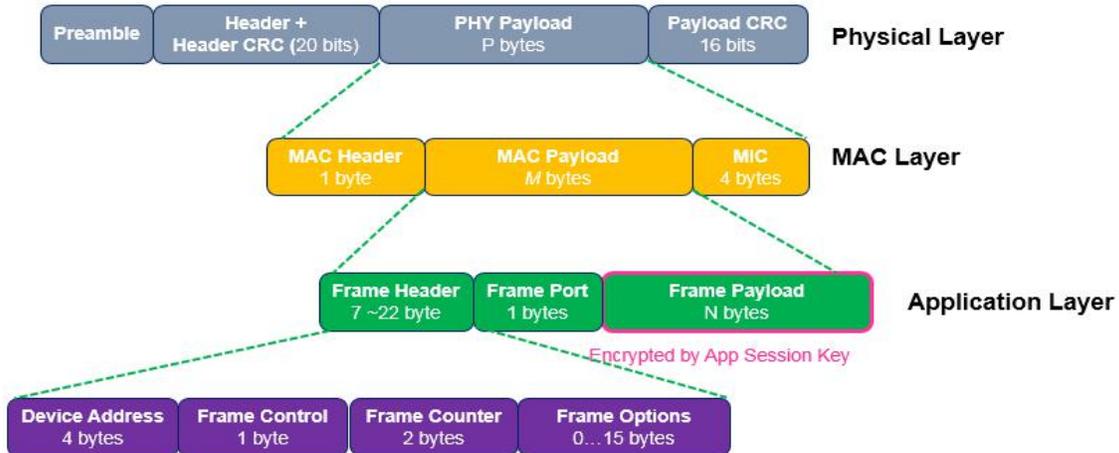


Figure 1.3.4.1 LoRa frame format [8]

The preamble is used to keep the receiver synchronized with the incoming data stream. It can remind the receiving chip that it is the valid signal that is going to be sent, pay attention to the reception to avoid losing the useful signal. After preamble is sent, the valid data will be sent immediately. The default preamble data size is 12 symbols in length, but it can be expanded according to the actual application. Receiver will detect the preamble periodically, therefore, the length of the preamble at the receiving and transmitting terminals must be the same. If it is unknown, the preamble length of the receiver should be set to the maximum value.

The default header in LoRa is explicit mode. In this mode, the header will contain information about the payload such as payload length, forward error correction coding rate and whether to use CRC (16 bits). The header is sent according to the maximum error correction code (4/8). In addition, the header also contains its own CRC, and the receiver can check this first to discard invalid header data packets.

Preamble	PHDR	PHDR CRC	PHYPayload					CRC*
			MHDR	MACPayload				MIC
			FHDR			FPort	FRMPayload	
			DevAddr	Fctrl	Fcnt	Fopts		

Figure 1.3.4.2 LoRa frame format [11]

The payload of the data packet is a field with unfixed length. The actual length and encoding rate CR can be determined by the header in the explicit header mode or determined by the register setting in the implicit mode.

In Fig. 1.3.4.2, PHDR, PHDR\_CRC and load CRC are joined via RF transceiver, and they are all determined by hardware. As for MHDR, it is only one byte, specifying the message type (MType) and the major version number (Major) of the LoRaWAN specification that the frame encoding follows.

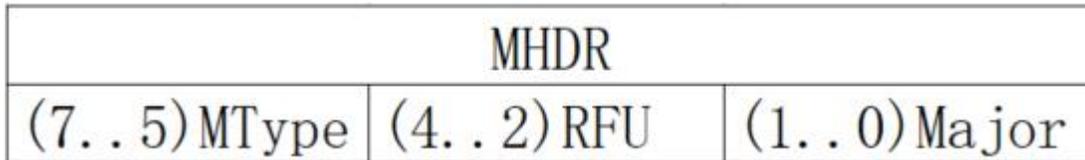


Figure 1.3.4.3 MHDR in PHY Payload [11]

Among them, (7..5) MType specifies the type of message:

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 1.3.4.4 The corresponding message type in MType [11]

(4..2) RUF is the reserved bit and (1..0) Major is shown in Fig. 1.3.4.5:

00	LoRaWAN R1
01..11	RFU

Figure 1.3.4.5 The corresponding major version number of LoRaWAN [11]

MIC is Message Integrity Code with four bytes, it is used to ensure that the data sent and received are consistent. And FHDR means Frame Header, it consists of four issues as shown in Fig. 1.3.4.2.

DevAddr : Four bytes, short terminal address, randomly determined by LoRa server / developer, one terminal address per device.

Fctrl : One byte, uplink and downlink are different. The difference is shown in Fig. 1.3.4.6.

Uplink:

<b>Bit#</b>	7	6	5	4	[3..0]
<b>FCtrl bits</b>	ADR	ADRACKReq	ACK	RFU	FOptsLen

Downlink:

<b>Bit#</b>	7	6	5	4	[3..0]
<b>FCtrl bits</b>	ADR	ADRACKReq	ACK	FPending	FOptsLen

Figure 1.3.4.6 The FCtrl bits in uplink and downlink [11]

Among them, ADR is used to control adaptive data rate. RFU is radio frequency unit. FPending is only used in downlink interaction, indicating that the gateway still has pending data waiting to be delivered, and the terminal needs to send upstream message as soon as possible to open another receiving window. And FOptsLen means that if FOptsLen is 0, FOpts is empty; when FOptsLen is not 0, FOpts is not empty.

Fcnt : Contains uplink and downlink counters.

Fopts : it stores MAC commands, the maximum length is 15 bytes.

Besides FHDR in MAC Payload ,there are also a FPort and a FRMPayload.

FPort : The port field cannot be left blank. The value range is 1 ~ 223.

FRMPayload : MAC frame payload encryption.

### 1.3.5 Why environment is important?

Although LoRa has many special advantages, there are still many challenging issues in practical applications. The environment has a great influence on the performance of wireless link, and it is the same for LoRa technology. Attenuation and distortion of the

wireless signal can have huge dynamics in different situations and scenes. Unfortunately, even there is plenty of anecdotal evidence about LoRa links being much shorter than claimed, channel models and empirical evidence are still largely lacking in the literature, with the exception of a few preliminary words [12, 13, 14, 15].

One reason is that we have to accumulate necessary experience and knowledge, even for a well defined deployment, the performance may not be good enough, because sometimes LoRa needs high requirements. Where there are many interference sources, such as metal and building intensive scenes, they will greatly affect the propagation quality of signals. And even for the power supplier, we must select the appropriate DC regulated current during the work process. If the power is too small, it will cause insufficient and shorten the transmission distance; and if it is too large, it will cause burnout of the LoRa wireless module. It is better to choose a power supply that can resist to interferences, with small ripple and strong load capacity. Excessive power ripple can also interfere with the transmission distance of the LoRa wireless module. The harsh working environment will affect the use of LoRa wireless modules, too. For example, too high or too low temperature will affect the service life of LoRa wireless modules. Furthermore, we should try to avoid using the LoRa module in humid air, since air humidity will affect the transmission distance of the wireless module. Unfortunately, in this thesis, there was a rainy season during our measurements which also affected our measurement results.

### **1.3.6 Applications of LoRa**

In recent years, the development of LoRa wireless technology has become more and more mature, and a complete ecosystem of IoT (the extension of the Internet from people to things) applications has gradually been pieced together. LoRa is a flexible and autonomous network that can be deployed wherever needed, and enterprises (even individuals) can become "operators". The technology can be used for many different applications such as smart city, smart industry, smart agriculture, smart buildings, smart logistics etc.

In the following two applications are applied to our daily life :

#### **A. LoRa for Smart Parking**

Nowadays, the number of cars are increasing day by day, and it becomes more and more difficult to find a parking space. Actually there are so many times that we don't even notice the available parking space because of the tightly placed vehicles around the spot. Smart parking systems are developed to help the drivers to solve this problem. Among these technologies, LoRa technology is one of the most efficient way for smart parking. Parking spaces can be monitored and managed more efficiently in this system and the city can monitor available spots to ensure ambulance, fire, and police services are always guaranteed.



Figure 1.3.6-a LoRa for smart parking system [16]

If there is no parking place around, the LoRa sensor can detect it and tell the incoming drivers with timely feedback. Therefore, it can reduce much more time for drivers who are looking for a parking spot. It can also monitor the whole parking lot in real time and offer available spots for people. With this convenient system we can even reduce more pollution of cars as it is.

## B. LoRa for Smart Agriculture



Figure 1.3.6-b LoRa for smart agriculture [16]

Modern agriculture relies more and more heavily on the ability to effectively manage resources to reduce the impact on the environment, minimize costs and maximize yields. IoT devices provide the ability to monitor crops and animals automatically, they can provide useful data that has traditionally been collected manually, and the ability to control systems and equipment throughout the operation.

For example, automated monitoring and maintenance models can be used to replace a large amount of manpower. Farmers can observe the necessity of irrigation based on weather forecasts, irrigation detection, humidity, etc. [16], while they can also observe the default values of the irrigation system. This avoids time-consuming on-site inspections of the system and potential crop losses.

### **1.3.7 Limitations**

Like all other technologies, LoRa has its own limitation too. Some of the imitations are listed below:

- A. It is not suitable for real-time communication applications which require very low latency.
- B. Spectrum interference. With the increase of LoRa equipment and network deployment, there will be some spectrum interference between them.
- C. LoRa technology is concentrated on Semtech too much. It is not conducive to the development of the entire industry.
- D. It needs to create a new network. In its network deployment boundary area, there may be only a small number of user equipment, but in order to achieve full coverage, base station deployment must be increased. These will cause problems such as increased network construction costs, difficult construction, and difficult maintenance.

## **1.4 Arduino**

Arduino is a convenient and flexible open source electronic prototype platform. It contains hardware (Arduino boards of various models) and software (Arduino IDE). Developed by a European development team in the winter of 2005 [17].

During this thesis, Arduino is used to develop the system, by using SX1276 chip, UNO board, Industruino IND.I/O D21G microcontroller board and USB to TTL serial converter. The programs are written and run on this platform, and then are loaded to the boards which controls the communications of different modules. Then the data can be monitored in the serial port window. If there is any problem, the platform will show it in details and data on the interface for the convince to check, fix, debug or build.

Here is the Arduino UNO board that we have used during our experiments:

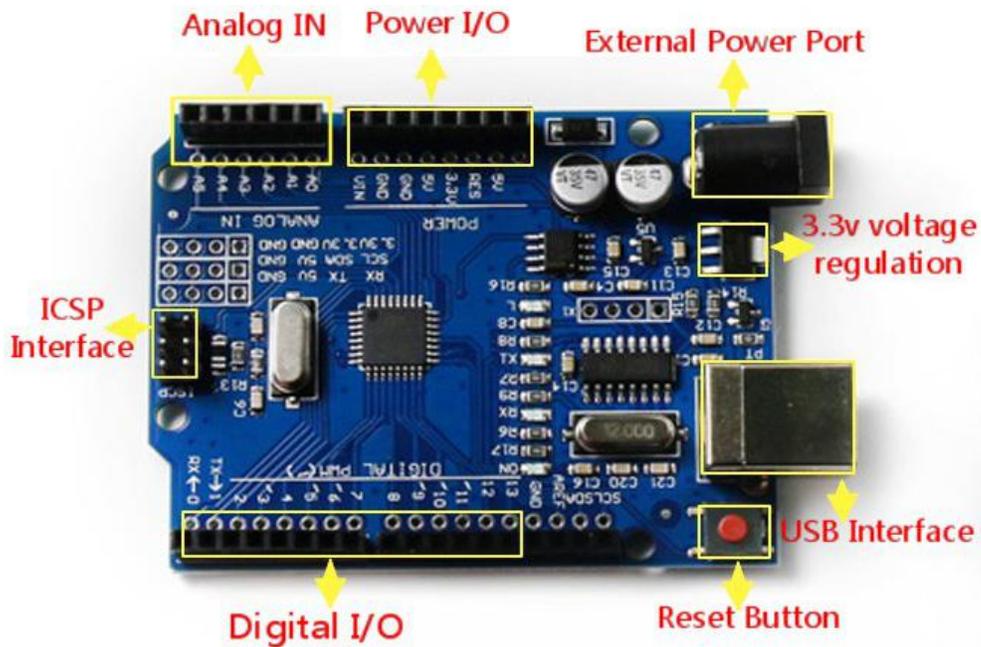


Figure 1.4.1 Arduino Board UNO-R3

And Table 1.4.1 shows the corresponding parameters of Arduino board:

Table 1.4.1 The corresponding parameters of Arduino board UNO-R3

Microprocessor controller	ATMEGA328P
Digital I/O	0~13
Analog I/O	0~5
USB Supply Voltage	5 V
External Supply Voltage	5 V~9 V
Maximum External Power Current	1 A
Maximum I/O Port Current	40 mA
FLASH Memory Capacity	32 KB
SRAM	2 KB
Clock Frequency	16 MHz
Size	75×54×15 mm
USB to Serial Chip	CH340

## 1.5 Industruino IND.I/O D21G

The Industruino IND.I/O D21G controller is a robust DIN-rail mountable Arduino IDE compatible controller with isolated level I/O capability [18].

IND.I/O is one of the PLC equivalents of an Arduino derived controller which is studied and developed in this thesis. It allows us to use the Arduino language for professional automation projects without sacrificing reliability, stability.

As can be observed in Fig. 1.5.1, there are two main parts inside it. The first one is the base board which has the connectors and the prototyping area and the second one is the top board which is the main controller. The two boards are connected to each other with connector. What's more, the top board is a completely self-contained controller board and has an on board LCD screen.

On the bottom side we can find the main components and on the middle of prototyping area we can put our own components and connect it to the external screw connectors.

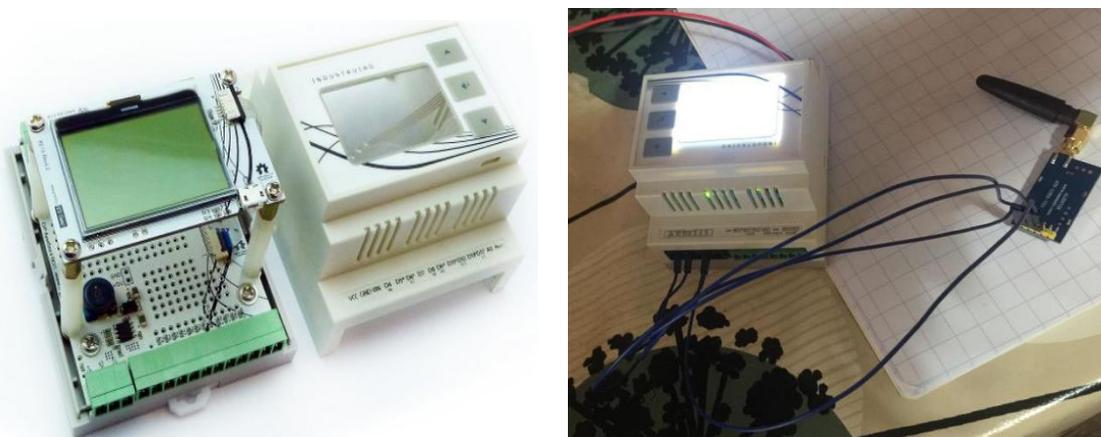


Figure 1.5.1 Industruino IND.I/O D21G [18]

The interface board offers isolated power zones and RS485 transceiver. With 8 channels of 24 V I/O, 4 channels of 0-10 V/4-20 mA 18 bit ADC, 2 channels of 0-10 V/4-20 mA 12 bit DAC.

Updating the code is very easy with this device. We can just plug in the micro USB connection, go to the Arduino Tools menu and select the appropriate serial port, then choose our corresponding boards. When we press the upload button the LCD screen will be flashing which shows that the upload process is taking place. And the on-board three button membrane panel and screen allows you to make your own graphical user interface. The following figures shows the configuration parameters of IND.I/O D21G:

Digital inputs	
Number of digital inputs	8 (shared with digital outputs)
Type of digital input	Galvanically isolated serializer with interrupt
Input voltage range	0-28V
Logic HIGH voltage	>11V
Logic LOW voltage	<3V
Maximum trigger frequency	10 KHz
Protection of digital outputs	Short-circuit, over-current, over-temperature, ESD, transients.
Digital outputs	
Number of digital outputs	8 (shared with digital outputs)
Type of digital output	Galvanically isolated high-side driver (Charge pump NFET)
Output voltage range	Tied to supply voltage (6.5-32V)
Maximum current per output	2.6 A
Maximum total current	6.5 A
Maximum switching frequency	400 Hz
Protection of digital outputs	Short-circuit, over-current, over-temperature, ESD, transients.
Analog inputs	
Number of analog inputs	4
Type of analog inputs	Buffered ADC
Range of voltage measurement	0-10V
Range of current measurement	0-20mA
Switching of voltage / current mode	Automatic - in software
Resolution	18Bit
Conversion rate	18bit: 3.75 Hz - 16bit: 15 Hz - 14bit: 60 Hz - 12bit: 240 Hz
Protection of analog inputs	ESD, transients.
Analog Outputs	
Number of analog outputs	2
Type of analog outputs	Buffered DAC
Range of output voltage	0-10V
Range of output current	0-20mA
Switching of voltage / current mode	Automatic - in software
Resolution	12Bit
Update rate	20 KHz
Protection of analog outputs	Short-circuit, over-current, over-temperature, ESD, transients.
Installation	
Mounting	on 35 mm DIN rail, 4 spacing units wide
Supply voltage	
Standard input voltage	12V / 24V
permissible range, lower limit (DC)	8 V
permissible range, upper limit (DC)	28 V

Figure 1.5.2 configuration parameters of IND.I/O D21G [18]

Communication ports	
<b>RS485</b>	
Isolation topology	Isolated from MCU and analog field section
Duplex type	Half duplex
Number of receivers on bus	32
Data rate	1 Mbps
<b>Expansion port (direct MCU control)</b>	
Isolation topology	Isolated from digital and analog field section
Number of pins	14
Voltage level	5V
Protocols supported	SPI, I2C, UART, 9 GPIO's
Protection of expansion port	ESD, transients.
User Interface	
LCD	128x64 pixel FSTN with dimmable backlight
Push buttons	3 - push button membrane panel

Figure 1.5.3 configuration parameters of IND.I/O D21G [18]

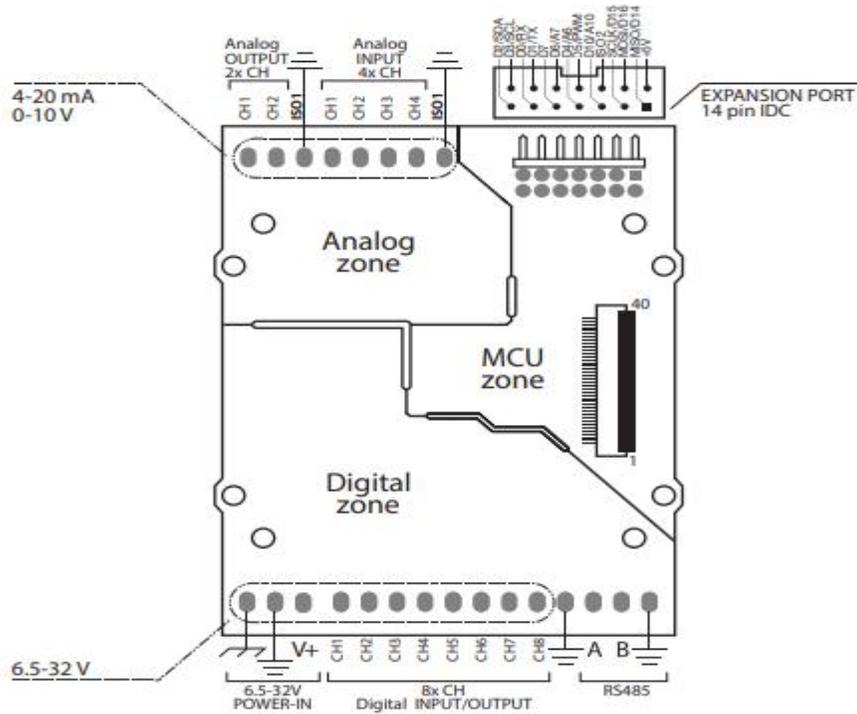


Figure 1.5.4 IND.I/O Baseboard Pinout [18]

The digital field part of the system provides power for the main functions. In order to get the best accuracy, the analog I/O section allows separate power supply in its field. The three zones are isolated from each other.

# Chapter2

## Background

Here, we discuss one of the theoretical propagation models and LoRa link budget estimation for the specific devices and environment that we will face. To be specific, we discuss and compare different empirical outside propagation models. Especially, we would like to discuss the corresponding empirical model which is used in this thesis in detail.

In actual mobile communications situation, the radio wave propagation is complex and changeable. Practice has proven that any attempt to use one or several theoretical formulas to calculate the results will introduce large errors, sometimes even far from the measured results. Therefore, researcher have concluded a variety of empirical models with amounts of actual field measurements and analysis. Usually under certain circumstances, using these models to estimate the propagation characteristics of mobile communication waves can obtain fairly accurate prediction results. The ability to accurately predict the coverage of the base station signal determines the strength and weakness of mobile communication network planning. The key to improving the accuracy of prediction is to choose the prediction model that is closest to the measured value. The Okumura model is currently widely used; but in order to make predictions with computer during system design, Hata compared the basic median field strength of the Okumura model through analysis and comparison of other prediction models, and comparison with the guidelines of actual data, found that the Okumura-Hata prediction model is closer to the actual measured value.

## 2.1 Empirical propagation models

### 2.1.1 Okumura propagation model

The Okumura model is the most widely used one for predicting urban signals. It has become the standard for system planning in Japan. It was developed during the mid 1960's as the result of large-scale studies conducted in and around Tokyo, Japan. This model is suitable for frequency range of 150~1920 MHz, distance 1~100 km, antenna height 30~1000 m. This model has been developed by Okumura et al using different frequencies and different antenna heights, with different distances for a series of tests, and finally drawing an empirical curve for data collected in Tokyo, Japan [19].

The corresponding path loss between TX and RX which is assumed by Okumura model can be formally expressed as:

$$L_P = L_{FS} + A_{mu} + h_{tu} + h_{ru} - \sum K_{correction} \quad (1)$$

where,

$L_P$  = the corresponding median path loss;

$L_{FS}$  = the free space path loss;

$A_{mu}$  = the basic median propagation loss;

$h_{tu}$  = the transmitter antenna height gain factor;

$h_{ru}$  = the receiver antenna height gain factor;

$K_{correction}$  = the correction factor gain.

The propagation loss in free space can be expressed as:

$$L_P = 32.45 + 20\log(d) + 20\log(f) - 10\log(G_t) - 10\log(G_r) \quad (2)$$

where,

$d$  = The distance between transmitter and receiver [km];

$f$  = signal propagation frequency [MHz];

$G_t$ ,  $G_r$  = The gains of transmitter and receiver antennas respectively.

### **Basic median propagation loss ( $A_{mu}$ ) in urban environment**

In order to calculate the median propagation loss in the mobile channel, we can divide the terrain into two categories, namely (i) medium undulating terrain and (ii) irregular terrain. Medium undulating terrain means that on the topographic profile of the propagation path, the height of the undulations on the ground does not exceed 20m, and the undulations are slow. Moreover, the horizontal distance between the peak and valley points is greater than the undulations. Other terrains such as hills, isolated mountains, slopes and mixed land and water are collectively called irregular terrain. Since our experimental location is in the city, we use the medium undulating terrain as the propagation benchmark.

When calculating the propagation loss on various terrains and ground objects, the median loss or field strength of the urban area on the medium undulations is used as the reference, so it is called basic median. The transmission loss depends on the propagation distance,

the operating frequency, the height of the base station antenna and the height of the mobile station antenna etc. Based on a large number of experiments and statistical analysis, prediction curve of basic median attenuation can be made. And Fig. 2.1.1 shows The curve of the relationship between the basic median, frequency, and distance in typical medium undulating urban area.

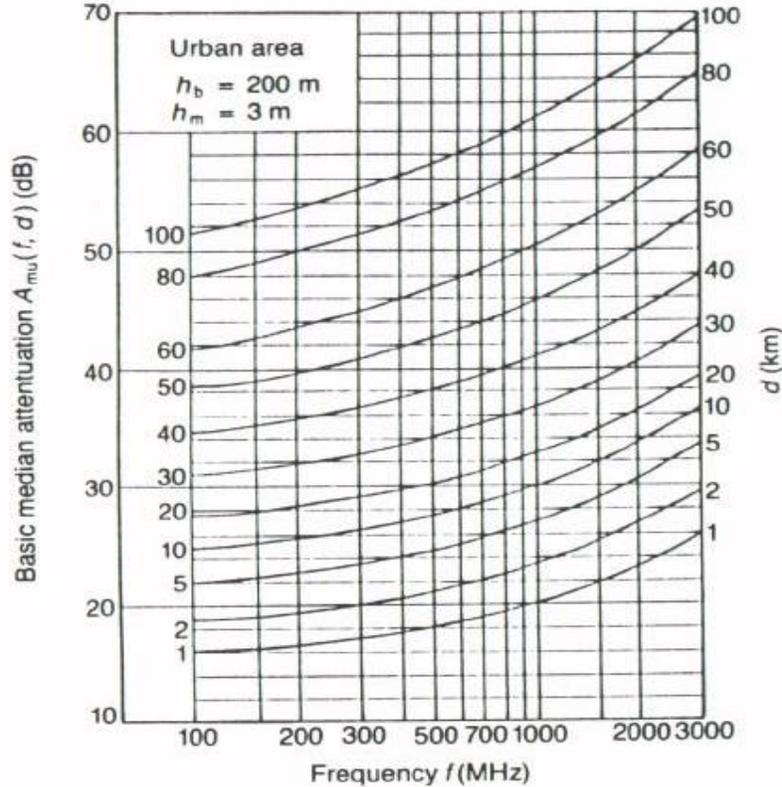


Figure 2.1.1 Basic median attenuation in typical medium undulating urban area [20].

From Fig. 2.1.1 we can see that with the frequency increases and the distance increases, the median basic propagation loss in urban areas will increase. The curve in the figure is measured with the reference antenna height, the base station antenna height is 200 m, and the mobile station antenna height is 3 m.

### The height gain factor of base station and mobile antenna

If the height of the base station antenna is not 200 m, the difference of median loss is expressed by the base station antenna height gain factor  $H_{tu}$ . Fig. 2.1.2 shows the relationship between  $H_{tu}$  and  $h_b$  at different communication distances, when  $h_{te} > 200$  m,  $H_{tu} > 0$  dB; on the contrary, when  $h_{te} < 200$  m,  $H_{tu} < 0$  dB.

Similarly, when the mobile station antenna height is not 3m, the data in Fig. 2.1.2 needs to be corrected by the mobile station antenna height gain factor  $H_{ru}$ , in Fig. 2.1.3 when  $h_{re} > 3$  m,  $H_{ru} > 0$  dB; On the contrary, when  $h_{re} < 3$  m,  $H_{ru} < 0$  dB. From Fig. 2.1.3 we can also see that when the antenna height of a mobile station is longer than 5m, its height gain

factor  $H_{ru}$  is not only related to the antenna height and frequency, but also to the environmental conditions. For example, in small and medium cities, the average height of buildings is relatively low, so its interference effect is small. When the mobile station height is more than 4m, as the antenna height increases, the antenna height gain factor increases significantly. If the height of the mobile station antenna is in the range of 1-4 m,  $H_{ru}$  is less affected by environmental conditions.

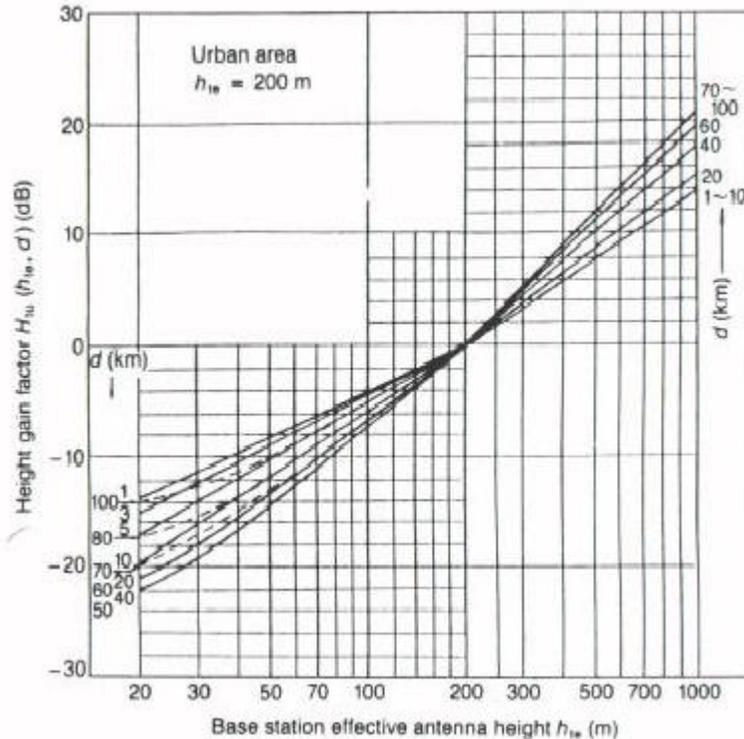


Figure 2.1.2 Base station height gain factor with its height [20].

One should notice that the median field strength in urban areas is also related to the direction of the street (relative to the direction of radio wave propagation from the TX to the RX). The median loss in the longitudinal route (parallel to the direction of radio wave propagation) is significantly smaller than the median loss in the transverse route (vertical to the radio wave direction). This is because the channel formed along the building facilitates the propagation of radio waves, which is called channeling effect. This makes the median field strength of the vertical route higher than the median reference field strength, and the median field strength on the horizontal route lower than the median reference.

In the model Okumura provides some additional corrections in graphical form. With what we mentioned above, we know that some corrections can be made to the overall slope of the terrain, street direction, the mix environment of land and sea to improve the accuracy of the model. However, these corrections are rarely used in practice as they are.

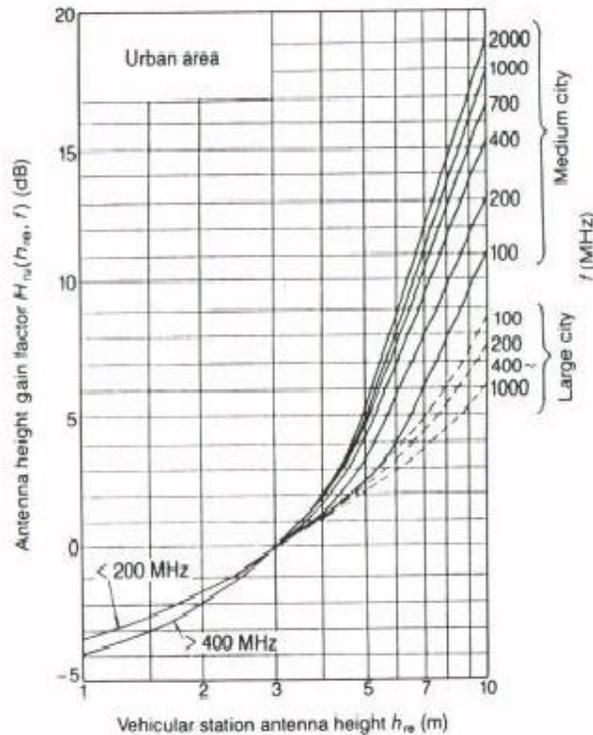


Figure 2.1.3 Mobile station height gain factor with its height [20].

In some complex environments, signal often encounters non-sighted paths caused by terrain obstacles. There is a correction factor in Okumura's model to solve these obstacles, which is called the "Isolated Ridge" factor. However, this correction applies only to obstacles that meet this description such as an isolated ridge. But if the terrain is more complex, this factor can not model it anymore. A number of more general models exist for calculating diffraction loss. However, none of these can be applied directly to Okumura's basic mean attenuation [20].

The Okumura model is based entirely on measured data. In many cases, the derived curve extrapolation can be performed to obtain values which are outside the measurement range. But this extrapolation effectiveness depends on the condition and smoothness of the curve in the specific problem.

What's more, in terms of the path loss prediction accuracy of cellular and land mobile radio systems in chaotic environments, we can say that it is the best and simplest model. The main disadvantage is that it responds slowly when the topography changes rapidly, so this model is ideal in urban and suburban areas, but not ideal in rural areas. In general, the common standard deviation between the predicted and measured path loss values is about 10 dB to 14 dB.

If the average terrain height is longer than the radiation center line height, the corresponding effective antenna height can even become negative; the use of effective

antenna height is only suitable for large cell radii. When the radius is less than 3 km, the effective antenna height becomes unsatisfactory; these characteristics of the Okumura model mean that its applicability is limited to the range of parameters which are used in model development. If the actual parameter value is out of range, the curves may need to be extrapolated depending on the specific situation.

The above shortcomings and difficulties have caused many adjustments in using it to address the details related to computer modeling. Here we will show an example to let the readers have a better understanding of this model.

### Example 1

Let us use the Okumura model to determine the received signal level at 2.3 miles with operating frequency of 870 MHz. The following numerical data are given:

Radiation center line of the BTS transmitter  $h_{bts} = 40\text{m}$ ;

The mobile receive antenna height  $h_m = 3\text{ m}$ ;

Terrain elevation of the BTS location  $E_{bts} = 340\text{ m}$ ;

The average terrain height in the area  $E_{terrain} = 312\text{ m}$ ;

Power delivered to the BTS antenna:  $P_{BTS} = 19.5\text{ W}$ ;

Base station antenna gain:  $10\log(G_t) = 10\text{ dB}$ ;

Mobile station antenna gain:  $10\log(G_r) = 0\text{ dB}$ .

The free space loss can be calculated as:

$$L_{FS} = 32.45 + 20\log(2.3 \cdot 1.609) + 20\log(870) - 10 = 92.61\text{ dB}$$

The basic median attenuation is determined from Fig. 2.1.1 as:

$$A_{mu} = 24\text{ dB}$$

The effective height of the BTS transmitter is given as:

$$h_{te} = 40 + 340 - 312 = 68\text{ m}$$

Correction for the base station height gain can be determined from Fig. 2.1.2 as:

$$H_{tu} = -9\text{ dB}$$

The total path loss is given as:

$$L = 92.61 + 24 + 9 = 129.61\text{ dB}$$

The received signal level is obtained as:

$$RSL = 10\log(19.5 \cdot 1000) - 129.61 = -82.7\text{ dB}$$

## 2.1.2 Okumura-Hata model

The Hata model is another radio propagation model used to predict the path loss of cellular transmissions in the external environment. It is suitable for frequencies from 150 to 1500 MHz. It is an empirical formula based on Okumura model data, so it is also commonly called the Okumura-Hata model [21]. This model combines the graphical information of the Okumura model and further improves it to study the effects of reflection, diffraction and scattering caused by urban structures [21].

Although the Hata model is based on the Okumura model, it does not cover the entire frequency range which is covered by the Okumura model. Okumura can provide support up to 1920 MHz, while Hata models will not exceed 1500 MHz. This model is suitable for broadcasting and point-to-point communication. The height of the corresponding base station antenna is in the range 30-200 m, the mobile station antenna height is 1-10 m, and the link distance is 1-10 km.

### 2.1.2.1 Okumura-Hata model for urban environment

The median formula of basic propagation loss of Okumura Hata model in urban environments is

$$L_P = 69.55 + 26.16\log f - 13.82\log h_T - a(h_R) + (44.9 - 6.55\log h_T)\log d \text{ [dB]}$$

where

$f$  = the corresponding operating frequency [MHz];

$h_t$  = the height of the transmitter [m];

$h_R$  = the height of the receiver [m];

$a(h_R)$  = the corresponding correction parameter with different area;

$d$  is the distance between the base and the receiver.

For small or medium sized city:

$$a(h_R) = [ 1.1\log f - 0.7 ]h_R - [1.56\log f - 0.8 ]$$

For large cities:

$$a(h_R) = 8.29(\log(1.54h_R) )^2 - 1.1 \text{ (} 150 \leq f \leq 200 \text{) or,}$$

$$a(h_R) = 3.2(\log(11.75h_R) )^2 - 4.97 \text{ (} 200 < f \leq 1500 \text{)}$$

### 2.1.2.2 Okumura-Hata model for suburban and open environment

The Hata model is suitable for transmission in suburban and rural areas, where there are buildings, but not as tall and dense as cities. More precisely, this model works well where buildings exist but the height of the mobile station does not change significantly. It can be expressed as:

$$L = L_P - 2(\log_{10} \frac{f}{28})^2 - 5.4 \text{ [dB]}$$

The Hata model which is suitable for open areas without dual transmission serial transmission is formulated as :

$$L = L_P - 4.78(\log_{10}f)^2 + 18.33(\log_{10}f) - 40.94 \text{ [dB]}$$

The Hata model is derived from the Okumura model. Therefore, it is somewhat confined to the propagation environment in Japan, but the environment in each region is different; the terms we mentioned such as "small city", "big city", "suburban" are not clearly defined, so people with different backgrounds can have different interpretations.

Initially, Okumura calculated the transmitter antenna height as the height of the TX antenna above the average terrain. Measurement results show that this method cannot effectively perform antenna calculation. In particular, when encountering sudden changes in terrain, the Hata model tends to average extreme changes in signal levels. To solve this problem, some prediction tools have been used to study alternative methods of calculating effective antenna height.

Here are some numerical examples of Okumura-Hata model in different environments:

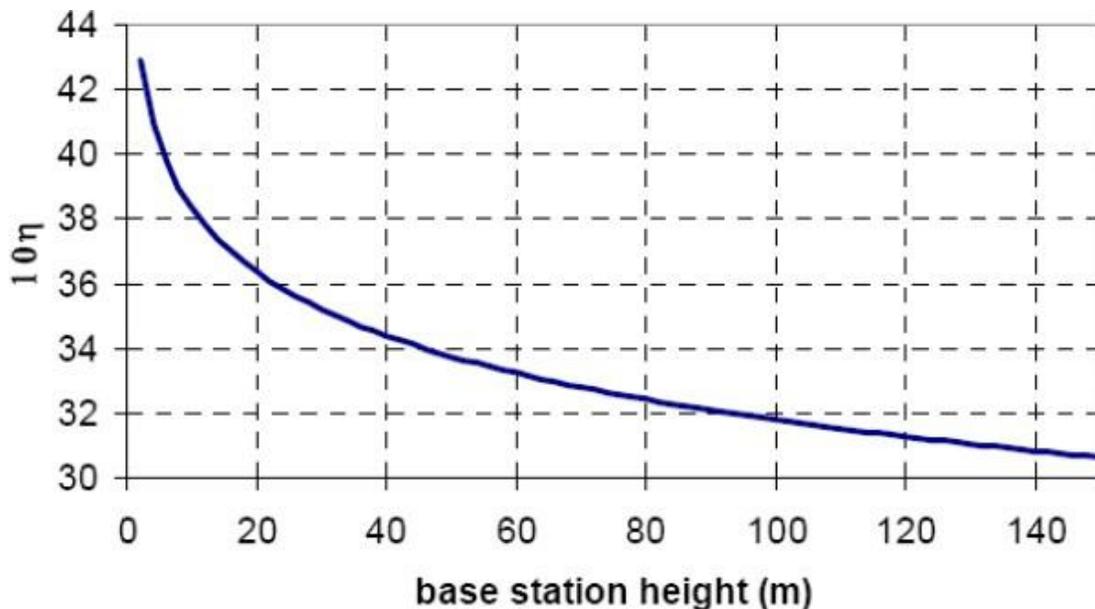
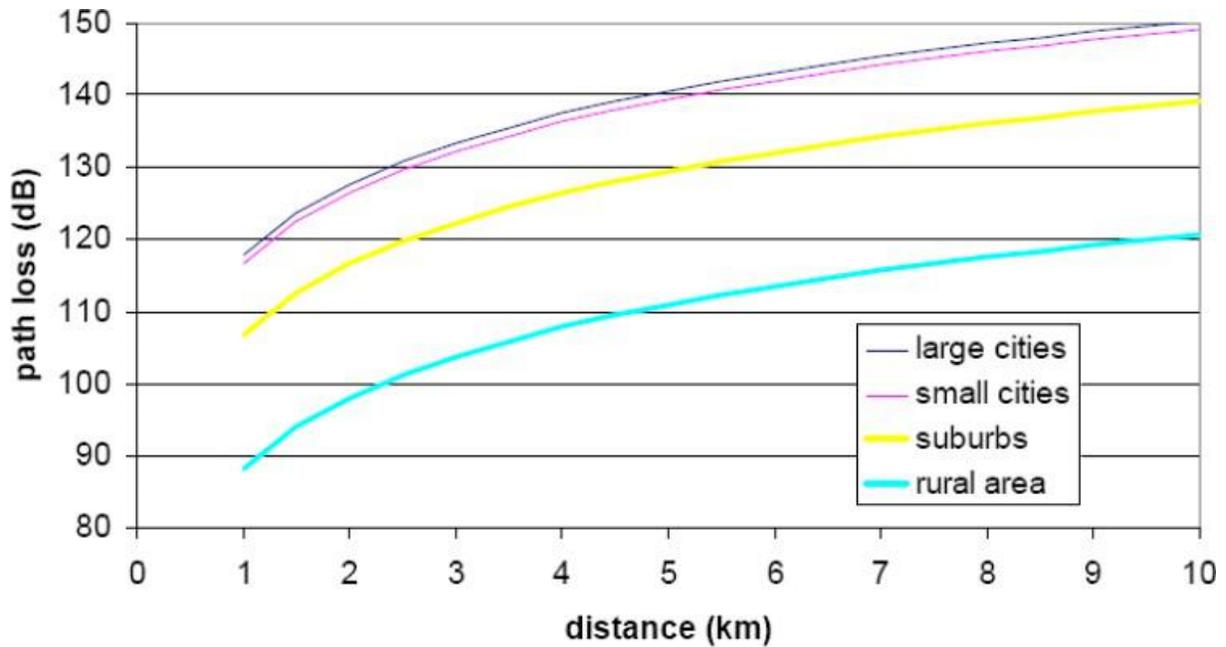


Figure 2.1.4 Okumura-Hata model - Propagation factor with base station height [22]



$$F=900\text{MHz}, h_{bs}=80\text{m}, h_{ms}=3\text{m}$$

Figure 2.1.5 Okumura-Hata model - Numerical example [22]

The propagation factor only depends on the height of the base stations:

$$(44.9 - 6.55\log h_T)\log d \Rightarrow (44.9 - 6.55\log h_T) = 10 \eta$$

### 2.1.2.3 Limitations for Okumura-Hata model

As discussed above, the Hata model is suitable for large area mobile systems, but not suitable for personal communication systems with a cell radius of 1 km.

When the base station antenna is above the surrounding clutter, it does not apply well to microcell plans with antennas below the roof height.

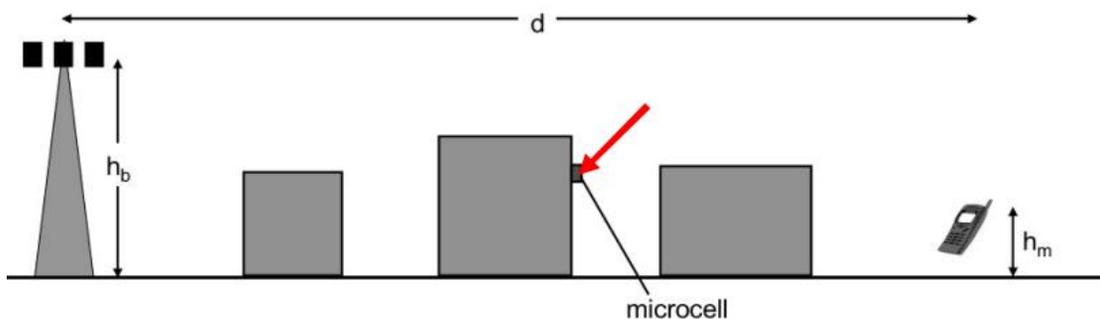


Figure 2.1.6 Microcell planning where antenna is below roof height [22]

Also, we show an example here which is related to example 1. Consider the prediction problem described in example 1, we assume that the propagation is an urban area of a small city. The mobile antenna height gain can be obtained as:

$$a(h_m) = (1.1\log(870) - 0.7) \cdot 3 - (1.56\log(870) - 0.8) = 3.81 \text{ dB}$$

Since the predictions are done in can urban area, no area correction is needed.

$$h_B = 40 + 340 - 312 = 68 \text{ m}$$

Finally, the received signal level is computed as:

$$L = 10\log(19.5 \cdot 1000) + 10 - 69.55 - 26.16\log(870) + 3.81 + 13.82\log(68) - (44.9 - 6.55\log(68))\log(2.3 \cdot 1.609) = -83.01 \text{ dBm}$$

Comparing the result with the value obtained in example 1, we find that the difference is negligible.

### 2.1.2.4 Some other main models based on published data

In the following, we will show some other main propagation models with a brief explanation.

- COST 231-Hata model, this model extends the Okumura-Hata model for medium to small cities to cover the 1500 to 2000 MHz band.
- COST 231-Walfish-Ikegami model, this model is based on considerations of reflection and scattering above and between buildings in urban environment, including some correction factors from measurements.

It considers both line of sight and non line of sight situations and it is designed for 800 MHz to 2 GHz; the corresponding base station height should be from 4 to 50 m, mobile height should be from 1 to 3 m, cell sizes up to 5 km.

- Erceg model, in 1999 Erceg et al. proposed a model derived from a vast amount of data at 1.9 GHz, which makes it a preferred model for picocells and higher frequencies. This model is popular with WiMAX suppliers for 2.5 GHz products, and even 3.5 GHz fixed WiMAX.

The model is usually restricted to frequency from 800 to 3700 MHz; base station antenna height from 10 to 80 m; mobile height is around 2 m; the cell range is from 0.1 to 8 km [22].

Besides the models we have mentioned above, there are also many other specific models for different special situations.

## 2.2 The LoRa link budget in theory

It is known that the communication distance between high-power radio equipment (such as satellites) is long, but the power consumption is large. While the communication between low-power radio equipment (such as ZigBee) is not large, the communication distance is too short. What people want is to use low-power radios for longer distance communications. LoRa is one of the low-power remote technologies which achieved this goal based on the link budget concept.

The link budget is to investigate and analyze various influencing factors in the uplink and downlink signal propagation paths in the system, estimates the system's coverage ability, and obtain the maximum propagation loss allowed by the link under a certain call quality. The purposes of the link budget are:

- Obtaining the maximum available transmit power to the base station, avoiding invalid downlink coverage, reducing interference and system noise, it is the basis for obtaining good network service quality;
- Get the maximum allowable indoor / outdoor path loss. The role of the link budget is that the signal is sent from the base station, the level value of each coverage point is calculated, and finally the coverage area of the base station can be obtained to confirm whether there is an imbalance between uplink and downlink, which is mainly used in coverage planning.

In simple terms, the link budget is a calculation of various losses and gains in a communication link. Network planning and design both require link budgets estimation. However, it should be noted that the link budget calculation is based on a large amount of empirical data. For different regions, the wireless environment in each region will vary, including the density of the buildings, the material of the buildings, and even the background noise, etc. are all different, so the result of the link budget can only provide a rough path loss value. In actual engineering design, it can only be used as a reference value, and cannot be used to guide engineering construction. To get a more accurate result of the impact of the wireless environment on electrical signals, model corrections must be performed locally.

As we mentioned above, there are so many factors in the real world that can have influence on the link budget. So in the fourth and fifth chapter, we will show the main affecting factors during measurement and evaluate their influences. Then we can get a more precise link budget value and compare it with the final path loss  $L$ , which is derived from the Okumura-Hata model.

# Chapter 3

## Experimental setup

### 3.1 Dataset

#### 3.1.1 Parameters of LoRa

As we discussed before, this thesis is related to measuring the performance of LoRa. The LoRa module we use is as shown in Fig. 3.1.1.1. We can see that the module can be connected to external circuit by a 1\*6 pin header connector, and the detailed description of each pin is showed in Table 3.1.1.1.



Figure 3.1.1.1 LoRa module

Table 3.1.1.1 Explanation of each pin

Sequence number	Name	Pin direction	Explanation
1	MD0	INPUT	Configuration entry parameter settings
2	AUX	INPUT/OUTPUT	Indicate module working status
3	RXD	INPUT	TTX serial input, match to external TXD
4	TXD	OUTPUT	TTX serial output,match to external RXD
5	GND		Ground wire
6	VCC		Mains input

The hardware connection of the module are shown in Fig. 3.1.1.2:

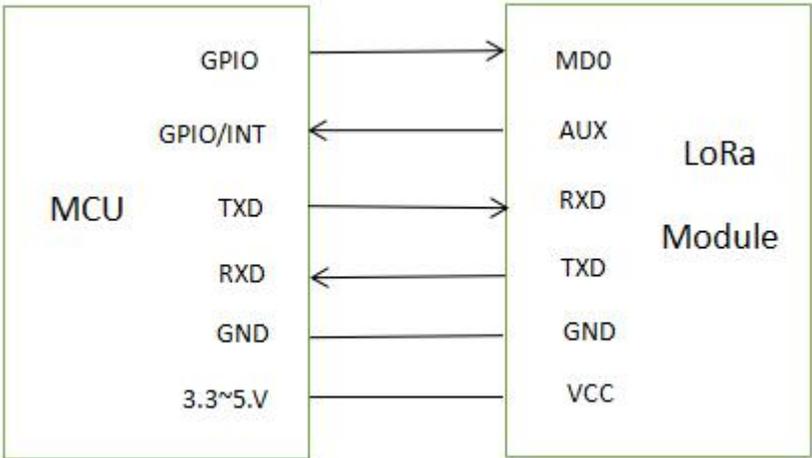


Figure 3.1.1.2 Hardware connection diagram

This LoRa module has 4 operating modes : general mode, wake up mode, power saving mode and signal intensity mode. The one we used during the thesis is the general mode, which also has two kinds of data transmission form: transparent transmission and beam transmission. Moreover, what we need to use is the transparent transmission.

Transparent transmission means the data forwarding device or module is unconscious and no need to care. It is used for devices which have the same address and communication between the same communication channel. The user data can be character or hexadecimal notation data form. For example, device A sends five characters data AA BB CC DD EE to device B, then what device B can receive is AA BB CC DD EE.



Figure 3.1.1.3 Transparent transmission (point to point)

In this thesis, we will use two devices to transmit data and another device to receive, just like Fig. 3.1.1.4 shows:

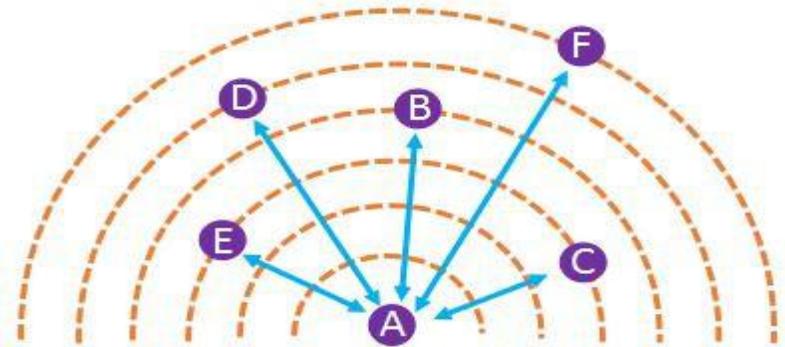


Figure 3.1.1.4 Transparent transmission (point to multi point)

The following table shows the basic LoRa module parameters:

Table 3.1.1.2 Electricity parameter of the LoRa module

Working Frequency	868 MHz
Transmitting Power	20 dBm
Data Rate	19.2 kbps
Modulation mode	LoRa SPSP
Working Voltage	3.3~5.2 V
Emission Current	118 ma
Receive Current	17 ma
Receiving Sensitivity	-136 dBm@0.3 kbps
SERBAUD	9600 bps
Tx Status	Transparent Transmission
Antenna Type	SMA
Antenna Gain	3 dBi
Module Address	0
Communication Channel	23
Transmission Length	Internal ring FIFO 512 characters cache, automatic subcontracting
Receive Length	Internal ring FIFO 512 characters cache, automatic subcontracting
Working temperature	-40~+85 °C

### 3.1.2 Annotation

To use the data which are showed above, annotation is needed in detail. In next part, we will show all the experiments that we did during the thesis. There are almost 6 experiments, from easy to difficult, that allow readers to fully understand the experimental process, especially

the measurements in urban environment. As we mentioned above, in the beginning we did some trials to be thoroughly familiar with the Arduino environment and the Industruino IND.I/O D21G controller. Then we can measure the LoRa performance in the urban environment and combine it with the Industruino IND.I/O D21G controller.

## 3.2 Experimental Process

### 3.2.1 Initial trial with Industruino hardware

As the beginning of all the experiments, the initial trial is the simplest one. In this experiment, our goal is to control the LCD, turn “on” or “off” with a automotive signal switch. Only when we press down the button will the LCD turns on, otherwise it will turns off all the time.

According to Fig. 3.2.1.1, we connected a switch with positive power supply and CH1, and we set CH1 as an input, which means we can use CH1 to read the switch signal.

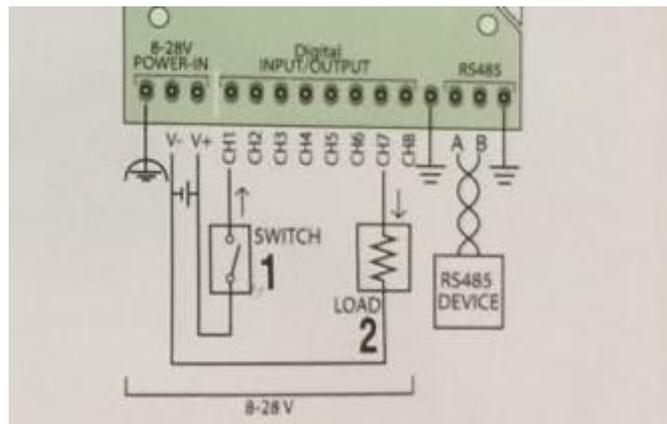


Figure 3.2.1.1 Circuit diagram of IND.I/O D21G

The corresponding C code implementation of the initial trial is shown blow:

```
#include <Indio.h>
int X = 0;

void setup()
{
  pinMode(LED_BUILTIN, OUTPUT);
  Indio.digitalMode(1, INPUT);
  Indio.digitalMode(7, OUTPUT);
}

void loop()
{
  X = Indio.digitalRead(1);
  if (X == HIGH)
```

```

{
    digitalWrite(LED_BUILTIN, HIGH);
    Indio.digitalWrite(7, HIGH);
}
else
{
    digitalWrite(LED_BUILTIN, LOW);
    Indio.digitalWrite(7, LOW);
}
}

```

When we press down the button, the state of CH1 will be high as it is connected to positive pole. Then the LCD can turn “on” with the state changed. And when we release the button, the state of CH1 will always be low, so the LCD turns “off”.



Figure 3.2.1.2 Photograph of the set up used for the implementation of the initial trial

### 3.2.2 Initial trial modification

The second trial is a modified version of the first one, so it still uses the same circuit diagram shown in Fig. 3.2.1.1. The difference is that this time when we press down the switch, the LCD will turn “on” all the time; and when we press down the switch again, it will turn “off” all the time, and so on in a similar fashion.

This time we still set CH1 as an input, but use an old state and a new state to achieve our goal. The corresponding C code implementation is shown below:

```

#include <Indio.h>
int X = 0;
int oldX=0;
int state = 0;

```

```

void setup()
{
  pinMode(LED_BUILTIN, OUTPUT);
  Indio.digitalMode(1, INPUT);
}

void loop()
{
  X = Indio.digitalRead(1);
  if ((X == HIGH)&&(oldX == LOW))
  { state = 1-state;
    delay(300);
  }
  oldX=X;
  if ( state ==1)
  {
    digitalWrite(LED_BUILTIN, HIGH);
  }
  else
  {
    digitalWrite(LED_BUILTIN, LOW);
  }
}

```

Every time when we press down the button, the state of CH1 will change from low to high, then to low again since the switch is a automotive signal switch. So when we press down the button each time, the state of CH1 will changes between 0 and 1, then the LCD will turns “on” or “off” with the changed state.

From Fig. 3.2.2.1, we can see that after we pressed down the button, the LCD turns on all the time.



Figure 3.2.2.1 Photograph of the set up used for the implementation of the second trial

### 3.2.3 Button counting trial with Industruino hardware

The third one is more complex than the first two experiments, because this time we need to use three buttons and press them down in sequence. It seems like to open a safe box with a specify password, we have to press down the first button two times, then the second button three times, then the third button two times. And only in this way will the LCD turns “on”; any other sequence is invalid.

What’s more, if we want to turn the LCD “off”, the only way we can do is that press down the first button, the second button and the third button in sequence, each one time. Only in this sequence will the LCD turns “off”, too. Besides, we increase one button as confirmation button, which means each time when we press down the buttons in sequence, we can press the confirmation button for confirmation.

Therefore, this time we set CH1, CH2, CH3, CH7 as input and use three variables for state judgement. The circuit diagram of button is the same as former, but in order to prevent instability and electrical short circuits we use four more resistances. The resistances are connected to the positive pole and pins separately.

Finally, each time when we press down the buttons, the counter i will increase by one, and we can read the value from the serial monitor to check experiment output. The corresponding C code implementation is shown blow:

```
#include <Indio.h>
#include <Wire.h>
int A = 0, B = 0, C = 0, D = 0;
unsigned int i = 0;
unsigned int MM = 0;
unsigned int Screen = 0;
byte password[7] = {};
byte password1[7] = {0x01, 0x01, 0x02, 0x02, 0x02, 0x03, 0x03};
byte password2[3] = {0x01, 0x02, 0x03};
void setup()
{
  pinMode(LED_BUILTIN, OUTPUT);
  Indio.digitalMode(1, INPUT);
  Indio.digitalMode(2, INPUT);
  Indio.digitalMode(3, INPUT);
  Indio.digitalMode(7, INPUT);
  SerialUSB.begin(9600);
}

void loop()
{
  A = Indio.digitalRead(1);
  B = Indio.digitalRead(2);
  C = Indio.digitalRead(3);
  D = Indio.digitalRead(7);
```

```

if (A == LOW)
{
    delay(100);
    if (A == LOW)
    {
        password[i] = 0x01;
        i++;
    }
}
if (B == LOW)
{
    delay(100);
    if (B == LOW)
    {
        password[i] = 0x02;
        i++;
    }
}
if (C == LOW)
{
    delay(100);
    if (C == LOW)
    {
        password[i] = 0x03;
        i++;
    }
}
if (i > 7)
{
    i = 0;
}
if ((D == LOW)&&(Screen == 0))
{
    delay(100);
    if ((D == LOW)&&(Screen == 0))
    {
        if ((password[0] == password1[0]) && (password[1] ==
password1[1]) && (password[2] == password1[2]) && (password[3] ==
password1[3])
        && (password[4] == password1[4]) && (password[5] ==
password1[5]) && (password[6] == password1[6]))
        {
            MM = 1;
            for (i = 0; i < 7; i++)
            {
                password[i] = 0;
            }
            i = 0;
        }
    }
}

```

```

else
{
  MM = 0;
  for (i = 0; i < 7; i++)
  {
    password[i] = 0;
  }
  i = 0;
}
}
if ((Screen == 1) && (D == LOW))
{
  delay(100);
  if ((Screen == 1) && (D == LOW))
  {
    if ((i==3)&&(password[0] == password2[0]) && (password[1] ==
password2[1]) && (password[2] == password2[2]))
    {
      MM = 0;
      Screen = 0;
      for (i = 0; i < 7; i++)
      {
        password[i] = 0;
      }
      i = 0;
    }
    else
    {
      MM = 1;
      Screen = 1;
      for (i = 0; i < 7; i++)
      {
        password[i] = 0;
      }
      i = 0;
    }
  }
}
if (MM == 1)
{
  Screen = 1;
  digitalWrite(LED_BUILTIN, HIGH);
  delay(100);
}
else
{
  Screen = 0;
  digitalWrite(LED_BUILTIN, LOW);

```

```
    delay(100);  
  }  
  SerialUSB.println(i);  
  delay(100);  
}
```

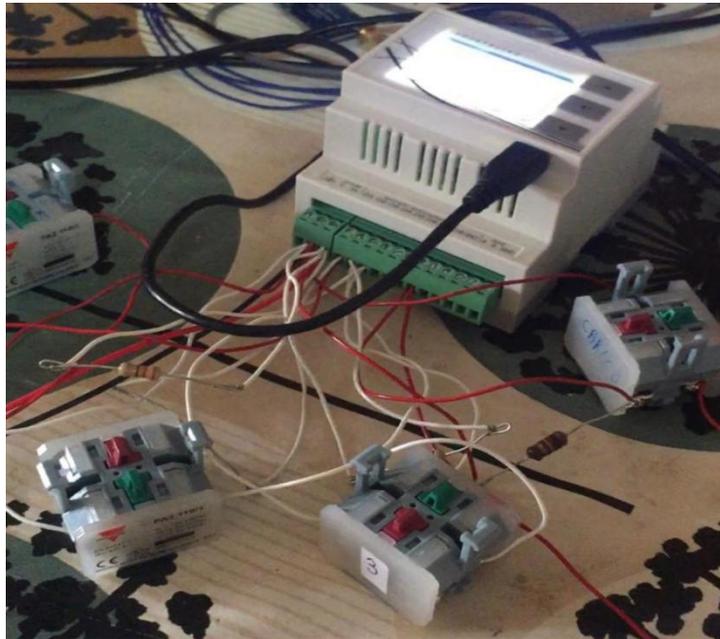


Figure 3.2.3.1 Photograph of the set up used for the implementation of the third trial

In Fig. 3.2.3.1, we can see that each button is matched with a resistor to achieve a pull-up resistor function. This function is used to keep the uncertain signals at “HIGH” state. In this way, the pull-up resistor can keep the pin at a certain logic level even when no external components are connected.

During this experiment, when the pin is under floating condition, the level is unstable; the state can always change without touching the button. So we matched each button with a resistor to solve this problem, each resistor is connected to the positive power supply and a CH port. One thing to note is that it is not good to solve this problem by setting the delay time, because the corresponding response speed will be limited. What’s worse, sometimes the state is still not stable even though the delay time is set.

### 3.2.4 Data reading trial

On the basis of the third experiment, the fourth experiment is about reading serial port value from serial monitor in two ways. Firstly, we will use USB cable to read the value, then use rj45 ( a router interface of the Ethernet module, is usually used for data transmission ) to see it from website. It is very easy to read value with USB cable, because the USB cable can connect the Industruino D21G with computer directly. But it’s more difficult to read value with rj45, since we need to use another Ethernet expansion module, which can adds network connectivity to Industruino.



Figure 3.2.4.1 Ethernet expansion module

In order to read the value with this module, we also need two network cables. One is used to connect the Ethernet expansion module with the router, the other one is used to connect the router with our computer. The corresponding C code implementation is shown below:

```
#include <Indio.h>
#include <Wire.h>
#include <SPI.h>
/*----- network module -----*/
#include <Ethernet2.h>
byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED};
byte gateway[] = {192, 168, 1, 1};
byte subnet[] = {255, 255, 255, 0};
IPAddress ip(192, 168, 1, 177);
EthernetServer server(80); // (port 80 is default for HTTP):
/*-----button -----*/
int A = 0, B = 0, C = 0, D = 0;
unsigned int i = 0; //state judgement
unsigned int MM = 0; //state judgement
unsigned int Screen = 0; //state judgement
byte password[7] = {};
byte password1[7] = {0x01, 0x01, 0x02, 0x02, 0x02, 0x03, 0x03};
byte password2[3] = {0x01, 0x02, 0x03};
void setup()
{
  /*----- button initialization -----*/
  pinMode(LED_BUILTIN, OUTPUT);
  Indio.digitalMode(1, INPUT);
  Indio.digitalMode(2, INPUT);
  Indio.digitalMode(3, INPUT);
  Indio.digitalMode(7, INPUT);
  /*----- network module initialization -----*/
  Indio.digitalMode(4, INPUT);
  Indio.digitalMode(6, OUTPUT);
  Indio.digitalMode(10, OUTPUT);
  delay(10);
}
```

```

Indio.digitalWrite(4, HIGH);
Indio.digitalWrite(6, HIGH);
Indio.digitalWrite(10, LOW);
SerialUSB.begin(9600);
Ethernet.begin(mac, ip); // start the Ethernet2 connection and
the server:
server.begin();
SerialUSB.print("server is at ");
SerialUSB.println(Ethernet.localIP());
}
void loop()
{
/*----- check buttons -----*/
Keys();
SerialUSB.println(i);
delay(100);
/*----- network module procedure -----*/
EthernetClient client = server.available();// listen for
incoming clients
if (client)
{
SerialUSB.println("new client");// http request
boolean currentLineIsBlank = true;
while (client.connected())
{
if (client.available())
{
char c = client.read();
SerialUSB.write(c);
if (c == '\n' && currentLineIsBlank)
{
client.println("HTTP/1.1 200 OK");// send a standard http
response header
client.println("Content-Type: text/html");
client.println("Connection: close"); // the connection
will be closed after completion of the response
client.println("Refresh: 5"); // refresh the page
automatically every 5 sec
client.println();
client.println("<!DOCTYPE HTML>");
client.println("<html>");
/*----- webpage print function -----*/
client.print("i= ");
client.print(i);
client.println("<br />");
client.println("</html>");
break;
}
}
if (c == '\n') // you're starting a new line

```

```

        {
            currentLineIsBlank = true;
        }
        else if (c != '\r') // you've gotten a character on the
current line
        {
            currentLineIsBlank = false;
        }
    }
}
delay(1); // give the web browser time to receive the data
client.stop();// close the connection:
SerialUSB.println("client disconnected");
}
}
/*----- button function -----*/
void Keys ()
{
    A = Indio.digitalRead(1);
    B = Indio.digitalRead(2);
    C = Indio.digitalRead(3);
    D = Indio.digitalRead(7);

    if (A == LOW)
    {
        delay(100);
        if (A == LOW)
        {
            password[i] = 0x01;
            i++;
        }
    }
    if (B == LOW)
    {
        delay(100);
        if (B == LOW)
        {
            password[i] = 0x02;
            i++;
        }
    }
    if (C == LOW)
    {
        delay(100);
        if (C == LOW)
        {
            password[i] = 0x03;
            i++;
        }
    }
}

```

```

}
if (i > 7)
{
    i = 0;
}
if ((D == LOW) && (Screen == 0))
{
    delay(100);
    if ((D == LOW) && (Screen == 0))
    {
        if ((password[0] == password1[0]) && (password[1] ==
password1[1]) && (password[2] == password1[2]) && (password[3] ==
password1[3]) &&
            (password[4] == password1[4]) && (password[5] ==
password1[5]) && (password[6] == password1[6]))
        {
            MM = 1;
            for (i = 0; i < 7; i++)
            {
                password[i] = 0;
            }
            i = 0;
        }
        else
        {
            MM = 0;
            for (i = 0; i < 7; i++)
            {
                password[i] = 0;
            }
            i = 0;
        }
    }
}
if ((Screen == 1) && (D == LOW))
{
    delay(100);
    if ((Screen == 1) && (D == LOW))
    {
        if ((i==3)&&(password[0] == password2[0]) && (password[1] ==
password2[1]) && (password[2] == password2[2]))
        {
            MM = 0;
            Screen = 0;
            for (i = 0; i < 7; i++)
            {
                password[i] = 0;
            }
            i = 0;
        }
    }
}

```

```

    }
    else
    {
        MM = 1;
        Screen = 1;
        for (i = 0; i < 7; i++)
        {
            password[i] = 0;
        }
        i = 0;
    }
}
}
if (MM == 1)
{
    Screen = 1;
    digitalWrite(LED_BUILTIN, HIGH);
    delay(100);
}
else
{
    Screen = 0;
    digitalWrite(LED_BUILTIN, LOW);
    delay(100);
}
}
}

```

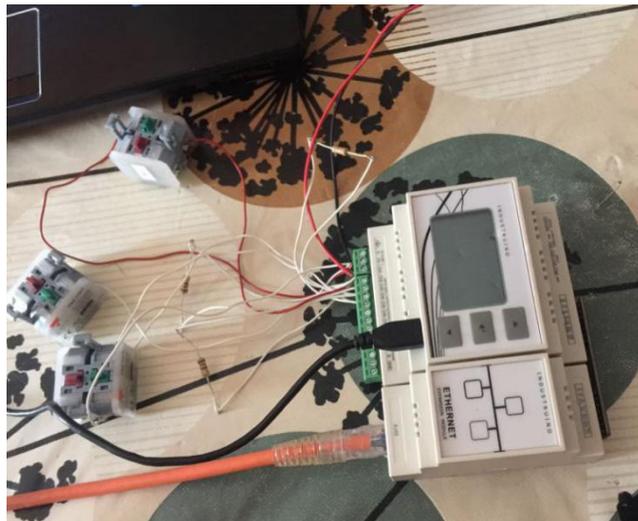


Figure 3.2.4.2 Photograph of the set up used for the implementation of the fourth trial

In fact, we can even read the value from website without the second network cable since our computer can connect the router with wireless network. But it still has big limitation, because in this method we must use a network cable to connect Industruino and router. So we need to achieve the same goal in other ways, and LoRa is a very good choice.

### 3.2.5 The combination of LoRa and D21G controller

This time, we are going to combine LoRa module with the D21G controller to implement the communication between D21G controller and the computer. As Fig. 1.5.4 shows, the expansion port 14 pin IDC, we connect the corresponding pins to LoRa pins. The circuit diagram is shown in Fig. 3.2.5.1.

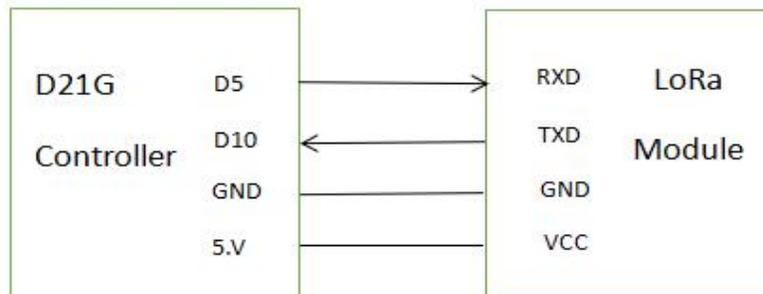


Figure 3.2.5.1 Circuit diagram of D21G controller and LoRa module

We also need to use another USB to TTL converter to connect another LoRa module to the computer, the corresponding circuit diagram is the same as Fig. 3.2.5.3 shows. Actually, in D12G controller, D5 represents its TXD and D10 represents its RXD.

When all the circuits are connected, the controller can communicate with our PC via LoRa modules. For example, when we upload the sketch of the third button experiment to the controller, we can see the serial port values from the serial monitor of Arduino. This communication style breaks the limitation of cable, which is very convenient to implement.

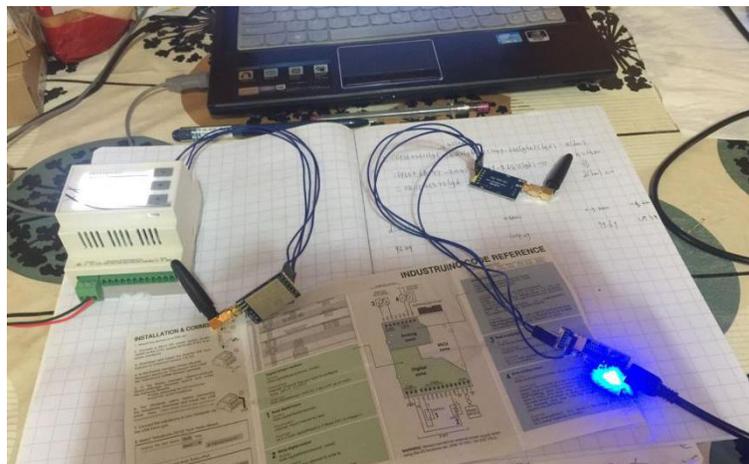


Figure 3.2.5.2 Implementation of the D12G controller and LoRa module

Finally we find that the combination is not perfect, since LoRa module is at the outside of the controller, the Dupont line (a special purpose sewing thread produced by DuPont, can be connected to pins very securely, no soldering required, circuit tests can be performed) can be broken easily. Then we can put the LoRa module in the shell and use electric soldering iron to connect it to the controller directly, which is much more convenient and stable.



Figure 3.2.5.3 The final combination of D21G controller and LoRa module

### 3.2.6 Measurement of LoRa in urban environment

This experiment is the most important one as our goal of the thesis is to study applications and validations of LoRa in complex urban environment. Originally, we were going to use the Industruino IND.I/O D21G for implementation; but for simplicity to perform effective measurements, we used Arduino UNO instead of the Industruino D21G. So during the implementation, we also need three more Arduino UNO boards and some USB to TTL converter modules. As we mentioned in chapter 1.4, Arduino is a convenient and flexible open source electronic prototype platform. Figs. 3.2.6.1 and 3.2.6.2 show the corresponding items we used in this experiment.

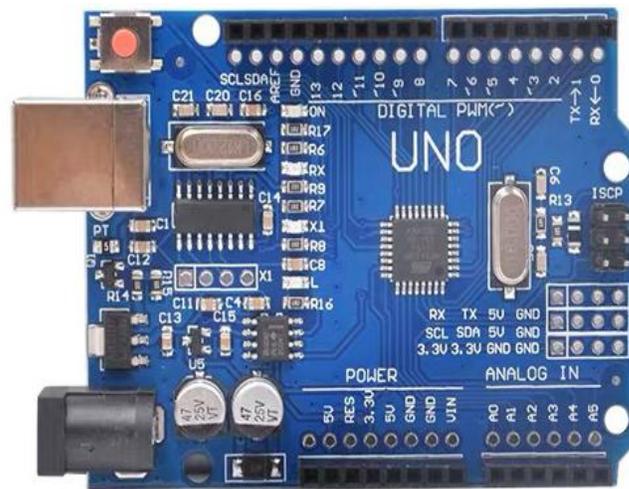


Figure 3.2.6.1 Photograph of Arduino UNO board (ATmega328P)

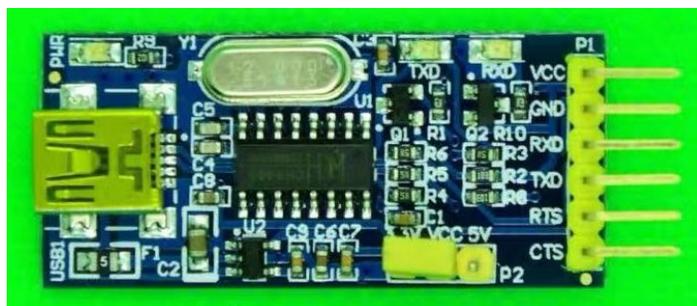


Figure 3.2.6.2 Photograph of USB to TTL converter (CH340)

### Configuration parameter

In order to implement the communication among LoRa modules, the first thing we should do is to set up the configuration parameters. Each USB to TTL converter is connected to a LoRa module and the PC, then we can set the parameters via a configuration software. Fig. 3.2.6.3 shows the circuit diagram and Fig. 3.2.6.4 shows the parameters of LoRa module.

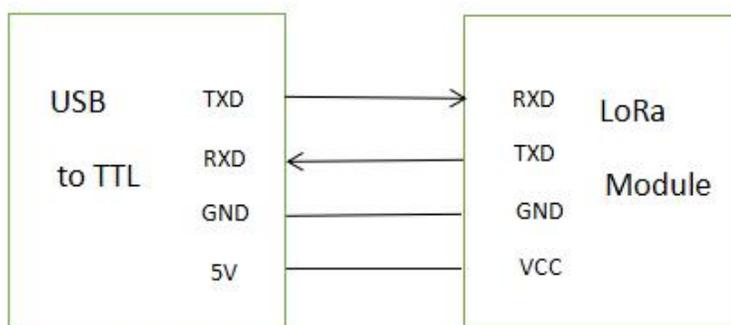


Figure 3.2.6.3 Circuit diagram of USB to TTL and LoRa module



Figure 3.2.6.4 The configuration parameters of LoRa module

From Fig. 3.2.6.4, we can see that the baud rate is 9600 bps, data rate is 19.2 kbps, module address is 0 and communication channel is 23. What's more, all the LoRa modules should have the same parameters, then they can communicate with each other.

After we set the parameters of LoRa module, we can start to connect the LoRa module to the UNO board. Fig. 3.2.6.5 shows the corresponding circuit diagram and Fig. 3.2.6.6 shows the corresponding implementation in real world.

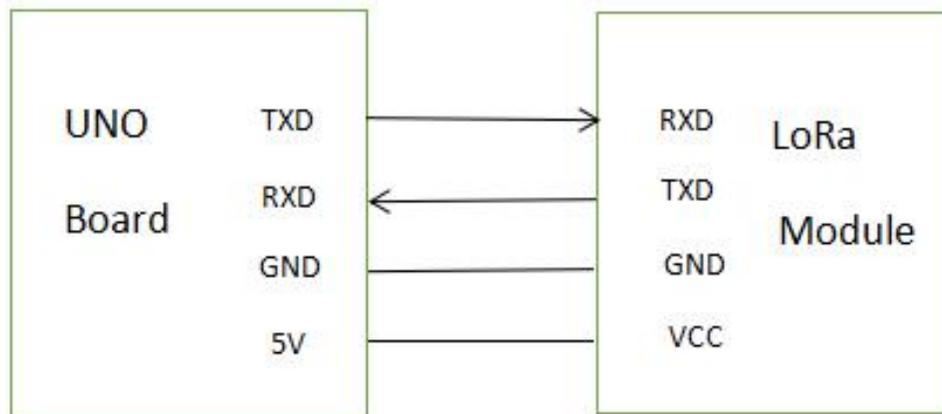


Figure 3.2.6.5 Circuit diagram of UNO board and LoRa module

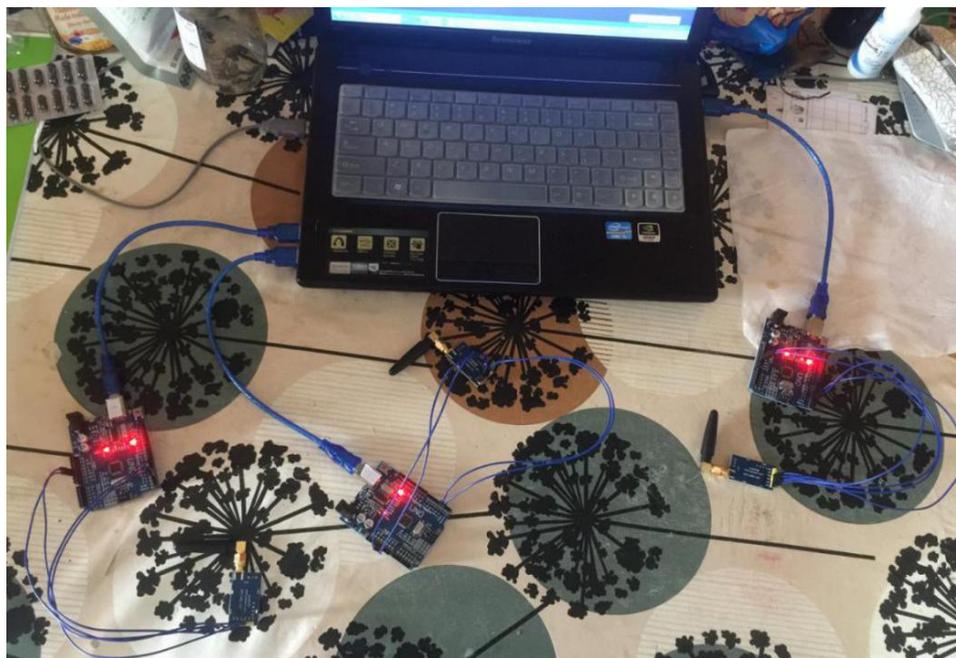


Figure 3.2.6.6 Implementation of LoRa modules with UNO board

As we mentioned above, the first two modules are used to transmit data A and data B, then the third one is used to receive data which are sent from the first two modules. Each time module A and B send the same length of data, but with different environments. When module C receives data, its counter will increase by 1.

## On-fied measurement scenarios

We conducted experiments in the vicinity of the Polytechnic of Turin University campus ( Figs. 3.2.6.7 and 3.2.6.8 ), which represents a large urban space with high-density high-rise buildings between the transmitter and the receiver. And the experiments were performed in a 1 km × 1 km area with high-density high-rise structures and natural vegetation.

To further test LoRa coverage, similar experiments were performed for indoor and semi-indoor environments. The indoor environment refers to placing the transmitter in a room in a building, while the semi-indoor environment refers to placing the same transmitter on a balcony, which is connected to the open space and has the same height as the previous one. The combination of indoor and semi-indoor environments allows us to make a comprehensive observation and comparison of the coverage of LoRa. The heights of the transmitters are 30 meters and the for the receiver is 1.5 m. The corresponding C code implementation of the transmitter is shown blow:

```
#include <SoftwareSerial.h>

SoftwareSerial mySerial(2, 3);

void setup()
{
  Serial.begin(9600);

  mySerial.begin(9600);
}

void loop()
{
  mySerial.print("A123");

  delay(120);
}
```

At the first time, transmitter A sends packets “A123” and transmitter B sends “B123”. Then they will send “A123456” and “B123456”, and finally send “A123456789” and “B123456789” separately, to verify that if different packet length have influence on the packet loss rate.

What’s more, we can see that the delay time is 120 ms, which means that each transmitter can send 500 packets per minute in theory.

The corresponding C code implementation of the receiver is shown blow:

```
#include <SoftwareSerial.h>

SoftwareSerial mySerial(2, 3); // RX, TX

String str1 = "";
int c=0;
int d=0;
```

```

void setup()

{
  Serial.begin(9600);

  mySerial.begin(9600);
}

void loop()

{

  while (mySerial.available() > 0)
  {
    str1 += char(mySerial.read());

  }
  if(str1.length() > 3 )
  {
    if( str1 == "A123")
    {
      c++;
      Serial.println(str1);

      Serial.println(c);
    }
    if (str1 == "B123")

    {
      d++;
      Serial.println(str1);

      Serial.println(d);
    }
    str1 = "";
  }
  delay(10);

}

```

In order to get a more precise measurement result, we chose 14 measurement points, which surround the transmitters in a circle. The receiver is connected to our computer, and while transmitters are sending packets, we should go outside and measure the received packets at each measurement point.

The measurement time is about 30 minutes at each point in this process, and then we can calculate a average number of received packets per minute to get a more precise measurement result.

The measurement points are shown as below:

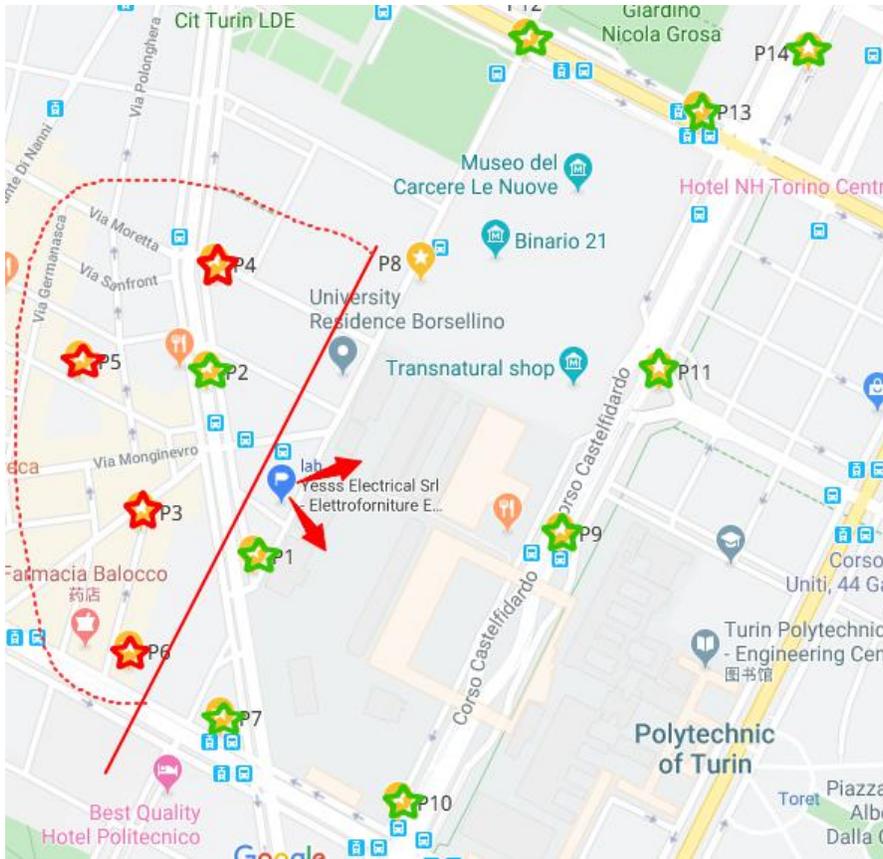


Figure 3.2.6.7 Measurement locations are star dots and base transmitters are flag dot

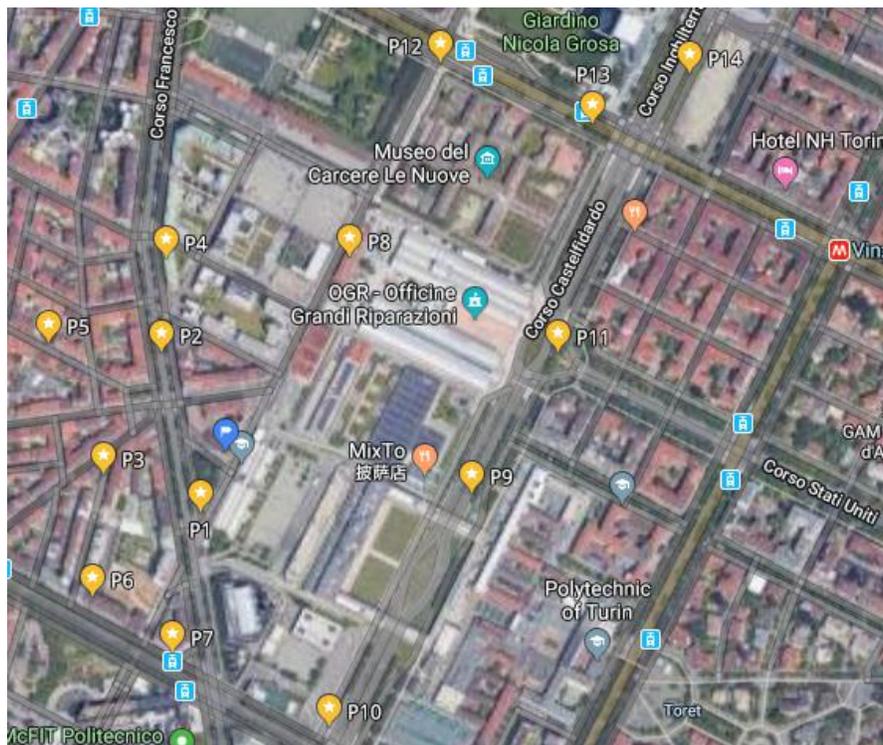


Figure 3.2.6.8 Measurement locations are star dots and base transmitters are flag dot

Polytechnic of Turin University and its surrounding urban environment are shown in Fig. 3.2.6.7 and 3.2.6.8. Under the above-mentioned urban conditions, the buildings are densely packed, and because of different natural conditions in different directions, the environment is quite different

The marked flag points on the map in the figure represent the transmitters in the experiment. Star points seen in different places are the corresponding locations for receiving signals. LoRa measurements are made at different distances and under different urban conditions. The communication range we measured in this thesis is 80 meters to 800 meters. Urban environmental conditions can be viewed as a moderately dense urban structure.

# Chapter 4

## Measurement results

In this chapter, we will show and discuss the measurement campaign results of the previous chapter. There are three measurement processes with different packet lengths. We show the corresponding data table of received packets and packets loss, then plot them as the distance of measurement points increase. The results show that in a similar environment most points have a good correlation between the packet loss and the Okumura-Hata model. Moreover, we also found that positions of the transmitters are important, as the obstacles have large interference with the LoRa signals.

During the measurement processes, the emission rate is 500 packets/min, and the average measurement time is 30 minutes, then we can get a average percentage of received packets per minute.

### 4.1 The first measurement campaign performance

The first measurement campaign is implemented with data “A123456789” and “B123456789”. As we mentioned before, the first transmitter was placed in the room; the second one is placed at the balcony, at the same height as the previous one. By default, the first transmitter was always transmitting data “A” and the second one was always transmitting data “B” during the whole measurements. Table 4.1.1 and Fig. 4.1.1 and Fig. 4.1.2 show the changes in the received packets and losses during the measurement process.

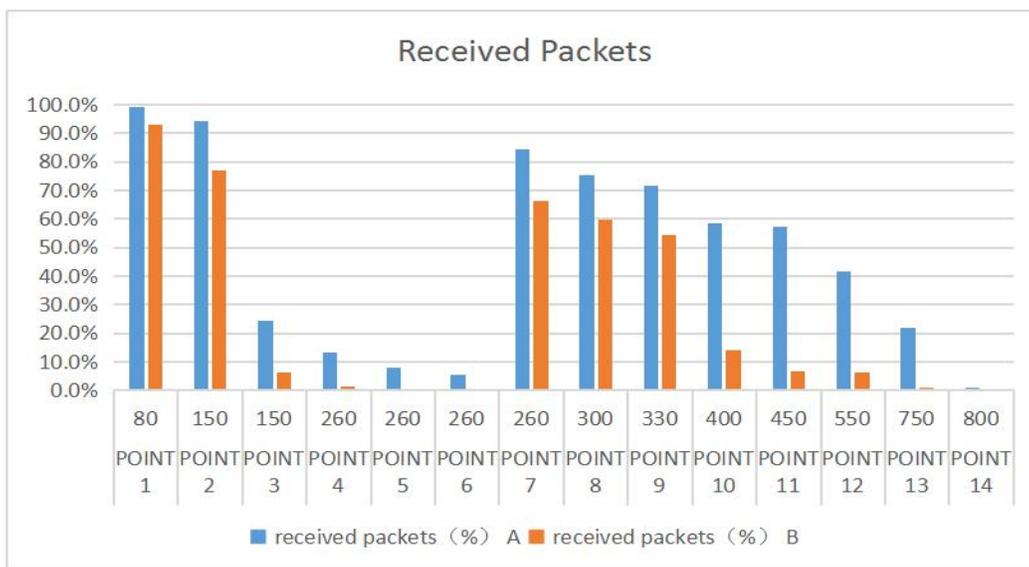


Figure 4.1.1 The percentage of received packets in first campaign

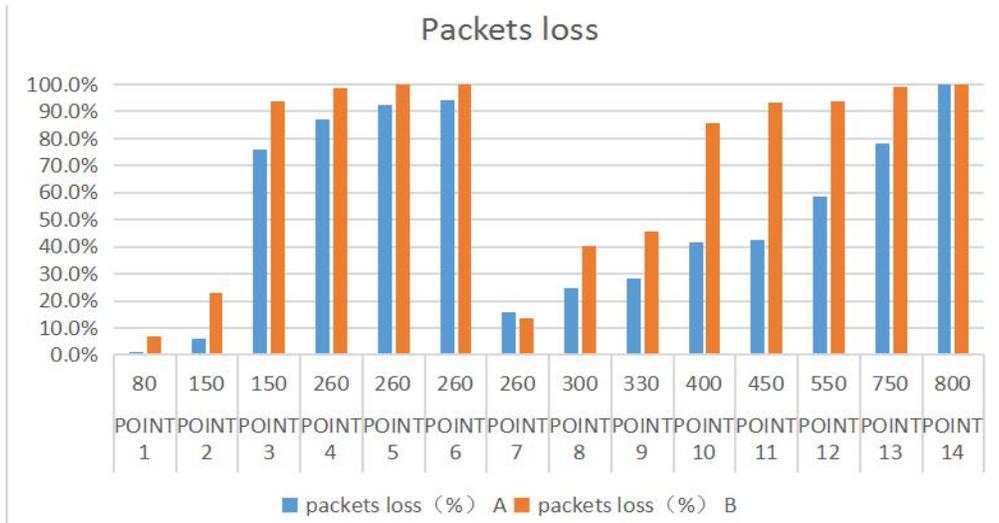


Figure 4.1.2 The percentage of packets loss in first campaign

Table 4.1.1 The percentage of received packets and packets loss in first campaign

POINTS	distance (m)	DATA “A123456789”		“B123456789”	
		received packets (%)		packets loss (%)	
		A	B	A	B
POINT 1	80	99.10%	93.20%	0.90%	6.80%
POINT 2	150	94.20%	76.90%	5.80%	23.10%
POINT 3	150	24.20%	6.40%	75.80%	93.60%
POINT 4	260	13.10%	1.20%	86.90%	98.80%
POINT 5	260	7.70%	0.00%	92.30%	100.00%
POINT 6	260	5.60%	0.00%	94.40%	100.00%
POINT 7	260	84.30%	66.30%	15.70%	13.70%
POINT 8	300	75.40%	59.60%	24.60%	40.40%
POINT 9	330	71.70%	54.50%	28.30%	45.50%
POINT 10	400	58.50%	14.10%	41.50%	85.90%
POINT 11	450	57.30%	6.70%	42.70%	93.30%
POINT 12	550	41.50%	6.30%	58.50%	93.70%
POINT 13	750	21.70%	1.10%	78.30%	98.90%
POINT 14	800	0.20%	0.00%	99.80%	100.00%

## 4.2 The second measurement campaign performance

The second measurement campaign is implemented with data “A123456” and “B123456”. Table 4.2.1 and Fig. 4.2.1 and Fig. 4.2.2 show the changes in the received packets and losses during the measurement process.

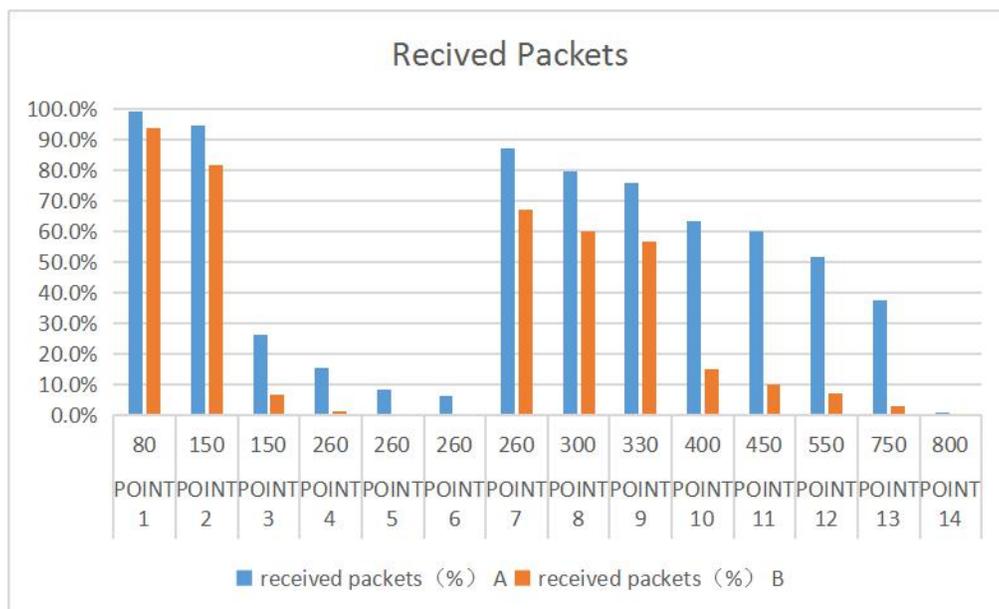


Figure 4.2.1 The percentage of received packets in second process

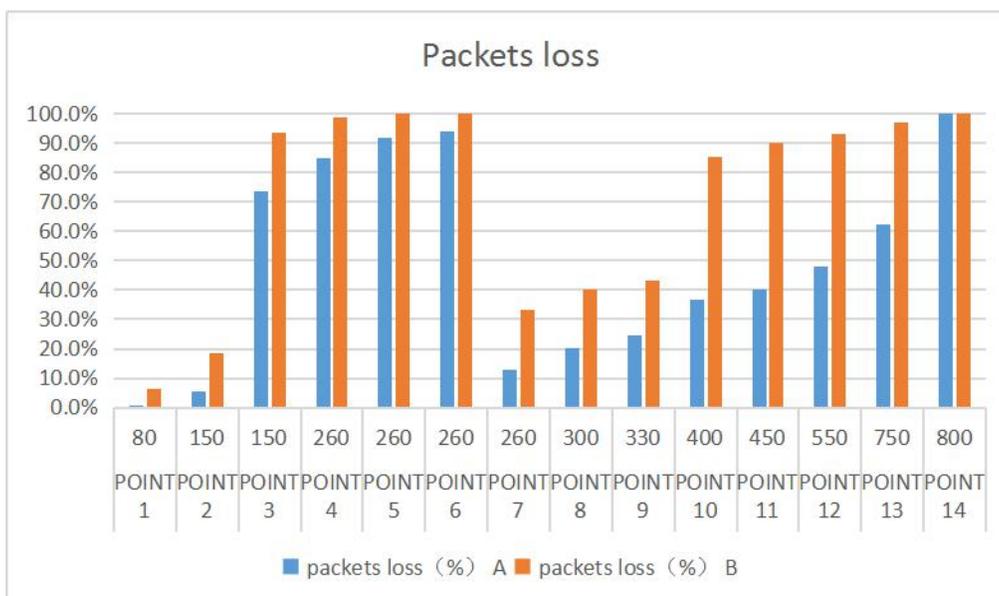


Figure 4.2.2 The percentage of packets loss in second process

Table 4.2.1 The percentage of received packets and packets loss in second campaign

POINTS	distance (m)	DATA “A123456” “B123456”			
		received packets (%)		packets loss (%)	
		A	B	A	B
POINT 1	80	99.3%	93.8%	0.7%	6.2%
POINT 2	150	94.5%	81.6%	5.5%	18.4%
POINT 3	150	26.4%	6.7%	73.6%	93.3%
POINT 4	260	15.3%	1.3%	84.7%	98.7%
POINT 5	260	8.4%	0.0%	91.6%	100.0%
POINT 6	260	6.1%	0.0%	93.9%	100.0%
POINT 7	260	87.2%	66.9%	12.8%	33.1%
POINT 8	300	79.6%	59.8%	20.4%	40.2%
POINT 9	330	75.6%	56.8%	24.4%	43.2%
POINT 10	400	63.3%	14.9%	36.7%	85.1%
POINT 11	450	59.8%	9.9%	40.2%	90.1%
POINT 12	550	51.8%	6.9%	48.2%	93.1%
POINT 13	750	37.5%	3.1%	62.5%	96.9%
POINT 14	800	0.2%	0.0%	99.8%	100.0%

### 4.3 The third measurement campaign performance

The third measurement campaign is implemented with data “A123” and “B123”. Table 4.3.1 and Fig. 4.3.1 and Fig. 4.3.2 show the changes in the received packets and losses during the measurement process.

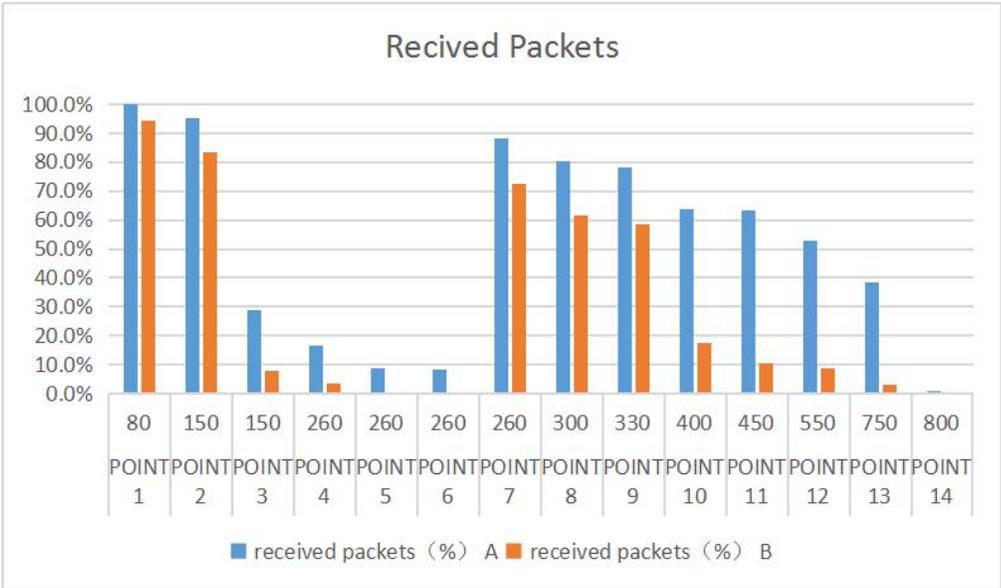


Figure 4.3.1 The percentage of received packets in third campaign

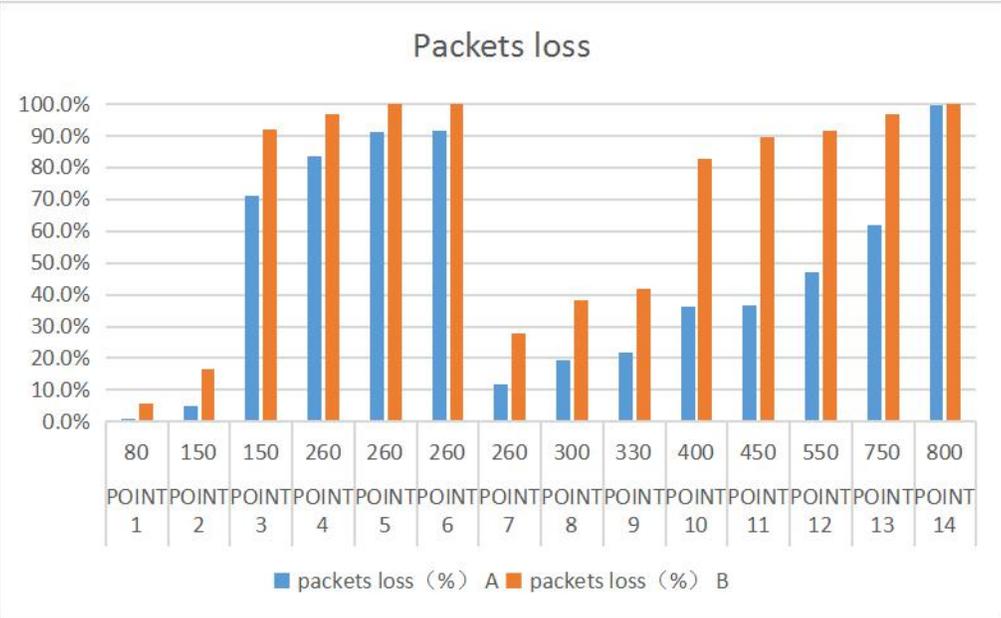


Figure 4.3.2 The percentage of packets loss in third campaign

Table 4.3.1 The percentage of received packets and packets loss in third campaign

POINTS	distance (m)	DATA “A123” “B123”			
		received packets (%)		packets loss (%)	
		A	B	A	B
POINT 1	80	99.8%	94.3%	0.2%	5.7%
POINT 2	150	95.2%	83.3%	4.8%	16.7%
POINT 3	150	28.9%	8.1%	71.1%	91.9%
POINT 4	260	16.4%	3.4%	83.6%	96.6%
POINT 5	260	8.8%	0.0%	91.2%	100.0%
POINT 6	260	8.3%	0.0%	91.7%	100.0%
POINT 7	260	88.4%	72.3%	11.6%	27.7%
POINT 8	300	80.5%	61.7%	19.5%	38.3%
POINT 9	330	78.1%	58.3%	21.9%	41.7%
POINT 10	400	63.7%	17.3%	36.3%	82.7%
POINT 11	450	63.4%	10.3%	36.6%	89.7%
POINT 12	550	52.9%	8.6%	47.1%	91.4%
POINT 13	750	38.3%	3.2%	61.7%	96.8%
POINT 14	800	0.3%	0.0%	99.7%	100.0%

## 4.4 Comparison among the three campaigns

In order to have a better comparison of how the packet length acts on the packets loss, we also have a parallel comparison of received packets and packets loss among the three measurement processes.

Figures 4.4.1, 4.4.2 and 4.4.3, 4.4.4 show the parallel comparison of received packets and packets loss among the three processes separately.

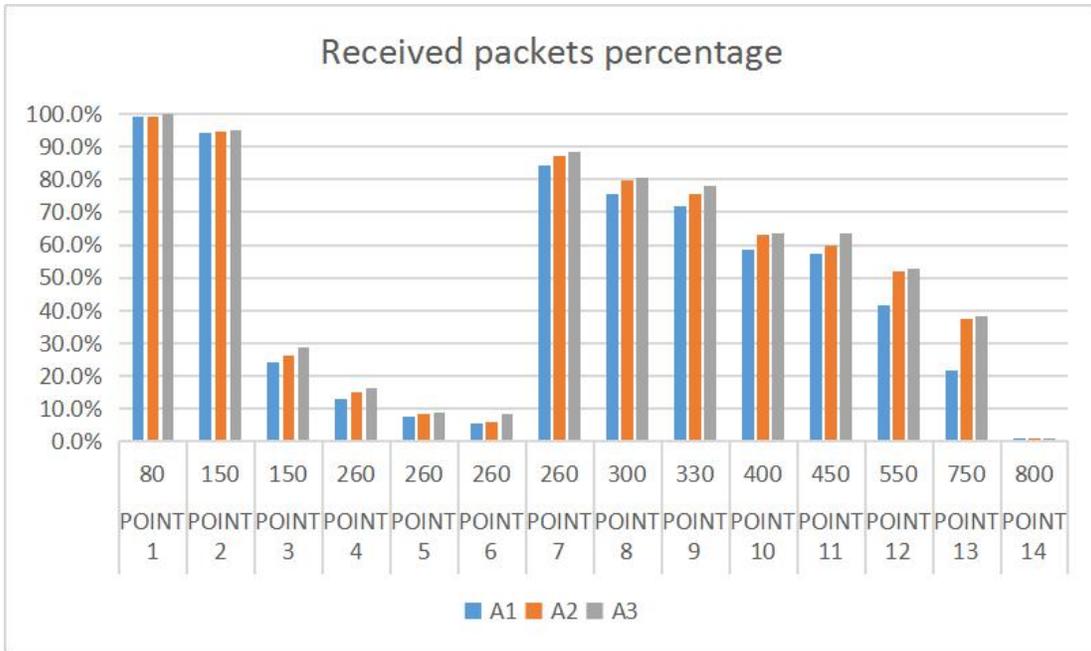


Figure 4.4.1 Comparison of A received packets percentage among three processes

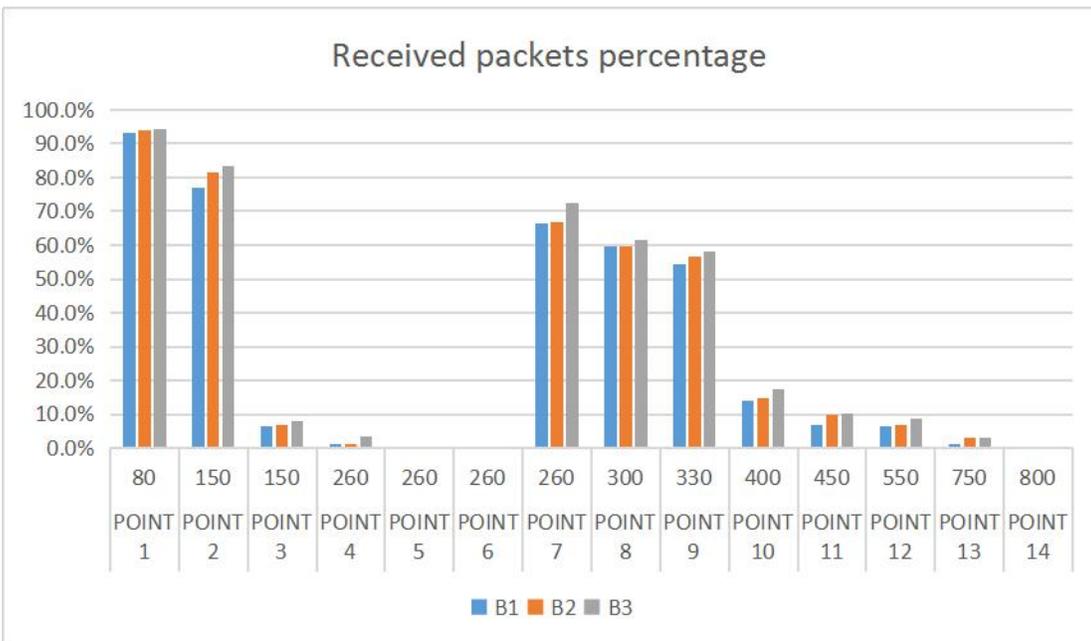


Figure 4.4.2 Comparison of B received packets percentage among three processes

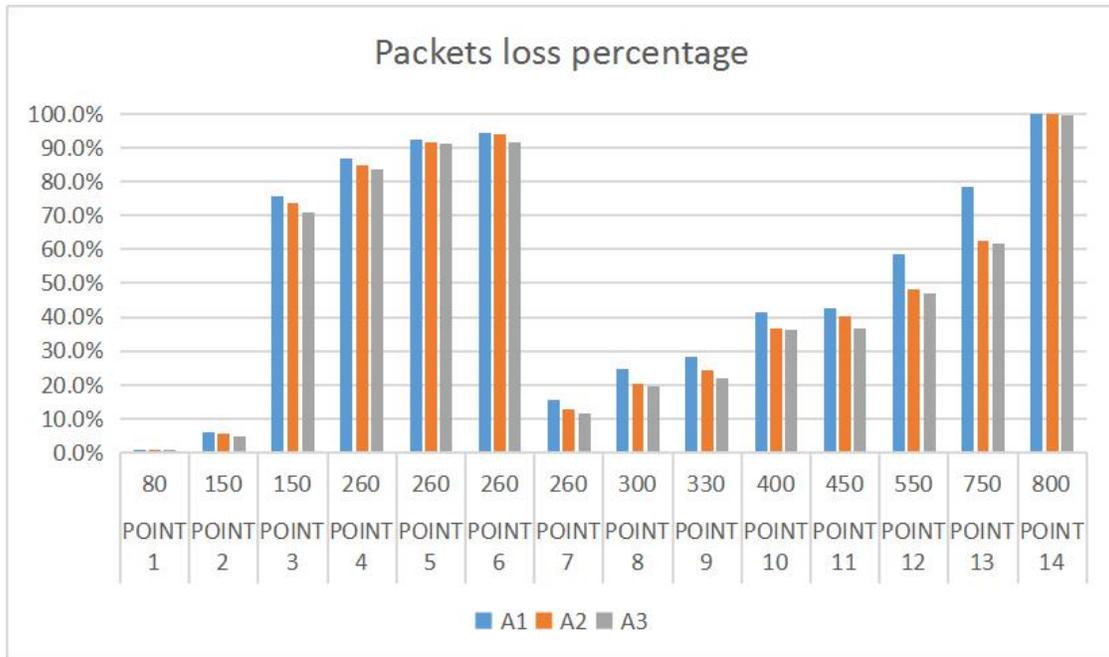


Figure 4.4.3 Comparison of A packets losses percentage among three processes

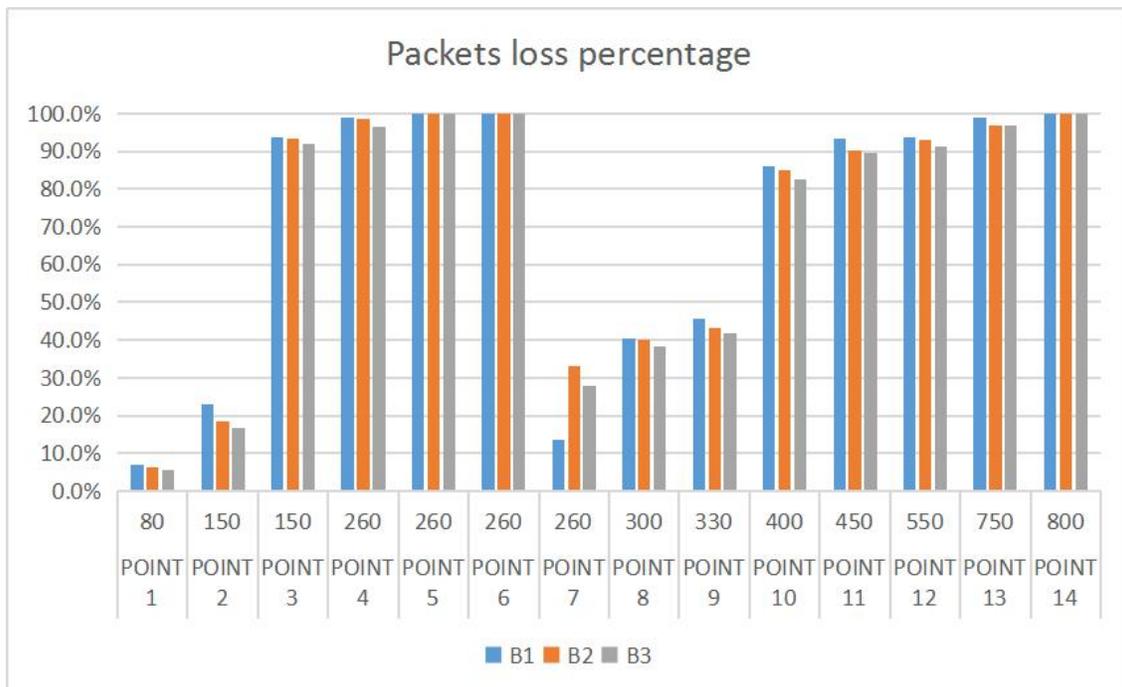


Figure 4.4.4 Comparison of B packets losses percentage among three processes

Through all the pictures shown above, we can have some preliminary judgments. Among the green stars in Fig. 3.2.6.7, we found that the longer the distance, the larger the packets loss; they have a good correlation with the Okumura-Hata model. While for the red stars P3, P4,

P5, P6, we also found that the longer the distance, the larger the packets loss. Therefore, we could say that they matched the Okumura Hata model well, too. On the whole, we found that in a specific area or in a similar environment, the measurements in this thesis can match the Okumura Hata model well.

From Figs. 3.2.6.7 and 3.2.6.8 we could see that in the red area, the buildings are quite dense, so there are more interference. Point 2 and point 3 have the same distance, but their packets loss are totally different. That's because there is an open road between transmitters and point 2, with much less obstacles. As for points 4, 5, 6, 7, although the distances are same, the packets loss are also quite different because of the different building density. From Figs. 4.4.1 and 4.4.2 we also found that the longer the packets length, the larger the packets loss.

What's more, the antennas' direction are set up as the red arrows indicated and the balcony is also at the right side of the building. Since the points on the left side of the red line have much more packets loss and smaller maximum transmission range, we believe that the direction of the antenna can also have big influence on the radio signals transmitting.

# Chapter 5

## Evaluation

In this chapter, we will evaluate the link budget and verify if there is a correlation between link budget and the Okumura-Hata model. What's more, we will also show the conclusions that we got from the output of Chapter 4.

### 5.1 Link budget

Radio Frequency planning for networks such as a cellular telephone system or wireless local area networks are a critical part of the network deployment. Inadequate planning can lead to a waste of resources or insufficient resources. Before planning a network, we should firstly understand the parameters that control the performance of each link. The essential parameters are the noise with the received signal, the received signal strength or any other factors that can cause signal attenuation [22].

A link budget is accounting of all of the gains and losses from the transmitter, through the medium (free space, cable, waveguide, fiber, etc.) to the receiver in a telecommunication system. It accounts for the attenuation of the transmitted signal due to propagation, as well as the antenna gains, feedline and miscellaneous losses [22].

The most general link budget equation is the following:

$$\text{Received Power (dB)} = \text{Transmitted Power (dB)} + \text{Gains (dB)} - \text{Losses (dB)}$$

Besides the link budget, we should also pay attention to the link margin. The link margin is obtained by comparing the expected received signal strength to the receiver sensitivity or threshold. It is a measure of how much margin there is in the communications link between the operating point and the point where the link can no longer be closed [22].

$$\text{Link margin} = \text{EIRP} - \text{L}_{\text{PATH}} + \text{G}_{\text{RX}} - \text{T}_{\text{H}_{\text{RX}}}$$

where,

EIRP is the effective isotropically radiated power in dBW or dBm;

$\text{L}_{\text{PATH}}$  is the total path loss, including miscellaneous losses, reflections, fade margins in dB;

$\text{G}_{\text{RX}}$  is the receive gain in dB;

$\text{T}_{\text{H}_{\text{RX}}}$  is the receiver threshold or the minimum received signal level that will provide reliable operation in dBW or dBm.

In this thesis, the transmitting power is 20 dBm, the receiver sensitivity is -118 dBm and both the transmitter and the receiver antenna gain is 3 dBi. So the maximum link budget without margin that this configuration can offer is

$$\text{Losses} = 20 \text{ dB} + 3 \text{ dB} + 3 \text{ dB} + 118 \text{ dB} = 144 \text{ dBm}$$

But in real world, before evaluating the distance with path loss, there are some other factors need to be considered:

- The loss between transmitter output and antenna can reduce the link budget (  $LOSS_{TX}$  ).
- The loss between receiver input and antenna can reduce the link budget (  $LOSS_{RX}$  ).

If the implementation plan is good enough, the loss of transmitter or receiver is about 1dB. If the circuits between antenna and transmitter or receiver are not matched well, there will be much more loss [23].

Taking all these factors into account provides a link budget that can be used for path loss is:

$$144 \text{ dB} - 1 \text{ dB} - 1 \text{ dB} = 142 \text{ dBm}$$

If the transmitter output impedance matches well with the input impedance, the power can be efficiently transmitted from the transmitter to the antenna. Contrary, if the antenna does not matches well with the amplifier, the transmitted signal will not be effectively transmitted to the antenna, then can reduce the communication range [23].

## 5.2 Okumura-Hata model

As we mentioned above, the formulation for urban environment is:

$$L_P = 69.55 + 26.16 \log f - 13.82 \log h_T - \alpha(h_R) + (44.9 - 6.55 \log h_T) \log d \text{ [dB]}$$

The parameters we used in this thesis are shown as blow:

$L_P$  = Median basic propagation loss in urban areas

$f$  = working frequency (868 MHz)

$h_T$  = base station antenna height (30 m)

$h_R$  = mobile station antenna height (1.5 m)

$d$  = distance between transmitter and receiver (km)

$\alpha(h_R)$  = mobile station antenna height factor

$\alpha(h_R) = 3.2 [\lg(11.75 h_m)]^2 - 4.97 \text{ dB}$   $f \geq 300 \text{ MHz}$ , if  $h_m = 1.5 \text{ m}$ ,  $\alpha(h_m) = 0$ .

From tables that are shown in the Chapter 4, we know that the maximum distance we can reach in this thesis is 0.8 km. Then with all these parameters we can calculate the basic propagation loss is  $L_p=122.6$  dB, which is smaller than the link budget 142 dB.

Then we can guess the maximum link margin can be

$$\text{Link margin} = 142 \text{ dB} - 122.6 \text{ dB} = 19.4 \text{ dBm}$$

Actually, our measurement environment was rainy day, there would be some attenuation due to rain. Moreover, the transmitters are working at the same frequency and use the same channel; if the interference is at the same frequency as the signal of interest, it is called co-channel interference, which is another source of external interference from other transmitters.

In order to provide margin to the operating conditions and operating stable in the real world, we should reduce at least 6 dB from the link budget [23]. So in this thesis, the maximum range of propagation loss is approximately

$$142 \text{ dB} - 6 \text{ dB} = 136 \text{ dBm}$$

So the deviation between predicted and measured path loss value is

$$136 \text{ dB} - 122.6 \text{ dB} = 13.4 \text{ dBm}$$

The output shows that there are still some other factors interfering with the signals. But the common standard deviations between predicted and measured path loss values are around 10 dB to 14 dB [24]. In this thesis, it is about 13.4 dB, which meets our expectations finally.

### 5.3 Analysis of measurement results

In Table 5.2.1 we also evaluated all the corresponding propagation losses of each measurement point with Okumura-Hata model in the urban environment. For each point, only the distance is changed, all other parameters are the same.

Table 5.2.1 Propagation loss of each measurement point with Okumura-Hata model

POINTS	distance(m)	Propagation loss (dB)
POINT 1	80	87.4
POINT 2	150	97.0
POINT 3	150	97.0
POINT 4	260	105.4
POINT 5	260	105.4

POINT 6	260	105.4
POINT 7	260	105.4
POINT 8	300	107.6
POINT 9	330	109.0
POINT 10	400	112.0
POINT 11	450	113.8
POINT 12	550	116.9
POINT 13	750	121.6
POINT 14	800	122.6

In order to have a better comparison of the relationship between packets loss and propagation loss of each point with Okumura-Hata model, we also have a parallel comparison of them among the three measurement processes.

Figures 5.2.1 and 5.2.2 show the parallel comparison of packets loss and propagation loss of each point with Okumura-Hata model among the three processes separately.

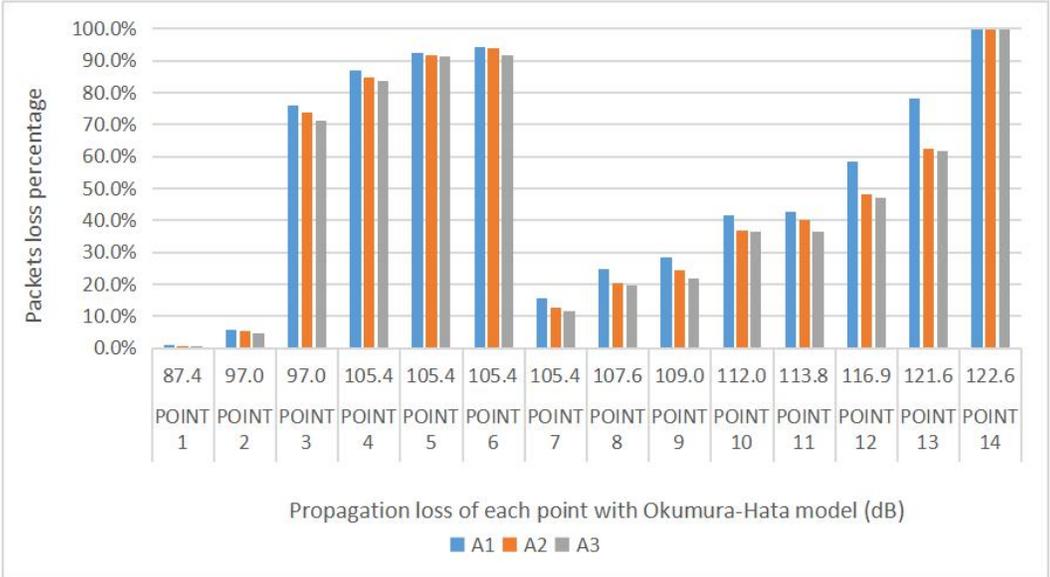


Figure 5.2.1 Comparison of A packets loss percentage with propagation loss of each point among three processes

From Fig. 5.2.1 and 5.2.2 we found that in a specific area or in a similar environment, the larger the path loss of each point with Okumura-Hata model, the larger the packets loss. In fact, as distance increases, the path loss will also increase; the total available link budget is 142 dB, so with path loss increases, the available margin will become smaller and smaller.

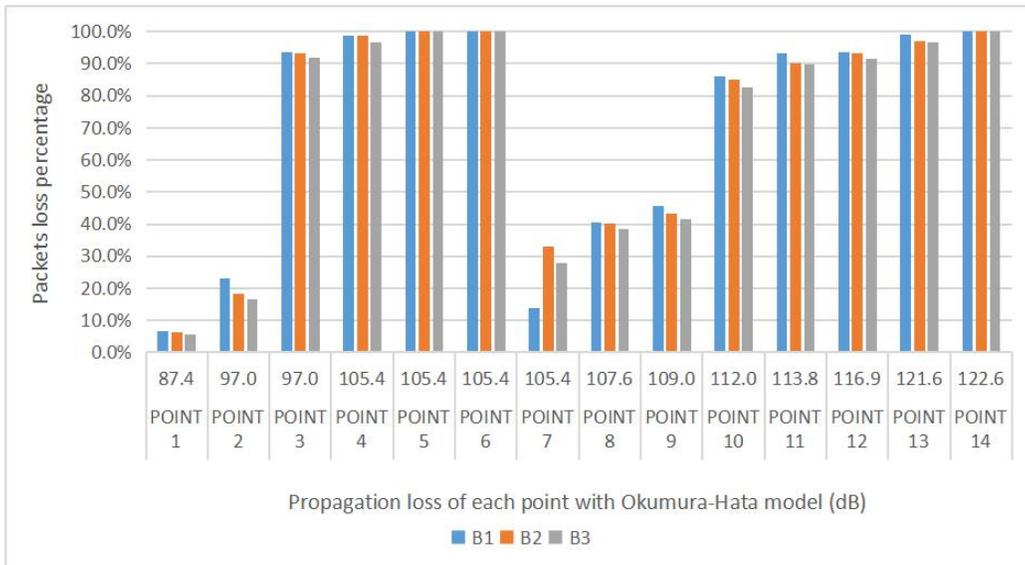


Figure 5.2.2 Comparison of B packets loss percentage with propagation loss of each point among three processes

The green stars in Fig. 3.2.6.7 show that the longer the distance, the larger the packets loss. They have a good correlation with the Okumura Hata model. While for the red stars P3, P4, P5, P6, we also found that the longer the distance, the larger the packets loss. Therefore, we can say that they matched the Okumura Hata model well, too. On the whole, in a specific area or in a similar environment, the measurements in this thesis can match the Okumura Hata model well.

From Fig. 3.2.6.7 and 3.2.6.8 we can see that in the red area, the buildings are quite dense, so there are more interference. Point 2 and point 3 have the same distance, but their packets loss are totally different. That's because there is an open road between transmitters and point 2, with much less obstacles. As for points 4, 5, 6, 7, although the distances are same, the packets loss are also quite different because of the different building density. From Fig. 4.4.1 and 4.4.2 we also found that the longer the packets length, the larger the packets loss.

What's more, the antennas' direction are set up as the red arrows indicated and the balcony is also at the right side of the building. Since the points on the left side of the red line have much more packets loss and smaller maximum transmission range, we believe that the direction of the antenna can also have big influence on the radio signals transmitting.

The manual shows that the maximum distance that the LoRa module we used can transmit is 3 km, with the test conditions of sunny day, open space, transmitting power 20 dBm, receiver's height is more than 2.5 m and data rate 2.4 kbps [25]. But our actual measurement environment was rainy day, urban space, transmitting power 20 dBm, receiver's height is 1.5 m and data rate 19.2 kbps. All these complex conditions led to a maximum transmission distance of only 0.8 km. And from the map we can also find that there are few buildings between the transmitter and point14, which are not high, either. That means fewer interference, so it's the point14 that get the maximum distance. In general, double the data rate will decrease the receiving sensitivity by 2-3 dBm. Since we use data rate 19.2 kbps, the corresponding receiver sensitivity is about -118 dBm.

# Chapter 6

## Conclusions and outlook

The main problem we studied in this thesis is to measure LoRa signal performance in complex urban environment and implementing an interactive interface between the Industruino IND.I/O D21G and other devices.

Firstly, trial experiments with Industruino IND.I/O D21G module are done to be thoroughly familiar with the devices and Arduino IDE environment. The trial experiments are from easy to difficult; at first it is just two trials for one button, then trial for three buttons are done to get the specific performance we wanted.

Next step we tried to observe the serial port values from the webpage by using the Ethernet module, the reason why we did this experiment is that it is a transitional phase from wire connection to wireless connection. We can monitor the values from the serial monitor of Arduino with universal serial bus and we can also monitor the values from webpage by using the module of Ethernet and network cable. But another problem arises, how can we observe the values without network cables but only with wireless? Then we started to use LoRa to solve this problem.

The main problem as we mentioned above is to measure the performance of LoRa and combine it with Industruino IND.I/O D21G. In our expected deployment, we wanted two transmitters in different places but sent the same length of packets each time, and one receiver to receive the data. But we had only one Industruino IND.I/O D21G module, it was not enough; in such situation, we bought three commercial Arduino UNO boards and other related devices to implement the measurement.

During the measurement, we used three LoRa modules which are combined with three Arduino UNO boards separately. Two of them are as transmitters and the other one is used to be the receiver. We set the two transmitters in different places but sent the same length of packets each time. Then we could receive the data at these specific measurement points.

With the measurement results we could calculate the path loss of the maximum distance we observed with Okumura-Hata model is 122.6 dB, and the corresponding maximum link budget without margin is 142 dB; the margin is evaluated to be 6 dB, therefor the maximum range of propagation loss is approximately 136 dB.

Through these experiments and comparisons we could also conclude that in a specific area or in a similar environment,

The longer the distance, the larger the path loss;

The longer the distance, the larger the packets loss;

The longer the packets length, the larger the packets loss;

The more obstacles and sources of interference, the larger the packets loss.

The actual maximum distance that the signal can be measured during our experiments is much shorter than the distance given by the manufacturer, which means the complex urban environment has a great impact on the radio signal.

Finally, we tried to combine the LoRa module with Industruino IND.I/O D21G controller together. We put the LoRa module in the shell of Industruino IND.I/O D21G controller module and use electric soldering iron to connect it to the controller directly, which is much more convenient and stable.

In future works, some other parameters of LoRa systems and various usage conditions should be considered for when the LoRa devices are used under urban environments. For example, the bandwidth (BW) of the module and the spreading factor (SF) can be changed. Therefore, users can find an optimal trade-off between link budget, interference immunity, spectral occupancy, data rate, and transmission range.

## REFERENCES

- [1] Alireza Zourmand, Chan Wai Hung, Andrew Lai Kun Hing, Mohammad AbdulRehman. (2019). "Internet of Things (IoT) using LoRa technology". *IEEE Internet of Things Journal*.
- [2] "Semtech Acquires Wireless Long Range IP Provider Cycleo". *Design And Reuse*. Retrieved 2019-10-17.
- [3] "Internet of Things: the complete IoT guide – benefits, risks, examples, trends" .  
[https://www.i-scoop.eu/internet-of-things-guide/#The\\_exponential\\_growth\\_of\\_the\\_Internet\\_of\\_Things](https://www.i-scoop.eu/internet-of-things-guide/#The_exponential_growth_of_the_Internet_of_Things).
- [4] L. A. T. C. R. P. Workgroup, "LoRaWAN™ 1.1 Regional Parameters," January 2018. [Online]. Available: <https://lora-alliance.org/resourcehub/lorawantm-regional-parameter-v11rb>.
- [5] "Mobile Experts White Paper for LoRa Alliance-Where does LoRa Fit in the Big Picture?," LoRa Alliance, 2015.
- [6] B. Reynders and S. Pollin, "Chirp spread spectrum as a modulation technique for long range communication," in Proc. SCVT, Mons, Belgium, Nov. 2016, pp. 1 – 5.
- [7] N. Sornin (Semtech), M. Luis (Semtech), T. Eirich (IBM), T. Kramp (IBM), O. Hersent (Activity) "LoRa Alliance, LoRaWAN Version: V1.0.1" , 2015 Oct.
- [8] Google, "LoRa packet format," [Online]. Available <https://cn.bing.com/images/search?q=lorawan%20packet%20format&qs=n&form=QBIR&sp=-1&pq=lorawan%20packet%20format&sc=0-35&sk=&cvid=B35C5FA9EC0D438B987C34CC2CFD094A>
- [9] K. Mikhaylov, J. Petaejaevaervi, and T. Haenninen, "Analysis of capacity and scalability of the LoRa low power wide area network technology," in Proc. 22nd Eur. Wireless Conf. (Eur. Wireless), Oulu, Finland, May 2016, pp. 1–6.
- [10] SEMTECH SX1276/77/78 data manual, the first edition, © 2013 Semtech Corporation. Available : [www.semtech.com](http://www.semtech.com)
- [11] Google, "MAC frame format of LoRaWAN". Available: [https://blog.csdn.net/qq\\_23940143/article/details/91380130](https://blog.csdn.net/qq_23940143/article/details/91380130)
- [12] Angel Perlesa, Eva Perez-Marin, Ricardo Mercado, J. Damian Segrelles, Ignacio Blanquer, Manuel Zarzo, Fernando J. Garcia-Diego, "An energy-efficient internet of things (IoT) architecture for preventive conservation of cultural heritage" in The International Journal of eScience, Spain: Universitat Politecnica de Valencia, June 2017.
- [13] M. Bor, J. E. Vidler, and U. Roedig. 2016. LoRa for the Internet of Things. Proc. of EWSN.
- [14] O. Iova, A. L. Murphy, G. P. Picco, L. Ghiro, D. Molteni, F. Ederico Ossi, and F. Cagnacci. 2017. LoRa from the City to the Mountains: Exploration of Hardware and Environmental Factors. In Proc. of MadCom.

- [15] S. Kartakis, B. D Choudhary, A. D. Gluhak, L. Lambrinos, and J. A. McCann. 2016. Demystifying low-power wide-area communications for city IoT applications. In Proc. of WINTECH.
- [16] Google, available : <https://www.semtech.com/lora/lora-applications/smart-cities>
- [17] WIKIPEDIA, "Arduino," [Online]. Available: <https://en.wikipedia.org/wiki/Arduino>.
- [18] Google, available :  
<https://industruino.com/forum/help-1/question/industruino-ind-i-o-d21g-manual-540>
- [19] WIKIPEDIA, "Okumura," [Online]. Available: <https://en.wikipedia.org/wiki/Okumura>.
- [20] Mohamed Shakir, N. Rakesh. "Performance Enhancement of Okamura Model using Correction Factor", 2019 International Conference on Communication and Signal Processing (ICCSP), 2019
- [21] Google, available : <https://centaur.reading.ac.uk>
- [22] Ladislau Matekovits. Marco Allegretti. "Radio Planning", 20016-2017, Propagation Models. DET, Politecnico di Torino.
- [23] Google, available : <https://www.eet-china.com/news/201803091057.html>
- [24] Student paper, Submitted to Engineers Australia
- [25] RF-82U Series Product Manual-V1.2, Web: [www.szrfstar.com](http://www.szrfstar.com)