

# On fast prototyping LoRaWAN: a cheap and open platform for daily experiments

ISSN 1751-8644  
doi: 0000000000  
www.ietdl.org

Mariano Pulpito<sup>1</sup>, Paolo Fornarelli<sup>1</sup>, Claudio Pomo<sup>1</sup>, Pietro Boccadoro<sup>1,2</sup>, Luigi Alfredo Grieco<sup>1,2</sup>

<sup>1</sup> Dep. of Electrical and Information Engineering (DEI), Politecnico di Bari, Italy, E-mail: m.pulpito@studenti.poliba.it, p.fornarelli@studenti.poliba.it, c.pomo@studenti.poliba.it, pietro.boccadoro@poliba.it, alfredo.grieco@poliba.it

<sup>2</sup> CNIT, Consorzio Nazionale Interuniversitario per le Telecomunicazioni

**Abstract:** Low-Power Wide Area Networks (LPWANs) are gaining momentum within Internet of Things (IoT) circles, thanks to their inherent capabilities to simplify network topologies and increase coverage in low rate low power communications. This manuscript focuses on a specific LPWAN technology, namely Long Range Wireless Area Network (LoRaWAN), which is being widely investigated by the research community worldwide. LoRaWAN offers many noticeable features, including: wide coverage areas, security by design, power consumption and bandwidth optimization, and a high degree of flexibility and reconfigurability. Unfortunately, in spite of these valuable characteristics, LoRaWAN appliances are often closed source and proprietary. To bridge the gap, this manuscript proposes a prototype, which has been designed and set up using open low-cost experimental boards and programming tools, only. Moreover, an experimental campaign has been carried out in order to demonstrate the effectiveness of the developed platform.

## 1 Introduction

LPWAN technologies are gaining momentum in many IoT application domains because of their inherent capabilities to provide long range coverage with low power and low rate communications. The different LPWAN solutions on the market seek different trade offs between modulation scheme, bandwidth occupation, latency, downlink and uplink data rates, and End-to-End (E2E) security functionalities [1]. Between LPWAN solutions, those working in Sub-1GHz bands, such as Weightless-W and Ingenu, offer robust and reliable communication, as these frequencies result to be less congested than around 2.4 GHz, which is mostly used by other commonplace wireless technologies (i.e., Bluetooth, ZigBee and Wi-Fi). Reducing bandwidth occupation is one of the leading principles of NB-IoT and Weightless-P, implementing narrowband modulation techniques, and Sigfox, that addresses noise reduction thanks to Ultra Narrow Band (UNB) solutions.

In this context, Long Range (LoRa) is a well known technology, which is acquiring a considerable interest in several use cases. One of the most noticeable example of LoRa's capabilities is the *inteliLIGHT* [2] project, an extensive lightning control solution that allows a punctual lamp control enhanced by long-range remote activation functionalities. The project proved manageability of street-light control in a real urban environment, and turned from a simple ON/OFF lamps switching tool to control and monitor all the lighting panels over a city. Another interesting case study is the *WAZIUP* [3] project, which employs LoRa technology to detect cattle position and prevent their theft. *WAZIUP* aims at developing a whole LoRa-compliant system while lowering design efforts and costs. The project has been conceived as a all-in-one solution, thus envisioning a complete infrastructure, from hardware components to web application, with a human appealing, as it has been thought to be deployed in rural scenario in Africa. The main difference between these two projects is the application field: the first proves LoRa's usability in urban scenario, whereas the second puts it at work in suburban environment.

The LoRa standardization process is cared by the LoRa Alliance [6]. LoRa's Physical Layer is based on Chirp Spread Spectrum (CSS) [4] modulation scheme. The LoRa protocol stack also offers custom characteristics at Media Access Control (MAC) layer, with LoRaWAN. It represents a long range and low power one-hop communication model. As a consequence, the network architecture is

simplified and resolves most of the issues encountered with multi-hop wireless networks [1]. Thanks to the LoRaWAN MAC, it is possible to build and maintain a network composed of functionally differentiated devices, connected to the Internet through dedicated gateways.

A LoRa gateway directly interact with network servers, via a backhaul relying on Transmission Control Protocol (TCP)/Internet Protocol (IP) Secure Sockets Layer (SSL) technological solutions. In LoRa, Network Servers and web services play a very important role, as they are in charge of data collection, management and elaboration. LoRaWAN also ensures security and provides E2E encryption capabilities. In particular, LoRaWAN is able to grant two different joining procedures for end-devices willing to connect to a reference gateway: Over-The-Air Activation (OTAA) and Application by Personalization (ABP). The main differences between the two regards the Network session key, generated at the beginning of the former procedure any time it is started. The latter joining procedure, instead, directly connects end-devices to the network without initiating the request, neither accepting the procedure. In fact, the end-device stores a set of dedicated parameters that allow a direct messages encryption.

LoRaWAN technology can provide many promising features, but currently available appliances implementing its features are commonly proprietary and closed source, thus hindering customization, development of new features, and wide spread adoption. Luckily, together with the diffusion of LPWAN technologies, in particular LoRaWAN, it is now possible to find LoRa-compliant hardware components that, properly orchestrated with services and programming tools, can lead to the development of open standard LoRaWAN platforms.

In this work, a complete LoRaWAN platform is designed, developed, and experimentally tested, using open and low cost hardware and software components, only. The number of components is use has been minimized in order to obtain a small sized prototype with reduced energy footprint. The developed platform has been conceived as an environmental monitoring unit: it knows its position, thanks to a Global Positioning System (GPS) antenna, and communicates geo-referenced environmental measurements of (i) temperature, (ii) relative humidity, and (iii) particulate matter. The sensing unit of the platform is able to communicate with a LoRa gateway, which receives measurements, and, since it is connected

to the Internet, populates a dedicated database with all sensed values. Data are organized into a 51 byte packet structure. This design criteria minimizes energy consumption of the sensor node, while communicating data, but also eases data elaboration on the receiver side. In order to concretely verify the functionalities of the envisioned platform, an experimental performance evaluation has been carried out. Considering different transmission parameter configurations, some Key Performance Indices (KPI) were taken into account. In particular, the experimental campaign was carried out to evaluate: (i) the energy consumption of the platform, (ii) Packet Loss Ratio (PLR), and (iii) coverage areas. While tuning peculiar LoRaWAN features, such as Bandwidth (BW) and Spreading Factor (SF), some interesting results were measured. For instance, the best values in terms of energy consumption have been achieved with SF=7 and BW=500 kHz configuration, which provides a measured autonomy of 208 h for the sensor node and 358 h for the gateway. As for PLR, instead, the best value, as low as 15%, has been measured with SF = 10 and BW = 500 kHz. The prototype has shown satisfying values in terms of distance covered. For instance, the longest communication range value is longer than 3 km. Nevertheless, lower but interesting values were measured with different SF and BW settings.

Still, the developed open platform has a high degree of modularity, in terms of both hardware and software, which grants several personalization possibilities and further applications support.

The remainder of the present work is as follows: Section 2 describes the technological background. Section 3 presents the realized platform, focusing on the main hardware and software design criteria and components. With Section 4 the experimental setup is tested in order to provide performance assessments. Section 5 concludes the work and highlights the main future work possibilities.

## 2 LoRa Technology Overview

IoT aims at creating and interconnecting devices able to exchange many types of data in any type of environment. In order to effectively support IoT applications, any proposed technological devices or solutions must respect some major constraints. In particular: (i) low power consumption, (ii) limited computational and storage capabilities, (iii) execution of small process and (iv) low form factor. One of the most interesting aspect of LoRa technology is the way network architecture is redefined to simplify the main building blocks and involved entities. Meanwhile, LoRa satisfies all the aforementioned constraints.

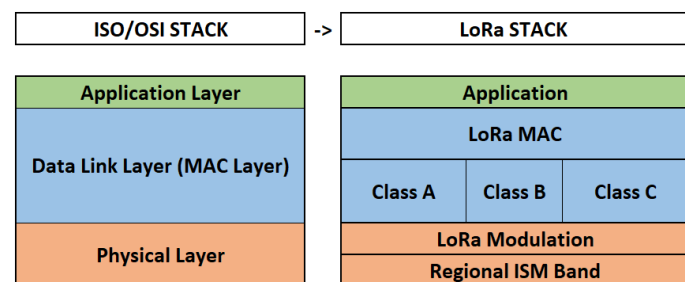


Fig. 1: LoRa Protocol Stack overview.

LoRa Protocol Stack, as defined and maintained by the LoRa Alliance, can be described as reported in Figure 1. It is mainly characterized by three layers:

- the LoRa Physical Layer (PHY), which implements LoRa modulation scheme;
- the MAC layer, which implements LoRaWAN technology;
- the Application layer, which acts like an interface between the end-nodes and the outer world (e.g Internet).

The LoRa's PHY layer is mainly characterized by the CSS modulation scheme [4]. This Spread Spectrum technique has been used

Data Rate	Spreading Factor	Physical Bit Rate [bps]	Maximum Payload Size [Bytes]
0	SF 12 / 125 kHz	250	51
1	SF 11 / 125 kHz	440	51
2	SF 10 / 125 kHz	980	51
3	SF 9 / 125 kHz	1760	115
4	SF 8 / 125 kHz	3125	242
5	SF 7 / 125 kHz	5470	242
6	SF 7 / 250 kHz	11000	242

Table 1 LoRaWAN Data Rates.

for military and space communications for many years as it grants interesting features. Among them, coverage areas granted are long range. Depending on the exploitation area, the covering range of a single base station may range up to 30 km, in a rural context [5]. In summary, among other CSS properties, it is important to highlight: (i) low power consumption, (ii) low latencies, (iii) high robustness, and (iv) resistance against Doppler effect [7].

LoRa is the one of the most interesting example of employment of such features in commercial LPWANs scenarios. Thanks to CSS, LoRa is able to support variable/adaptive data rate, allowing a trade-off between covering range and power consumption, maintaining constant bandwidth [8]. To better understand LoRa's adaptive behavior, it is important to consider three main parameter: SF, Code Rate (CR) and BW. The basic premise of Spread Spectrum is that each piece of information (e.g., a bit) is encoded using multiple chips. The substitution of a bit for multiple chips means that the SF directly impacts the duration of a LoRa packet (from 39.2 ms with SF=8, to 528.4 ms with SF=12). Another design principle recalled here is bandwidth occupancy. LoRa allows the selection of SF trading-off with the BW parameter. LoRa modulation sends the spread data stream at a certain chip rate which corresponds to the selected bandwidth, which can assume three different values, for instance 125, 250 or 500 kHz (respectively influencing time to air values, with the resulting variations from 66 ms to 264.2). Bandwidth is controlled by means of the BW parameter. To improve LoRa transmissions' robustness, the CR has to be taken into account. In particular, with a small overhead of additional encoding, it is possible to implement Forward Error Correction (FEC), that allows to recover bit information in case of corrupted packets. CR's allowed values range from 1 to 4, with a time to air which spans from 148.5 to 123.9 ms. As a result, designers are able to effectively control modulation bit rate (using SF), the noise resistance (thanks to CR) and encoding. For the sake of completeness, with Table 1, some numbers are given in terms of LoRaWAN data rates.

LoRa's network architecture foresees different kind of devices:

- the End-devices, which are the end-nodes of the IoT network, typically in charge of executing "smart" tasks, such as sensing or actuating;
- the gateway, also referred to as the base station. It represents a bridge between the IoT network and the broader Internet. It is in charge of interacting to the Network Server relying on 3G/4G backbone connection;
- the Network Server, indeed a central entity in charge of collecting and storing data, while making them available for the users;
- the Application Server, for instance the interface between the developers and the Network Server. It offers user-friendly functionalities and interfaces to develop specific web services related to a certain specific subset of the amount of data available on the Network Server.

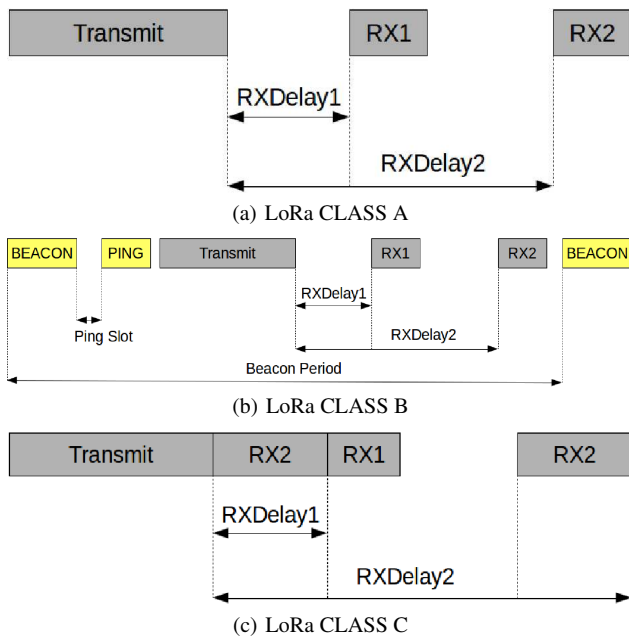
In LoRaWAN scenarios, end-node devices are not associated to a single base station but the data flow reaches all the gateways into the coverage range, for instance, the so called uplink phase. It is important to specify that LoRa solutions are resilient to topology modifications and that the handover of an end-device between different base station can be easily managed, thus limiting Quality of Service (QoS) decay. The data manipulation state is associated to

servers and not a gateway, that is, instead, in charge of checking data correctness and detect duplicated packets, if any. Occasionally, it sends a request to end-device through the appropriate gateway [9], for instance, the downlink phase.

LoRaWAN is the MAC Layer of the LoRa Protocol Stack. These specifications are functionally identifying LoRa devices dividing them into three different classes. Each class carries out specific tasks in a strict timeline, represented in Figure 2 and described in details in what follows:

- CLASS A devices, that use pure ALOHA access for the uplink phase, as depicted in Figure 2.a. They enable asynchronous communication between end-device and the base station. A CLASS A device typically respects a schedule for transmitting and receiving messages. After any transmission from the end node to the base station, the end devices open two reception-dedicated time windows to listen to possible messages coming from the server. CLASS A devices are designed to improve energy efficiency, therefore they are typically low power devices.
- CLASS B devices, whose communications rely on Beacon frames of specified length. In particular, CLASS B base stations provide a Beacon message to wake-up specific nodes in the network. In this way, the LoRa system indicates to the server when end devices can listen. After receiving the Beacon message, an End Node enables its radio transceiver to both send and receive. Between a Beacon and a transmitting time period, it exists a time period called ping slot. Within the inter-Beacon period, CLASS A communication scheme is used as the core of CLASS B devices communications. This mechanism is reported in Figure 2.b. CLASS B devices are preferred for applications with additional downlink traffic needs.
- CLASS C devices, that are different from the others as End-Device are listening in a continuous way, with the only exception of the transmitting period. In fact, these kind of nodes are able to switch between the two types of activities at any time. As a consequence, CLASS C nodes represent the worst case in terms of energy consumption, while allowing the best performances in terms of data transmission. A detail of the communication timeline for CLASS C devices is depicted in Figure 2.c.

It is really important to specify that, within the same network, devices belonging to the three classes can coexist. Moreover, it is also possible for the devices to commute between one class to another, if they are designed to do so.



**Fig. 2:** LoRa class device communication mechanisms differentiation.

In LoRaWAN, communications are strongly centralized and hierarchical. In fact, end devices are not allowed to communicate directly one with the other. All data transmissions are made possible for end devices only towards the base station and, obviously, in the reverse path. This happens thanks to an activation procedure, which mainly consists of four unique keys [9]:

- end device address (namely, *DevAddr*): a 32 bit id of each end device. 7 bits over 32 are used to identify the network id, whereas the remaining 25 bits refer to each individual device. It is important to note that this is a non-unique id;
- Application identifier (namely, *AppEUI*): a global application id;
- Network Session Key (namely, *NwkSKey*): this key is used by the Network Server and the end device in order to verify data integrity;
- Application session key (namely, *AppSKey*): this key is used by Network Server and end device to provide payload encryption and decryption activities.

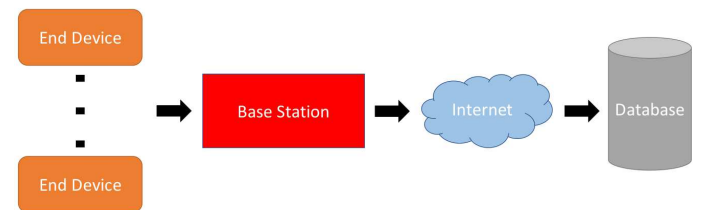
The peculiar characteristics of LoRa's MAC layer allow several advantages [10] in terms of: (i) low-power consumption, (ii) scalability, (iii) high efficiency (with respect to 3G/4G [1]), and (iv) low transmission rate.

The LoRaWAN protocol specifications not only define network architecture and communication protocol but also both signing and encryption for LoRaWAN packets. These operations are completed relying on symmetric keys that are known by both the end-node and the Network Server. Another entity which could potentially be aware of such keys is the Application Server. Keys are distributed in different ways, depending on the way nodes join the network. It is worth noting that security functionalities are out of the scope of the present work.

### 3 The Proposed Open Platform

This Section provides the design rationale and implementation details of the open platform proposed in this manuscript.

With respect to the network architecture described in Section 2, Figure 3 explains the main building blocks that have been implemented. For the sake of simplicity, the developed open platform foresees two different devices: (i) the end device and (ii) the base station. The former is able to communicate but it is also able to sense real environmental data. To this aim, it has been enhanced with sensing units, that will be presented later on in this Section. The latter obeys to the design principles and the main characteristics of LoRa network architecture by providing gateway functionalities.



**Fig. 3:** Prototyping architecture.

The end device has been equipped with several sensors to measure temperature, pressure, relative humidity, and particulate matter (PM10, PM2.5 PM1) [11]. Moreover, it is able to acquire (thanks to a GPS receiver) and transmit its position in order to enable mobile environmental monitoring applications.

Functionally speaking, after sensing activities, the end node has to establish a connection with the reference gateway using the LoRa modem. The latter is, then, in charge of communicating the content of the received packets to a main server with Message Queue Telemetry Transport for Sensors Network (MQTT-SN) [12][13] by

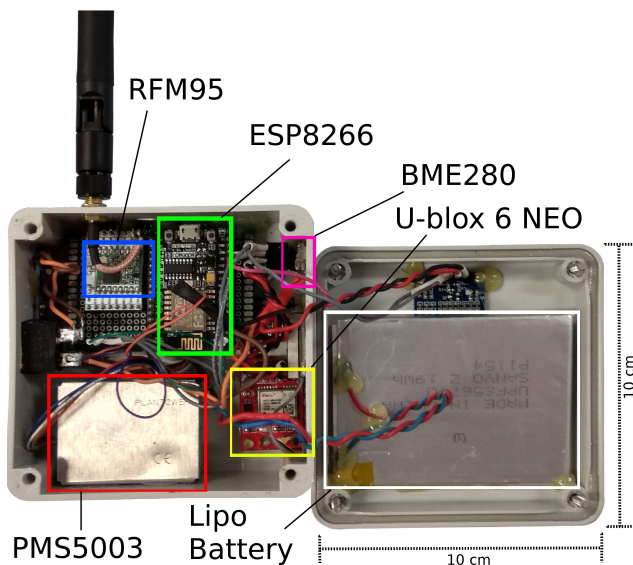
means of the User Datagram Protocol (UDP) protocol. After reception is completed, the main server stores the packets in a local database for any further use.

Even if they are functionally different, these prototypes communicate as they both use a LoRa-compliant radio transceiver, in particular, the RFM95 [16]. To orchestrate operations, an ESP8266 [17] Micro-Controller Unit (MCU) oversees all hardware components. These two equipments are used in both the end node and the gateway and will be presented in details later on in this Section.

### 3.1 Sensor Node

The Sensor Node is based on the idea of integrating common and cheap hardware components. The resulting part list is as follows:

- LoRa Modem: HopeRF RFM95;
- MCU: ESP8266;
- Environmental parameters sensor: BME280;
- Particulate matter sensor: PMS5003;
- GPS antenna: U-blox 6 NEO;



**Fig. 4:** A detail of the main building blocks of the realized end node device.

In order to reach the goal of fast prototyping, the NodeMcu V3 and BME280 [18] open-source development platforms have been used. The former integrates the ESP8266 (v. 12E). The latter, instead, integrates three types of environmental sensors, thus allowing temperature, humidity and pressure measurements. Such variables are of great interest in a large variety of IoT applications, from Smart Cities to Machine-to-Machine (M2M) applications [14]. The particulate matter monitoring can be done using several modules. Some of them are really expensive. To keep the overall cost as low as possible, precision and accuracy of the available solutions have been evaluated. The outcome lead to choose PMS5003 [19]. This sensor is widely adopted in a certain number of commercially available solutions, and is indeed quite popular in the open hardware community. Its internal structure is composed of a fan, used for canalizing the air flow from external environment to the unit which reads the particulate. the reading principle is laser light scattering. A GPS module allowed to geo-localize the environment data useful in the test phase for maximum distance monitoring. The choice of this module, as in previous case, was made because it is not lot expensive and easy-to-find on market. U-blox 6 Neo [20] is the best choice in this way, it is a family of stand-alone GPS receivers featuring the high performance.

By means of a deep investigation of each hardware components, it has been possible to properly operate a choice. The detailed analysis of each of the listed part follows.

**3.1.1 ESP8266:** this Integrated Circuit (IC) is an MCU working at the predefined frequency of 80 MHz and it offers 16 General Purpose Input Output (GPIO) pins with a small form-factor, for instance 5 x 5 mm. It integrates and fully supports the 802.11 b/g/n [15] (WiFi) protocol stack. In terms of memory, it has 64 kB of Random Access Memory (RAM) memory and up to 4 MB of Flash memory. The latter value results to be considerably higher than other popular MCUs (e.g. Arduino's Core, the ATmega328 MCU). The ESP8266 also shows interesting numbers in terms of power consumption, with a Deep sleep mode value  $\leq 10\mu A$  and a Power down leakage current  $\leq 5\mu A$ . In terms of interfaces, the ESP8266 natively supports the most commonly used communication protocols, for instance Inter Integrated Circuit (I2C) [21], Serial Peripheral Interface (SPI) [22] and Universal Asynchronous Receiver-Transmitter (UART) [22]. These interfaces have been useful to connect all the modules to the sensor node. In fact, HopeRF RFM95 uses SPI, BME280 uses I2C, whereas PMS5003 and U-blox 6 NEO use UART. This module operates with a voltage input value equal to 3.6 V.

**3.1.2 HopeRF RFM95:** this popular and widely used hardware module implements LoRa<sup>TM</sup> modulation technique. Among the technical characteristics of this device, one of the most interesting is the high sensitivity value, that reaches -148 dBm. This small sized component, whose dimensions are as small as 16 x 16 mm, operates with a voltage input level equal to 3.6 V. The reference power consumption value is quite limited, as it would be experimentally evaluated and clarified in Section 4 with a dedicated Table. For the sake of completeness, it is important to specify that the LoRa devices composing our open platform belong to those with embedded Network authentication key. For instance, they will use the ABP procedure.

**3.1.3 BME280:** this IC has three different sensors to read environmental parameters. It operates reading values in a range from  $-40^{\circ}C$  to  $+85^{\circ}C$ , as for temperature, from 0 to 100%, as for humidity, and from 300 to 1100 hPa for atmospheric pressure. The sensor's accuracy is equal to  $\pm 3\%$ . In terms of current, its consumption equals  $\leq 3.6\mu A$ , whereas the operating voltage at 3.6 V. This cheap IC has been taken into consideration as it has a small form factor of 2.5 x 2.5 mm.

**3.1.4 PMS5003:** the sensor, used for particulate matter, can be successfully employed in the range of 0.3  $1.0\mu m$ , for measuring PM1, between 1.0 and 2.5  $\mu m$ , as for PM2.5, and from 2.5 to 10  $\mu m$ , for PM10. The resolution equals  $1\mu g/m^3$  and it is characterized by a working temperature range between  $-40^{\circ}C$  and  $+60^{\circ}C$ . As highlighted in Figure 4, the physical dimensions of this component are larger than the others, as they reach 50 x 38 x 21 mm. In terms of power supply, the IC requires a voltage input value equal to 5 V. As for current, this sensor drains an average value of  $\leq 100mA$  during the reading phase. While on standby, instead, current drawn fall down to  $\leq 200\mu A$ . Given the interesting features, the usage of the PMS5003 sensor has been taken into consideration as it allows a wide variety of use cases.

**3.1.5 U-blox 6 NEO:** this GPS antenna provides 50-channel positioning engine, with horizontal velocity and heading equal to 2.5m, 0.1m/s and  $0.5^{\circ}$ , respectively. Between the main features, this piece of equipment shows low values in terms of Time-To-First-Fix (TTFF) in hot start conditions\*, below 1 second. Its power

\*The hot start condition is verified as long as satellite orbits' information do not have to be received by the antenna. If this synchronization phase still has to take place, the procedure goes under the name of cold start and may take some minutes to be completed.



supply equals 3.3 V and the current drawn ranges from 47mA during the acquisition state to 11mA in tracking power save mode. Also in this case, the form factor is limited to 16 x 12.2 x 2.4 mm.

### 3.2 Gateway

In a LoRa solution, the gateway represents a connecting unit that allows the communication between different types of networks. In this scenario, the gateway acts to bridge the LoRa end node and a WiFi network. It is worth noting that the latter is allowed as the ESP8266 supports WiFi protocol. As previously said, the gateway not only integrates the ESP8266 but also features the RFM95. The latter component receives packets from the LoRa network and the MCU ESP8266 sends them to a dedicated web server thanks to the WiFi connection. The recalled web server will be discussed in details later on in the present Section. Nevertheless, it is important to highlight that the gateway is not in charge of elaboration tasks. Data management is only carried out on the server side. Differently from what happened for the End node, the gateway does not include any sensing unit. Moreover, given its fixed and stateful nature, the gateway is not supposed to change its current position over time. Therefore it has not been equipped with GPS.

Functionally speaking, the gateway continuously listens to one channel; the end node coherently transmits over the same one. It is worth noting that multi channel is not supported by the envisioned prototype. This design choice implies two main things: (i) the gateway has very low power consumption values, and (ii) an extremely reduced form factor, for instance 10 x 10 cm, as shown in Figure 4). It is worth noting that the overall dimensions are common for both devices, as they are mainly due to the external coverage.

When comparing this gateway with other platforms, such as WAZIUP [3], it is possible to note some major differences. The central units are single-board computers (e.g. Raspberry Pi), which are undoubtedly more powerful in terms of computational capabilities. But they also are more expensive, in terms of cost per unit. Moreover, the usage of a single-board computer implies an increase in physical size and higher values in terms of power consumption. Accordingly, the aggregated power consumption value of a gateway made with an ESP8266 results to be lower.

Given the Class differentiation described in Section 2, it is important to specify that in the envisioned configuration, communications between *node* and *gateway* are based on CLASS A specifications.

### 3.3 Power supply

One of the most challenging aspect in developing an IoT platform is power supply. Given the set of transmission and reception tasks that repeats themselves over time, current is consumed by the MCU and the radio transceiver. The more frequent the radio activities, the faster batteries will discharge. Indeed, both the node and the gateway have been designed in order to be properly powered up thanks to a battery supply that could grant reliability and long-term functionality even in the case of frequent transmission.

In general, it is possible to estimate the time taken by the envisioned system to discharge it. The battery used for this prototype is a Lithium-ion Polymer battery (LiPo) with 4800 mAh capacity. With this value in mind, it is possible to estimate the Autonomous Time (AT) as the ratio between the Battery Capacity (BC) and the Device Power Consumption (DPC) as shown in Formula 1.

$$AT[hour] = \frac{BC[mAh]}{DPC[mA]} \quad (1)$$

### 3.4 Data Structure

One of the leading design criteria of the envisioned platform has been network efficiency. In order to prove the effectiveness of the transmission activities, the open platform described sends a payload with a fixed number of bytes. This value has a double motivation: protocol constraints and real data coming from sensors. The first can be considered a direct consequence of the architectural description

given in Section 2; in particular, the close relationship between the payload size and the SF leads to a maximum size of 51, the lowest value, and a maximum payload size of 222 bytes in the best datarate case. Given this span, it has been chosen to keep the amount of data fixed to the lowest value in order to concretely prove transmission effectiveness using the worst case as the leading one. For the sake of completeness, it has to be highlighted that the LoRaWAN protocol adds 13 bytes (at least) to the application payload. As for the second, instead, the whole dataset can be optimized without using separators any specific notation.

The structure of the dataset transmitted to the LoRa gateway is reported in Figure 5. In particular, the payload is composed by:

- 8 bytes for longitude information;
- 7 bytes for latitude information;
- 5 bytes for altitude information;
- 2 bytes for PM10 concentration;
- 2 bytes for PM2.5 concentration;
- 2 bytes for PM1 concentration;
- 5 bytes for temperature;
- 5 bytes for humidity;
- 6 bytes for pressure;
- 1 byte to identify the end of the string.



Fig. 5: Structure of the transmitted packet.

### 3.5 Web Architecture

In LoRa, Network Servers and web services play a very important role in terms of data collection, management and elaboration. In order to enrich the prototype functionalities with web services, several possibilities have been evaluated. Among them: (i) containers, (ii) virtual machines, and (iii) microservices. A container can be considered as a *black box* in which software are packaged to run in an isolated environment on a shared operating system. Many solutions could be evaluated for implementing containers and Docker [23] resulted to be one of the best candidates. Unlike Virtual Machines (VMs), containers do not wrap an entire operating system. In other words, containers virtualize the operating system instead of hardware components. As a consequence, containers are more flexible and efficient. To achieve the goal of rapid prototyping, the development of web architecture appeals to microservice architecture [29]. This approach is a popular trend in software development scenario, as this architectural choice allows to obtain a rapid overview of all component needed [24], with a perfectly integrated approach [25].

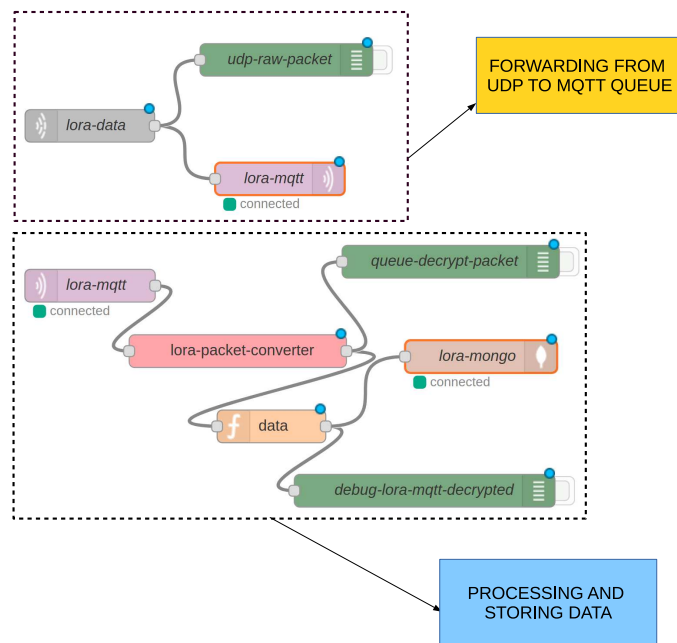
The envisioned architecture relies on three containers: the first one hosts Mosquitto, in order to support MQTT-SN. A second one holds the MongoDB database, used to store data coming from the gateway. The third and last one is dedicated to Node-RED. One of the most important challenge to face in an IoT scenario is messages exchange between smart objects belonging to a network. Communications might happen in both synchronous and asynchronous way. Given the differences between the two, a useful approach is represented by the Publish/Subscribe communication scheme, based on a message broker [26]. With this solution, devices in the network send and receive messages along specific channels, which goes under the name of *topic*. Based on message broker pattern [30], these messages can be either transformed with aggregation technique, rerouted in other channel, or sent in multicast mode.

Mosquitto [27] is an open source message broker that implements the Message Queue Telemetry Transport (MQTT) protocol. MQTT is a lightweight approach to provide message exchange using

a Publish/Subscribe model. This strategy is useful in IoT messaging scenario to provide intercommunication between low power sensors or mobile devices such as phones, embedded computers or microcontrollers.

MongoDB [31] was chosen as the approach used to the system design has been model-driven. All data made available by the open platform are processed in order to organize them into JavaScript Object Notation (JSON) format, that is highly recommended as it is easily readable from both human and machines. Given these premises, MongoDB represents the ideal solutions, as it belongs to NoSQL document oriented family [32], a kind of database that perfectly manages JSON format.

The web architecture also foresees Node-RED [28], an open source project initially developed by IBM, which uses a dataflow programming model for building IoT applications and services. Node-RED enables a prototyping environment which follows a flow-based programming paradigm [33] that defines applications as a network of *black box* processes. Node-RED can be easily used by functionally specifying each building block and tiding them up to reconstruct the data-flow. This approach helps building single pieces of an IoT application that can be wired together to effectively implement every single function of the application.



**Fig. 6:** Main building blocks of the developed processing real data.

In order to properly configure the joint usage of MQTT-SN protocol and UDP connections, the dataflow in Node-RED has been created as reported in Figure 6. A first box starts a UDP connection to a specific target port. This box represents the fact that the port is up and running and ready to get data. Afterward, data are communicated to an MQTT box implementing a queue. This block also wraps Mosquitto in the same container, a solution adopted as, for the best of our knowledge, there is not any implementation of MQTT-SN queue developed in NodeJS. As soon as data are populating the MQTT queue, the following box receives them and parse incoming messages in order to decode information within each packet. Afterward, a payload is built in order to store data in the box dedicated to wrap MongoDB container.

## 4 Experimental Results

In this Section, the experimental campaign is presented, in one with the obtained results. The main aim of evaluating rapid prototyping technologies in a LoRa platform development has been reached with the realization of the open platform described in Section 3.

The experimental campaign took place in a urban environment monitoring environmental parameters. In particular, here are reported both the KPI and the measured results related to different LoRa transmission parameters configurations.

The protocol stack allows the designer to choose the spreading factor, the bandwidth and the code rate, in order to better support the specific application. For the sake of compatibility and completeness, the presented solution provides three different parameters to be set: SF, BW and CR. These degrees of freedom can assume the set of values reported in Table 2.

SF	BW [Hz]	CR
7	125 / 250 / 500	4/5 4/6 4/7 4/8
8	125 / 250 / 500	4/5 4/6 4/7 4/8
9	125 / 250 / 500	4/5 4/6 4/7 4/8
10	125 / 250 / 500	4/5 4/6 4/7 4/8

**Table 2** LoRa communication parameters.

In particular, according to LoRa specifications, CR is set to maximum value (4/8) to guarantee minimum error effect on transmission of packets. In this way it is possible to tune SF and BW to investigate the behavior of the protocol. Tests have been carried out in the suburbs of an urban area, in order to grant that no physical obstructions could neither impede or affect propagation. In particular, tests have been done while on the move, walking along a specific path, common to every realization. Packets are considered lost in two cases: (i) every time they are not delivered to the gateway, or (ii) they are in the queue but they are incomplete.

One of the main aim of the tests is to verify prototype's lifetime, as a consequence of power consumption. It is well known that radio transceivers are draining the largest amount of current while performing transmission and/or reception tasks. To properly address this aspect, during the test phase, both the end node and the gateway radio interfaces are continuously communicating one with the other. Indeed, as a preliminary assessment, devices have been evaluated in terms of battery capacity and discharge time. The AT concrete values can be found in Table 3. For a better comprehension of the values, it has to be clarified that the tests configuration foresees:

- TEST1, with SF = 7 and BW = 125 kHz;
- TEST2, with SF = 9 and BW = 125 kHz;
- TEST3, with SF = 10 and BW = 125 kHz;
- TEST4, with SF = 7 and BW = 500 kHz;
- TEST5, with SF = 10 and BW = 500 kHz.

	Node	Gateway
<b>TEST1</b>	87 h	428 h
<b>TEST2</b>	40 h	428 h
<b>TEST3</b>	40 h	428 h
<b>TEST4</b>	208 h	358 h
<b>TEST5</b>	54 h	358 h

**Table 3** Battery Discharge Time.

The results highlight the usability of the prototype, in the worst case, for a total amount of 40 hours, which means that in a matter of two days, batteries should be replaced. The longest duration has been registered with the configuration of TEST4, with a total duration of 208 hours, which corresponds to more than 8 days longevity.

Accordingly, the five testing configurations have been evaluated in terms of PLR and communication range. Transmission effectiveness has been stressed with the end node getting far from the gateway in line of sight. With the TEST1 configuration, the expected result is the worst in terms of coverage range and delivery ratio. The latter resulted in a value of about 75%, which means that the PLR is 25%.

In terms of coverage capabilities, the platform was able to effectively complete communication task in a 1.5 km range, as it is reported in Figure 7. During this test, delivery ratio reached a maximum value equal to 70%. In this configuration, the current absorption of the LoRa modules result in different values between the end-node and the gateway. In particular, the measured values are:

- end-node: 55 mA and power consumption up to 181 mW;
- gateway: 11.2 mA and power consumption up to 37 mW.



**Fig. 7:** Maximum distance registered in TEST1 conditions, SF = 7 and BW = 125 kHz.

In the case of TEST2 configuration, the expected values are different, both in terms of coverage areas and PLR. In particular, expected values should be higher, for both of them. The measured values in this configuration, respectively, are 2.13 km and 35.7%. The respective delivery ratio is about 65.3%. In this configuration, the measured absorption values of both the LoRa end-device and the gateway demonstrated a growing trend. Measured values are reported herein:

- end-node: 120 mA, thus resulting in a power consumption up to 396 mW;
- gateway: 11.2 mA, thus resulting in a power consumption up to 37 mW.

As for the third configuration, also referred to as TEST3, the expected result is to reach the best performance in terms of coverage range and the worst values in terms of delivery ratio. In fact, as expected, the maximum coverage area was sensibly larger, for instance 3.30 km, as reported in Figure 7. As for the delivery ratio, the measured value has been 58%, which implies a PLR value of 42%. The power consumption related values measured in TEST3 configuration are :

- end-node: 120 mA, thus resulting in a power consumption up to 396 mW;

- gateway: 11.2 mA, thus resulting in a power consumption up to 37 mW.



**Fig. 8:** Maximum distance registered in TEST3 conditions, SF = 10 and BW = 125 kHz.

The rest of the experimental campaign was related to the BW value of 500 kHz. In TEST4 configuration, the SF has been set to 7 whereas in TEST5 configuration, this value has been setup to the value of 10.

The expected results are related with an improvement of both the coverage area and the delivery ratio. The values related with TEST4 were 2.02 km as a maximum distance, and delivery ratio equal to 68% (which obviously implies a PLR equal to 32%). The measured absorption values of the LoRa system in this configurations are:

- end-node: 23 mA, with a resulting power consumption up to 75.9 mW;
- gateway: 13.4 mA, with a resulting power consumption up to 44.22 mW.

In the last testing configuration, namely TEST5, the highest values for both SF and BW have been set, respectively 10 and 500 kHz. The expected result is an improvement of the coverage area and a reduction of the delivery ratio. The test outcomes demonstrated coverage area similar to TEST3 configuration and a delivery ratio equal to 82%. In this case, the measured power consumption values have been:

- node: 88mA and power consumption up to 290 mW;
- gateway: 13.4mA and power consumption up to 44.22 mW.

A summary of the main outcome in terms of registered power consumption is reported in Table 4.

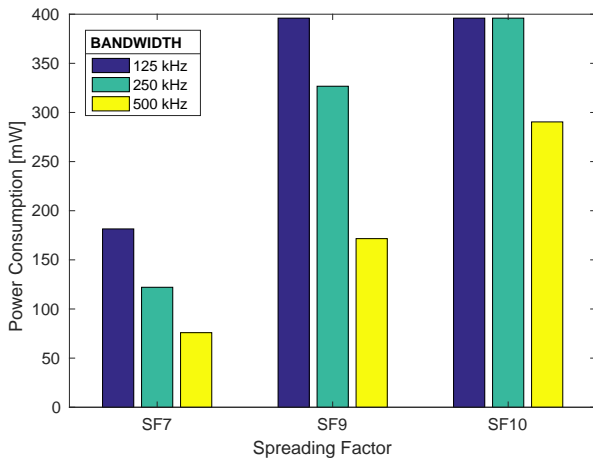
What clearly emerged from the experimental evaluation is that the envisioned prototyping architecture match to LoRa and physical constraints and low-power requirements.

	Node	Gateway
TEST1	181 mW	37 mW
TEST2	396 mW	37 mW
TEST3	396 mW	37 mW
TEST4	75.9 mW	44.22 mW
TEST5	290 mW	44.22 mW

**Table 4** Registered Power consumption values.

Some specific considerations can be done as regards the SF: when it increases, a direct proportional bound to the distance up to physical limit can be identified. Instead, when keeping the SF fixed, it clearly results that the maximum coverage distance value can be reached with the higher value of BW. Measurements are reported in Figure 10.

Another important inspected theme is power consumption. The analysis has been carried out on both the end-node and the gateway. Values are shown in Table 4. The power consumption of the gateway always results to be lower than those of the end-node. This is clearly due to the number of component embedded in the end-node. Therefore, the power consumption values are clearly related with GPS and sensors supply. Values are summarized in Figure 9.

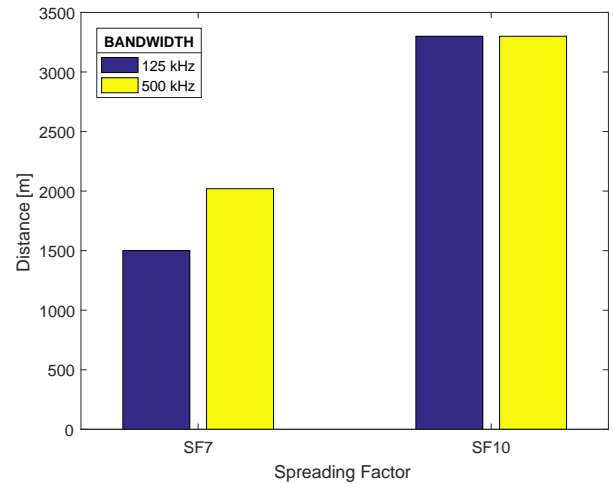


**Fig. 9:** Power consumption values over the three SF configurations.

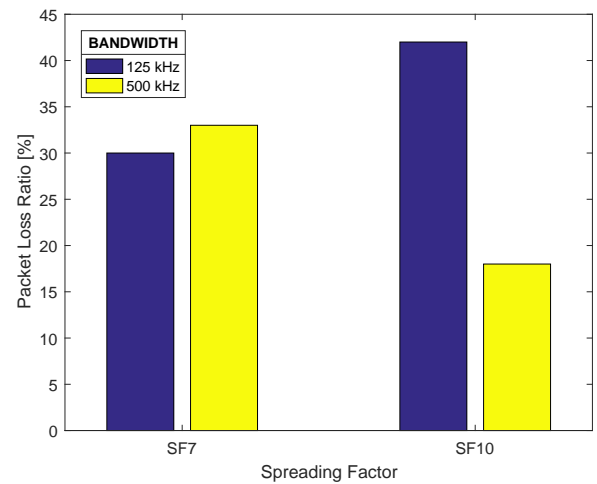
Important considerations can be done as for the relationship between power consumption and BW. Indeed, when the testing conditions foresaw the same values for SF but different settings for the BW, the power consumption has resulted to be lower with higher BW values. This is clearly shown in Figure 9, and it happens for each SFs values. Given this law, it is possible to derive trade-off considerations: larger coverage area and distances can be reached with low power consumption values. This conclusion is demonstrated by the values reported in Figure 10. The maximum distance that the presented platform has reached is 3.3 km. This happened with a SF equal to 10 and two different values of bandwidth, for instance BW = 125 kHz and BW = 500 kHz. The difference between these two configurations is in the measured PLR and power consumption values, which leads to trade-off considerations. The best value has been obtained in the case of BW = 500 kHz, where the PLR value has been 15%, as showed in Figure 11.

## 5 Conclusions

The present contribution represents a clear example of the noticeable possibilities offered by rapid prototyping. Low-cost embedded boards and open source software solutions can effectively ease technology accessibility. To demonstrate their suitability in realizing a



**Fig. 10:** Maximum distances reached over the experimental campaign.



**Fig. 11:** Packet loss ratio over the experimental campaign.

LoRa-compliant, fully functional and complete solution, a prototype has been conceived and realized using an ESP8266 MCU and a RFM95 transceiver.

The platform presented in this work concretely demonstrates such possibilities and leverages LoRa protocol stack to open source development. To validate the guiding intuition, an experimental campaign has been carried out in urban environment. Between the main outcome, it is remarkable that battery lifetime guarantees prototype usability and midterm duration, especially for the gateway. The obtained results demonstrated that in a range of 2 km it is possible to reach significant delivery ratio. In terms of mere distance, the best measured value was higher than 3 km. One of the most relevant conclusion that can be derived from the results is that spreading the signal bandwidth implies an increase in the communication range, even though power consumption increases thus limiting battery lifetime and autonomy of the platform.

Despite these encouraging results, a number of enhancements can be pursued. First of all, the prototype needs to be tested in a larger scale rural environment. Then, it would be of great interest to test communication effectiveness over a network composed of more than only one end node. In the near future, in fact, a deeper and extensive experimental campaign will foresee (i) a larger area of interest, (ii) a higher number of nodes, (iii) security functionalities and (iv) CR variation will be taken into consideration. As a further enhancement, the prototype will also be given energy harvesting capabilities.



## 6 References

- 1 U. Raza, P. Kulkarni, and M. Sooriyabandara, "Low power wide area networks: An overview," *IEEE Communications Surveys Tutorials*, vol. 19, no. 2, Secondquarter 2017.
- 2 FLASHNET, InteliLIGHT Project, <https://intelilight.eu/>, URL Consulted 03/04/2018
- 3 C. Pham, F. Ferrero, M. Diop, L. Lizzi, O. Dieng, and O. Thiaré, "Low-cost antenna technology for lpwan iot in rural applications," in *2017 7th IEEE International Workshop on Advances in Sensors and Interfaces (IWASI)*, Jun. 2017.
- 4 A. Springer, W. Gugler, M. Huemer, L. Reindl, C. C. W. Ruppel, and R. Weigel, "Spread spectrum communications using chirp signals," in *IEEE/AFCEA EURO-COMM 2000. Information Systems for Enhanced Public Safety and Security (Cat. No.00EX405)*, 2000.
- 5 J. Petajajarvi, K. Mikhaylov, A. Roivainen, T. Hanninen, and M. Pettissalo, "On the coverage of lpwans: range evaluation and channel attenuation model for lora technology," in *2015 14th International Conference on ITS Telecommunications (ITST)*, Dec. 2015.
- 6 LoRa Alliance, <https://www.lora-alliance.org/>, URL Consulted 03/04/2018
- 7 P. Zhang and H. Liu, "An ultra-wide band system with chirp spread spectrum transmission technique," in *2006 6th International Conference on ITS Telecommunications*, Jun. 2006.
- 8 O. Georgiou and U. Raza, "Low power wide area network analysis: Can lora scale?" *IEEE Wireless Communications Letters*, vol. 6, no. 2, Apr. 2017.
- 9 A. Augustin, J. Yi, T. Clausen, and W. M. Townsley, "A study of lora: Long range & low power networks for the internet of things," *Sensors*, vol. 16, no. 9, 2016.
- 10 D. H. Kim, J. Y. Lim, and J. D. Kim, "Low-power, long-range, high-data transmission using wi-fi and lora," in *2016 6th International Conference on IT Convergence and Security (ICITCS)*, Sep. 2016.
- 11 "Elemental characterization of pm10, pm2.5 and pm1 in the town of genoa (italy)," *Chemosphere*, vol. 62, no. 2, 2006.
- 12 K. Grgic and I. Speh and I. Hedi, "A web-based IoT solution for monitoring data using mqtt protocol," in *2016 International Conference on Smart Systems and Technologies (SST)*, Oct. 2016.
- 13 U. Hunkeler, H. L. Truong, and A. Stanford-Clark, "Mqtt-s - a publish/subscribe protocol for wireless sensor networks," in *Communication Systems Software and Middleware and Workshops, 2008. COMSWARE 2008. 3rd International Conference on*, Jan. 2008.
- 14 L. D. Xu, W. He, and S. Li, "Internet of things in industries: A survey," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2233–2243, Nov. 2014.
- 15 C. Shao, D. Hui, R. Pazhyannur, F. Bari, R. Zhang, and S. Matsushima, "IEEE 802.11 Medium Access Control (MAC) Profile for Control and Provisioning of Wireless Access Points (CAPWAP)," RFC 7494, Apr. 2015.
- 16 Hoperf Electronic, RFM95/96/97/98(W) - Low Power Long Range Transceiver Module V1.0, [http://www.hoperf.com/upload/rf/RFM95\\_96\\_97\\_98W.pdf/](http://www.hoperf.com/upload/rf/RFM95_96_97_98W.pdf/), URL Consulted 03/04/2018
- 17 Espressif Systems IOT Team, ESP8266EX Datasheet Version 4.3, [https://cdn-shop.adafruit.com/product-files/2471/0A-ESP8266\\_\\_Datasheet\\_\\_EN\\_v4.3.pdf](https://cdn-shop.adafruit.com/product-files/2471/0A-ESP8266__Datasheet__EN_v4.3.pdf), URL Consulted 03/04/2018
- 18 Bosch Sensortech, BME280 Final data sheet, [https://ae-bst.resource.bosch.com/media/\\_tech/media/datasheets/BST-BME280\\_DS001-12.pdf](https://ae-bst.resource.bosch.com/media/_tech/media/datasheets/BST-BME280_DS001-12.pdf), URL Consulted 03/04/2018
- 19 PLANTOWER, PMS5003 series data manual, [http://www.aqmd.gov/docs/default-source/eq-spec/resources-page/plantower-pms5003-manual\\_v2-3.pdf](http://www.aqmd.gov/docs/default-source/eq-spec/resources-page/plantower-pms5003-manual_v2-3.pdf), URL Consulted 03/04/2018
- 20 U-Blox, Neo-6 datasheet, [https://www.u-blox.com/sites/default/files/products/documents/NEO-6\\_DataSheet\\_\(GPS.G6-HW-09005\).pdf](https://www.u-blox.com/sites/default/files/products/documents/NEO-6_DataSheet_(GPS.G6-HW-09005).pdf), URL Consulted 03/04/2018
- 21 V. Carvalho and F. L. Kastensmidt, "Enhancing i2c robustness to soft errors," in *2017 IEEE 8th Latin American Symposium on Circuits Systems (LASCAS)*, 2017.
- 22 M. Poorani and R. Kurunjimalar, "Design implementation of uart and spi in single fpga," in *2016 10th International Conference on Intelligent Systems and Control (ISCO)*, 2016.
- 23 C. Anderson, "Docker [software engineering]," *IEEE Software*, vol. 32, no. 3, pp. 102–c3, May 2015.
- 24 J. Thönes, "Microservices," *IEEE Software*, vol. 32, no. 1, pp. 116–116, Jan. 2015.
- 25 A. Sill, "The design and architecture of microservices," *IEEE Cloud Computing*, vol. 3, no. 5, pp. 76–80, Sep. 2016.
- 26 M. Shah and D. B. Kulkarni, "Enabling qos support for multi-core message broker in publish/subscribe system," in *2014 IEEE International Advance Computing Conference (IACC)*, Feb. 2014, pp. 774–778.
- 27 Mosquitto, <https://mosquitto.org/>, URL Consulted 03/04/2018
- 28 Node-RED, <https://nodered.org/>, URL Consulted 03/04/2018
- 29 Martin Fowler, Microservices, <https://martinfowler.com/articles/microservices.html>, URL Consulted 03/04/2018
- 30 Enterprise Integration Patterns, <http://www.enterpriseintegrationpatterns.com/patterns/messaging/MessageBroker.html>, URL Consulted 03/04/2018
- 31 Y. Gu, S. Shen, J. Wang, and J. U. Kim, "Application of nosql database mongodb," in *2015 IEEE International Conference on Consumer Electronics - Taiwan*, Jun. 2015, pp. 158–159.
- 32 S. Chickerur, A. Goudar, and A. Kinnerkar, "Comparison of relational database with document-oriented database (mongodb) for big data applications," in *2015 8th International Conference on Advanced Software Engineering Its Applications (ASEA)*, Nov. 2015, pp. 41–47.
- 33 J. P. Morrison, "Flow-based programming," in *Proc. 1st International Workshop on Software Engineering for Parallel and Distributed Systems*, 1994, pp. 25–29.