

# POLITECNICO DI TORINO

## Master of Electronics Engineering Embedded Systems

MASTER THESIS

---

### Design and implementation of a Wireless M-bus - Lora gateway for remote water consumption metering

---



*Supervisor:*

Prof. Daniele Trincherò

*Signature:*

.....

*Applicant:*

Maria Luisa Michelangeli

*Signature:*

.....

Autumn-Winter 2018/2019



MASTER THESIS

**Design and implementation of a  
Wireless M-bus - Lora gateway for  
remote water consumption  
metering**

MICHELANGELI MARIA LUISA



### Abstract

The goal of this thesis is the design and implementation of an interface able to connect a set of water meters equipped with a Wireless M-Bus radio module to a LoRa network. In the future, thanks to this gateway, it could be possible to acquire the meter's readings from remote, without direct human interaction on each measurement device.

This thesis took place in the framework of a collaboration between the SMAT (Società Municipale Acque Torino) and the iXem Laboratory of the Politecnico di Torino.

The water meters in use during the tests were provided by the municipal company, whereas the laboratory and electronic equipment used are iXem's and Politecnico's property.

The architecture of the designed system is shown in figure 0.0.1. It links the water meters to the final application server, through a Wireless M-Bus, and then a LoRaWAN network. In the middle of this interconnection the gateway works as a translator both of frequency and protocol.

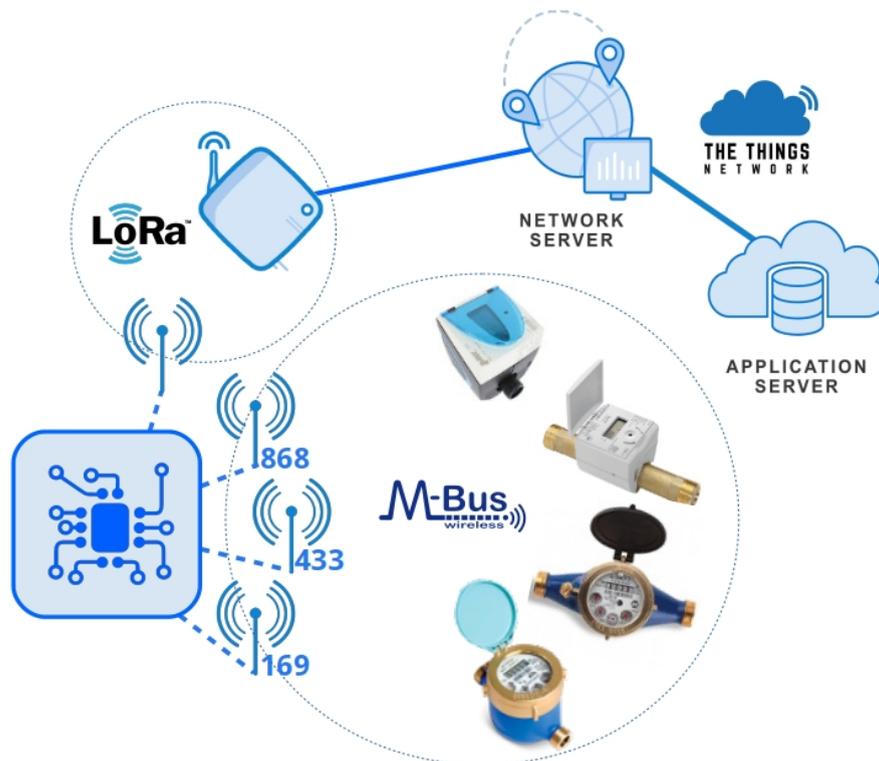


Figure 0.0.1: System architecture overview

This thesis is composed of three main parts:

- Wireless M-Bus: Description of the basics of the Wireless M-Bus communication theory, underlining the specifications used. Illustration of the water meters and the transceivers taken into account during the design. Evaluation boards range measurement (chapters 2 to 4).
- LoRa: Description of the basics of LoRa communication theory, the transceiver and the evaluation board used (chapter 5).
- Prototyping: Description of the prototypes developed: by interconnection of the evaluation boards and with the design of a PCB. Further possible improvements(chapters 6 to 8).

## Contents

<b>1</b>	<b>Introduction</b>	<b>9</b>
1.1	iXemLabs . . . . .	9
1.2	SMAT . . . . .	10
<b>I</b>	<b>Wireless M-Bus</b>	<b>12</b>
<b>2</b>	<b>Wireless M-Bus</b>	<b>12</b>
2.1	Stack architecture . . . . .	13
2.2	WM-Bus and Open Metering System specifications . . . . .	15
2.2.1	Supported CI-fields . . . . .	15
2.2.2	Supported C-fields . . . . .	15
2.2.3	Timing . . . . .	17
2.2.4	Access of a bidirectional meter . . . . .	20
2.2.5	Addressing of a meter . . . . .	21
<b>3</b>	<b>Water meters</b>	<b>23</b>
3.1	Sensus iPerl . . . . .	23
3.2	Maddalena DS TRP MID with Arrow . . . . .	24
3.3	Itron Dihel Hydrus . . . . .	25
3.4	Watertech Pegasus with Smart Metering MultiReader-C-169 . . . . .	26
<b>4</b>	<b>WM-Bus transceivers and evaluation boards</b>	<b>28</b>
4.1	Telit ME70-169 . . . . .	28
4.1.1	AT-Commands (or Hayes command set) . . . . .	29
4.1.2	Evaluation phase and final withdrawal . . . . .	31
4.2	ST-Microelectronics SPIRIT1 . . . . .	32
4.2.1	SPIRIT1 GUI . . . . .	34
4.2.2	IAR project ST-Link programming . . . . .	35
4.3	ST-Microelectronics evaluation boards range test . . . . .	36
4.3.1	STEVAL-IDS001V4M (868MHz) . . . . .	38
4.3.2	STEVAL-IDS001Vx (433MHz) . . . . .	40
4.3.3	STEVAL-IKR002V1 (169MHz) . . . . .	41
<b>II</b>	<b>LoRa</b>	<b>44</b>
<b>5</b>	<b>LoRa and LoRaWAN basics</b>	<b>44</b>
5.1	LoRa and LoRaWAN basics . . . . .	44
5.1.1	LoRa physical layer . . . . .	45

5.1.2	LoRaWAN network architecture . . . . .	46
5.1.3	Classification of end nodes . . . . .	48
5.2	St-Microelectronics B-L072Z-LRWAN1 evaluation board . . . . .	49
5.2.1	Firmware by iXem Labs . . . . .	49
5.2.2	Tests by iXem Labs . . . . .	50
<b>III</b>	<b>Prototyping</b>	<b>53</b>
<b>6</b>	<b>First prototype</b>	<b>53</b>
6.1	HW Structure . . . . .	54
6.2	FW structure . . . . .	55
6.2.1	WM-Bus . . . . .	55
6.2.2	LoRa . . . . .	57
6.3	Power consumption test . . . . .	61
<b>7</b>	<b>Second prototype</b>	<b>62</b>
7.1	Schematic . . . . .	62
7.2	PCB . . . . .	66
7.3	Soldering and Mounting . . . . .	67
<b>8</b>	<b>Further improvements</b>	<b>74</b>

## List of Tables

2.0.1 WM-BUS modes . . . . .	13
2.2.1 Some supported CI-fields . . . . .	15
2.2.2 C-fields master to slave . . . . .	16
2.2.3 C-fields slave to master . . . . .	16
2.2.4 Update interval of consumption data for different resources . . . . .	20
2.2.5 Certifiable device types of OMS-Meter . . . . .	22
4.1.1 Telit ME70-169 supported AT-commands [13] . . . . .	30
4.3.1 Fixed parameters of evaluation board range measurement . . . . .	37
4.3.2 STEVAL-IDS001V4M (868MHz) range measurement . . . . .	38
4.3.3 STEVAL-IDS001Vx (433MHz) range measurement . . . . .	40
4.3.4 STEVAL-IDS001Vx (169MHz) range measurement . . . . .	41
5.1.1 LoRa standard datarates . . . . .	45
7.3.1 IP code classification: Solid protection . . . . .	72
7.3.2 IP code classification: Moisture protection . . . . .	72

## List of Figures

0.0.1 System architecture overview . . . . .	1
1.1.1 iXemLabs logo . . . . .	9
1.2.1 h . . . . .	10
2.0.1 Wireless M-bus logo . . . . .	12
2.1.1 Basic Wireless M-Bus architecture . . . . .	13
2.1.2 WM-BUS frames structure, format A and B . . . . .	14
2.2.1 Timing diagram: Unidirectional meter with synchronous and asyn- chronous transmission . . . . .	17
2.2.2 Timing diagram: RF connection with SND-UD and short TPL . . . . .	18
2.2.3 Timing diagram: Access demand from meter . . . . .	19
2.2.4 WM-BUS address internal content organization overview . . . . .	21
2.2.5 Manufacturer's ID - M-Field value to ISO 22158 capital letters . . . . .	22
3.1.1 Sensus Iperl water meter . . . . .	23
3.1.2 Sensus iPerl WM-Bus packet . . . . .	24
3.2.1 Maddalena Arrow water meter . . . . .	24
3.2.2 Maddalena Arrow WM-Bus packet . . . . .	25
3.3.1 Dihel Itron Hydrus water meter . . . . .	25
3.3.2 Dihel Hydrus WM-Bus packet . . . . .	26
3.4.1 Watertech MultiReader-C-169 water meter . . . . .	27
3.4.2 Watertech MultiReader-C-169 WM-Bus packet . . . . .	27
4.1.1 Telit ME70-169 . . . . .	28
4.1.2 MikroElektronika M-BUS RF click board . . . . .	31
4.2.1 STEVAL IDS001V4M overview . . . . .	33
4.2.2 ST-Microelectronics SPIRIT1 GUI . . . . .	34
4.2.3 ST-Microelectronics WM-Bus stack FSM . . . . .	35
4.3.1 Bird's eye view of the measurement field . . . . .	36
4.3.2 STEVAL-IDS001V4M (868MHz) range measurement . . . . .	39
4.3.3 STEVAL-IDS001Vx (433MHz) range measurement . . . . .	41
4.3.4 STEVAL-IKR002V1 (169MHz) range measurement . . . . .	42
5.0.1 LoRa logo . . . . .	44
5.1.1 LoRa preamble and synchronization fields . . . . .	46
5.1.2 Standard LoRaWAN network architecture . . . . .	47
5.2.1 Map of the RSSI measurements performed by the iXemLabs . . . . .	50
6.0.1 Photo of the first gateway MW-Bus (868MHz) - LORA prototype . . . . .	53
6.1.1 Example of WM-Bus packet output . . . . .	54
6.1.2 The Things Network console overview . . . . .	55
6.2.1 Format of the buffer sent from the WM-Bus board to the LoRa one . . . . .	56
6.2.2 Meters table's element struct: single meter identification parameters and WM-Bus frame variable storage . . . . .	57

6.2.3 Overview of the LoRa prototype 1 board software structure . . . . .	59
6.2.4 Overview of the flags used by LoRa prototype 1 board software structure	60
6.3.1 Power consumption test instrumentation . . . . .	61
7.1.1 Schematic: top overview . . . . .	64
7.1.2 Schematic: WM-Bus MCU overview . . . . .	64
7.1.3 Schematic: WM-Bus 868MHz transceiver overview . . . . .	64
7.1.4 Schematic: WM-Bus 169MHz transceiver overview . . . . .	65
7.1.5 Schematic: LoRa core overview . . . . .	65
7.2.1 Top view of the final PCB . . . . .	67
7.2.2 Bottom view of the final PCB . . . . .	67
7.3.1 868MHz LoRa antenna (left), 169MHz WM-Bus antenna (right) . . .	68
7.3.2 Bare PCB of the second prototype, top and bottom view . . . . .	68
7.3.3 Solder placed on PADS . . . . .	70
7.3.4 Pick and Place task . . . . .	70
7.3.5 Board placed inside the oven . . . . .	71
7.3.6 Photo of the cleaned board . . . . .	71
7.3.7 Photo of the final device . . . . .	73



# 1 Introduction

## 1.1 iXemLabs



Figure 1.1.1: iXemLabs logo

"*iXemLabs*" [1] is a laboratory of Politecnico di Torino, established in 2004 by Daniele Trincherò, Riccardo Stefanelli and Ludwig dei Ghermanti.

During the past years they conducted research projects, aimed to the development of radio systems apt to overcome the digital divide, to improve the efficiency of productive methods and to facilitate sustainability processes. In fact, their motto is "Wireless anytime, anyhow, anywhere, for anyone".

Their main activity is the development of wireless sensor networks with low cost, low power consumption and wide range. Thus the main fields of action are in the IOT domain: smart metering, smart city, smart agriculture and more.

In the lab two main projects are in progress:

- **iXemWine**: Innovative platform for precision agriculture that help farmers in everyday activities in the vineyard. It makes use of low power sensors that control environmental and meteorological parameters.
- **SenzaFiliSenzaConfini** (literally "no wires no boundaries"): Association of social promotion registered as internet service provider that aims to reduce the digital divide in rural zones. It established the widest network owned by a University institution in the world.

During the final months of the development phase, this thesis project has become the third main activity of the laboratory, under the name of **Me-RI** (Metering Remote Interpreter).

## 1.2 SMAT



Figure 1.2.1: SMAT logo

SMAT (Società Municipale Acque Torino) [2] is the municipal water company of the Turin metropolitan area. They collaborate with universities, national and international research centers, and other industries and companies working in the same sector.

Its research center was founded in 2008, but even in the previous years they dedicated resources to improve their hydro-systems. They also produce water for the ISS (International Space Station), and they are studying methods to produce water for long lasting space missions.

In 2018 they won the "TOP UTILITY - research and innovation" award.

This thesis is based on the collaboration between SMAT and the iXemLabs. It aims to improve their integrated system of tele-control, which monitors the behavior of their water supply network: data such as the water consumption of each unit, possible leakage and various problems will be available in a daily update. Thus the the tele-control will be more reliable and the billing system for their user will be improved.



## Part I

# Wireless M-Bus

## 2 Wireless M-Bus



Figure 2.0.1: Wireless M-bus logo

Wireless M-Bus [3] is the standard communication protocol widely used in Europe for energy and water meters. It is based on the EN13757-4 document [4] and it makes use of the license free sub-GHz frequencies in the bands 169MHz, 434MHz and 868MHz. Despite its growing popularity, Wireless M-Bus continues to remain a standard without an independent certification authority to accredit product compliance. Therefore each EU state has its own additional requirements on top in order to optimize the standard for their needs and environment. For this reason smart meters may not be truly inter-operable across all Europe.

In Italy the CIG requirements have to be applied: that means that the WM-Bus has to follow the UNI-TS-1129 standard which limits the maximum admitted EIRP to +27dBm. The protocol is founded on a star network, with a 32kB stack size. The channel access has to be based on ALOHA or other Listen Before Talk mechanisms to minimize collisions, but there could be a broadcast window for firmware download. Moreover the DLMS/COSEM is the optimized application layer for the standard.

There are few modes that can be used when a WM-BUS node is configured, and all have a unidirectional and bi-directional sub-methods (see table 2.0.1).

Table 2.0.1: WM-BUS modes

Mode	Band	Comments
S	868	Meter send data few times a day
T	868	Meter send data several times a day
C	868	Higher DR then T
N	169	Long range, narrow band
R	868	Collector reads multiple meters on more channels
F	434	Frequent bi-directional communication

On the hardware side, there are several options available for WM-Bus metering solutions. There are four core components required for a high-performance WM-Bus solution:

- A low power microcontroller;
- A high-performance sub-GHz transceiver;
- A modular stack architecture;
- Development tools to design and configure the metering system.

Selecting the right MCU and radio transceiver is a key point when designing smart metering devices featuring wireless connectivity and energy-efficiency. In the following chapters the choices made for this project will be described.

## 2.1 Stack architecture

The basic stack architecture of the Wireless M-Bus protocol is straightforward. There are three layers: application, data link and physical (see figure 2.1.1).

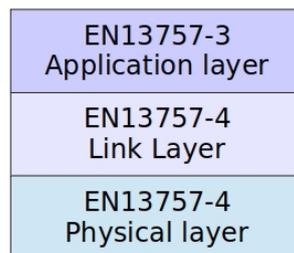


Figure 2.1.1: Basic Wireless M-Bus architecture

The physical layer defines the frames structure, which includes the preamble (header plus synchronization fields), payload and postamble. As stated in the EN13757-4 standard the various fields have different length according to the configured mode. There are also two different frame formats A and B (see figure 2.1.2). In both of them:

- The first block has 10 fixed bytes which include the frame length (L-field), the control information (C-field), the manufacturer ID (M-field) and the sender address (A-field);
- The second block starts with the CI-field which declares the data structure and continues with the Data-field, ending with the Cyclic Redundancy Check (CRC-field);
- The third block is optional and contains an additional Data-field and a CRC-field.

L-field	C-field	M-field	A-field	CRC-field	First block (A)
1 byte	1 byte	2 bytes	6 bytes	2 bytes	
CI-field	Data-field			CRC-field	Second block (A)
1 byte	15 or if it's the last block $((L-9)\text{ modulo }16)-1$ bytes			2 bytes	
Data-field				CRC-field	Optional block (A)
16 or if it's the last block $((L-9)\text{ modulo }16)$ bytes				2 bytes	
L-field	C-field	M-field	A-field	First block (B)	
1 byte	1 byte	2 bytes	6 bytes		
CI-field	Data-field			CRC-field	Second block (B)
1 byte	115 or if it's the last block $(L-12)$ bytes			2 bytes	
Data-field				CRC-field	Optional block (B)
$(L-129)$ bytes				2 bytes	

Figure 2.1.2: WM-BUS frames structure, format A and B

The data link layer connects the application and physical ones by providing transfer data services. Moreover it generates the outgoing CRC and frames and verifies them for incoming messages, provides the WM-Bus addressing and acknowledges transfers for bidirectional communication modes.

## 2.2 WM-Bus and Open Metering System specifications

On top of the basic normative, the gateway WM-Bus to LoRa has to work with the OMS (Open Metering System) standard [5] in order to communicate with the given water meters. Those specifications set boundaries for the frames' internal content, physical requirements, application protocol, security and more. In the following sub-sections some of those aspects will be analyzed.

### 2.2.1 Supported CI-fields

The CI-field (refer to figure 2.1.2) is the application header and it defines the type of data in the application data payload. Moreover it declares (if needed) the communication layer, transport direction and application protocol.

Only some values are allowed for the CI-field. Table 2.2.1 shows some possibilities, for the complete reference see the OMS standard [5].

Table 2.2.1: Some supported CI-fields

CI-field	Function	Up/Down link
53h	Application reset or select	Down
6Ch	Time Synch	Down
72h	Response	Up
74h	Alarm	Up

### 2.2.2 Supported C-fields

The C-field is used to declare the message type. In fact, there are different message types for data exchange:

- Spontaneous message without reply;
- Commands from master to slave with acknowledge;
- Data request with response from slave to master;
- Commands from master to slave with an immediate response;
- Special messages for installation or alarm.

The C-field may be generated by the master (gateway or other communication device) and shall be accepted by the slave (meter or actuator). Others instead are sent spontaneously or as a reaction by the slave.

In tables 2.2.2 and 2.2.3 the various possibility for the C-field are shown.

Table 2.2.2: C-fields master to slave

Message type	C-field	Comments	Expected response
SND-NKE	40h	Reset after communication	-
SND-UD2	43h	Send command with response	RSP-UD, NACK
SND-UD	53h, 73h	Send command	ACK, NACK
REQ-UD1	5Ah, 7Ah	Alarm request	ACK, RSP-UD
REQ-UD2	5Bh, 7Bh	Data request	RSP-UD
ACK	00h	Acknowledge reception ACC-DMD	-
CNF-IR	06h	Confirm registration of slave	-

Table 2.2.3: C-fields slave to master

Message type	C-field	Comments	Expected response
SND-NR	44h	Send spontaneous/periodical appl data without request	-
SND-IR	46h	Send installation data	CNF-IR, SND-NKE
ACC-NR	47h	Signal empty transmission or provide access bidirectional meter	-
ACC-DMD	48h	Request application data	ACK
ACK	00h, 10h, 20h, 30h	Acknowledge reception of SND-UD or response to REQ-UD1 if no alert	-
NACK	01h, 11h, 21h, 31h	Not-Acknowledge in case of error	-
RSP-UD	08h, 18h, 28h, 38h	Response of data after request	-

Only messages of type RSP-UD and SND-NR can be used to send data from the meter to the gateway. Moreover the spontaneous/periodical application data transmission must be supported by both uni and bidirectional slaves.

If the gateway receives a request of communication from a meter (SND-IR), it must generate a SND-NKE or CNF-IR in order to confirm the ability to receive. In order to avoid collisions when trying to access a specific slave, each meter must be assigned to a single gateway. In case of an erroneous multiple assignment each gateway shall support a collision avoidance mechanism.

### 2.2.3 Timing

The communication protocol should pursue the following timing diagram to properly exchange data between a meter and a collector. These examples are mainly for the S and T modes.

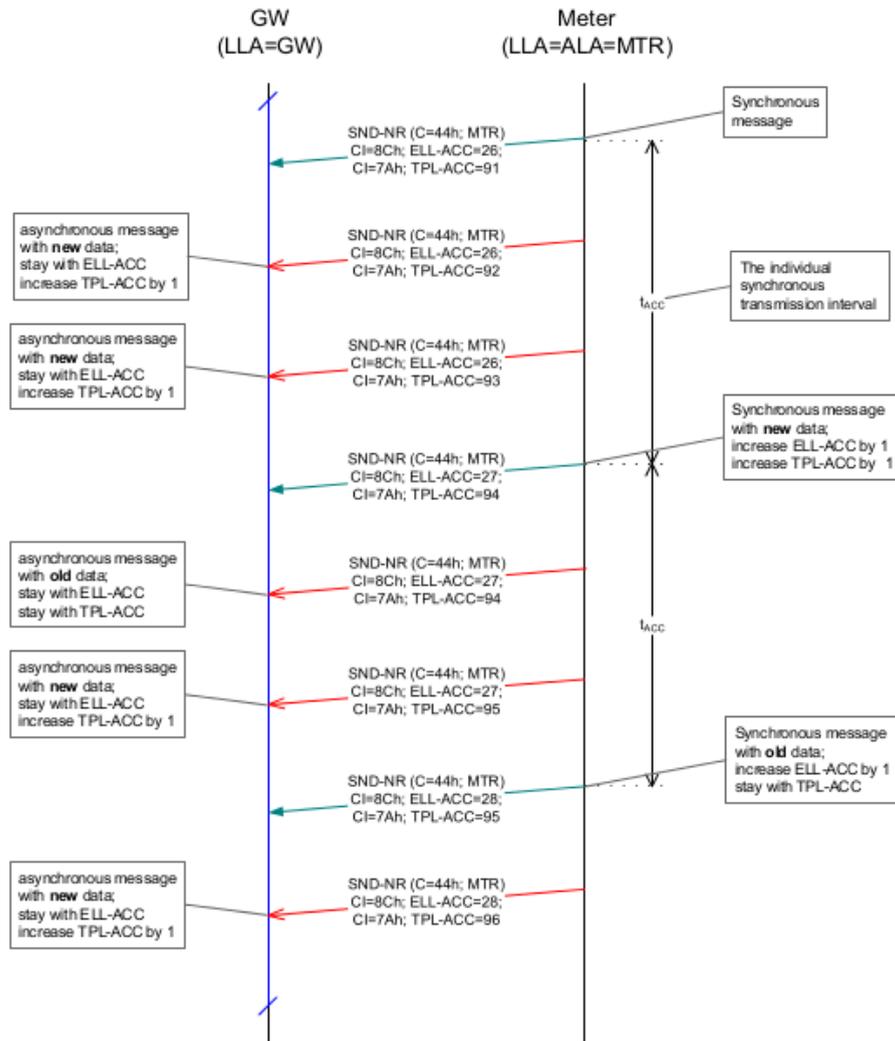


Figure 2.2.1: Timing diagram: Unidirectional meter with synchronous and asynchronous transmission

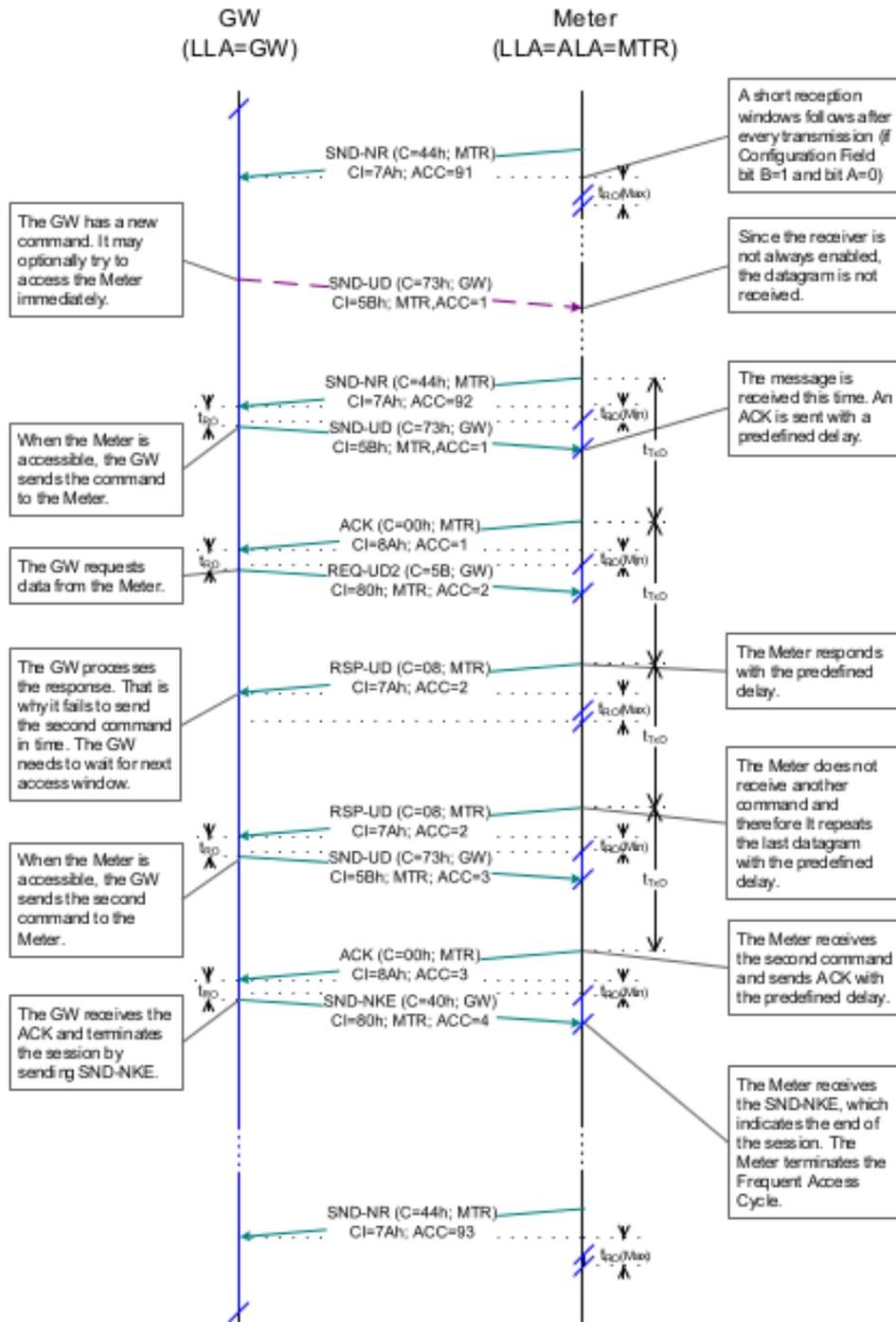


Figure 2.2.2: Timing diagram: RF connection with SND-UD and short TPL

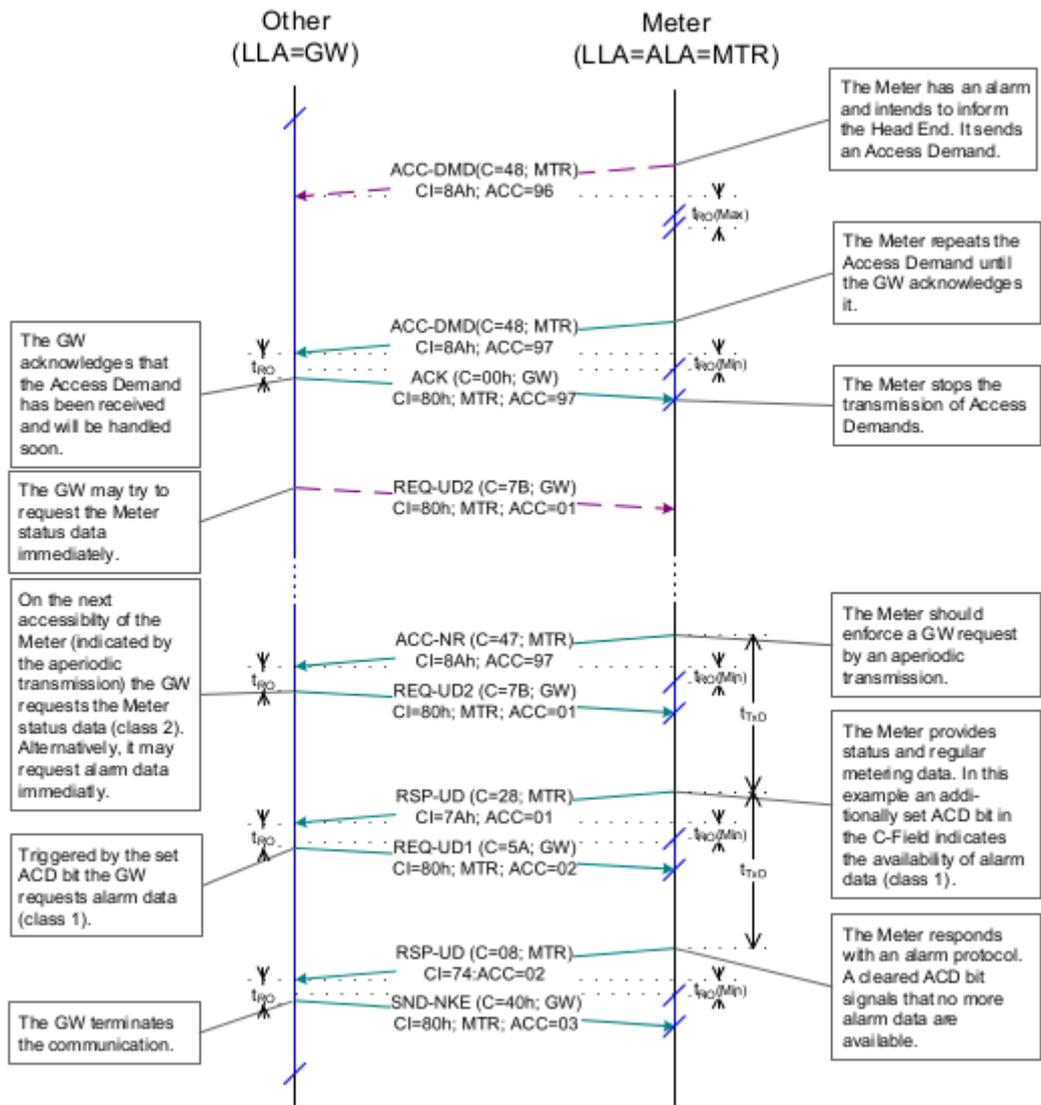


Figure 2.2.3: Timing diagram: Access demand from meter

The transmission intervals cannot be chosen arbitrarily, they have to follow some requirements. Asynchronous transmissions are allowed, but they need to follow the diagram 2.2.1 and respect the interval given by  $t_{ACC}$ . Meter message types RSP-UD, ACK, NACK, SND-IR shall be transmitted asynchronously.

An update of consumption data with every synchronous transmission is recommended. However it shall be transmitted at least with the average interval as listed in table 2.2.4.

Table 2.2.4: Update interval of consumption data for different resources

Meter type	Avg update interval maximum [minutes]	Visualization interval for provider [hour]
Electricity	7.5	1
Gas	30.0	1
Heat (district heating)	30.0	1
Water/warm water	240.0	24
Heat cost allocators	240.0	24
Heat/cold (sub metering)	240.0	24

Furthermore there should be a minimum time delay between successive transmissions different for each mode:

- Mode T: 0.72s meter to other, 1.8s other to meter;
- Mode C: 0.72s meter to other, 3.6s other to meter;
- Mode S: 1.8s in both directions.

#### 2.2.4 Access of a bidirectional meter

If a gateway wants to transmit a message to a meter it checks the Link Control Bits present inside the C-field to know whether the meter is accessible.

- LCB = 00: Meter do not provide access window (unidirectional);
- LCB = 01: Meter do not provide access window after this transmission even if it supports bidirectional communication in general;
- LCB = 10: Meter provide a short access window immediately after this transmission;
- LCB = 11: Meter provide unlimited access at least until the next transmission (only main powered devices).

Battery powered bidirectional devices usually are very restricted in power consumption. For this reason they usually provides a short access window only immediately after the transmission. Thus the gateway (as master) may initiate the communication to the meter (as slave) only during this timeslot.

### 2.2.5 Addressing of a meter

Each WM-BUS transfer has to be addressed in the same way. The fields involved in the process are M and A (see figure 2.1.2).

The two bytes of M-field contain the manufacturer ID, while the A-field incorporates the serial number, the version (or model) and the type of the meter (water, gas, electricity...). This last field content order is not univocal for all the water meters, so figure 2.2.4 has to be intended just as a graphical overview of the header's internal organization.

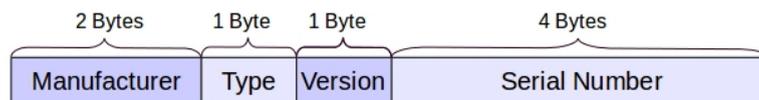


Figure 2.2.4: WM-BUS address internal content organization overview

The manufacturer ID consists in a two bytes value computed starting from the 3 capital letters univocal ISO 22158 [6] code of the company. This code is registered and administered by the FLAG ID association [7]. The equation 2.2.1 [8] connects those two values:

$$M_f = [A - 2^6] * 2^5 * 2^5 + [B - 2^6] * 2^5 + [C - 2^6] \quad (2.2.1)$$

where  $M_f$  represent the two bytes of M-field, and A, B and C the three capital letters.

The formula 2.2.1 has been then reversed to decrypt the M-field and get easier to read informations about the manufacturer. Starting from the assumption that the capital letters in ASCII are encoded between the decimal values 65 (binary 01000001) and 90 (binary 01011010), subtracting  $2^6$  means that the difference can be encoded on 5 bits. Then each multiplication by  $2^5$  shifts that binary value to the left by five bits. The final sum of the three letters modified and shifted is on 15 bits.

At the end the two bytes of the M-field contains the bits of the letters in the order shown by the following image (the communication uses big endian).

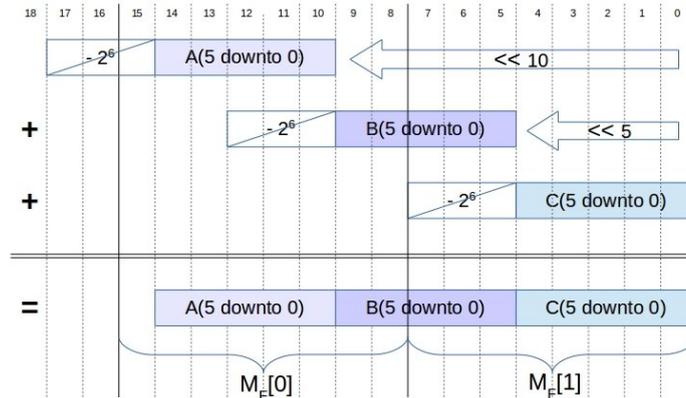


Figure 2.2.5: Manufacturer’s ID - M-Field value to ISO 22158 capital letters

By means of masks, shifts and sums the three letters are finitely decrypted in equation 2.2.2.

$$\begin{aligned}
 A &= ((M_f[0] \& 0xFC) / 2^2) + 2^6 & (2.2.2) \\
 B &= ((M_f[1] \& 0xE0) / 2^5) + ((M_f[0] \& 0x03) * 2^3) + 2^6 \\
 C &= (M_f[1] \& 0x1F) + 2^6
 \end{aligned}$$

Each manufacturer then describes the version of the device in a byte (“Version” frame in figure 2.2.4) and the serial number in 4 bytes using a proprietary encoding. The “Type” frame is instead described inside the OMS standard [5]. The byte value is restricted to the ones listed in table 2.2.5.

Table 2.2.5: Certifiable device types of OMS-Meter

Meter type	Code [hex]
Electricity meter	02
Gas meter	03
Heat meter	04
Warm water meter (30-90°C)	06
Water meter	07
Heat cost allocator	08
Cooling meter	0A
Combine heat/cooling meter	0D
Hot water meter (>90°C)	15
Cold drinking water meter	16
Waste water meter	28

### 3 Water meters

Four water meters have been provided for the purpose of this thesis by the SMAT group. Three of them work in the 868MHz band, the latter in the 169MHz one.

There was some communication problem between the two actors, so not all the datasheets of the radio module of the meters have been provided. For this reason some of the information illustrated in the following pages were inferred using the WM-BUS theory from the frames received from the devices.

For the same reason the WM-Bus packets have never been decrypted, since the encryption keys are unknown to the writer of this thesis.

All of the present meters are built for water consumption and should be pipe mounted in a domestic/city environment.

Moreover they all ensures a unidirectional communication with the WM-Bus - LoRa gateway in project.

#### 3.1 Sensus iPerl



Figure 3.1.1: Sensus Iperl water meter

The Sensus iPerl (figure 3.1.1) water meter uses remanent magnetic field technology which provides a linear measurement range even down to very low flow rates. The magnetic field acting on the water flowing through the channel generates an electrical voltage; this is proportional to the velocity of the water (principle of magnetic-inductive flow measurement). It has up to 15-year expected operational life depending on communication frequency.

It is equipped with low power 868 MHz radio module compliant with the WM-Bus OMS specifications. It is a unidirectional meter working in WM-Bus mode T1. The

packets sent by it are configured with the structure described in figure 3.1.2 and with a periodicity of 1 hour.

Manufacturer	Serial Number	Type	Ver	
0xAE 0x4C	0x87 0x64 0x43 0x20	0x68	0x07	PAYLOAD (4+16Bytes)

Figure 3.1.2: Sensus iPerl WM-Bus packet

In the image the manufacturer ID field correspond to the FLAG ID "SEN", the serial number is written LSB first in the frame and it is the one printed on the case of the device. Moreover the 0x07 Type field correspond to the one expected as it works on the OMS specifications.

The payload contains the following information encrypted using AES-128: Counter (1Byte), Status (1Byte), Signature (2Bytes), AES-Verify (2Bytes), DIF/VIF reading (2Bytes), Reading LSB first (4Bytes), DIF/VIF flow (2Bytes), Flowrate (2Bytes), Padding (4Bytes).

For more information about this meter it is suggested to consult the datasheet and the manufacturer website [9].

### 3.2 Maddalena DS TRP MID with Arrow



Figure 3.2.1: Maddalena Arrow water meter

The Maddalena DS TRP MID (figure 3.2.1) is a sealed register permanently protected dry dial multi-jets water meter. It is designed to combine high performance

at low flow rates and maximum resistance to high flow rates and pressure. It is equipped with the Arrow module, which consists on some electronic sensors to assess the water flow and a low power 868MHz radio module compliant with the WM-Bus OMS specifications. It has up to 12-year expected operational life. In the overall it is a unidirectional meter working in WM-Bus mode T1. The packets sent by it are configured with the structure described in figure 3.2.2 and with a periodicity of 15 seconds.

Manufacturer	Serial Number	Type	Ver	
0x24 0x34	0x69 0x10 0x41 0x16	0x81	0x07	PAYLOAD (4+20Bytes)

Figure 3.2.2: Maddalena Arrow WM-Bus packet

In the image the manufacturer ID field correspond to the FLAG ID "MAD", the serial number is written LSB first in the frame and it is the one printed on the case of the device. Moreover the 0x07 Type field correspond to the one expected as it works on the OMS specifications.

We don't have any information regarding the content and the encryption of the payload. We only know that its length is 20 bytes, detail discovered empirically. For more information about this meter it is suggested to consult the manufacturer website [10].

### 3.3 Itron Dihel Hydrus



Figure 3.3.1: Dihel Itron Hydrus water meter

The Dihel Hydrus (figure 3.3.1) is an electronic water meter designed to determine potable water consumption. It is based on ultrasonic technology which works reliably and yields precise results even if exposed to dirty water and sand. Air in the pipe is not measured. This means that the meter has been filled with water in order to be used in the laboratory.

It is equipped with a 868MHz low power radio module compliant with the WM-Bus OMS specifications. It has up to 15-year expected operational life. It is a unidirectional meter working in WM-Bus mode T1. The packets sent by it are configured with the structure described in figure 3.3.2 and with a periodicity of 8 seconds.

Manufacturer	Serial Number	Type	Ver	
0xA5 0x11	0x80 0x95 0x64 0x58	0x70	0x07	PAYLOAD (4+64Bytes)

Figure 3.3.2: Dihel Hydrus WM-Bus packet

In the image the manufacturer ID field correspond to the FLAG ID "DME", the serial number is written LSB first in the frame and it is the one printed on the case of the device. Moreover the 0x07 Type field correspond to the one expected as it works on the OMS specifications.

The payload contains the following information encrypted using AES-128: Counter, current volume, annual accounting data, current flow rate, battery lifetime, water temperature in °C, date and log volume.

For more information about this meter it is suggested to consult the datasheet and the manufacturer website [11].

### 3.4 Watertech Pegasus with Smart Metering MultiReader-C-169

The Watertech Pegasus (figure 3.4.1) is a mechanical water meter. It's a multi-jet turbine water meter, for cold potable water. The water consumption measured is directly readable thanks to numbered rollers encapsulated in a special liquid for protection and lubrication.



Figure 3.4.1: Watertech MultiReader-C-169 water meter

It is equipped with Smart Metering MultiReader-C-169. That device consists on some electronic sensors to assess the water flow and a low power 169MHz radio module compliant with the WM-Bus OMS specifications. It has up to 10-year expected operational life. It is a unidirectional meter working in WM-Bus mode N1. The packets sent by it are configured with the structure described in figure 3.4.2 and with a periodicity of 6 hours, night time excluded.

Manufacturer	Serial Number	Type	Ver	
0xA5 0x11	0x80 0x95 0x64 0x58	0x70	0x07	PAYLOAD (4+64Bytes)

Figure 3.4.2: Watertech MultiReader-C-169 WM-Bus packet

In the image the manufacturer ID field correspond to the FLAG ID "SMM", the serial number is written LSB first in the frame and it is the one printed on the case of the device. Moreover the 0x07 Type field correspond to the one expected as it works on the OMS specifications.

The payload contains the following information encrypted using AES-CBC (Mode 5): current date and volume at transmission time, date and volume at midnight of the same day, of the previous day, of two and three days before, 2am-5am consumption, minimum and maximum flow rate from midnight of the transmission day, daily total volume and other diagnostic information.

For more information about this meter it is suggested to consult the datasheet and the manufacturer website [12].

## 4 WM-Bus transceivers and evaluation boards

Knowing the basic principles of the WM-Bus communication and some details about the water meters radio system, some suitable development boards have been identified to try to establish a link to the meters in order to receive the packets that they send.

Unfortunately there's not an wide range of possibilities on the market, in fact some of them could not be sent to Italy from the dispatcher or the shipping time is a few months long.

For this reason only two manufacturer have been identified: Telit and St-Microelectronics. The first one produces a series of chip with the WM-Bus stack already flashed on their ROM and programmable through AT-Commands. The second manufacturer produces a wide range of development boards based on their low-power sub-GHz transceiver SPIRIT1 and programmable through St-Link debug probe with a series of software examples.

At the end of the evaluation phase, after the initial tests, the ST-Microelectronic products have been chosen, due to difficulties with the Telit chips.

In the following sections a detailed description of the two systems can be found.

### 4.1 Telit ME70-169

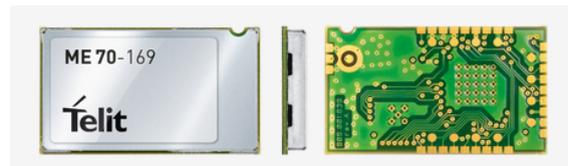


Figure 4.1.1: Telit ME70-169

The Telit ME70-169 (figure 4.1.1) is a certified WM-Bus module compliant with the EN13757 part 4 2013 standard [4]. It operates in the 169MHz band, running N mode protocols. It is suited for use in one or two-way data links with water meters.

There are more versions available covering all the WM-Bus bands, but for the scope of th project only the 169MHz version has been bought for testing due to costs limitations.

The datasheet [13] underlines some expected features that pinpoint this chip as desirable for the gateway in development:

- Range up to 20km;
- Power supply 2.3 to 3.6V;

- Transmitted power up to 1W with sensitivity up to -119 dBm;
- Current consumption of less than  $1.5\mu\text{A}$  in sleep mode with an RTC clock running;
- Configurable output power from 0dBm to 30dBm;
- Serial Interface RS232 TTL (Tx, Rx, CTS, RTS);
- Multichannel mode available;
- Multiple Listen-Before-Talk (LBT) methods (ALOHA, AFA);
- Frequent-Access-Cycle (FAC) management.

It is controllable via AT-Commands, which must be sent through UART serial line, with a default baudrate of 19200 bit/s, word length of 8 bits, no parity bit and 1 stop bit.

Its registers can be read and written to program the device as needed. The parameters that could change are: the mode (N unidirectional/bidirectional), activation or deactivation of C, M, A or CI-field, channel, preamble and postamble length, output power, serial speed, meter key and address (if used as a meter), UART parameters, LBT and FAC options.

#### 4.1.1 AT-Commands (or Hayes command set)

The AT-Commands (or Hayes command set) [14] are a specific command language consisting on a series of short characters strings sent and received by an host to interact with a connected device.

The Telit ME70-169 can receive a set of those commands through the UART, with some expedients: the command, except the escape code '+++', must always be ended with a carriage return <CR> (character '\r'), the string must be sent as a whole with a maximum time distance of 5ms between one character and the other. If those precautions are not followed, the device doesn't recognize the command but enters serial timeout. If a command is correctly received, the device answer on the same serial line to the host.

On the next page a complete list of all the AT-Commands with their responses and functionalities could be found in table 4.1.1. They can be divider in 3 blocks:

- +++ and ATO: commands used to enter and exit from Hayes mode. With ATO the device starts working in WM-Bus mode as stated in the internal registers.

- Read and Write: commands used to read or write a specific register, retrieve information from the device such as date and time or reset the registers to the default values.
- ATTx: commands used to test the radio functions. They start transmitting continuously after the "OK". The device stops the radio function only when a command is received on the UART.

Table 4.1.1: Telit ME70-169 supported AT-commands [13]

Command	Response	Description
+++	OK	Enter in Hayes mode to configure parameters, inactivates radio functions, no carriage return
ATO	OK	Go in operating mode
AT/V	pp.UP3.MM.mm.Bbbb <CR>pp.B00.NN	Display modem firmware and bootloader version
ATSn?	Sn=x or ERROR	Display content of register n
ATSn=m	OK or ERROR	Write value m inside register n
ATR	OK	Reset modem parameters and meter list
ATM	OK	Reset only the meter list
ATBL	OK	Runs the bootloader to update firmware
ATDT=MMDD hhmmYYss	OK	Set date and time as specified
ATDT?	MMDDhhmmYYss	Display date and time
ATT	OK	Continuous modulating carrier simulating transmission of '01' data
ATT0	OK	Pure carrier transmission at center frequency
ATT1	OK	Continuous modulated carrier simulating transmission of '01110010' if 4GFSK
ATT2	OK	As ATT
ATT3	OK	Continuous modulated carrier of random data

#### 4.1.2 Evaluation phase and final withdrawal

Due to the ease of use and its features the Telit ME70-169 seemed the best choice for the application in design. Nevertheless a lot of problems were encountered during the preliminary and evaluation phases, thus leading to a final withdrawal of the transceiver in favor of the ST-Microelectronics technology.

First of all, the purchase of the device seemed impossible: it is difficult to find both the transceiver and the existing development kit on the market.

Fortunately we found the MikroElektronika M-BUS RF click board [15], a very simple circuit that allows the access to the transceiver's pins and connects it to the antenna (see figure 4.1.2).



Figure 4.1.2: MikroElektronika M-BUS RF click board

The communication of the AT-commands between the PC and one of these boards has been tested. Promptly a new problem arose: the Telit software includes a GUI which helps to connect and send commands to the ME70-169, but it only works with the development board kits. There is no way to connect the bare chip to the manufacturer tool as there is not the microcontroller interface, so other ways have been tried.

The initial step was to use Putty together with a USB-to-serial 3.3V adapter to connect the PC to the transceiver's UART, but it was not practical as the AT-Command has to be sent as a whole string in order to respect the timeout, and it is not straightforward to do so manually. Moreover there is no way to have an history of the received and sent messages as all the commands and responses do not include a line feed.

Thus an interface using Arduino technology has been developed. The Arduino board sends and receives commands and responses through its serial port and displays them on the serial monitor of the Arduino IDE. In this way there can be a chronology of the commands and it is also possible to check the behavior of the chip through the responses. The final version of the Arduino interface includes a menu to help selecting between the AT-commands without the use of the datasheet.

At this point new difficulties were encountered: to have a receiver and a transmitter two Telit transceivers have to be connected to the board, but the Arduino UNO

[16] has only one serial port, so the Arduino Mega [17] was taken into account. Nevertheless both the Arduino models work on 5V signals, so a voltage divider has been inserted on the serial line to translate the signals on 3.3V. However the two boards are not able to supply enough power to the Telit transceiver in order to activate its radio communication tests, so an external power supply has been included.

At this point the system was really complex, so the Arduino Intel Galileo board [18] has been taken into consideration in order to avoid the use of the external devices. However one of its serial lines was connected to the 5V audio jack output and moreover it suffers of over heating problems, so it was not a good choice.

For this reason the Arduino DUE board [19] working on 3.3V has been used. The external power supply was still necessary, but the voltage divider could be avoided. Then, during the test of the radio functions, the final problem which led to the withdrawal of the technology was found.

The Telit allows to test the radio transmission with a set of commands (see table 4.1.1). The signal emitted should be continuous until the user sends a stop character via UART, but that did not happened. Using the spectrum analyzer Agilent E4402B and the Saleae logic analyzer it was found that the radio module stops transmitting after a short and not predictable delay.

The radio communication has then been tested using the examples of configuration and communication stated in the datasheet, but the link between the two boards was never established even if the steps were followed in details.

The ST-Microelectronics STEVAL-IKR001V1 evaluation kit has then been bought and used in the frequency 169MHz.

## 4.2 ST-Microelectronics SPIRIT1

ST-Microelectronics manufactures a complete set of evaluation boards to cover the whole set of WM-Bus frequencies. It also provides a set of proprietary software interfaces to communicate with them.

For those reasons it is a very strong competitor on the WM-Bus applications.

The evaluation boards taken into account for this project are different for each Wireless M-Bus frequency:

- **169MHz:** STEVAL-IKR002V1 [20]. It is a kit containing two daughter boards mounting the SPIRIT1 transceiver linked to the antenna SMA connector through a discrete RF circuit, and two STM32L1 microcontroller-based motherboards featuring a USB connector for PC GUI interaction. A JTAG connector allows the update of specific firmware on the microcontroller.

- **433MHz:** STEVAL-IDS001Vx [21]. It is an USB dongle mounting the SPIRIT1 transceiver linked to a ceramic antenna through a discrete RF circuit, and a STM32L1 microcontroller. It features a USB connector for PC GUI interaction and firmware updates.
- **868MHz:** STEVAL-IDS001V4M [22] (figure 4.2.1). It is an USB dongle mounting the SPIRIT1 transceiver embedded into the SPSGRF-868 module together with a balun and a chip antenna. Moreover a STM32L1 microcontroller is present on the board. It features a USB connector for PC GUI interaction and firmware updates.



Figure 4.2.1: STEVAL IDS001V4M overview

As mentioned all the evaluation boards mount a low power RF SPIRIT1 transceiver [23]. This device is intended for RF wireless applications in the sub-GHz band. It is designed to operate both in the license-free ISM and SRD frequency bands at 169, 315, 433, 868, and 915 MHz. The air data rate is programmable from 1 to 500 kbps, and the SPIRIT1 can be used in systems with channel spacing of 12.5/25 kHz, complying with the EN 300 220 standard. It uses a very small number of discrete external components and integrates a configurable baseband modem, which supports data management, modulation, and demodulation. The data management handles data in the proprietary fully programmable packet configuration, but it also allows the WM-Bus standard compliant format.

Furthermore, the SPIRIT1 can perform Cyclic Redundancy Checks (CRCs) on the data as well as FEC encoding/decoding on the packets. The SPIRIT1 provides an optional automatic acknowledgement, re-transmission, and timeout protocol engine in order to reduce overall system costs by handling all the high-speed link layer operations.

Moreover, the SPIRIT1 supports an embedded CSMA/CA engine. An AES 128-bit encryption co-processor is available for secure data transfer. The SPIRIT1 fully supports antenna diversity with an integrated antenna switching control algorithm. The SPIRIT1 supports different modulation schemes: 2-FSK, GFSK, OOK, ASK, and MSK. Transmitted/received data bytes are buffered in two different three-level FIFOs (TX FIFO and RX FIFO), accessible via the SPI interface for host processing.

There weren't any major difficulties during the evaluation phase of the ST-Microelectronics technology, unlike for the Telit chip. Thus the evaluation boards listed into this chapter were used for prototyping.

One of the reasons of their success was the existence of proprietary software tools.

### 4.2.1 SPIRIT1 GUI

[24] The GUI allows to connect to the ST-Microelectronics SPIRIT1 evaluation boards in order to upload new firmware or testing purposes in a simple way.

If a specific memory image is written inside the evaluation board, it is possible to access the internal parameters, read the content of the memory and change both of them run time using the GUI.

First of all it is possible to access the radio parameters and change frequency base, data rate, frequency deviation, channel filter, modulation and output power.

Moreover the packet format can be set: preamble and postamble length and payload width can be switched manually. Also the WM-Bus packet format can be set.

However, the most important feature of the GUI is the possibility to test the communication link between two evaluation boards of the same kind. This element has been used to evaluate the range of the various boards. In fact it is possible to set the number of packets, their content and their time period and then start the transmission on one end and the reception on the other. At the end of the test the GUI shows the chronology of all the received packet with their content and received RSSI. Moreover the medium RSSI and percentage of packet loss is computed automatically. Figure 4.2.2 shows an overview of the test interface.

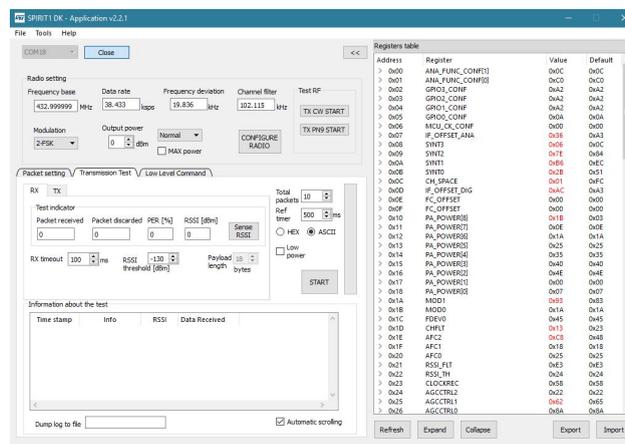


Figure 4.2.2: ST-Microelectronics SPIRIT1 GUI

### 4.2.2 IAR project ST-Link programming

ST-Microelectronics provides the WM-Bus stack and a set of example projects [25] based on the IAR IDE. These projects implement the basic unidirectional and bi-directional schemes of communication defined by the OMS WM-Bus standard [5] (figures 2.2.1, 2.2.2 and 2.2.3).

The basic communication scheme used by all the water meter is unidirectional, so that example project was the starting point for further implementations.

The software is organized in order to work on all the WM-Bus evaluation boards, so there is a simple way to modify the radio parameters.

It is, in fact, possible to change the frequency just by setting a different WM-Bus mode (see table 2.0.1). If it is set to T1/T2 the transceiver works in the 868MHz band, F1/F2 on the 433MHz, while N1/N2 on the 169MHz. This is the fastest way to configure the radio, but it is also possible to access to the low level variables.

The reception, transmission and management of the WM-Bus packets is deputized to a FSM working in loop (see figure 4.2.3). When a frame is received, according to its C-field the FSM behaves in a different way in order to interpret data correctly and if needed reply to the sender.

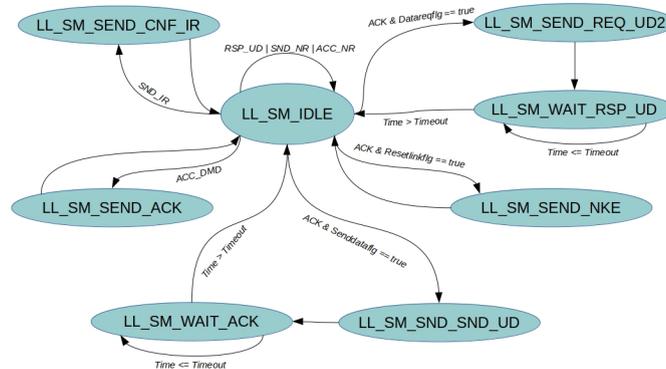


Figure 4.2.3: ST-Microelectronics WM-Bus stack FSM

The received WM-Bus frame is unpacked in its various field automatically. In this way it is simple to access the address and payload information. Moreover, if the payload is encrypted with AES-CBC or AES 128 and the key is available, it is possible to decrypt it.

In addition drivers for the various peripheral of the microcontroller are provided. Thanks to them it has been possible to set the UART serial line and connect the evaluation board to the LoRa board in the first prototype.

**STM32L151CB** The whole project is created to work on the STM32L151 microcontroller [26] which is the one mounted on the evaluation boards.

This device extends the ultra-low-power concept with no compromise on performance, using the Cortex-M3 core and a flexible CPU clock from 32 kHz up to 32 MHz. In addition to the dynamic run and low-power run modes, two additional ultra-low-power modes bring very low power consumption while keeping an RTC, backup register content and low-voltage detector.

It can work with a power supply voltage in the range 1.65V to 3.6V and can reach a current consumption of  $0.3\mu\text{A}$  in standby mode (3 wakeup pins).

It has 8 peripheral communication interfaces: a USB 2.0, three USARTs (ISO 7816, IrDA), two SPIs (data rate 16 Mbit/s) and two I2Cs.

Moreover it is equipped with up to 128 Kbytes Flash memory with ECC, and an LCD driver.

From the analog point of view it provides a 12-bit ADC with up to 24 channels, a 12-bit DAC with 2 channels and output buffers, and two ultra-low-power-comparators with window mode and wake up capability.

Its development support comprehends the serial wire debug (SWO), JTAG and trace.

### 4.3 ST-Microelectronics evaluation boards range test

The free space range interconnection of the ST-Microelectronics evaluation boards has been measured in the Politecnico di Torino campus in front of the GM building (see figure 4.3.1), during the last weeks of October 2018.



Figure 4.3.1: Bird's eye view of the measurement field

A couple of devices of the same type were employed to test the Received Sense Strength Indication (RSSI) using one of them as receiver and the other as transmitter. The measurements has been performed with the support of the SPIRIT1-GUI. For each point in distance the mean of the RSSI values of 50 received packets has been recorded. The RSSI of each packet is collected from the appointed byte in the transmission frame. As stated into the device's datasheet [23], the received frame should be translated into the correct value using the equation 4.3.1, as the accuracy is half of dBm for each bit.

$$RSSI = \frac{BYTE}{2} - 130 \quad (4.3.1)$$

This computation and the mean of the 50 values are automatically executed by the GUI software. It also estimates the percentage of errors on the overall transaction: each lost or corrupted packet received contributes with a 2%.

The measurements were performed applying some fixed parameters, depending on the frequency base, as shown in table 4.3.1.

Table 4.3.1: Fixed parameters of evaluation board range measurement

Frequency base [MHz]	868	433	169
Frequency deviation [kHz]	20.629	19.836	19.836
Channel filter [kHz]	100.5	102.115	102.115
Data rate [ksps]	38.383	38.433	38.433
Modulation	2FSK	2FSK	2FSK
Payload lenght [bytes]	18	18	18
WM-Bus mode	T1, T2	F1, F2	N1, N2
Preamble '01' sequence	2	2	2
Postamble '01' sequence	1	1	1

Moreover the Noise Floor (NF) was sensed for each frequency in order to fix the threshold. That measurement have been carried out thanks to the internal SPIRIT1 register "RSSI.LEVEL". The accuracy achieved is the same as before, as the RSSI is acquired using again equation 4.3.1. The threshold was fixed 5 dBm over the NF. For each band the following NF was computed using an average of 10 different single measurements:

- 868MHz: -110dBm
- 433MHz: -90dBm
- 169MHz: -86dBm

Given this framework the measure of the RSSI was carried out for each frequency, for different output power (0dBm, 3dBm, 7dBm, 12dBm) and with increasing distance between the receiver and the transmitter.

To check the correctness of the so taken measurements, the computed RSSI x distance points were then interpolated using a logarithmic trendline to have a comparison with the ideal behavior given by the Frii's equation (4.3.2)(4.3.3), in which the antenna gain is set to -4dBi.

$$RSSI = P_{TX} \frac{G_{TX} G_{RX}}{\left(\frac{4\pi\lambda}{d}\right)^2} \quad (4.3.2)$$

$$RSSI|_{dBm} = P_{TX}|_{dBm} + G_{TX}|_{dBi} + G_{RX}|_{dBi} - 20 \log(d|_{km}) - 20 \log(f|_{GHz}) - 92.45 \quad (4.3.3)$$

### 4.3.1 STEVAL-IDS001V4M (868MHz)

In table 4.3.2 the measurements taken in the 868MHz band with the method described in the in the previous section are listed. For each output power taken into account (0dBm, 3dBm, 7dBm and 12dBm) the RSSI measured and its correspondent percentage of lost packets are shown. Moreover the ideal value given by the Frii's equation is displayed.

Table 4.3.2: STEVAL-IDS001V4M (868MHz) range measurement

Meters	RSSI[0]	%[0]	friis	RSSI[3]	%[3]	friis	RSSI[7]	%[7]	friis	RSSI[12]	%[12]	friis
4	-66,1	0	-51,3			-48,3			-44,3			-39,3
8	-77,6	0	-57,3	-81,5	0	-54,3	-73,7	0	-50,3	-66,7	0	-45,3
12	-74,2	0	-60,8			-57,8			-53,8			-48,8
16	-84,3	0	-63,3	-73,9	0	-60,3	-68	0	-56,3	-62,1	0	-51,3
20	-74	0	-65,2			-62,2			-58,2			-53,2
24	-79,8	0	-66,8	-82,2	0	-63,8	-78,5	0	-59,8	-73,6	0	-54,8
28	-83,1	0	-68,2			-65,2			-61,2			-56,2
32	-82,7	0	-69,3	-84	0	-66,3	-77,7	0	-62,3	-73,1	0	-57,3
36	-87,4	0	-70,3			-67,3			-63,3			-58,3
40	-88,1	0	-71,3	-87,3	0	-68,3	-80,9	0	-64,3	-76,4	0	-59,3
44	-91,9	2	-72,1			-69,1			-65,1			-60,1
48	-93,8	14	-72,8	-88,4	0	-69,8	-80	0	-65,8	-79,8	0	-60,8
52	-94,1	6	-73,5			-70,5			-66,5			-61,5
56	-90,4	0	-74,2	-90,3	2	-71,2	-82,5	0	-67,2	-81,3	0	-62,2
60	-91	0	-74,8			-71,8			-67,8			-62,8
64	-91,5	2	-75,3	-89,6	0	-72,3	-90,7	0	-68,3	-84,5	0	-63,3
68	-92,9	2	-75,9			-72,9			-68,9			-63,9
72	-99,5	20	-76,4	-91,9	0	-73,4	-90,4	0	-69,4	-83,6	0	-64,4
76	-97,2	6	-76,8			-73,8			-69,8			-64,8
80	-99,2	66	-77,3	-95	4	-74,3	-88,2	2	-70,3	-83,5	0	-65,3
84						-74,7			-70,7			-65,7
88				-93,2	2	-75,1	-89,5	0	-71,1	-85,1	2	-66,1
92						-75,5			-71,5			-66,5
96				-95	12	-75,9	-89,8	0	-71,9	-94,2	6	-66,9
100						-76,2			-72,2			-67,2
104				-91,6	2	-76,6	-88,1	0	-72,6	-86,5	0	-67,6
112					100	-77,2	-90,9	0	-73,2	-85,7	0	-68,2
120							-94,1	2	-73,8	-79,9	0	-68,8
128							-98,6	16	-74,4	-82,9	0	-69,4

The table has been translated in graphical form to have better understanding of the measurements trend (see figure 4.3.2). As can be seen the behavior is correct, but there is an offset between the taken values and the ideal ones. That could be ascribed to the non ideal environment in which the measurements have been performed. The noise floor in fact was high due to the presence of radio disturbances in the Politecnico ambient. Moreover, as can be seen in the photo (figure 4.3.1), the maximum distance in free space that can be physically reach is limited by the two Politecnico's bridges.

For this reason a further measure has been done on the rooftop, reaching a distance of 160m, with -97,8dBm of RSSI and 34% of packet lost. The ideal Frii's value for that range is -71.3dBm.

The measurements should have been done in another environment to achieve better performance.

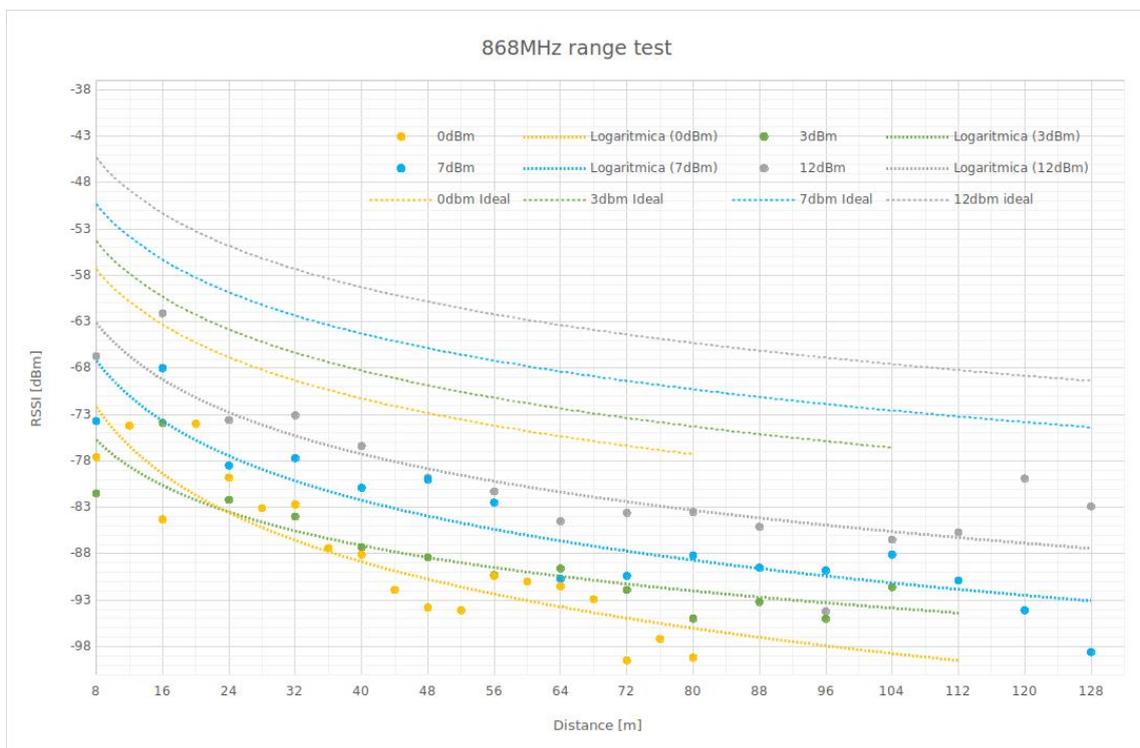


Figure 4.3.2: STEVAL-IDS001V4M (868MHz) range measurement

### 4.3.2 STEVAL-IDS001Vx (433MHz)

In table 4.3.3 the measurements performed in the 433MHz band with the method described previously are listed.

Table 4.3.3: STEVAL-IDS001Vx (433MHz) range measurement

Meters	RSSI[0]	%[0]	friis	RSSI[3]	%[3]	friis	RSSI[7]	%[7]	friis	RSSI[12]	%[12]	friis
8	-59	0	-51,2	-51	0	-48,2	-52,1	0	-44,2	-40,9	0	-39,2
16	-64,7	0	-57,3	-62,6	0	-54,3	-57,8	0	-50,3	-51,5	0	-45,3
24	-69,6	0	-60,8	-64,4	0	-57,8	-72,6	4	-53,8	-61,9	0	-48,8
32	-75	2	-63,3	-74,9	10	-60,3	-66,4	0	-56,3	-63,6	0	-51,3
40	-78,1	10	-65,2	-71,3	0	-62,2	-77,3	18	-58,2	-66,7	0	-53,2
48	-75	8	-66,8	-75,5	4	-63,8	-76	6	-59,8	-65,5	0	-54,8
56	-76,1	0	-68,1	-74,7	0	-65,1	-73,3	0	-61,1	-68,8	0	-56,1
64	-78,6	14	-69,3	-75,7	0	-66,3	-78,7	24	-62,3	-71,7	0	-57,3
72	-82,5	98	-70,3	-78,4	68	-67,3	-80,5	98	-63,3	-70,8	0	-58,3
80			-71,2	-79,5	56	-68,2			-64,2	-67,8	0	-59,2
88			-72,1	-77,9	64	-69,1			-65,1	-70,3	0	-60,1
96			-72,8	-79,7	56	-69,8			-65,8	-75,2	6	-60,8
104			-73,5			-70,5			-66,5	-72,5	0	-61,5
112			-74,2			-71,2			-67,2	-67,5	0	-62,2
120			-74,8			-71,8			-67,8	-76,4	4	-62,8
128			-75,3			-72,3			-68,3	-79,7	58	-63,3
136			-75,9			-72,9			-68,9	-73,2	0	-63,9

The table has been translated in graphical form to have better understanding of the measurements trend (see figure 4.3.3).

As can be seen the behavior is not perfect. There is an offset between the taken values and the ideal ones as before, but the 7dBm output power range series trend goes much lower than expected. That could be ascribed to the non ideal environment in which the measurements have been taken. The noise floor in fact was very high (-90dBm) due to the presence of radio disturbances in the Politecnico ambient. The distance reached is still limited by the Politecnico bridges, but with the lower power the range is shorter than the one of the 868MHz band, and this is not an ideal behavior.

Even in this case a final measurement has been performed on the rooftop, but the distance reached was lower than the range acquired previously.

The whole measure should have been done in another environment to achieve better performance.

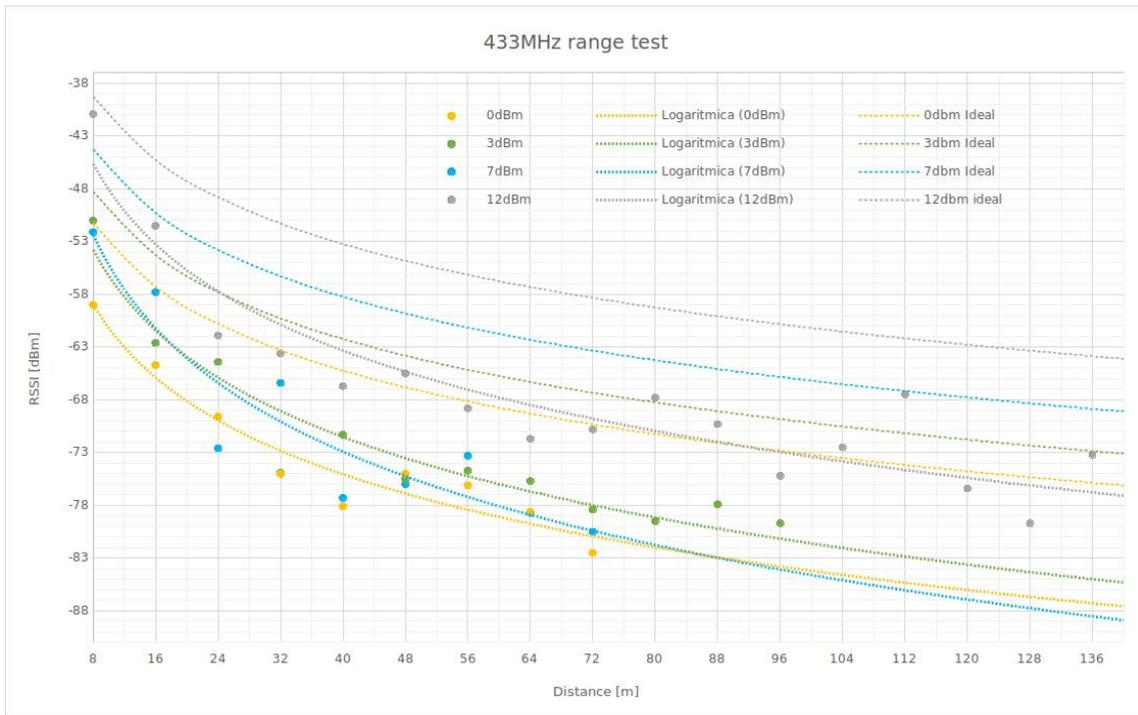


Figure 4.3.3: STEVAL-IDS001Vx (433MHz) range measurement

### 4.3.3 STEVAL-IKR002V1 (169MHz)

In table 4.3.4 the measurements performed in the 169MHz band with the method described previously are listed.

Table 4.3.4: STEVAL-IDS001Vx (169MHz) range measurement

Meters	RSSI[0]	%[0]	friis	RSSI[3]	%[3]	friis	RSSI[7]	%[7]	friis	RSSI[12]	%[12]	friis
8	-48,6	0	-43,1	-51,4	0	-40,1	-47,4	0	-36,1	-45,3	0	-31,1
16	-60,1	0	-49,1	-65,9	0	-46,1	-58,9	0	-42,1	-61,3	0	-37,1
24	-63,6	0	-52,6	-60	0	-49,6	-66,8	0	-45,6	-63,7	0	-40,6
32	-76,4	0	-55,1	-62	0	-52,1	-64,7	0	-48,1	-65,4	0	-43,1
40	-68,1	0	-57,0	-69,3	0	-54,0	-67	0	-50,0	-65,1	0	-45,0
48	-77,9	6	-58,6	-76,2	0	-55,6	-69,2	0	-51,6	-68,6	0	-46,6
56	-76,8	4	-60,0	-68	0	-57,0	-63,2	0	-53,0	-65,3	0	-48,0
64	-70,2	0	-61,1	-73,5	0	-58,1	-72,5	0	-54,1	-73,1	0	-49,1
72	-78,1	0	-62,2	-80	4	-59,2	-72,8	0	-55,2	-76,3	0	-50,2
80	-78,7	2	-63,1	-85,4	90	-60,1	-69,3	0	-56,1	-72,2	0	-51,1
88	-86,8	52	-63,9	-76,6	0	-60,9	-76,6	6	-56,9	-77,8	0	-51,9
96	-86,4	32	-64,7	-79,3	66	-61,7	-73,2	0	-57,7	-75	0	-52,7
104	-86,3	46	-65,3	-82	8	-62,3	-76,1	4	-58,3	-73,9	0	-53,3
112	-77,1	2	-66,0	-80	7	-63,0	-82,6	28	-59,0	-74,9	0	-54,0
120	-87,2	44	-66,6	-77,6	2	-63,6	-76,7	4	-59,6	-77	6	-54,6
128	-88,8	68	-67,2	-86,9	76	-64,2	-74,4	0	-60,2	-73,6	0	-55,2

The table has been translated in graphical form to have better understanding of the measurements trend (see figure 4.3.4).

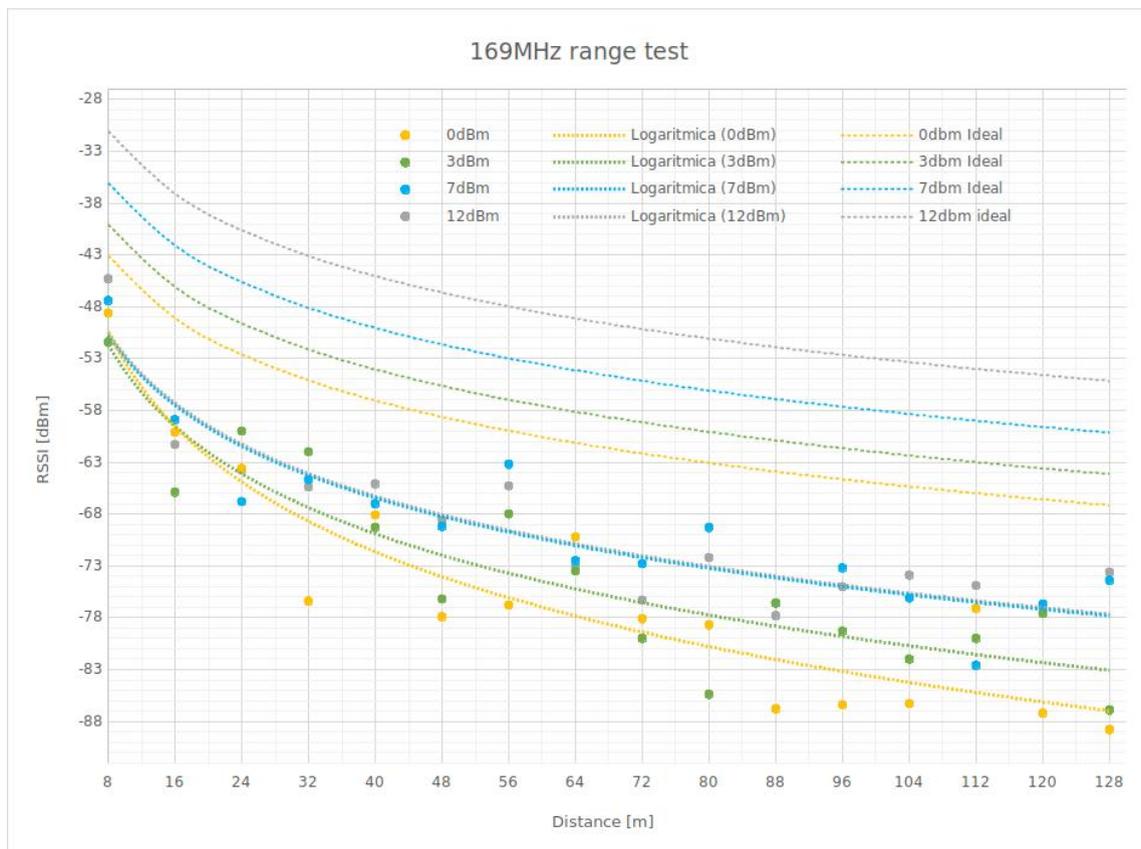


Figure 4.3.4: STEVAL-IKR002V1 (169MHz) range measurement

As can be seen the behavior is not perfect. There is an offset between the taken values and the ideal ones as before, but the 3 and 7dBm output power range series trend are superimposed. That could be ascribed to the non ideal environment in which the measurements have been taken. The noise floor in fact was very high (-87dBm) due to the presence of radio disturbances in the Politecnico ambient. The distance reached is still limited by the Politecnico bridges, but with the lower power the range is shorter than the one of the 868MHz band and this is not an ideal behavior.

In this case a final measurement on the rooftop has not been performed. The whole measure should have been done in another environment to have a more meaningful result and better performance.



## Part II

# LoRa

## 5 LoRa and LoRaWAN basics



Figure 5.0.1: LoRa logo

LoRa (**L**ong **R**ange) is a low power, long range wireless communications technology working in the 169MHz, 433MHz and 868MHz or 915MHz bands, developed for IoT and M2M applications.

In fact, it is ideal for providing intermittent low data rate connectivity over significant distances. The radio interface has been designed to enable extremely low signal levels reception, and as a result even low power transmissions can be received at significant ranges. The receiver is usually cheap and it is able to reach a very good sensitivity and a low bit error rate.

For those reasons, LoRa is a very strong competitor on the LPWAN (Low Power Wide Area Network) market.

As a result, the LoRa Alliance was set up develop and promote the LoRa wireless system across the industry with the goal of providing an open global standard for secure, carrier-grade IoT LPWAN connectivity.

In this way the LoRaWAN standard was born.

### 5.1 LoRa and LoRaWAN basics

[27] The LoRa protocol must follow some directives to be classified as standard. The first limitation is given by the band. The transceiver must in the bands centered at frequencies 169MHz, 433MHz and 868MHz (EU) or 915MHz (USA).

The LoRa Alliance has also defined an open protocol stack. The creation of the open source stack has enabled the concept of LoRa to grow because all the different types of companies involved in LoRa development, use and deployment have been able to come together to create an easy to use, low cost solution for connectivity of IoT devices.

The apparatuses inside the LoRa network are linked throughout a simple star-network, where the end-devices are connected to a central gateway which collects the transactions and communicate them to the network provider.

The directives for the physical and networking layer will be listed in a following sections.

### 5.1.1 LoRa physical layer

The Lora physical layer is based on a modulation scheme that implements a chirp spread-spectrum (CSS) technique [28].

A chirp (Compressed High Intensity Radar Pulse) is a signal which frequency either increase or decrease with time. They have constant amplitude and pass the whole bandwidth in a linear or non-linear way from one end to another end in a certain time period. If the frequency changes from lowest to highest, it is called up-chirp, instead if the frequency changes from highest to lowest, it's a down-chirp.

Thanks to this mechanism LoRa protocol is robust against the Doppler Effect.

The spreading factor (SF) is the ratio between the band of the chirp (frequency span between the starting and stop frequency) and the time period in which it evolves.

This parameter has been fixed by the LoRa Alliance to a set of standard values.

LoRa uses three different bandwidth for the chirps: 125kHz, 250kHz and 500kHz.

The spreading factor combined with the bandwidth gives a set of standardized datarates (DR), as listed in table 5.1.1.

Table 5.1.1: LoRa standard datarates

DR	Bandwidth	Modulation	SF	Bitrate
7	500kHz	FSK	-	50kbit/s
6	250kHz	CSS	7	11kbit/s
5	125kHz	CSS	7	5.5kbit/s
4	125kHz	CSS	8	3.1kbit/s
3	125kHz	CSS	9	1.8kbit/s
2	125kHz	CSS	10	1kbit/s
1	125kHz	CSS	11	500bit/s
0	125kHz	CSS	12	290bit/s

As can be seen the spreading factor and the datarate are inversely proportional. Thus the SF must be incremented to reduce over-the-air time.

For instance, if DR is equal to 0 it is possible to transmit only 51 bytes of payload over an hour, meanwhile with DR equal to 6 the same value could increase to more than 100 bytes. This duty cycle limitations for the transmission are ensured by the standard in order to avoid the implementation of LBT (Listen Before Talk) mechanisms.

The standard frame of the LoRaWAN is very simple: it includes 8 preamble symbols, 2 synchronisation symbols, physical payload and optional CRC. Figure 5.1.1 shows the behavior in time x frequency of the LoRa header followed by 5 random data bits.

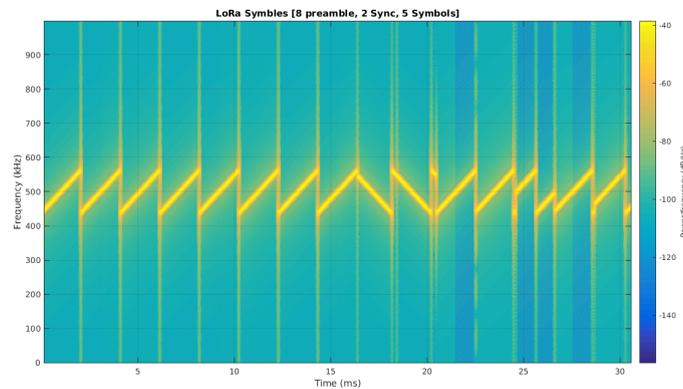


Figure 5.1.1: LoRa preamble and synchronization fields

As can be seen in the image, the up-chirp can be modulated by changing the starting frequency, In this way it is possible to increase the number of symbols in a simple way.

### 5.1.2 LoRaWAN network architecture

LoRaWAN is the standard network architecture of LoRa. It provides the routing of data from the end node via a LoRaWAN gateway to the required entities. LoRaWAN also defines the way in which data is sent around the network, detailing the responses of the LoRaWAN gateways, and the LoRa network server.

Figure 5.1.2 underlines the various actors involved in the LoRaWAN network:

- **End Node:** It is classified in this way any sensor or actuator, which is connected to the LoRa network. It is always the initiator of the communication and does not need an acknowledge signal from the gateway in order to proceed.

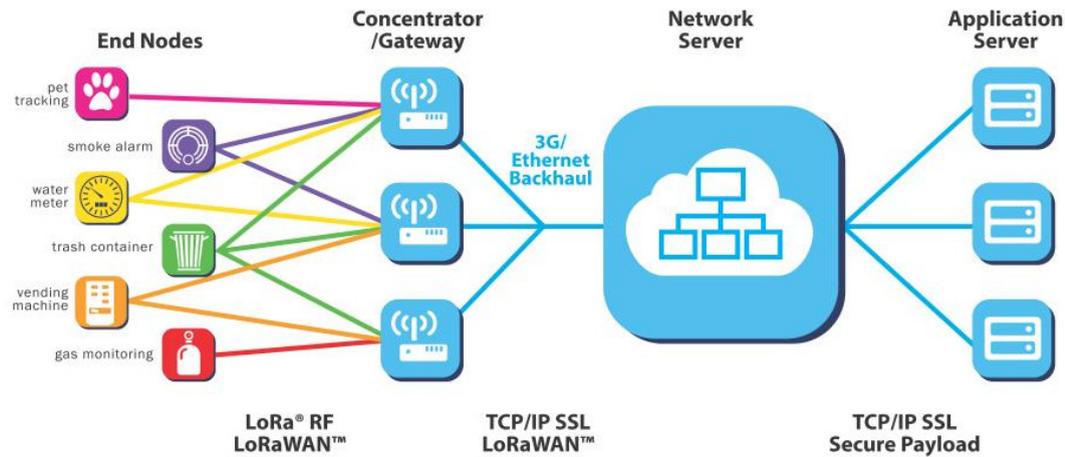


Figure 5.1.2: Standard LoRaWAN network architecture

The end node could be in the range of more than one gateway, in that case its messages are received and forwarded by all the reached gateways. This capability enhances the resilience of the network, improving the number of messages that are successfully received.

- **Gateway:** This type of device receives the communications from the end nodes and transfers them to the network server via ethernet, cellular or any other telecommunication link wired or wireless. The gateways are connected to the network server using standard IP connections. In this way data using the standard protocol can be connected to any telecommunications network, whether public or private.  
Each gateway can answer to a single end node at a time after it receives a frame from it and a successive acknowledgement, and solely if the sensor is programmed to receive. Moreover it must answer in a predefined time period starting from the end-node acknowledgement. A single gateway could be potentially connected to 250'000 end nodes.
- **Network server:** The LoRa network server manages the network and, as part of its function, it acts to eliminate duplicate packets, schedules acknowledgements, and adapts data rates.
- **Application server:** It is a proprietary online application for every end node deployer. It reads and process the data collected by the network server from the end nodes. If necessary it schedules the acknowledgements.

### 5.1.3 Classification of end nodes

The LoRa end nodes can be classified according to their power consumption in 3 main categories.

**Class A** The default class which must be supported by all LoRaWAN end-devices. Class A communication is always initiated by the end-device and is fully asynchronous. Each uplink transmission can be sent at any time and is followed by two short downlink windows, giving the opportunity for bi-directional communication or network control commands if needed. This is an ALOHA-type protocol. The end-device is able to enter low-power sleep mode for as long as defined by its own application: there is no network requirement for periodic wake-ups. This makes class A the lowest power operating mode, while still allowing uplink communication at any time. Because downlink communication must always follow an uplink transmission with a schedule defined by the end-device application, downlink communication must be buffered at the network server until the next uplink event.

In this class are categorized the battery power sensors without energy harvester.

**Class B** In addition to the class A initiated receive windows, class B devices are synchronized to the network server using periodic beacons, and open downlink ‘*ping slots*’ at scheduled times. This provides the network the ability to send downlink communications with a deterministic latency, but at the expense of some additional power consumption in the end-device. The latency is programmable up to 128 seconds to suit different applications, and the additional power consumption is low enough to still be valid for battery powered applications.

In this class are usually categorized the battery power sensors with energy harvester.

**Class C** In addition to the class A structure of uplink followed by the two downlink windows of class B, class C further reduces latency on the downlink by keeping the receiver of the end-device open at all times that the device is not transmitting (half duplex). Thus the network server can initiate a downlink transmission at any time on the assumption that the end-device receiver is open. In this way there is no latency. The compromise is the power consumption of the receiver (up to 50mW) and so class C is suitable only for applications where continuous power is available. For battery powered devices, temporary mode switching between classes A and C is possible, and it is useful for intermittent tasks such as firmware over-the-air updates. In this class are categorized the main powered sensors and actuators.

## 5.2 St-Microelectronics B-L072Z-LRWAN1 evaluation board

The LoRa evaluation board used for the project is the one adopted by the iXem Laboratory. In this way it has been possible to use the modified and tested firmware for the LoRa stack as a starting point.

During the development of this project that same firmware has been updated with the new version of the LoRa stack realized by the ST-Microelectronics.

The development board in use is the B-L072Z-LRWAN1 LoRa Discovery kit [29]. It mounts the CMWX1ZZABZ-091 LoRa module produced by Murata [30]. A STM23L0 microcontroller is embedded into the device.

The board also includes a LoRa RF interface, LEDs, push-buttons, antenna, Arduino UNO V3 connectors, USB 2.0 connector and a ST-Link/V2-1 debugger and programmer for the microcontroller.

The kit comes with a firmware package compliant with the LoRaWAN stack, and it is certified for class A and C LoRa devices.

**STM32L072** [31] The STM32L072 features autonomous peripherals, real-time clock, low-power time clock, hardware encryption and ultra-low-power 12-bit ADC with 48  $\mu\text{A}$  budget consumption at 100 Ksps.

It offers up to 192-Kbytes of Flash memory, up to 20 Kbytes of RAM and up to 6 Kbytes of embedded EEPROM, and comes in 32- to 100-pin packages in UQFN, LQFP, BGA, and WLCSP.

Crypto libraries covering AES 256-bit, DES, 3DES, RSA and ECC are available.

### 5.2.1 Firmware by iXem Labs

The firmware has been developed inside an Eclipse workspace. It is based on the IAR project developed by the ST-Microelectronics, adopting its LoRa network core. It is designed to be able to interconnect easily the LoRa core to different existing sensors.

For this reason a series of preprocessor defines has been used. To change the configuration, and thus the type of sensor connected to the LoRa core, a simple exchange of active constant must be performed.

In this way it has been possible to declare a standard initialization function for the sensors peripherals and stop mode configuration.

This operation has been simplified thanks to the existence of a set of standard libraries developed by the ST-Microelectronics and able to work on all the versions of the STM32 microcontroller. They are thus the same used for the WM-Bus evaluation boards.

At the end the main function is not impacted with the complexity of the system, but each connectable sensor has its own tasks defined in a proper file untethered from the others. Moreover the interchange of the devices is managed by another intermediary level of functions.

The firmware of WM-Bus-LoRa gateway in design will be inserted in the main project as a new type of sensor, with new initialization and operational functions created in designated "c" and "h" files.

The LoRa frame sent to the network gateway will be filled with data coming from the WM-Bus network using a translation function inserted into the "LoRa send" software routine.

### 5.2.2 Tests by iXem Labs

The tests described in this section were performed by the iXemLabs researchers [32]. The development board has been used to test the range of the LoRa module. The range has been tested using atmospheric balloons in order to have a free range interconnection.

In a rural zone in Monferrato, a transmitting device have been placed at a height of 100m, while the receiving device was on the ground. The results show that:

- In a non-line-of-sight the maximum distance for transmission is 2 Km.
- When in a line-of-sight data follow the Friis Law until the maximum distance of about 40 km.

The map in figure 5.2.1 show the measurements of the RSSI collected on the ground.

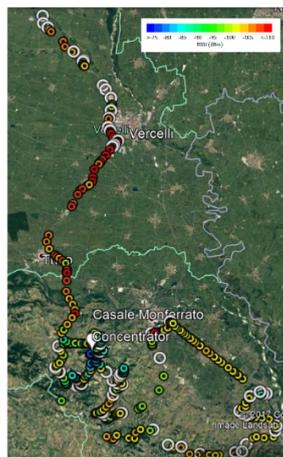


Figure 5.2.1: Map of the RSSI measurements performed by the iXemLabs

The development board has been modified in a more simplified version of hardware by the iXem Lab researcher, in order to have a simple and defined connection for the sensors. That version moreover allowed them to reach a very low current consumption in stop mode when not receiving: about 3nA.



## Part III

# Prototyping

## 6 First prototype

The first prototype of the WM-Bus - LoRa gateway (figure 6.0.1) has been implemented with a straightforward approach using the ST-Microelectronics evaluation boards described in the previous chapters and firmware based on the one provided by the manufacturer.

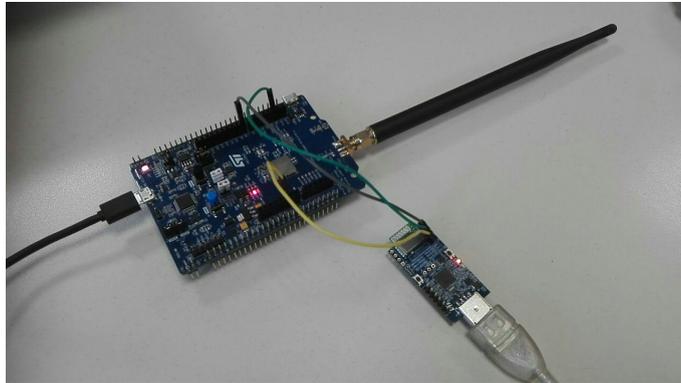


Figure 6.0.1: Photo of the first gateway MW-Bus (868MHz) - LORA prototype

This approach has been chosen in order to reach the goal faster and with ease. The drawback of this system is that it is able to process only one WM-Bus frequency band at a time, due to the lack of existing board-to-board interconnections in the evaluation boards.

That disadvantage has been overstepped because the only two water meters with a transmission delay of a few seconds (therefore easily readable during the development stage) are both in the 868MHz band: the Maddalena's and the Itron's ones. The other two meters have a transmission period of an hour or more, so they are not appropriate for testing.

In the next sections the hardware and firmware structure of the prototype will be described in detail.

## 6.1 HW Structure

The two evaluation boards chosen are the STEVAL-IDS001V4M for the 868MHz WM-Bus network and the B-L072Z-LRWAN1 for the LoRa. They are connected through the only UART interface available: the connection of the USART1 of the LoRa module to the USART2 of the WM-Bus USB-dongle. The UART is set to work with 1 stop bit, no parity and 8 bits data length. The bauderate is set to 9600bits/s, with no hardware control protocols.

The power is supplied to both boards by USB interconnection to a PC.

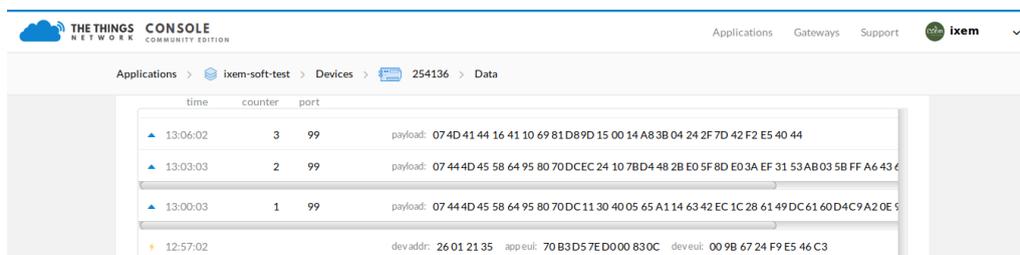
The gateway WM-Bus - LoRa works as a LoRa end-node and it is connected to the online LoRaWAN server provider "The Things Network" [33] through a LoRa gateway placed inside the iXem laboratory. This connection is established by setting, inside the LoRa firmware, the device address, App EUI and Device EUI keys to the values pointed by the server provider. The transmission delay has been set to 3 minutes in order to avoid duty cycle related problems.

With this prototype structure it is possible to see both the LoRa and the WM-Bus packets on the PC screen. The STEVAL-IDS001V4M USB interconnection, in fact, allows to access through a serial terminal to the data received by the WM-Bus transceiver using a simple *printf* C function. The output is shown in figure 6.1.1.

TYPE	MAN	SERIAL NUMBER	VER	RSSI	PAYLOAD
WMAT	IME	98 64 95 80	70	-35,000000	15 30 40 05 16 6c 62 c3 09 01 e7 64 b5 97 28 7c a9 fc 04 26 0b d3 34 86 7c 27 c6 aa 43 f0 b5 88 7a ea 88 d3 9a 22 e4 96 c 0f 06 34 0e 8c b3 a0 c8 54 4d fa
WMAT	MAD	16 41 10 69	81	-37,000000	cd 15 00 14 a8 70 96 4b 28 db de 4c 52 3d 4c
WMAT	IME	98 64 95 80	70	-35,000000	15 30 40 05 18 2a 69 bc b1 ff 87 93 07 f6 ac 99 b8 bf 22 96 a8 22 44 cb 8d 23 5c b5 1a 4a 19 bd 77 82 e1 3b 23 8f 22 9e 1 79 83 ed da 7e 7a ee ca 48 84 d7
WMAT	MAD	16 41 10 69	81	-37,000000	dd 15 00 14 a8 67 b9 02 83 fa 0d 95 89 24 4e
WMAT	IME	98 64 95 80	70	-35,000000	15 30 40 05 54 6d 00 d1 93 da 5c c7 53 06 6e 7b 10 91 ca 64 6a 99 41 40 64 3b 36 d3 7e 10 59 82 29 23 2a c7 a5 ed 3b b4 3 23 d6 76 48 d4 0b 46 50 6c 94 43
WMAT	MAD	16 41 10 69	81	-37,000000	ed 15 00 14 a8 5e c8 d8 7e 98 79 ff e4 0f 49
WMAT	IME	98 64 95 80	70	-35,000000	15 30 40 05 ab eb 35 e8 df 37 5b 90 49 02 e3 4c c0 dc 5a b6 c7 ee 8f 3c 42 b1 d5 0e 7c 37 62 e2 40 83 40 17 34 71 34 5a 5 81 98 13 8c 9a 93 76 0f 95 13 9a
WMAT	MAD	16 41 10 69	81	-37,000000	fd 15 00 14 a8 49 e7 91 d5 b9 aa 26 3f 16 4b
WMAT	IME	98 64 95 80	70	-35,000000	15 30 40 05 2d a7 b2 b9 90 32 2c 6a b1 e8 c1 4d a0 06 e4 6d b9 de e9 01 43 f4 2f 30 68 53 2d 53 84 8b 57 90 22 9e f1 f1 2 82 9c 8f 43 6e da fb 4c 9f 20 ce
WMAT	MAD	16 41 10 69	81	-37,000000	8d 15 00 14 a8 2c 2b 6d 84 5c 91 2b 3e 59 46

Figure 6.1.1: Example of WM-Bus packet output

On the other end, on *The Things Network* online console, it is possible to see the LoRa packets sent by our end-node. In figure 6.1.2 the bottom line shows the application packet with which the end-node apply to the network. The other lines shows three LoRa data packets containing the WM-Bus frames. The first one refers to the Maddalena's meter, the other two to the Itron's one which sends with each transmission more data bytes than the maximum admitted length of the LoRa frame. For this reason the packet is split in half.



time	counter	port	payload
13:06:02	3	99	074D41441641106981D89D150014A83B04242F7D42F2E54044
13:03:03	2	99	07444D455864958070DCEC24107BD44828E05F8DE03AEF3153AB0358FFA6436
13:00:03	1	99	07444D455864958070DC1130400565A1146342EC1C286149DC6160D4C9A20E5
12:57:02			devaddr: 26012135 appeui: 70B3D57ED000830C deveui: 009B6724F9E546C3

Figure 6.1.2: The Things Network console overview

More information can be found in the next section.

## 6.2 FW structure

The prototype involve two microcontrollers mounted on two different boards working together connected trough a UART serial line. Thus there are two different firmware: one manages the WM-Bus frames reception and sends the data on the UART, the second one receives from the UART, manages the information, stores it and sends it via LoRa to the network provider.

### 6.2.1 WM-Bus

This version of the WM-Bus firmware is based on the project and examples provided by ST-Microelectronics. Using their WM-Bus FSM (figure 4.2.3) and various driver libraries, it is easier to create a working system.

The board is set to operate in T1 mode, receiving from the meters the WM-Bus packets with C-field SND-NR unidirectionally (see figure 2.2.1 for the schematic of the transmission).

When a new frame is received, the ST-Microelectronics firmware divides it into its internal fields (see figure 2.1.2) and adds information like the time stamp and the RSSI read from the appointed byte, the same used during the board's range test. Those information are managed in order to retrieve the meter address and payload information.

The issue that arises at this point is that the A-field of the meters is not uniquely structured, as explained in the previous chapters. Thus a set of rules to translate A-field into Type - Version - Serial Number needs to be created for each known meter manufacturer.

For this scope a table is created and stored inside the STEVAL-IDS001V4M's flash memory. Each entry of this table corresponds to a set of parameters apt to retrieve the essential information of one known meter:

- Manufacturer FLAG-ID, the three letters code: this information is always contained into the M-field of the received frame. The translation from M-field to FLAG-ID is computed with the implementation of equation 2.2.1. After the conversion this information is used to access the right entry of the table.
- Position and value of the type: the position allows to find the right byte inside the A-field where the type information is stored. This is used to check that the WM-Bus frame collected refers to the type "water" of meter produced by the manufacturer pointed by the FLAG-ID.
- Position and value of the version: the position allows to find the right byte inside the A-field where the version information is stored. This is used to check that the WM-Bus frame collected refers to the right version of water meter produced by the manufacturer pointed by the FLAG-ID. In fact the version of meter could potentially change the A-field information order. Thus leading to a wrong interpretation of the frame.

The total length of the table is stored in the flash as well.

At the reset, thanks to the information about its length, the whole rules table is read from the flash to the RAM. In this way it is possible to interpret run-time the received WM-Bus frames. If the obtained frame is not conform to any of the rules, because the manufacturer is not known, or the type or version are not the same as the ones pointed by the rules, then it is discarded.

Alternately the frame is packed inside a buffer with the information ordered in a fixed format, as described in figure 6.2.1. Then the buffer is sent via UART to the LoRa board.

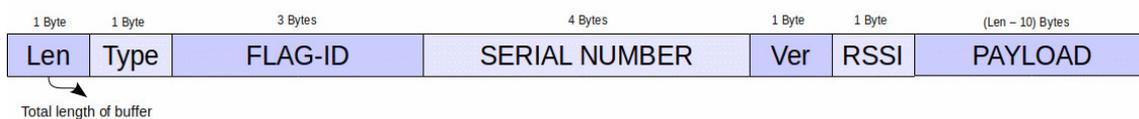


Figure 6.2.1: Format of the buffer sent from the WM-Bus board to the LoRa one

A new entry for the rules table can be received via UART, as the LoRa board receives an update. This new entry is stored in flash, and the length of the table is updated. As a recall, the working state of the WM-Bus board consists in a continuous reception of frames from any WM-Bus device in the 868MHz band. The received frames conform to the known meter format are then packed and sent to the LoRa board.

### 6.2.2 LoRa

The LoRa's microcontroller firmware is based on the ST-Microelectronics LoRaWAN core and drivers libraries.

However the initial project has already been used and modified by the iXem Lab researchers. For this reason the version of the firmware used as a starting point of the development is not the same available on the manufacturer's website. On top of the LoRa core structure a new "sensor" has been created in order to manage the WM-Bus frames reception.

First of all the WM-Bus board sends all the conform frames received, but there is no need to store or send to the network provider each WM-Bus packet received, because in this way more LoRa traffic then necessary is created. That could lead to a blockage in case there's more end-nodes connected to the gateway in use.

To avoid the problem a meters' table is created inside the flash memory of the device, storing the address information about each known physical meter. Each entry of the table contains the following data: Manufacturer FLAG-ID, type, version, serial number and payload length. As for the rules table stored in the WM-bus board flash.

At the reset the whole meters table is read from the storage into the RAM. Each entry value is used to fill a struct variable, in which the payload and the RSSI of the received WM-Bus frame related to the meter taken into account are stored as they are collected.

```

91 struct MT_element{
92     //address
93     char man_flagID[3];           //3 letters FLAG-ID to identify the company
94     unsigned char type;          //type value of the meter: WATER OMS = 0x07
95     unsigned char version;       //version value of the meter
96     unsigned char serial_number[4]; //Serial Number of the meter
97
98     //payload
99     unsigned char* payload;       //pointer to the payload
100    unsigned char payloadlen;     //length of the payload
101
102    //parameters
103    unsigned char rssi;           //RSSI of the received WMBUS packet
104
105    //flags
106    MTelm_acquired_flag acquired; //flag to identify if the meter has been read or not
107 };

```

Figure 6.2.2: Meters table's element struct: single meter identification parameters and WM-Bus frame variable storage

When a meter frame is received for the first time the "acquired flag" is set. In this way any following frame with the same dispatcher address is discarded. When all the known meters are acquired, namely the meters table is full, the LoRa end-node then pack and sends one by one all the table entries through the LoRa gateway to the network provider.

The prototype is a state machine which cycles between multiple states. At the reset

the LoRa board starts in ”*Operational Mode*” in which it receives the conform WM-Bus frames from the UART and stores them in the meter table.

When the table is full the device switches into another state in which sends one by one the received frames while clearing the acquired flags of each entry.

At the end of this cycle, when the table is empty again, the LoRa board goes in stop mode. It’ll wake up after a preset period of time in Operational mode.

Ideally the LoRa board could switch off the WM-Bus one when the meter table is full, and switch it on at the wake up. Due to the lack of possible interconnections between the boards, this option has not been pursued. This optimization could be implemented in a future version of the device.

Moreover the LoRa end-node is able to receive from the network provider updates for the meters or rules tables, consisting in new entries or removal of old ones. Each update related to the rules table is automatically sent through the UART to the WM-Bus board.

It is also possible to go in a *dummy mode* in which the meters table is ignored and each conform WM-Bus meter frame is sent directly to the application server. In this way it’s possible to evaluate the RSSI of the received packets in order to choose which are the best meters to connect to the end-node in observation, and thus filling the meters table with valid entries.

Switching between these modes is managed through a set of flags. In the next pages some useful images to understand better the software structure can be found (figures 6.2.3 and 6.2.4).

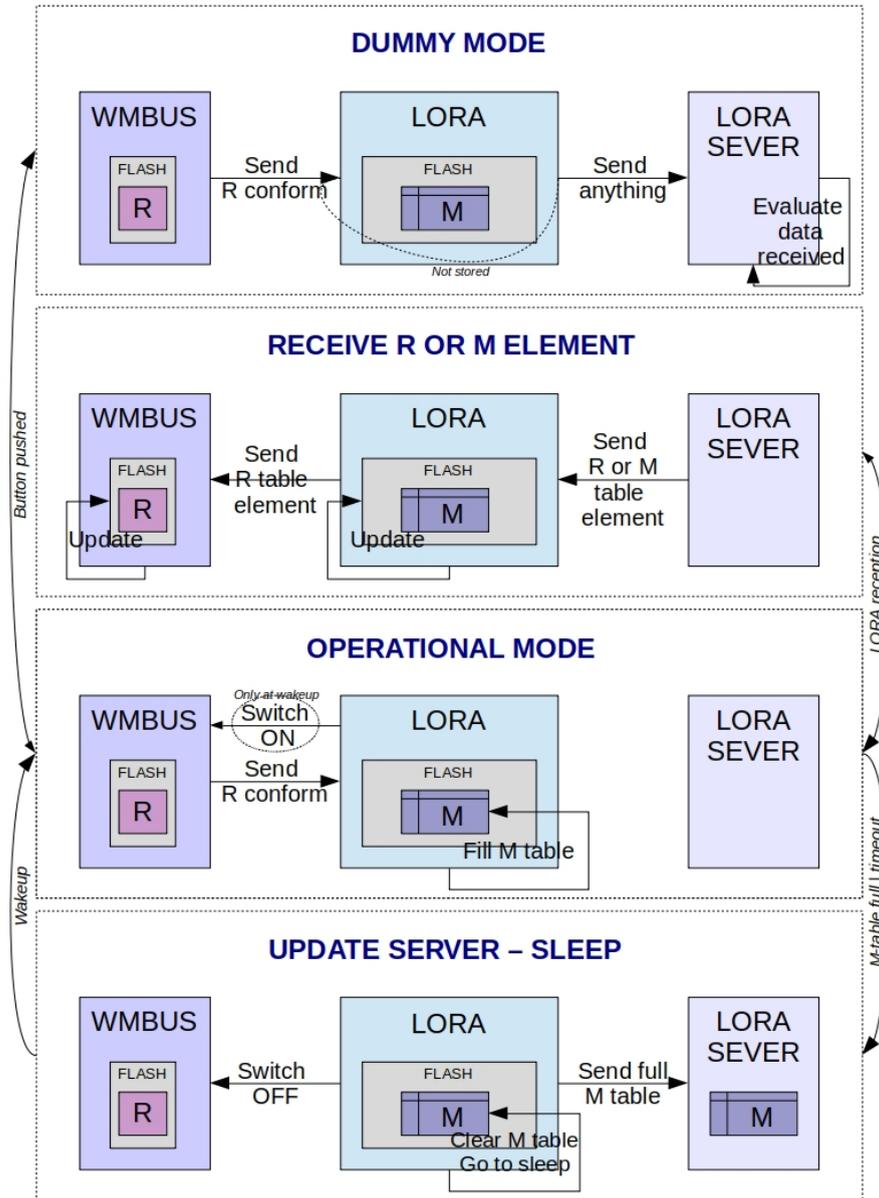


Figure 6.2.3: Overview of the LoRa prototype 1 board software structure

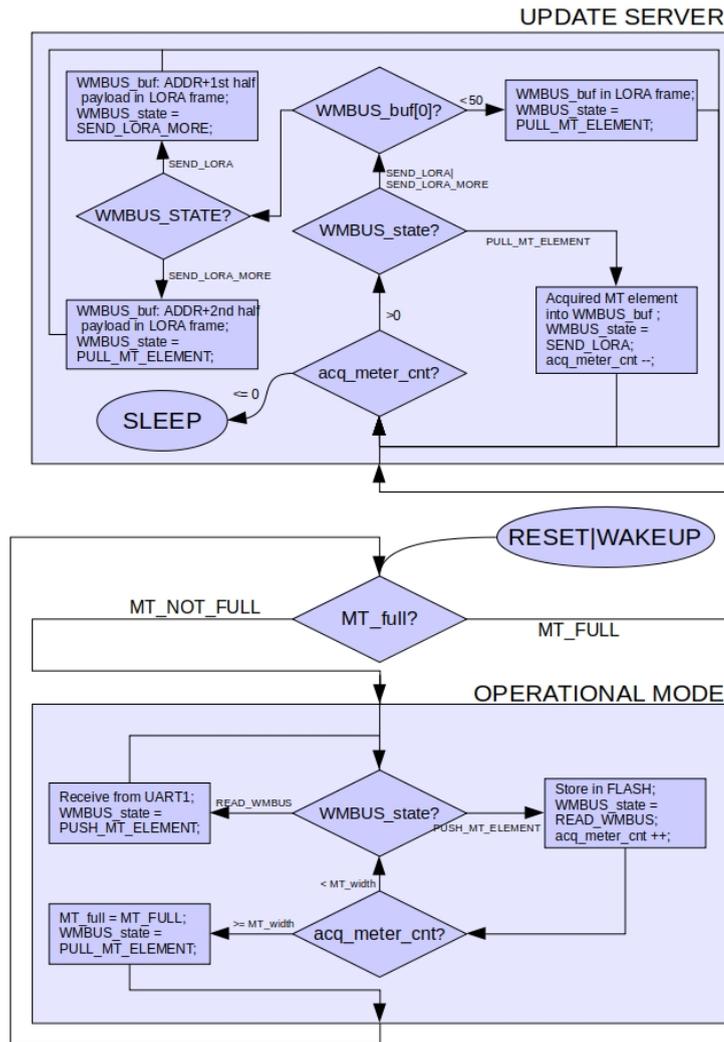


Figure 6.2.4: Overview of the flags used by LoRa prototype 1 board software structure

### 6.3 Power consumption test

The prototype have been subject to a power consumption test using a regulated DC power supply ATTEN TPR3003-3C and a RIGOL DM3068 digital multimeter (see figure 6.3.1).



Figure 6.3.1: Power consumption test instrumentation

The test showed what expected: the hardware structure of the prototype is expensive in terms of current consumption; the WM-Bus 868MHz evaluation board consumes around 25mA steadily. This is probably due to the USB chip present on the board which is not possible to disconnect without altering the board interconnection possibilities.

Moreover the LoRa evaluation board in stop mode consumes more then the 3nA measured by the iXem Lab researchers (see chapter 5.2.2).

For this reason a proper power consumption test has been delayed to the final designed PCB, because it will not infer a lot of the hardware architecture of the evaluation board, like the USB chips and the ST-Link debugger chip.

## 7 Second prototype

Starting from the first prototype, and thus the hardware structure of the evaluation boards, a new device has been designed. It will be the starting point for the development of the final water consumption WM-Bus to LoRa gateway.

Actually, only the hardware structure of the prototype has been developed, but the software architecture will be derived from the previous prototype, with only minor adjustments.

The evaluation boards modified and joined are:

- B-L072Z-LRWAN1 for the LoRa core;
- STAVAL-IDS001V4M for the WM-Bus core at 868MHz;
- STAVAL-IKR002V1 for the WM-Bus transmission at 169MHz.

A 433MHz WM-Bus element has not been integrated, because there are no water meters working on that band between the available ones.

Instead, the existence of two separate cores, WM-Bus' and LoRa's, leads to the presence of two distinct microprocessors:

- For the LoRa core the STM32L072CZ, which is embedded in the CMWX1ZZABZ-091 module (see chapter 5.2 for more information);
- For the WM-Bus core the STM32L151CB (see paragraph 4.2.2 for more information).

On the evaluation boards there is an EEPROM memory chip, which is not included in the design since both the microcontrollers embed a FLASH memory wide enough for our implementation.

The development of the hardware structure of the prototype followed the classic steps: schematic design, PCB design, soldering and mounting.

### 7.1 Schematic

The schematic has been developed using Eagle by Autodesk. It has been divided in more blocks, using modules, in order to simplify the readability.

A block has been dedicated to the LoRa core. This core is based on the iXemLab's architecture.

The CMWX1ZZABZ-091 module has been connected to the oscillator and to bypass capacitors. There are three separated power supply lines one for the microcontroller, another for the USB and the last one for the RF. In this way it is possible to connect or disconnect each module from the main power supply. This choice can be done by

soldering  $0\Omega$  resistors on the final PCB.

Moreover a reset push button has been connected to the active low reset line of the LoRa module through a pull-up network.

The RF circuit is not integrated in a chip balun, but a discrete  $\pi$  net has been designed.

The LoRa core is connected to the WM-Bus one through the USART1 serial line of the microcontroller. Moreover the PB7 pin has been connected to the main reset line of the WM-Bus microcontroller, giving the LoRa module the power to control the STM32L151CB behavior.

From the outside the LoRa microcontroller is accessible through the SWO debug and programming port. There are also two LEDs connected to output pins of the STM32L072CZ to have a visual confirm of the transmission on the network.

The WM-Bus section, on the other hand, is composed of the microcontroller (as stated before) and the two transceivers:

- SPSGRF-868 integrated chip for the 868MHz band. It embeds the SPIRIT1 transceiver, with an integrated balun and chip antenna.
- SPIRIT1 for the 169MHz band. Its discrete RF circuit has been designed starting from the schematic of the evaluation board. The antenna will be connected through an SMA.

The first transceiver is connected to the STM32L151CB SPI2 port, the other to the SPI1. The WM-Bus core is connected to the LoRa one through the USART2 port. Both are connected to the same reset signal coming from the MCU.

From the outside the WM-Bus microcontroller is accessible through the SWO debug and programming port. There are also two LEDs connected to output pins of the STM32L151CB to have a visual confirm of the transmission on the networks. The reset of the core is accessible from a push button connected via a push up circuit.

For debug purposes the UART and both the SPI serial lines are accessible thanks to a connection to pins.

The power supply will be provided using two AA batteries or a set of pins, moreover there is a test pad for the GND.

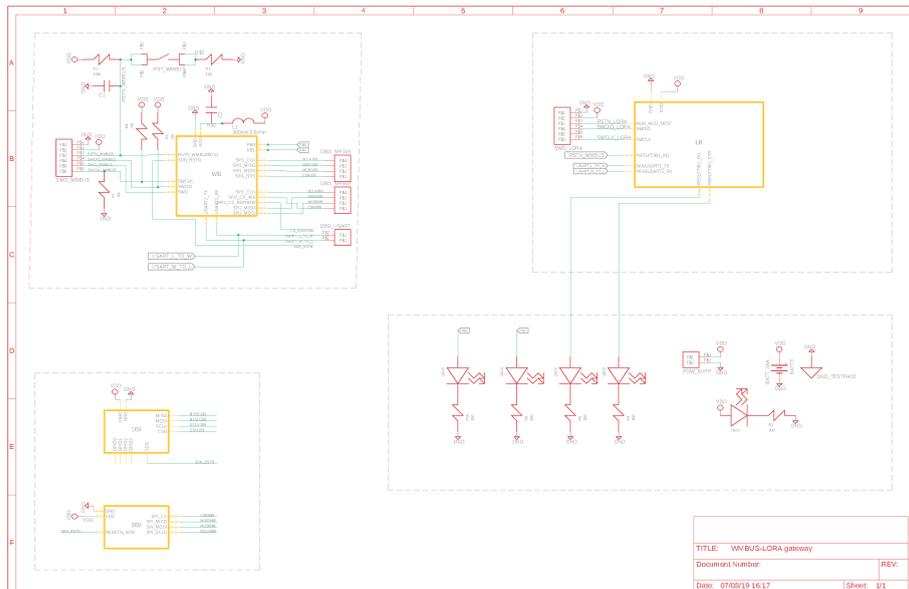


Figure 7.1.1: Schematic: top overview

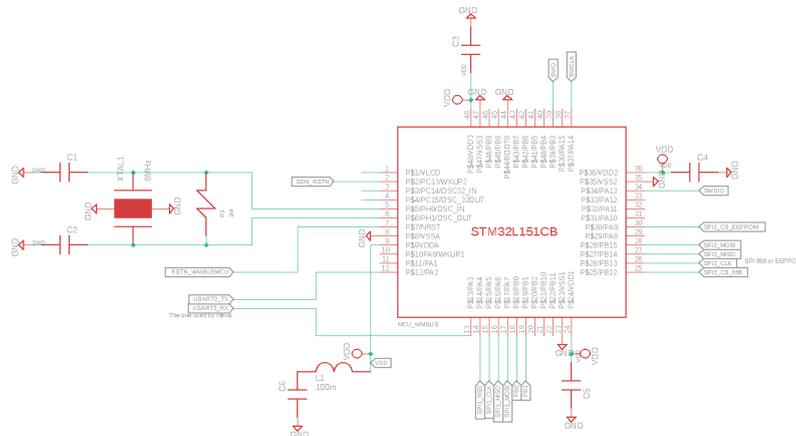


Figure 7.1.2: Schematic: WM-BUS MCU overview

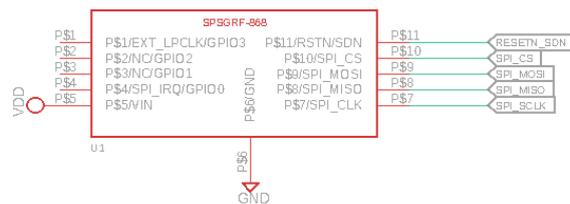


Figure 7.1.3: Schematic: WM-Bus 868MHz transceiver overview



## 7.2 PCB

Starting from the schematic, the PCB has been designed using Eagle. It will be manufactured by PCBWay prototype service [34], thus it must comply to its capability:

- Board Size Tolerance:  $\pm 0.2\text{mm}/\pm 0.5\text{mm}$ ;
- Board Thickness: 0.4-2.4mm (10%);
- Minimum Trace Width and spacing: 0.1mm/4mil;
- Drill Sizes (CNC): 0.2-6.3mm;
- Minimum Character Width x Height(Legend): 0.15mm x 0.8mm;

A script to check the DRC errors during the design is provided. With it is also possible to create the Gerber files.

The board is designed on 4 layers. The top and the bottom ones contain all the various component, the second layer is filled with a ground plane and the third with the VDD plane.

The battery case is placed on the bottom. The LEDs and pins could be mounted either on top or bottom.

The top layer contains the two MCU and two transceivers. To have a compact final shape of the device, the SPSGRF-868 has been placed in a more central position, and a cutout has been performed on the edge of the PCB to free the antenna of further ground planes as required by the device datasheet.

The bypass capacitors are placed near the dedicated MCUs power supply pins. The LEDs, buttons and pins are instead placed on the edges of the board.

All the RF circuits and interconnections are placed on the top. For this reason that layer has been filled with another GND plane, in order to shield the RF routes. For a more accurate RF behavior, the main RF interconnection has been surrounded with vias connected to the GND plane on the second layer. The same vias are also placed around the whole board to connect the two ground planes.

The width of the RF routes has been computed using "Saturn PCB" tool and the PCBWay constraint for the routing minimum width, conductor material and substrate parameters. Both the 169 and 868MHz routes have a width of 0.21mm since the parameter is not really affected by the frequency.

Here the final top and bottom view of the gerber files is shown.

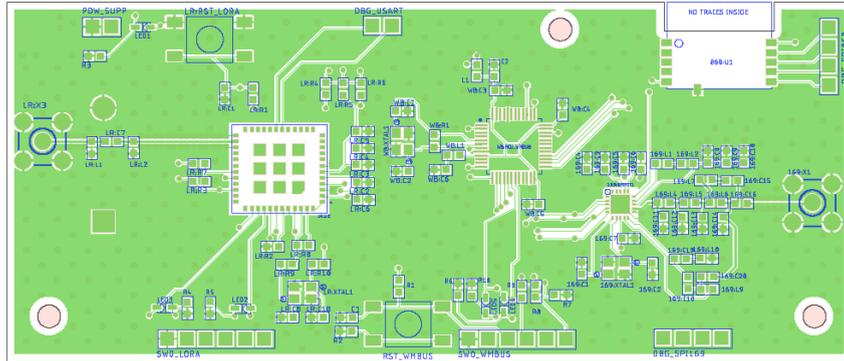


Figure 7.2.1: Top view of the final PCB

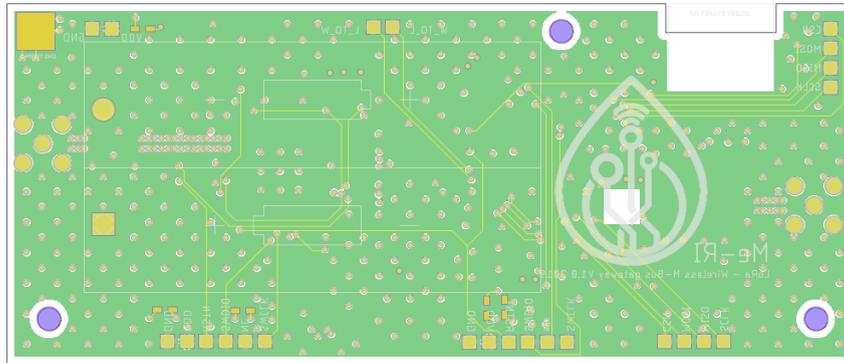


Figure 7.2.2: Bottom view of the final PCB

### 7.3 Soldering and Mounting

The various components and the bare board has been purchased. Moreover a box have been chosen to contain the prototype hardware and protect it during the further in-field tests.

The single components have then been manually soldered onto the PCB using the Politecnico's instrumentation.

Since the 868MHz WM-Bus module embeds a chip antenna, only two antennas have to be chosen in order to be mounted on the device (see figure 7.3.1):

- 868 MHz LoRa: a 5.2cm long pigtail antenna;
- 169 MHz WM-Bus: a 7cm long pigtail antenna.

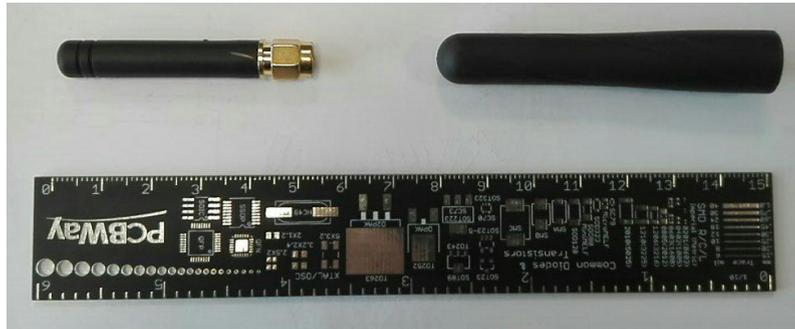


Figure 7.3.1: 868MHz LoRa antenna (left), 169MHz WM-Bus antenna (right)

**Soldering** When the bare boards (figure 7.3.2) and all the components were delivered to the lab, the assembly process began.

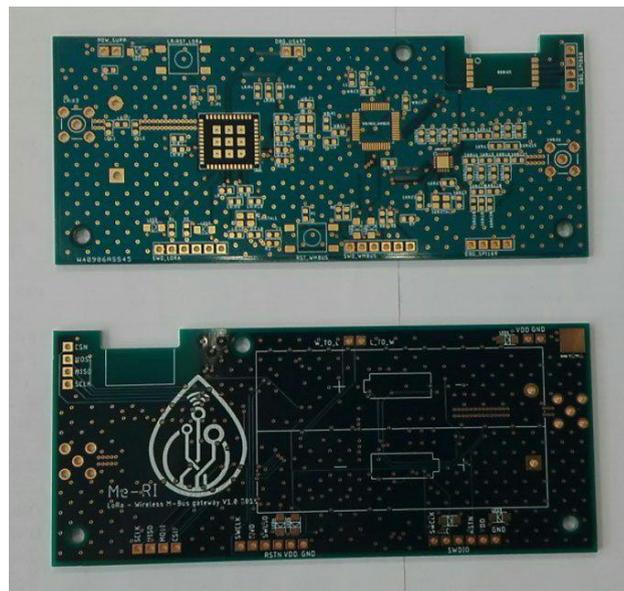


Figure 7.3.2: Bare PCB of the second prototype, top and bottom view

Soldering was performed inside the "Laboratorio hardware microelettronica" of the electronics department DET of Politecnico di Torino.

The process was divided into four main steps and performed first on the top layer, using a solder with higher fusing temperature, and then on the bottom with lower fusing temperature. In this way when the board is flipped bottom side up to work on the bottom part, the soldered top components do not detach due to fusing.

The soldering process steps are:

1. Place the solder on the pads using a needle. The solder chemical composition is different for the two faces of the PCB (low temperature: Sn42/Bi58, high temperature: Sn96/5Ag3.0/Cu0.5), but it is always in a micro-ball form. The flux-liquid is spread on the pads together with the microball of solder. The tool used to perform this job is a Martin Clever Dispense-4 (figure 7.3.3).
2. Use a pick and place machine to place the components on position. The machine is an Essemtec. It is operated by hand. The single components are picked using a suction needle thanks to the void that creates when the hole is obstructed by the component surface. The devices are then placed in the position with the correct orientation thanks to a microscope-camera connected to a screen. The PCB silkscreen contains the names of the components, the edges and orientation symbols in order to help this process. The solder placed on the pads prevents them from escaping the position like a drop of water, but the boards must be handled with care, because a fast movement or a touch on the surface could ruin all the work (figure 7.3.4)
3. Put the boards inside an oven to fuse the solder. The oven is a T-962A Infrared IC Heater. Multiple heating curves, differentiated by peak temperature and time period, can be selected. All the curves have a slow warm-up phase, followed by a short peak when the solder fuses, and then follow a cooling trend. When the oven finishes its work the components are soldered to the PCBs, but there can be faults on the single solder points so a check must be performed (figure 7.3.5).
4. Adjust by hand any faulty point using more fluxant, tin, "tin-eating" tools and welder. This last step was mainly performed by Mattia Poletti, one of the lab researchers. It is a complex task, because there are many possible kinds of fault, and the correction approach is different for each of them. The short-circuit and open circuit point were fixed at the end.

All the steps are documented by the pictures contained in this paragraph.

At the end of the whole process on both sides, the board was cleaned from the excess of flux liquid using a brush and a proper detergent (figure 7.3.6).

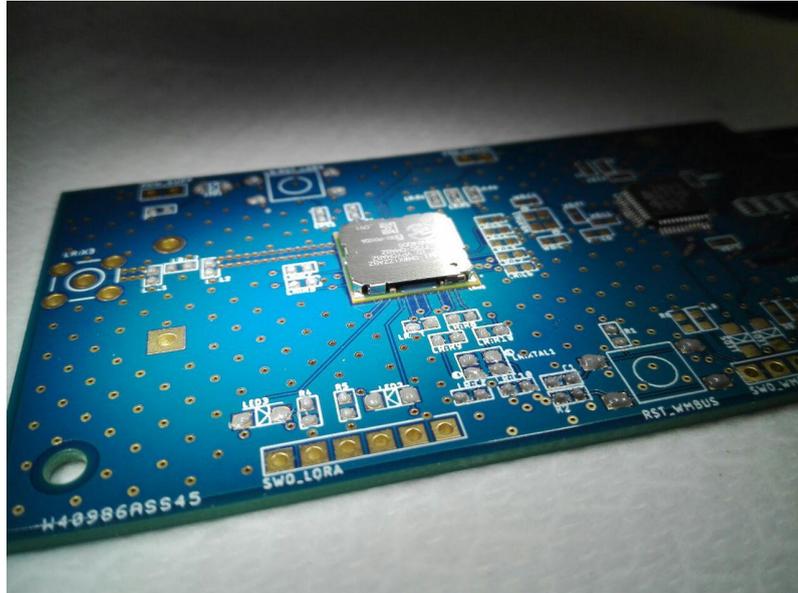


Figure 7.3.3: Solder placed on PADS

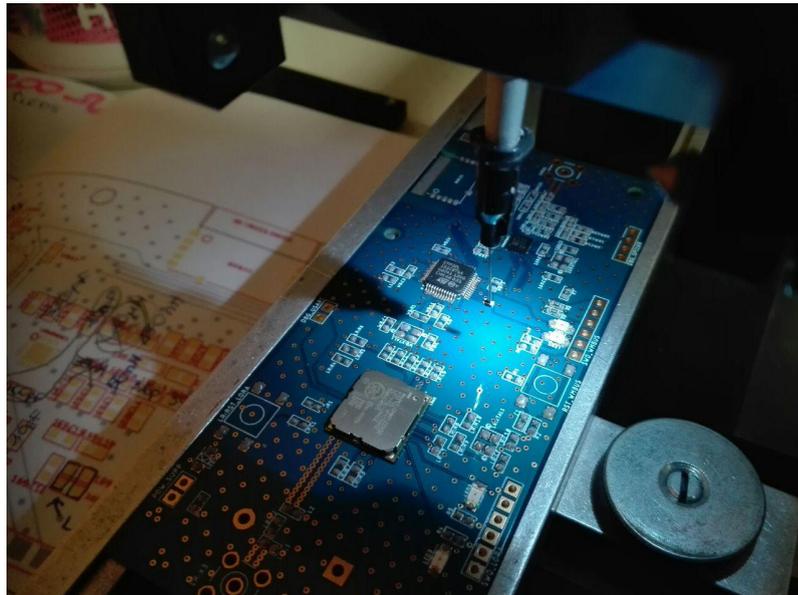


Figure 7.3.4: Pick and Place task



Figure 7.3.5: Board placed inside the oven

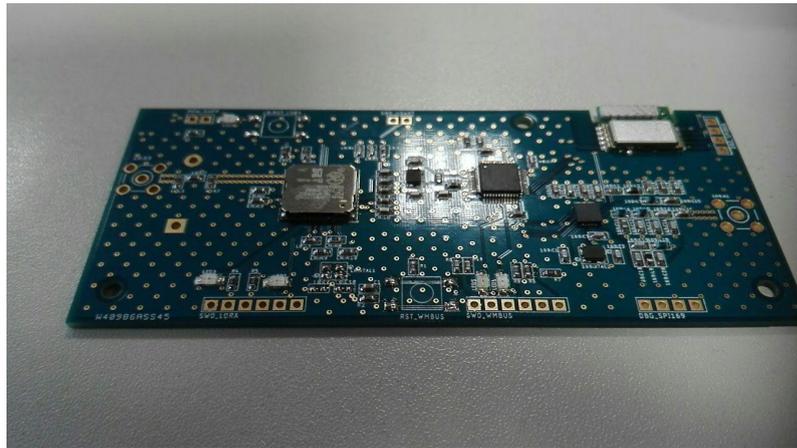


Figure 7.3.6: Photo of the cleaned board

**Mounting** The soldered device now has to be mounted inside a box and connected to the LoRa 868MHz and WM-Bus 169MHz antennas.

First of all the dimension of the final device have been measured in order to choose an appropriate box. It have been identified three possible type of antenna mounting:

- TYPE A: the two antennas are mounted perpendicular to the PCB, impacting the total depth of the device, but not the height and width (15.5x5x95 cm);
- TYPE B: the two antennas are mounted parallel to the shorter side of the PCB;

in this way the height is maximized but the length and depth are minimized (15.5x10x3 cm);

- TYPE C: the two antennas are placed as an extension of the length of the PCB, minimizing depth and height (24,5x5x3 cm).

In addition to these possible device dimensions, the box have been chosen for its impermeable characteristics. Those parameters are rated with the international IP standard which is based on the classification defined by the tables 7.3.1 and 7.3.2.

Table 7.3.1: IP code classification: Solid protection

IP Code	Protection
1	Protection from contact with any large surface of the body
2	Protection from fingers or similar objects
3	Protection from tools, thick wires or similar objects
4	Protection from most wires, screws or similar objects
5	Partial protection from contact with harmful dust
6	Protection from contact with harmful dust

Table 7.3.2: IP code classification: Moisture protection

IP Code	Protection
1	Protection against vertically dripping water
2	Protection against vertically dripping water (device tilted up to 15°)
3	Protection against direct sprays of water (device tilted up to 60°)
4	Protection from sprays and splashing of water in all directions.
5	Protection from low-pressure water from a nozzle with a 6.3mm diameter
6	Protection from powerful water jets from a nozzle with a 12.5mm diameter
7	Protected from immersion in water with a depth of up to 1 meter
8	Protected from immersion in water with a depth of more than 1 meter

The device could be placed in a manhole under the city street, so the protection has to be at least IP67, since it could be subjected to sinking.

The boxes ABS FIBOX TEMPO TA191209 and Bopla Bocube 96025225 have been identified, choosing the TYPE B style of mounting.

The final device is shown in picture 7.3.7.

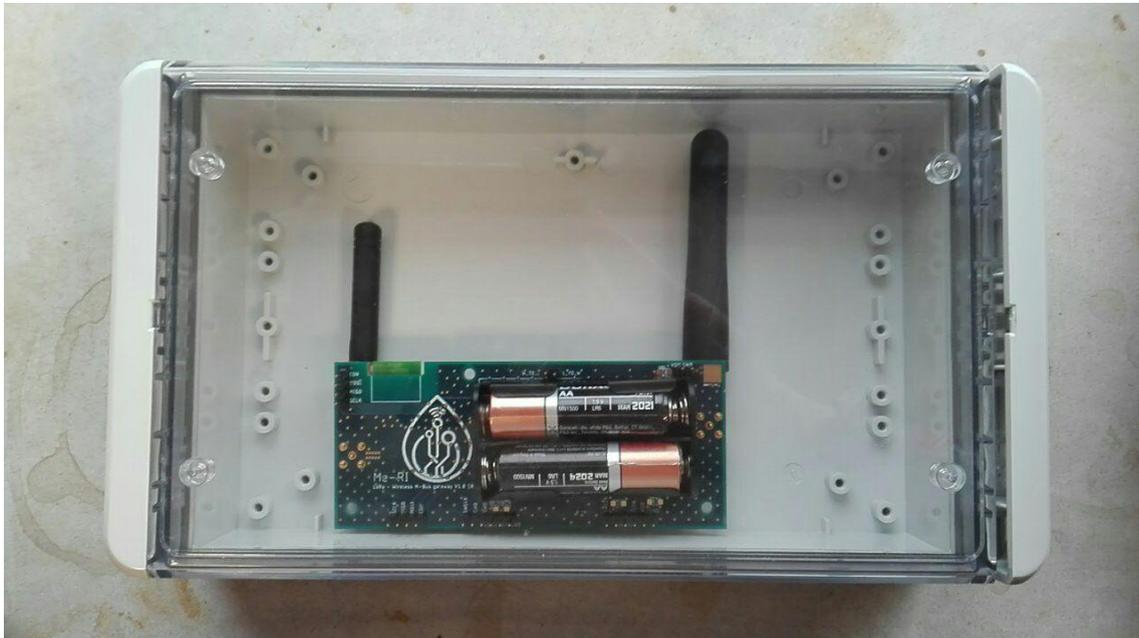


Figure 7.3.7: Photo of the final device

**Tests** The final assembled prototype must be subject to range and power consumption tests.

This must be done to assess if the device could be solely battery powered, and choose the proper antenna to have the desired range.

Unfortunately due to lack of time, it has not been possible to perform these tests during the thesis months.

They will be carried out by the author of the elaborate during a research associate period.

## 8 Further improvements

At the end of this thesis the Wireless M-Bus - LoRa gateway in design is still in development phase. A lot of steps have been taken but the device could be still vastly improved.

From the software point of view:

- The reception through the LoRa network must be improved to acquire new elements for the meters and rules tables present in the FLASH memory of the two microcontrollers.
- The “dummy mode” must be implemented (see figure 6.2.3). In this way the device could be used during in field tests when the meters in range are not known.
- The application server layer must be developed, linking to the SMAT server the received data on *The Things Network*. In this way the database of the measurements acquired with the system could be created.

From the hardware point of view:

- The whole system could be implemented using a single microcontroller, in order to lower the consumption and simplify the implementation. This means that the LoRa software core and the WM-Bus one could be collapsed in a single entity. The design could be vastly simplified this way .
- A single 868MHz antenna could be employed, because the WM-Bus and LoRa communications are exclusive.

Moreover the prototype must be tested first in laboratory to assess its power consumption and test the battery life, but also in field to estimate its range. The test must refer both to the WM-Bus reception from known meters deployed in Turin and to the LoRa gateway interconnection.

*This project will proceed under a research associate contract.*



## Acknowledgements

First I would like to acknowledge Professor Daniele Tincherò for the possibility to work on this project and his support throughout all the thesis work.

I'd like particularly to thank Mattia Poletti and Giovanni Colucci for their patience, help and problem solving skills that they provided to me during this months.

I'd also like to remember Tiziana Gasparoni, Mattia Gregolin, Paolo Cielo, Aiman El Damaty, Michele Rodighiero and all the other people that I met in the iXem headquarter for their company in the laboratory and during all the delayed lunches and coffees.

I'm really grateful to Federico Panicco, my partner in crime, for always been there for me, even despite the distance, through all the final years of my student career.

I like to thank also my mum and my dad, my grandma, my aunts and all my family for all the support that I had since I started my student career in primary school.

I'd also like to mention my old friends Enrica Baio, Alice Melis, Marco Belli and Eleonora Susanna, thank you for all the laughter during our late nights together.

I'd like to appreciate also Elena Migliorin, Erica Raviola and Paolo Michelotti, misfortune mates since the beginning, I never could have done through all the classes and projects without your company and friendship.

Thank also to Flavio Tanese, Andrea Casalino, Pietro Inglese, Fabio Castagno, Andrea Mongardi, Luca Sasselli and all the other people that I met during my years here at Politecnico.

*"So long, and thanks for all the fish!"*

## References

- [1] iXemLabs website: [www.ixem.polito.it](http://www.ixem.polito.it)
- [2] SMAT website: [www.smatorino.it](http://www.smatorino.it)
- [3] Vivek Mohan, "An Introduction to Wireless M-Bus", Silicon Labs
- [4] EU EN13757 standard online resources:  
["ec.europa.eu/eip/ageing/standards/ict – and – communication/data/en – 13757\\_en"](http://ec.europa.eu/eip/ageing/standards/ict_and_communication/data/en_13757_en)
- [5] "Open Metering System Specification, Volume 2, Primary Communication", Issue 4.1.2/2016-12-16, OMS
- [6] ISO 22158 standard online resources: [www.iso.org/standard/44291.html](http://www.iso.org/standard/44291.html)
- [7] List of registered FLAG ID: [www.dlms.com/flag-id/flag-id-list](http://www.dlms.com/flag-id/flag-id-list)
- [8] Prof.Dr.H.Ziegler, "Dedicated Application Layer (M-Bus)", REV4
- [9] Sensus iPerl online resources: [sensus.com/products/iperl-international/](http://sensus.com/products/iperl-international/)
- [10] Maddalena Arrow online resources: [www.maddalena.it/prodotti/moduli – di – comunicazione/arrow – 868 – 915 – mhz/82](http://www.maddalena.it/prodotti/moduli_di_comunicazione/arrow_868_915_mhz/82)
- [11] Dihel Hydrus online resources: [www.diehl.com/metering/us/en/hydrus/](http://www.diehl.com/metering/us/en/hydrus/)
- [12] Watertech MultiReader-C-169 online resources: [www.smartmetering.it/](http://www.smartmetering.it/)
- [13] "Telit Wireless M-Bus 2013 Part 4 User Guide", REV14 2016-01-11
- [14] AT-Commands wikipedia page:  
[en.wikipedia.org/wiki/Hayes\\_command\\_set](http://en.wikipedia.org/wiki/Hayes_command_set)
- [15] MickroElettronica Wireless M-BUS-Click board online resources:  
[www.mikroe.com/m – bus – rf – click](http://www.mikroe.com/m-bus-rf-click)
- [16] Arduino UNO online resources:  
[store.arduino.cc/arduino – uno – rev3](http://store.arduino.cc/arduino-uno-rev3)
- [17] Arduino MEGA online resources:  
[store.arduino.cc/mega – 2560 – r3](http://store.arduino.cc/mega-2560-r3)
- [18] Arduino Intel Galileo online resources:  
[www.arduino.cc/en/ArduinoCertified/IntelGalileo](http://www.arduino.cc/en/ArduinoCertified/IntelGalileo)
- [19] Arduino DUE online resources:  
[store.arduino.cc/arduino – due – without – headers](http://store.arduino.cc/arduino-due-without-headers)

- [20] STEVAL-IKR002V1 online resources:  
[www.st.com/en/evaluation-tools/steval-ikr002v1.html](http://www.st.com/en/evaluation-tools/steval-ikr002v1.html)
- [21] STEVAL-IDS001Vx online resources:  
[www.st.com/en/evaluation-tools/steval-ids001v4.html](http://www.st.com/en/evaluation-tools/steval-ids001v4.html)
- [22] STEVAL-IDS001V4M online resources:  
[www.st.com/en/evaluation-tools/steval-ids001v4m.html](http://www.st.com/en/evaluation-tools/steval-ids001v4m.html)
- [23] SPIRIT1 transceiver online resources:  
[www.st.com/en/wireless-transceivers-mcus-and-modules/spirit1.html](http://www.st.com/en/wireless-transceivers-mcus-and-modules/spirit1.html)
- [24] SPIRIT1-GUI online resources:  
[www.st.com/content/st\\_com/en/products/embedded-software/evaluation-tool-software/stsw-connect009.html#overview](http://www.st.com/content/st_com/en/products/embedded-software/evaluation-tool-software/stsw-connect009.html#overview)
- [25] ST-Microelectronics WM-Bus stack software user guide:  
[/www.st.com/content/ccc/resource/technical/document/application\\_note/3f/fb/35/5a/25/4e/41/ba/DM00233038.pdf/files/DM00233038.pdf/jcr:content/translations/en.DM00233038.pdf](http://www.st.com/content/ccc/resource/technical/document/application_note/3f/fb/35/5a/25/4e/41/ba/DM00233038.pdf/files/DM00233038.pdf/jcr:content/translations/en.DM00233038.pdf)
- [26] STM32L1 microcontroller online resources:  
[www.st.com/en/microcontrollers-microprocessors/stm32l1-series.html](http://www.st.com/en/microcontrollers-microprocessors/stm32l1-series.html)
- [27] LoRaWAN alliance website: [lora-alliance.org](http://lora-alliance.org)
- [28] CSS for LoRa online resources: [www.sghosly.com/p/lora-is-chirp-spread-spectrum.html](http://www.sghosly.com/p/lora-is-chirp-spread-spectrum.html)
- [29] B-L072Z-LRWAN1 LoRa Discovery kit online resources:  
[www.st.com/en/evaluation-tools/b-l072z-lrwan1.html](http://www.st.com/en/evaluation-tools/b-l072z-lrwan1.html)
- [30] CMWX1ZZABZ-091 online datasheet: [wireless.murata.com/pub/RFM/data/type\\_abz.pdf](http://wireless.murata.com/pub/RFM/data/type_abz.pdf)
- [31] STM32L0 microcontroller online resources: [www.st.com/en/microcontrollers-microprocessors/stm32l0-series.html](http://www.st.com/en/microcontrollers-microprocessors/stm32l0-series.html)
- [32] Colucci, Poletti, Stefanelli, Trincherò - *"Internet of Things as a Means to Improve Agricultural Sustainability"*
- [33] The Things Network website: [www.thethingsnetwork.org](http://www.thethingsnetwork.org)
- [34] PCBway website: [www.pcbway.com](http://www.pcbway.com)