

River Water Quality Monitoring Using LoRa-Based IoT

Luís Miguel Pires ^{1,2,*}  and José Gomes ^{1,*}

¹ Department of Electronical Engineering, Telecommunications and Computers (DEETC), Instituto Superior de Engenharia de Lisboa (ISEL), 1959-007 Lisbon, Portugal

² Technologies and Engineering School (EET), Instituto Politécnico da Lusofonia (IPLuso), 1700-098 Lisbon, Portugal

* Correspondence: luis.pires@ipluso.pt (L.M.P.); a28386@alunos.isel.pt (J.G.)

Abstract: Water pollution presents one of the biggest challenges in the world today, as the degradation of water quality of rivers in many instances is increasing so fast and poses a big danger to all forms of life, eventually causing many aquatic species and other species that depend on them to be endangered. Hence, with the development of Internet of Things (IoT) and Wireless Sensor Networks (WSNs), there arises a need to monitor river waters for a timely response in protecting the rivers, which is the aim of this paper. With respect to this project, we searched a little bit for some existing IoT technologies and other related work. In this paper, we propose a practical low-cost solution based on Long Range (LoRa) technology to obtain real-time observations of, with certain sensors, such water parameters as temperature, pH, conductivity and turbidity. Data gathered at a sensor node are transmitted via LoRa modulation to a gateway for processing and local storage on a Message Queuing Telemetry Transport (MQTT) server, visualization on a Node-RED interface, or transmission to the cloud. The prototype system created is employed in the actual field and demonstrates that the water quality monitoring in the river can be carried out effectively within a small scale of the area of roughly 20 km² depending on the location of the study site.

Keywords: river; water; sensors; long range; Node-RED; monitoring; embedded systems



Citation: Pires, L.M.; Gomes, J. River Water Quality Monitoring Using LoRa-Based IoT. *Designs* **2024**, *8*, 127. <https://doi.org/10.3390/designs8060127>

Received: 24 October 2024

Revised: 20 November 2024

Accepted: 25 November 2024

Published: 28 November 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Water quality control has been traditionally carried out by collecting some samples and taking them to a laboratory where tests would be performed and a conclusion drawn based on the collected samples [1]. This takes a lot of time and resources. While this traditional approach is appropriate for carrying out detailed and broad surveys over a long period of time, it is ideally not practical in day-to-day operations aimed at protecting rivers and even being proactive [1].

This experience aims at creating and prototyping an efficient and effective river water monitoring system that employs modern technology. This prototype takes the benefits of LoRa technology such as large coverage area, low power requirement and not relying on a third party. Lastly, the prototype is designed to be rugged, lightweight, easy to assemble, and inexpensive.

With this developed prototype, the following water quality parameters will be monitored:

- pH—Indicates the acidity or alkalinity of a solution [2]. It does not represent the measure of the quantity of acids or bases, but the relationship between them. The pH varies between 0 and 14. High or low pH values can be indicative of pollution, with normal values being considered between 6.5 and 8.5.
- Conductivity—This is an important parameter due to the ease of detecting contamination levels when measuring water conductivity [3]. High conductivity means that the water contains a high level of contaminants, and the opposite means that drinking water is practically incapable of conducting electrical current. The unit of measurement for electrical conductivity is mS/cm and the normal value is up to 2 mS/cm.

- Turbidity—Is a measure of water transparency [4]. Cloudy water is often caused by suspended particles such as silt, (seaweed, etc.), organic materials, etc. These particles soak up (like a towel) and scatter the light rays instead of letting them pass through the water, which is detrimental to water-based plants and animals. A high turbidity value indicates cloudier water, and a low value means clear water. Turbidity measurements are represented in Nephelometric Turbidity Units (NTU) and normal values are up to 5 NTU.
- Temperature—There is no ideal temperature for river water, but many water-based organisms are sensitive to high temperatures and the oxygen solubility is lower in warmer waters, thus limiting the supply of oxygen [5]. The unit for temperature is degrees Celsius (°C).

This article begins by presenting the current state of IoT usage and other related works in Section 2. Section 3 resumes an investigation that was carried out about some communications technologies used in IoT and explains LoRa controlling/adjusting. In Section 4, the earlier developed model is explained in some detail, and in Section 5, results are presented. In Section 6, we write conclusions and some statements about future work.

2. State of the Art

IoT refers to a big network of interconnected physical devices containing sensors, microcontrollers and other circuitry meant for data movement (from one place to another) to and from other devices and the Internet. These devices consist of simple home units and more sophisticated complex industrial structures [6].

In recent years, IoT has become one of the most important technologies of the century. Now that we can connect everyday devices (home appliances, cars, sensors, watches, phones, etc.) to the internet, communication between people, processes and “things” is possible. With the use of low-cost processing systems (computers, embedded systems), the “cloud”, data analysis tools and mobile technologies, physical devices can share data with minimal human intervention. In this ultra interconnected world, these systems can record, monitor and adjust each interaction between them. The physical world meets the digital world, and they cooperate [5].

The applications of IoT are huge and varied, and its effect is noted across multiple areas, including manufacturing, transportation, healthcare, and farming.

To provide great support to IoT, WSN plays a very important role, due to the advancement of sensor and microcontroller systems, as well as to ensure high-quality wireless communication rules of conduct [7].

In Portugal, in the residential segment, 38% of personal equipment and 22% of household appliances are already connected to the Internet. In the business segment, 23% of equipment is connected to the Internet, with the main uses being facility security, energy consumption management, logistics management and production processes [6].

As the number of Internet-connected devices continues to grow, IoT is likely to play a more and more important role in shaping our world, changing the way we live, work and interact with each other and machines.

In the article “Water Quality Monitoring System Based on the Internet of Things” [7], the authors develop a very simple system composed of sensors (temperature, turbidity, pH and total dissolved solids), a microcontroller with Wi-Fi and a Global System for Mobile Communications (GSM) module. The data are sent to a “cloud” via WIFI connection, or in the case of need (alarm), an SMS is sent to the user. Note the presence in this system of an Analog-to-Digital Converter (ADC) to interface between the sensors and the microcontroller (NodeMCU ESP8266), as it does not have analog inputs. Due to the use of a WIFI network, this system is very limited in terms of monitoring coverage.

In the system described in the article “IoT Based Water Quality Monitoring System Using Solar Powered and LoRaWAN” [8], a water quality monitoring system is also developed, but the data acquired are sent via LoRa to the LoRaWAN network, and there must be a LoRaWAN Gateway close to the site under study. In this system, the microcontroller used

is the Arduino Nano, but as it does not have analog inputs for the sensors (temperature, turbidity, conductivity and pH), the authors used a module (CAT S767S) that communicates with the Arduino via Inter-Integrated Circuit (I2C). This module, in addition to having analog inputs, also includes the LoRa module. As the system is mobile (floating), a GPS module was also incorporated to send the location of the measurements.

3. IoT Technologies

In IoT, many devices are wireless, forming part of WSNs, and to support these communications, the Low Power Wide Area Network (LPWAN) is used. LPWANs are low-energy-consumption technologies that provide long-range communications. For these reasons, they are the focus of this analysis, with special emphasis on LoRa technology.

3.1. LPWAN

Narrow Band-IoT (NB-IoT) is a technology developed by the 3rd Generation Partnership Project (3GPP) and is implemented over existing Long-Term Evolution (LTE) cellular networks [9]. It occupies a 200 kHz block of an LTE carrier and can be implemented on the mobile network in three different modes; see Figure 1.

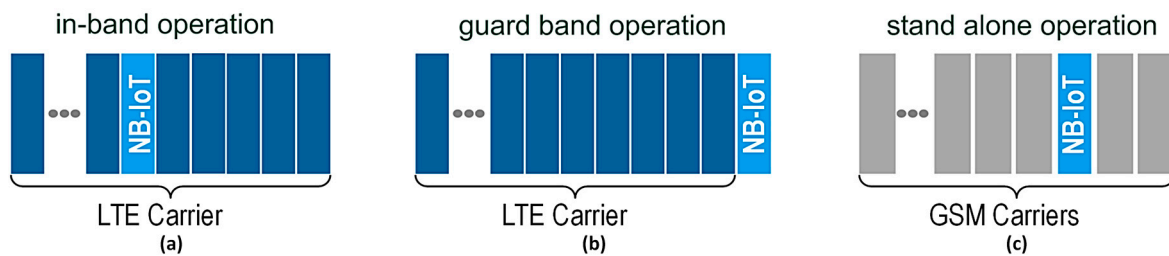


Figure 1. LTE carrier operation modes for NB-IoT: (a) in-band; (b) guard band; (c) stand-alone (adapted from [10]).

In any of the operating modes, there is a 10 kHz guard band between the blocks, leaving 180 kHz of usable bandwidth for data.

The great advantage of this technology, as it is integrated into mobile networks, is its great coverage in both external and internal environments. The disadvantage is that there are costs associated with mobile operators.

Sigfox technology, also called “Sigfox 0G Technology”, was the first LPWAN to be developed in 2010 by the French company with the same name, but it was acquired in 2022 by UnaBiz [11]. Communications use Binary Phase Shift Keying (BPSK) modulation, with a bandwidth of just 100 Hz, allowing low energy consumption and high receiver sensitivity at the expense of a maximum throughput of just 100 b/s.

The devices (sensors) communicate with the base stations, using the free Industrial Scientific and Medical (ISM) frequency band, 868 MHz in Europe. Devices send messages over the air to the base stations, then these messages go through a backhaul (Internet connection) to Sigfox Cloud, where they are stored and can be accessed by the customer’s end platform, as shown in the Sigfox network architecture in Figure 2. It is also possible to send messages in the opposite direction.

The Sigfox network has good coverage, especially in Europe, but with operator costs as the main disadvantage.

LoRa is a protocol developed by the French company Cycleo (Grenoble, French), which was acquired by Semtech (Semtech Corporation, Camarillo, CA, USA) in 2012. Like Sigfox, LoRa uses the ISM band for communications, but using a proprietary modulation technique derived from existing Chirp Spread Spectrum (CSS) technology [12]. LoRaWAN is an LPWAN standard managed by LoRa Alliance [13], based on LoRa, designed to connect wireless devices to the Internet, and meets the main IoT requirements, such as bidirectional communications, security, mobility and location services. There are some LoRaWAN operators, and they have good coverage.

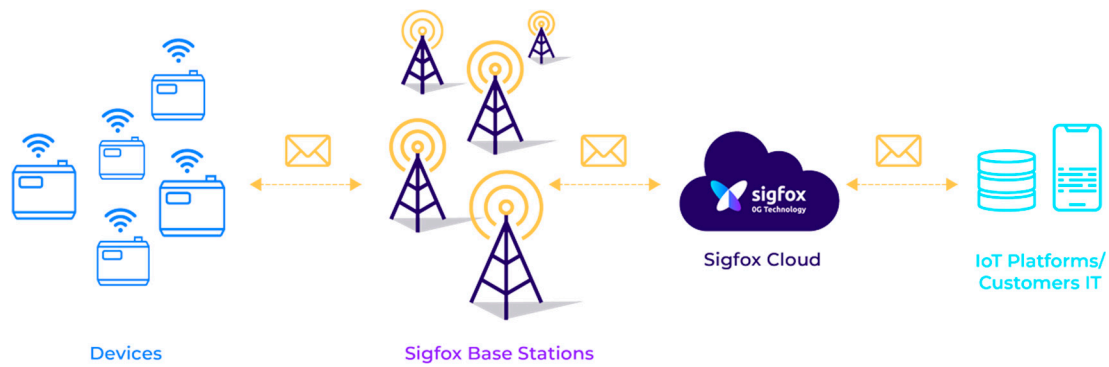


Figure 2. Sigfox network architecture: A device broadcasts a message using its radio antenna; multiple base stations in the area will receive the message, and the base stations then send the message to the Sigfox Cloud, which eventually sends the message to the customer’s end platform. (adapted from [11]).

The LoRaWAN network architecture (Figure 3) uses a star-of-stars topology, where gateways relay messages between end-devices and a central network server [13]. Connected via standard Internet connections (backhaul), gateways convert RF packets to IP packets and vice versa [13].

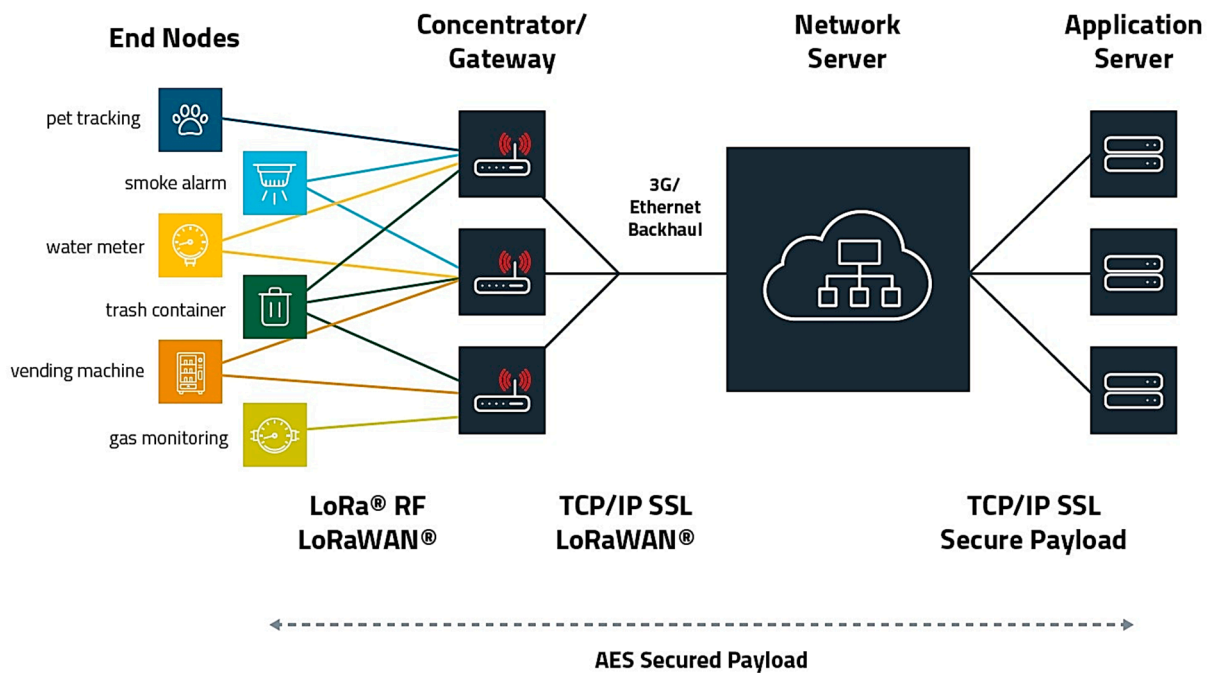


Figure 3. LoRaWAN network architecture: Gateway receives messages from any end node, forwards these data messages to the network server, and they are finally accessed by the application server (adapted from [14]).

The specification defines three classes of operation for the devices [13]. Class A supports (going in both directions) communication, requires the most energy (producing a lot with very little waste) and is required in all devices. The downlink is available only after a device TX (uplink). Class B tools (going in both directions) communicate with scheduled downlink slots. Class C has the same uplink schedule as Class A but always keeps the receiving window open, providing less latency but more energy use.

A summary of the characteristics of the LPWAN networks covered here is presented in Table 1.

Table 1. Overview of LPWAN network technical aspects (adapted from [15]).

	Sigfox	LoRaWAN	NB-IoT
Modulation	BPSK	CSS	QPSK
Freq. Band	ISM	ISM	LTE
BW	100 Hz	125/250 kHz	180 kHz
Bidirectional	Limited Half-duplex	Yes Half-duplex	Yes Half-duplex
Messages/day (max)	140 (UL) 4 (DL)	Limited (duty-cycle)	Unlimited
Message size (max)	12 Byte (UL) 8 Byte (DL)	243 Byte	1600 Byte
Throughput (max)	100 bit/s	50 kbit/s	160 kbit/s (UL) 120 kbit/s (DL)
Range	10 km (urban) 40 km (rural)	5 km (urban) 20 km (rural)	1 km (urban) 10 km (rural)
Interference immunity	High	High	Low
Encryption	No	Yes (AES 128 bits)	Yes (LTE)
Private networks	No	Yes	No
Standard	Sigfox	LoRa Alliance	3GPP

These LPWAN networks present a compromise between some parameters [15] in terms of some IoT factors, which are shown in Figure 4.

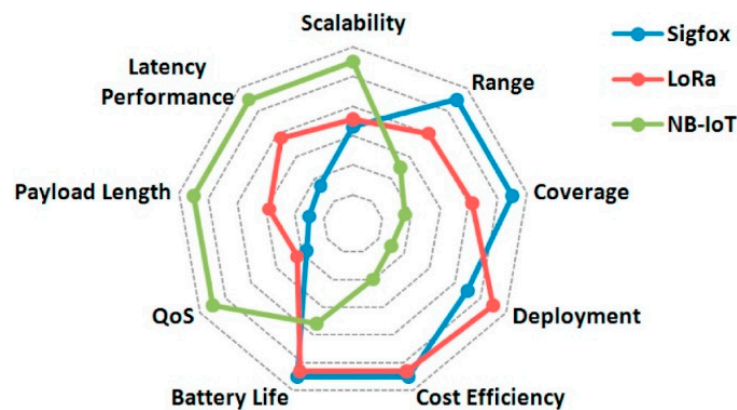


Figure 4. LPWAN advantage compromise in terms of some IoT factors (adapted from [15]).

In summary, each technology has a place in IoT networks. Sigfox and LoRa present the devices with the lowest cost, best range, best coverage and lowest energy consumption. LoRa will be best suited for the development of local and private networks. On the other hand, NB-IoT networks have lower latencies and better quality of service, but at higher costs.

3.2. LoRa Modulation

LoRa uses a proprietary modulation technique derived from existing Chirp Spread Spectrum (CSS) technology [12], which spreads a narrowband signal over a wider bandwidth on the transmitting channel, allowing low power requirements, noise immunity, robustness against channel degradation (multipath, jamming, fading and Doppler effect).

In CSS, spectrum spread is obtained by generating a chirp signal (Compressed High-Intensity Radar Pulse) that continuously varies in frequency, between the minimum and

maximum values of the transmission channel bandwidth (BW). These deviations can increase (up-chirp) or decrease (down-chirp).

Each chirp will transmit a symbol with a certain number of bits, which represents the Spreading Factor (SF). Each chirp is composed of 2^{SF} chips, which is the number of values that it is possible to encode with the number of bits. In the examples in Figure 5, two up-chirps are represented, with decimal information symbols of 32 (Figure 5a) and 64 (Figure 5b), with an SF = 7, which each corresponds to 128 chips.

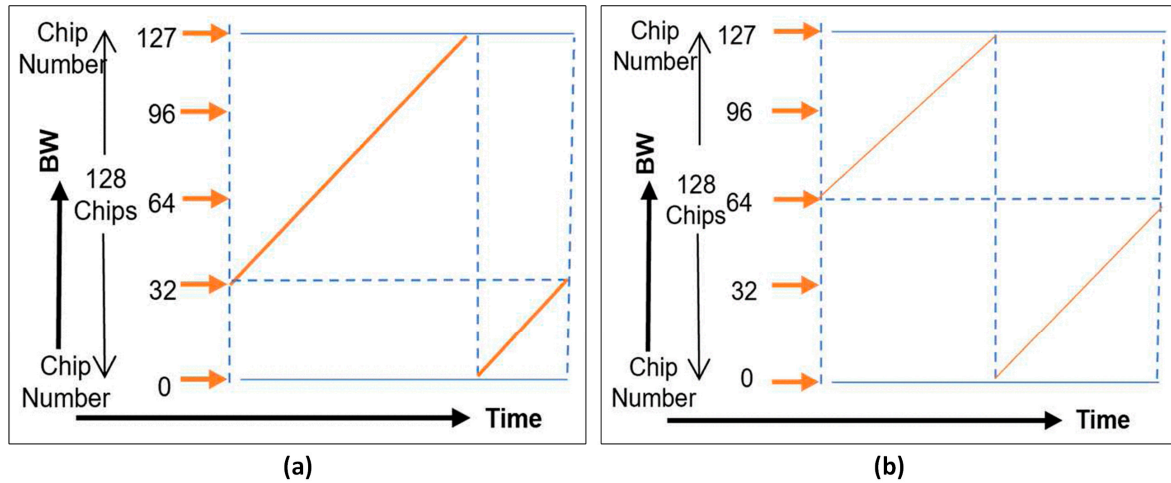


Figure 5. Up-chirp signals, with SF = 7: (a) decimal information symbol of 32; (b) decimal information symbol of 64 (adapted from [16]).

The SF can go from 7 to 12. A low SF means that more chirps are being sent per second. This way, more information can be coded per second; however, the range of the transmission drops. On the other hand, a high SF provides a better range, but the chirps are more spread out in time, lowering the bit rate.

Bandwidths of 125 kHz, 250 kHz and 500 kHz are normally used, the latter being applied only in the downlink. With higher bandwidths, the binary rhythm increases, reducing the symbol period, but on the other hand, this sacrifices receive sensitivity due to the introduction of more noise.

The symbol period, T_{sym} , is defined by (1)

$$T_{sym} = \frac{2^{SF}}{BW} [s], \tag{1}$$

where:

- SF is the spreading factor (7..12);
- BW is the modulation bandwidth (Hz).

The symbol rate, R_{sym} , is defined by (2)

$$R_{sym} = \frac{1}{T_{sym}} = \frac{BW}{2^{SF}} [sym/s], \tag{2}$$

where:

- SF is the spreading factor [7..12];
- BW is the modulation bandwidth [Hz];
- T_{sym} is the symbol period [s].

The chip rate, R_c , is defined by (3)

$$R_c = R_{sym} * 2^{SF} = \frac{BW}{2^{SF}} * 2^{SF} = BW [chips/s], \tag{3}$$

where:

- SF is the spreading factor [7...12];
- BW is the modulation bandwidth [Hz];
- R_{sym} , symbol rate [sym/s].

This leads to the following statement. One chip is sent per second per Hz of bandwidth [17].

And because LoRa modulation also includes an error correction scheme, Forward Error Correction (FEC) is implemented. This improves transmission robustness by including redundant bits. The final bitrate calculation, R_b , is calculated according to (4) below:

$$R_b = SF * \frac{BW}{2^{SF}} * \frac{4}{4 + CR} \text{ [bit/s]}, \tag{4}$$

where:

- SF is the spreading factor [7...12];
- BW is the modulation bandwidth [Hz];
- CR is the code rate (number of redundant bits) [1..4].

There is an increase in bitrate for lower SF and higher BW; see Figure 6. And of course, if we increase the number of redundant bits (CR), the bitrate will also increase.

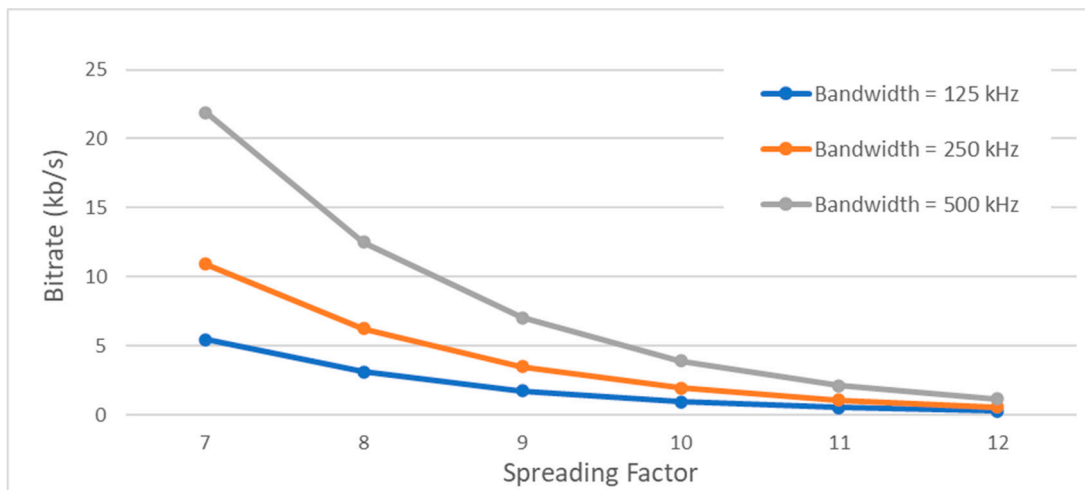


Figure 6. Bitrate and spreading factor relationship (CR = 1).

For the transmission of LoRa messages, Semtech defined a structure for the packet [17]. The packet comprises several elements, as shown in Figure 7.

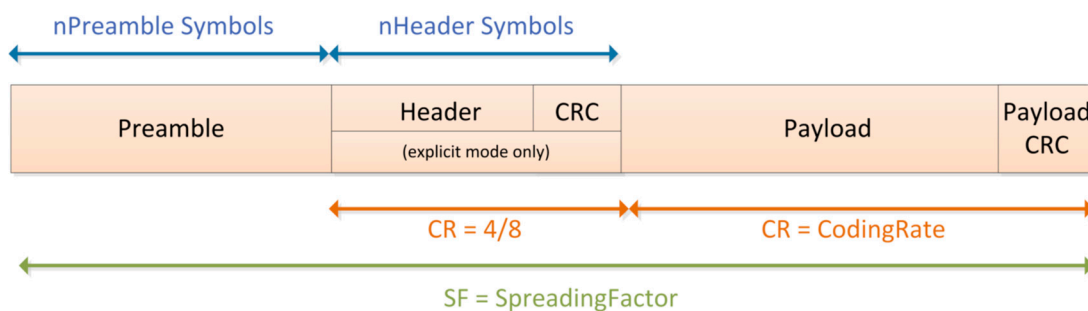


Figure 7. LoRa packet format (adapted from [18]).

The preamble is used to synchronize the receiver with the received data stream and begins with a sequence of $n_{preamble}$ up-chirps, followed by two up-chirps that contain a

synchronization byte (sync word) and ends with two and a fourth down-chirps. The value of the sync word is to differentiate LoRa networks. In the case of LoRaWAN, this value is 0x34 [19]. The $n_{preamble}$ size can vary between 6 and 65,535, but by default, on Semtech transceivers it is 8.

The receiver begins the detection process, and this process occasionally restarts. For this reason, the length should be configured to be identical to the transmitter length. Where the length is not known, or can change/differ, the maximum length should be programmed on the receiver side.

The preamble period, $T_{preamble}$, can be calculated with (5).

$$T_{preamble} = (n_{preamble} + 4.25) * T_{sym} [s], \tag{5}$$

where:

- $n_{preamble}$ is the number of programmed preamble symbols;
- T_{sym} is the symbol period [s].

Depending upon the chosen mode of operation, two types of header mode are available:

- Explicit mode—This is the default mode of operation [20]. Here, the header provides information on the payload, namely payload length in bytes, forward error correction code rate and presence of an optional 16-bit CRC for the payload [20]. The header is transmitted with maximum error correction code (4/8) [20]. It also has its own CRC to allow the receiver to throw out invalid headers.
- Implicit mode—In certain pictures/situations, where the payload, coding rate and CRC presence are fixed or known in advance, it may provide an advantage to reduce transmission time by calling for/using an understood header mode [20]. In this mode, the header is removed from the packet [20]. In this case, the payload length, error coding rate and presence of the payload CRC must be manually configured on both sides of the radio link [20].

The payload contains the data to transmit, which may or may not be followed by the CRC (two bytes) and can have a size of up to 255 bytes (including the CRC). Equation (6) calculates the number of symbols, $pSymN$, in the payload.

$$pSymN = 8 * \max \left(\text{ceil} \left(\frac{8PL - 4SF + 28 + 16 - 20H}{4 * (SF - 2DE)} \right) * (CR + 4), 0 \right), \tag{6}$$

where:

- PL is the number of payload bytes;
- SF is the spreading factor [7..12];
- $H = 0$ when the header is enabled, and $H = 1$ when no header is present;
- $DE = 1$ when low data rate optimization is enabled, and $DE = 0$ when disabled;
- CR is the code rate (number of redundant bits) [1..4].

And the total duration of the LoRa packet, T_{packet} , is given by Equation (7):

$$T_{packet} = T_{preamble} + (pSymN * T_{sym}) [s], \tag{7}$$

where:

- $T_{preamble}$ is the preamble period [s];
- $pSymN$ is the number of symbols;
- T_{sym} is the symbol period [s].

As seen in Figure 8, the packet duration is directly proportional to the number of payload bytes and to the SF.

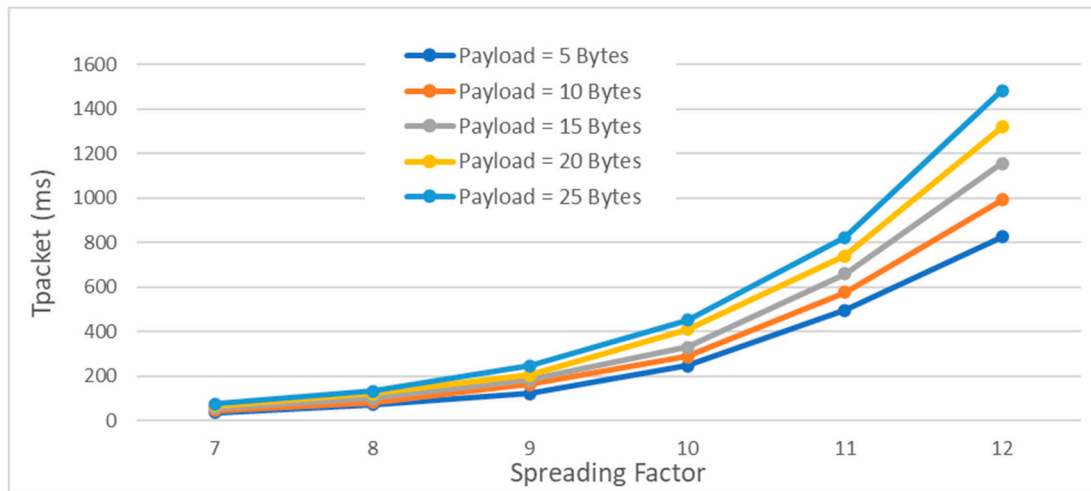


Figure 8. Packet duration and spreading factor relationship (CR = 1, BW = 125 kHz).

And the packet duration decreases for higher BW; see Figure 9.

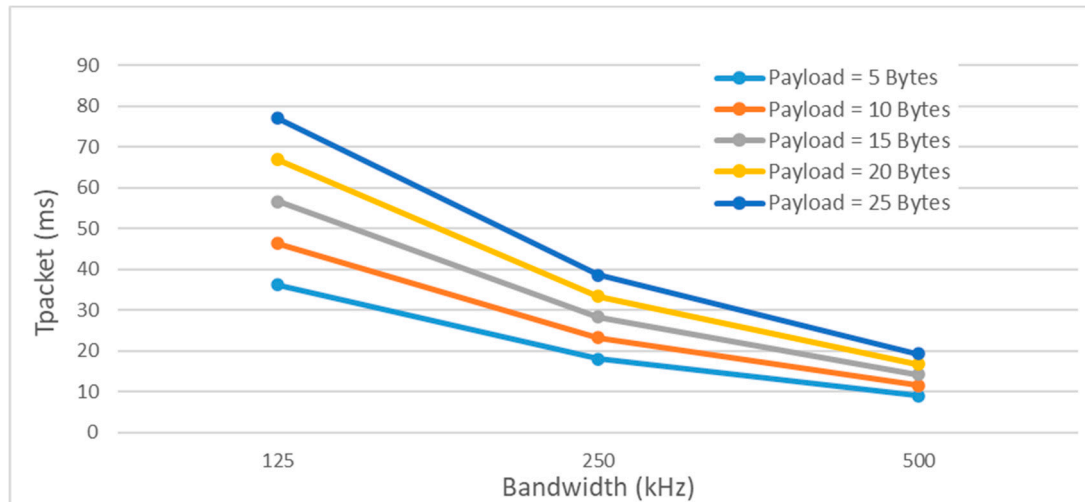


Figure 9. Packet duration and bandwidth relationship (CR = 1, SF = 7).

T_{packet} also increases slightly to the highest CR values, as more additional redundancy bits are included.

The packet duration, T_{packet} , also means the Time on Air (ToA) and is the time a LoRa device is transmitting. To comply with legal requirements as to the time of occupancy of the channel (duty-cycle), there is a need to use the duration of ToA to calculate the time between successive transmissions, T_{TX_off} , by Equation (8).

$$T_{TX_off} = ToA * \left(\frac{1}{duty_cycle} - 1 \right) [s] \tag{8}$$

where:

- ToA is the Time on Air (equal to T_{packet}) [s];
- $duty_cycle$ is the channel occupancy time ratio.

4. System Design

The developed prototype is divided into two isolated subsystems, IoT Node and Gateway, as shown in the block diagram in Figure 10.

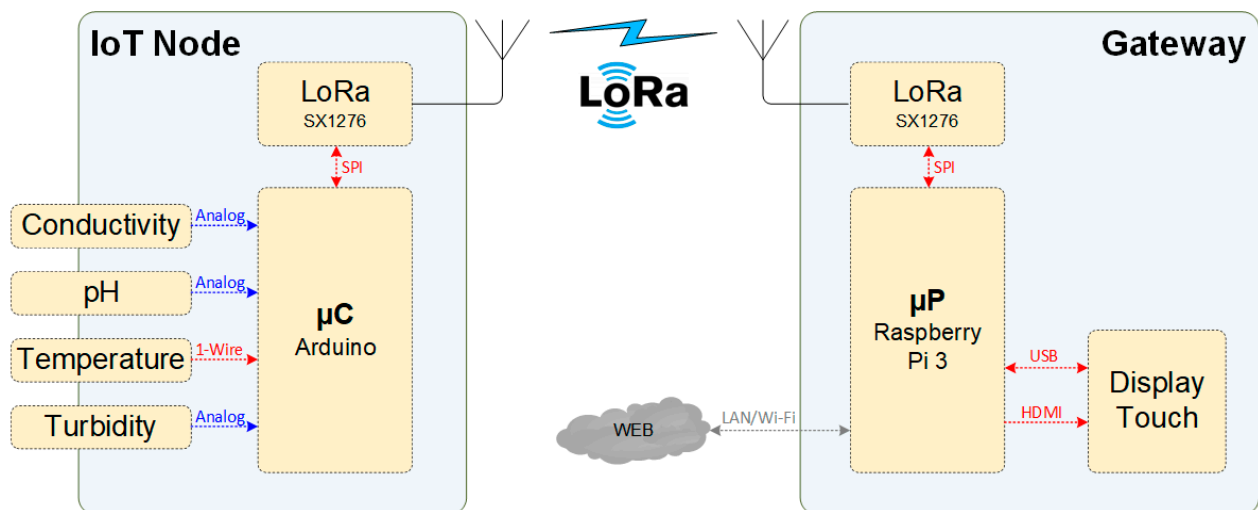


Figure 10. System block diagram of the developed prototype, with the two supporting, IoT Node and Gateway.

In the IoT Node, which is installed next to the river under study, the water quality parameters (temperature, pH, conductivity and turbidity) are measured using independent and specific sensors. The data acquired are processed on a microcontroller board and then sent using LoRa modulation with a radio module (LoRa SX1276 transceiver (Semtech Corporation, Camarillo, CA, USA)).

The microcontroller board, used in IoT Node, is the Arduino Mega 2560 Rev3 [21], which is based on the Atmega2560. This is an eight-bit microcontroller at 16 MHz, with 256 kB Flash program memory, 8 kB Static Random Access Memory (SRAM), and 4 kB Electrically Erasable Programmable Read-Only Memory (EEPROM). This board offers several interfaces: four Universal Asynchronous Receiver/Transmitter (UART), I2C, Serial Peripheral Interface (SPI), 54 digital inputs/outputs and 16 analog inputs.

On these analog inputs, it is possible to measure signals that are multiplexed and converted to digital, using the microcontroller's ADC. The ADC has ten-bit resolution and a sampling rate of 15,000 samples per second and accepts input values of up to five volts, the same value as the supply voltage.

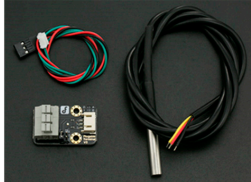
In the Gateway, these data are received from the LoRa network with an equal transceiver, processed, stored and displayed in a dashboard developed in Node-RED [22].

In this subsystem, the single-board computer Raspberry Pi 3 is used [23]. It is equipped with a 64-bit quad-core 1.4 GHz processor and 1 GB of Synchronous Dynamic Random Access Memory (SDRAM) memory. It has WIFI in 2.4 GHz and 5 GHz bands, Bluetooth 4.2, Gigabit Ethernet, four USB 2.0, HDMI ports and is fed to five volts. It has a Micro SD card slot where the operating system is installed. It also has a 40-pin connector where multiple pins of General-Purpose Input/Output (GPIO), a UART serial port, an I2C interface, and an SPI interface are made available.

4.1. Sensors and LoRa Module

For a better understanding of the sensors and LoRa module used in this prototype, images with the main characteristics and sensor calibration procedure are shown and explained in this section.

The temperature sensor used is the DFRobot DFR0198 [24], which is a waterproof sensor based on the DS18B20 [25]. It communicates with the Arduino board via the 1-Wire [26] protocol in digital pin D3. Figure 11 shows the parameters of this sensor.




Parameter	Value
Operating Voltage [V]	3.0 ~ 5.0
Current Consumption [μ A]	< 3
Operating Temperature [$^{\circ}$ C]	-55 ~ 125
Precision [$^{\circ}$ C]	± 0.5

Figure 11. DFRobot DFR0198, temperature sensor, parameters (adapted from [24]).

This temperature sensor, DS18B20, does not require calibration, as it is calibrated at the source.

The pH sensor used is the DFRobot SEN0161-V2 [27], which sends a voltage value to the analog input A1 of the Arduino board, and with software, converts it to the pH value of the solution being tested. The parameters of the pH sensor are shown in Figure 12.



Parameter	Value
Operating Voltage [V]	3.3 ~ 5.5
Output Voltage [V]	0 ~ 3.0
pH, values	0 ~ 14
Operating Temperature [$^{\circ}$ C]	5 ~ 60
Precision [pH]	$\pm 0.1 @ 25^{\circ}$ C
Internal Resistance, probe (M Ω)	< 250

Figure 12. DFRobot SEN0161-V2, pH sensor, parameters (adapted from [27]).

This sensor is calibrated with software adopting the two-point calibration method using two standard buffer solutions (pH = 7.0 and pH = 4.0). This calibration procedure is shown in Figure 13.

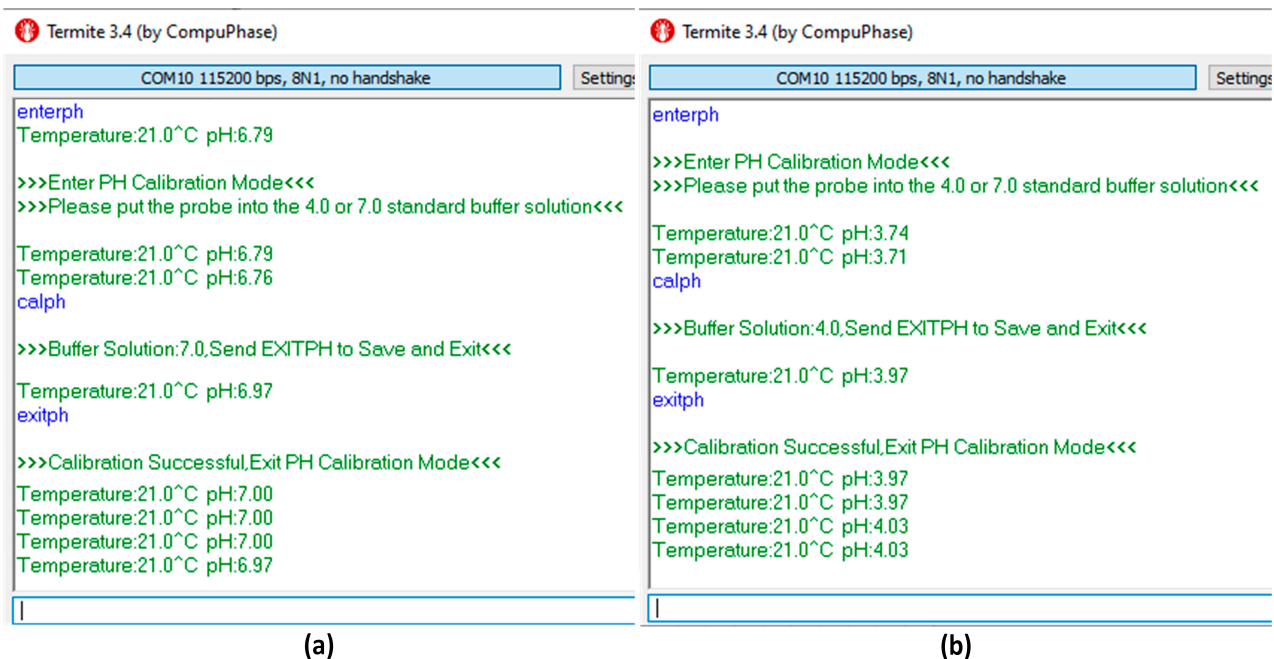


Figure 13. pH sensor calibration steps: (a) pH = 7 point; (b) pH = 4 point.

The conductivity sensor used is the DFRobot DFR0300 [28], which sends a voltage value to the analog input A2 of the Arduino board, and with software, converts it to the conductivity value of the solution being tested. Figure 14 shows the conductivity sensor parameters.

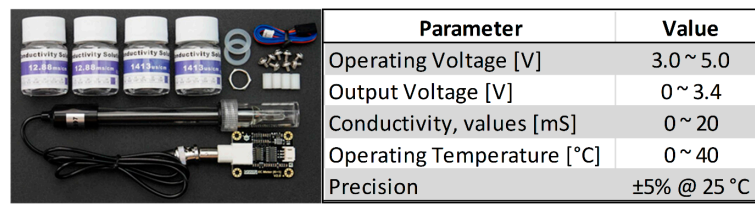


Figure 14. DFRobot DFR0300, conductivity sensor, parameters (adapted from [28]).

This sensor is calibrated with software, adopting the two-point calibration method using two standard buffer solutions (EC = 12.88 mS and EC = 1413 µS). This calibration procedure is shown in Figure 15.

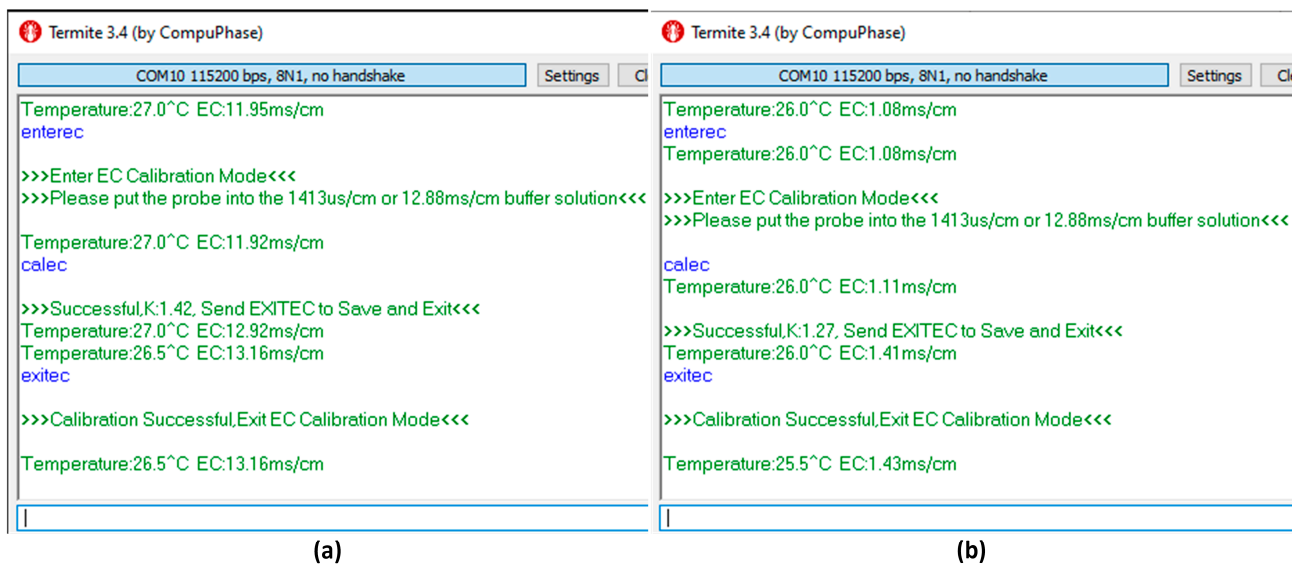


Figure 15. Conductivity sensor calibration steps: (a) EC = 12.88 mS point; (b) EC =1413 µS point.

The turbidity sensor used is the Seed Studio 101020752 [29], which sends a voltage value inversely proportional to the turbidity value to an analog input A3 of the Arduino. This voltage is converted to turbidity values (0 to 3000 NTU) with software. The parameters of this sensor are shown in Figure 16.

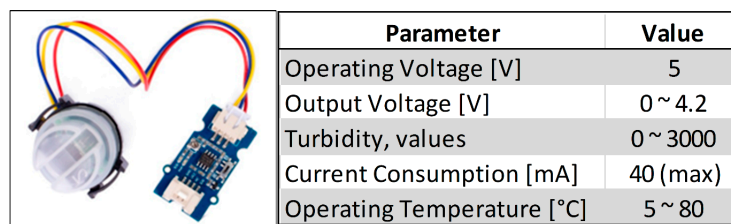


Figure 16. Seed Studio 101020752, turbidity sensor, parameters (adapted from [29]).

No calibration process was performed on this sensor, but the voltage-to-turbidity conversion formula (Figure 17) provided by the manufacturer was used.

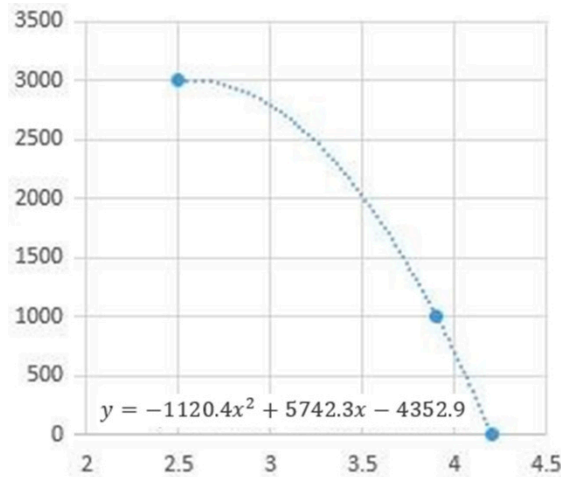


Figure 17. Relationship between turbidity and voltage (adapted from [29]).

The LoRa module used in both subsystems is based on Semtech chip SX1276 [20], a LoRa half-duplex transceiver that operates in the ISM frequency band. The basic characteristics are shown in Figure 18.

Parameter	Value
Operating Voltage [V]	1.8 ~ 3.7
Current Consumption Tx [mA]	120 (max)
Current Consumption Rx [mA]	12 (max)
Operating Temperature [°C]	-40 ~ 85
Power (Max) Tx [dBm]	20
Sensibility (Max) Rx [dBm]	-148
Operation Band [MHz]	862 ~ 1020

Figure 18. SX1276, LoRa module characteristics (adapted from [20]).

In both subsystems, IoT Node and Gateway, this LoRa module interconnects with the controller board (Arduino and Raspberry Pi) by SPI for communication and other auxiliary pins for control and interrupting handling.

4.2. IoT Node

For the IoT Node, an Arduino shield, or a Hardware Attached on Top (HAT), is designed, and its electrical schematic is shown in Figure 19. As the Arduino works at 5 V and the LoRa module works at 3.3 V, to ensure a good adaptation of logic levels, a voltage-level shifter (TXB0108 [30]) is inserted between these two devices.

The Printed Circuit Board (PCB) of this Arduino shield is shown in Figure 20. This board is also compatible with Arduino Uno.

The final prototype of the IoT Node subsystem, with the components assembled in a box and prepared for installation close to a river, is shown in Figure 21.

In IoT Node, a program is developed for the Arduino board in C/C++, whose main function is to take the sensor readings and create a payload with these values to be sent to the LoRa network. Figure 22 shows the program flowchart.

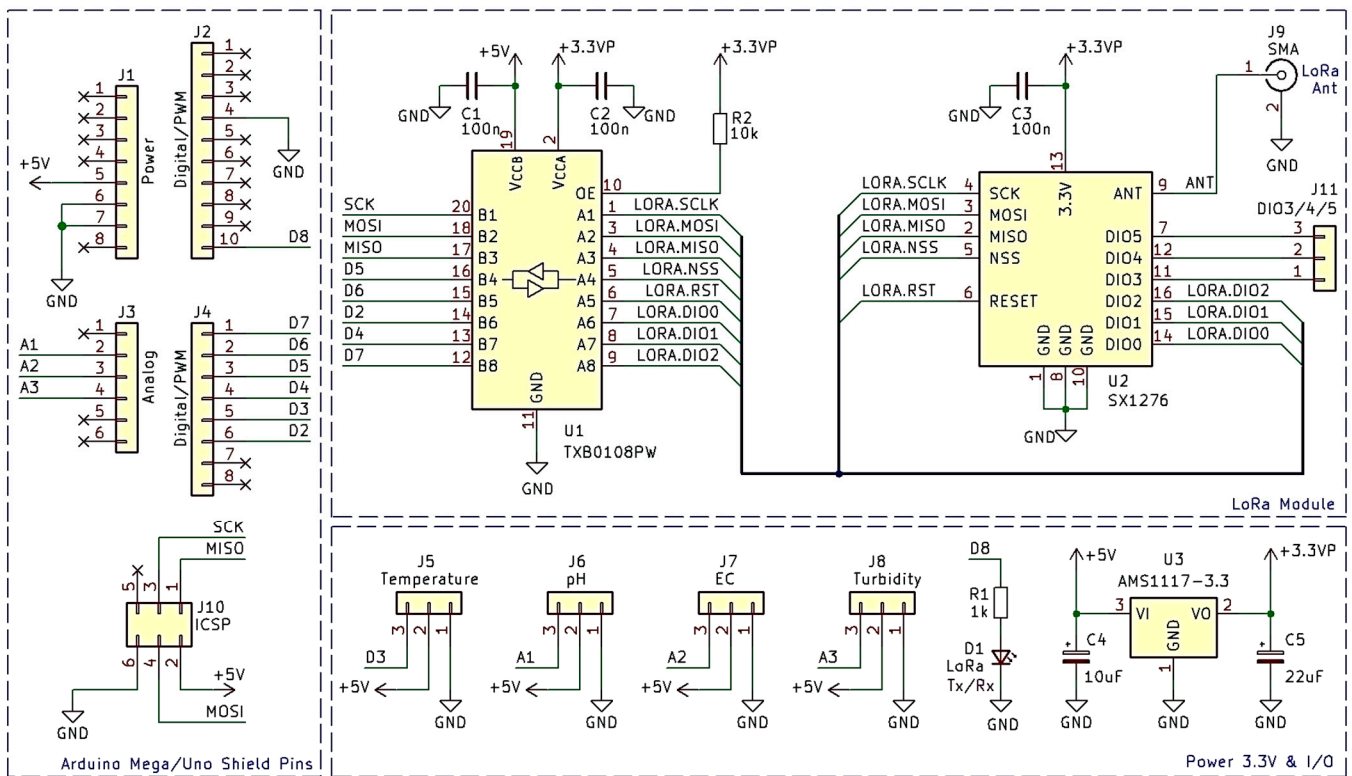


Figure 19. Electrical schematic of IoT Node subsystem.

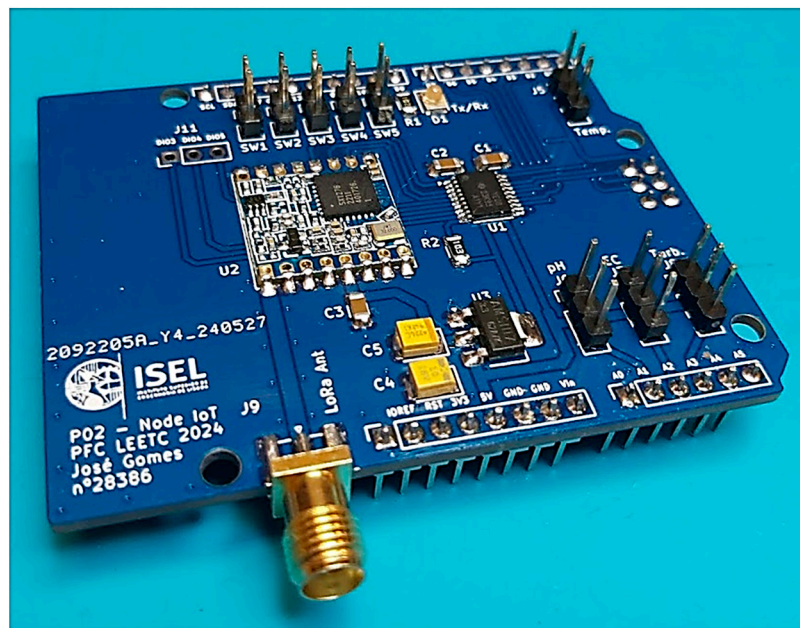


Figure 20. PCB developed for IoT Node subsystem (Arduino shield).

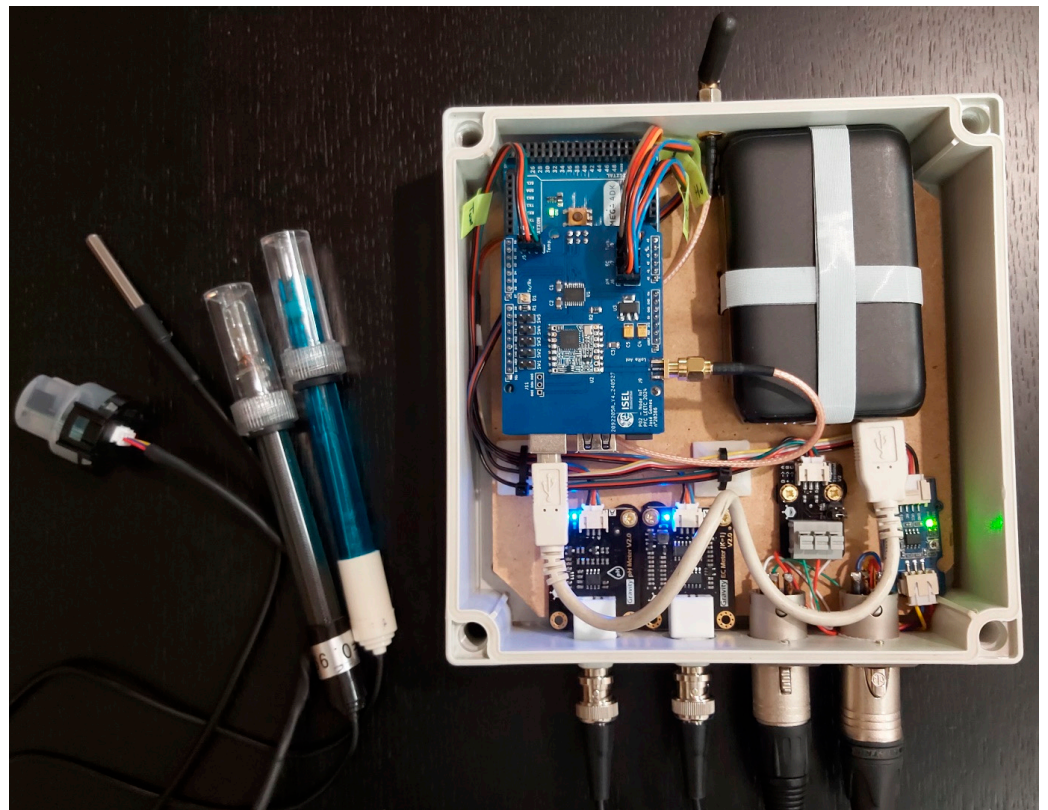


Figure 21. IoT Node subsystem prototype, practical assembly.

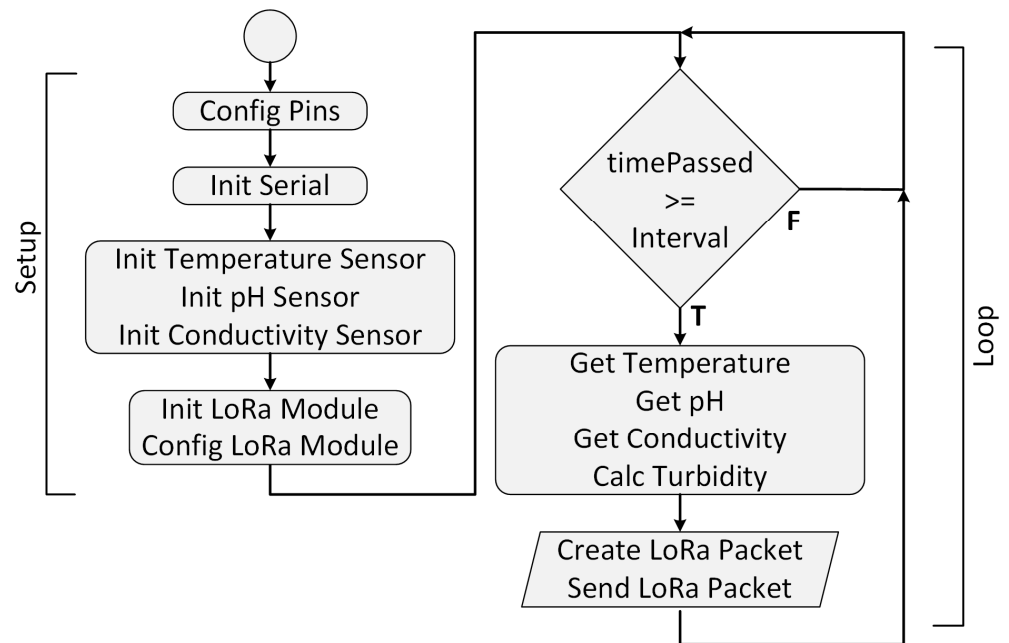


Figure 22. IoT Node program flowchart. After the peripherals are initialized (setup), it periodically sends LoRa messages with sensor data (loop).

4.3. Gateway

For Gateway, an HAT is designed for the Raspberry Pi 3 board, and its electrical schematic is shown in Figure 23.

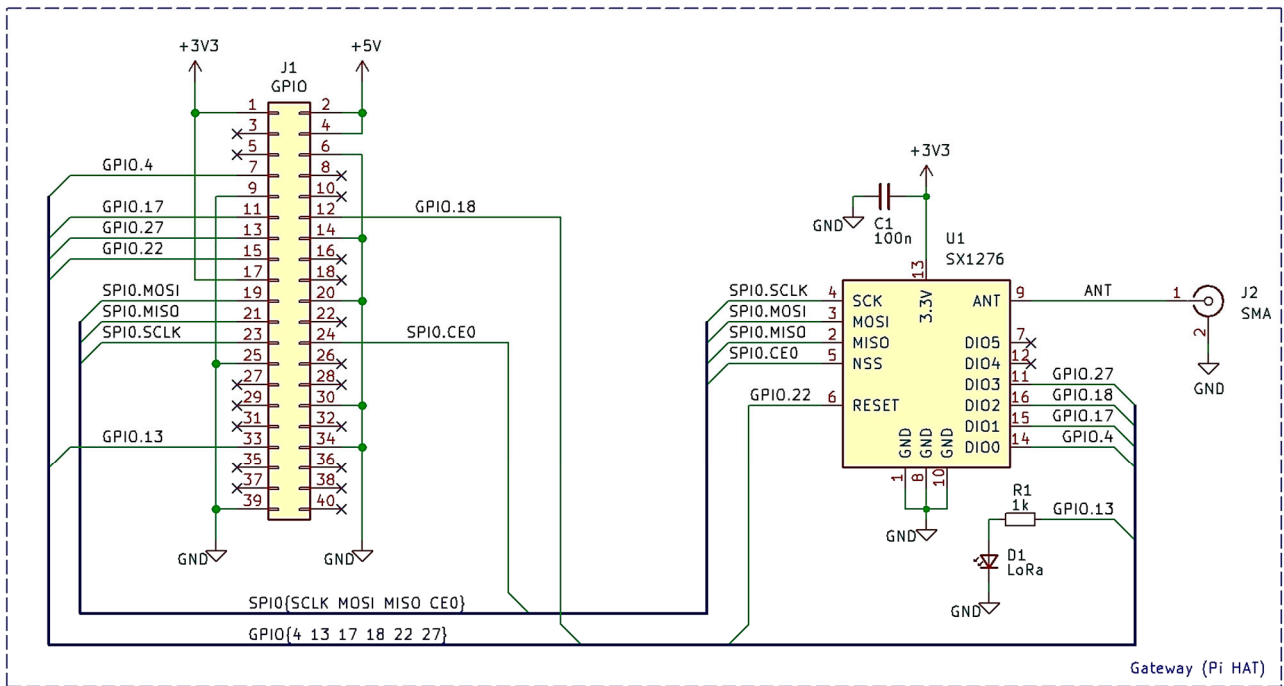


Figure 23. Electrical schematic of Gateway subsystem.

The PCB (Pi HAT) developed for the Gateway subsystem is shown in Figure 24.

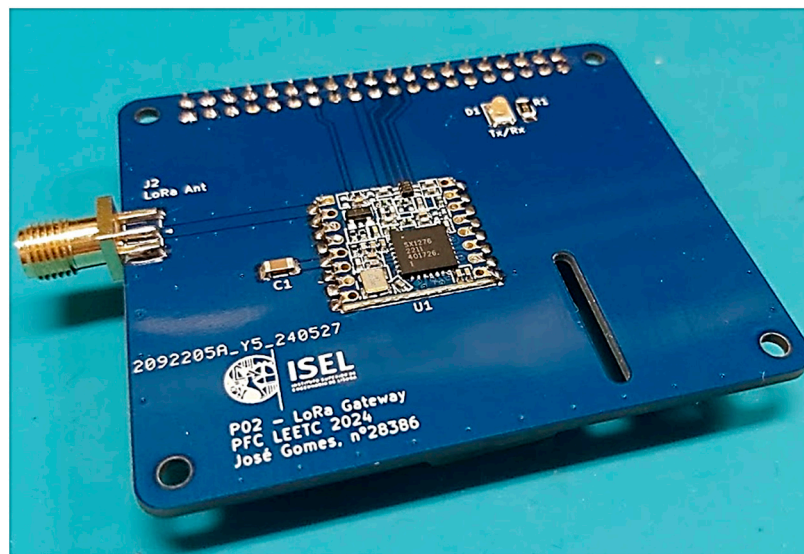


Figure 24. PCB developed for Gateway subsystem (Pi HAT).

The final prototype of the Gateway subsystem, with the display, the Raspberry Pi and the HAT assembled, is shown in Figure 25.

For the Gateway, a program is developed in Python [31] where the data are received with another LoRa transceiver. Figure 26 shows the program flowchart.

In the Main section of the program, only the LoRa module is initialized, and the connection from the MQTT client to the MQTT Broker is carried out.

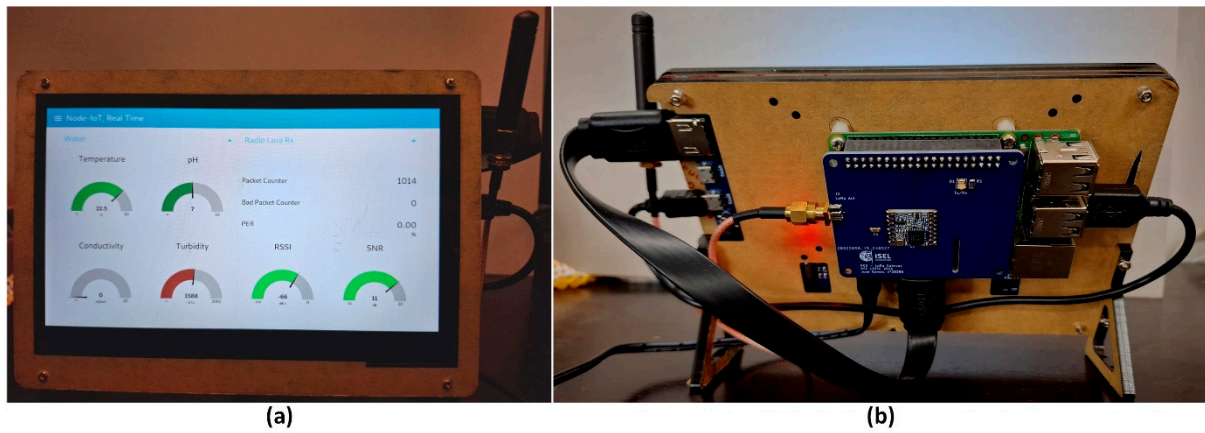


Figure 25. Gateway subsystem prototype, practical assembly: (a) front view; (b) rear view.

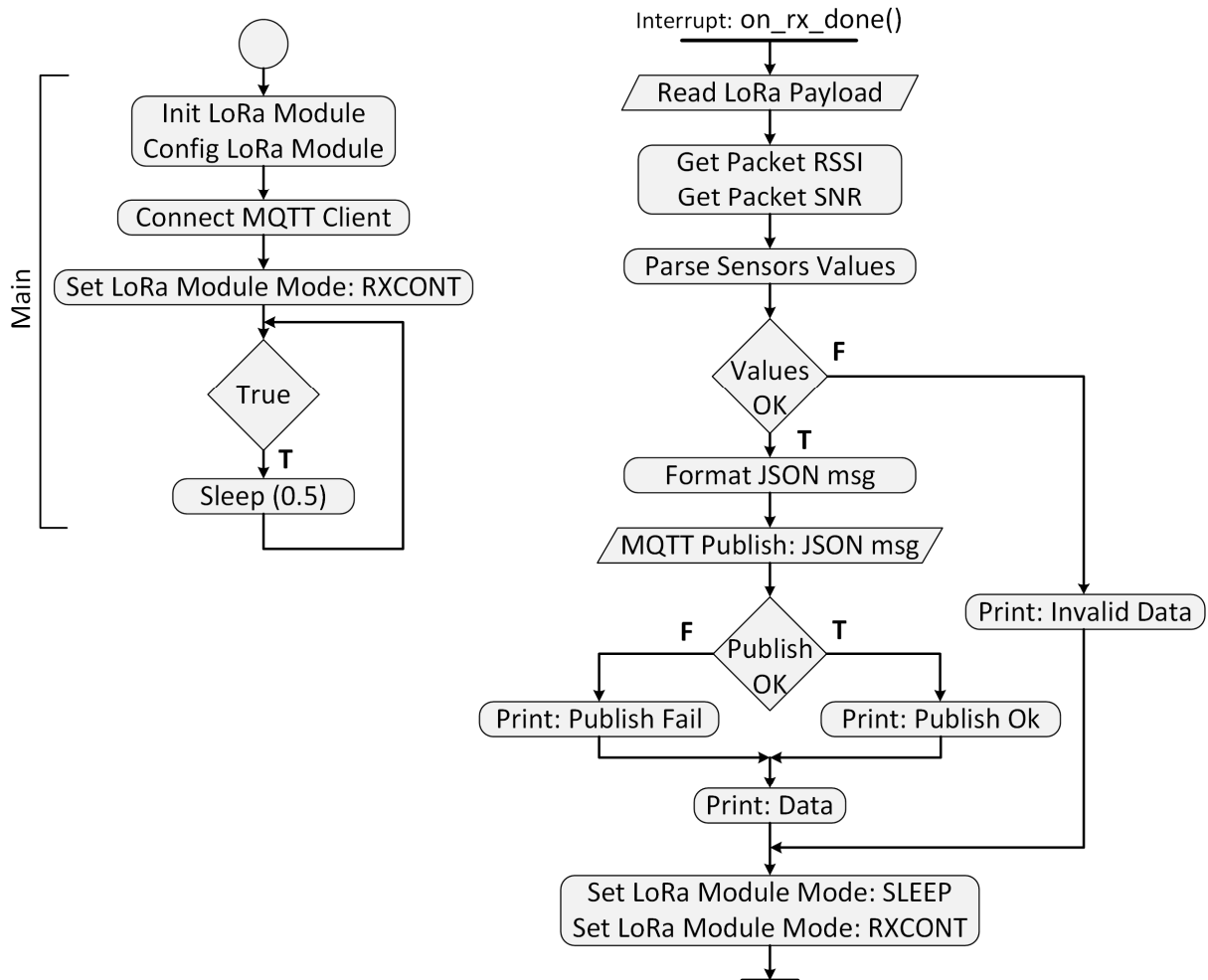


Figure 26. Gateway program flowchart, initialization and receive interrupt handler.

When the LoRa module receives a message, it will cause an interruption in the program, through the DIO0 pin, and then in the interrupt handler (`on_rx_done()`), the data are processed and published in a local MQTT Broker (Mosquito) [32] by an MQTT Client (Paho) [33,34]. The data are subscribed by another MQTT Client in a local Node-RED [22] and finally displayed on the dashboard; see Figure 27.

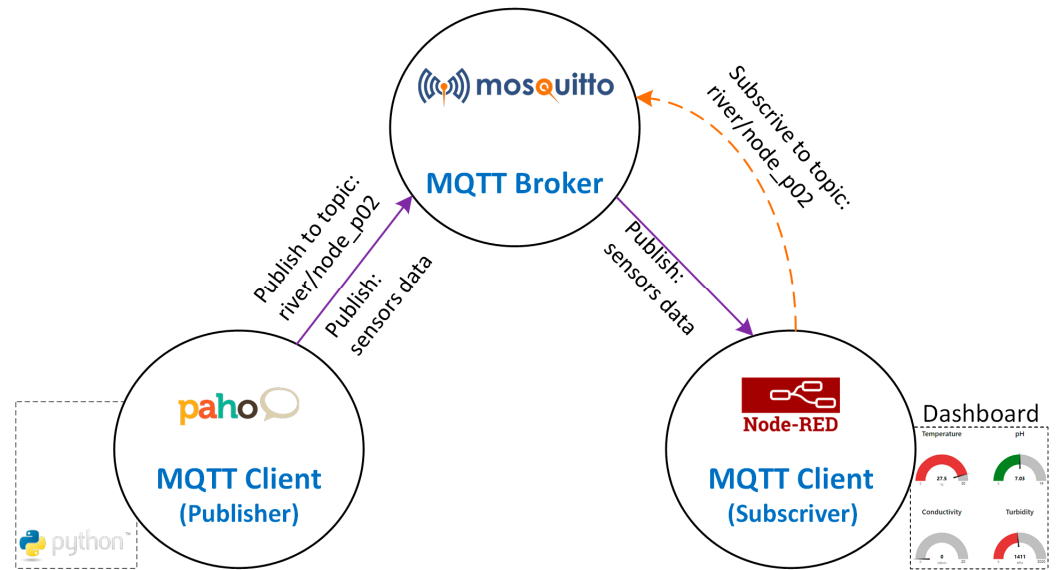


Figure 27. MQTT architecture flowchart in Gateway subsystem.

In Node-RED, a flow is developed to allow us to present the data on a dashboard. In the Real Time page, the sensor data (Figure 28a) and the radio LoRa reception parameters (RSSI, SNR, packet counter and Packet Error Rate (PER)) (Figure 28b) are shown.

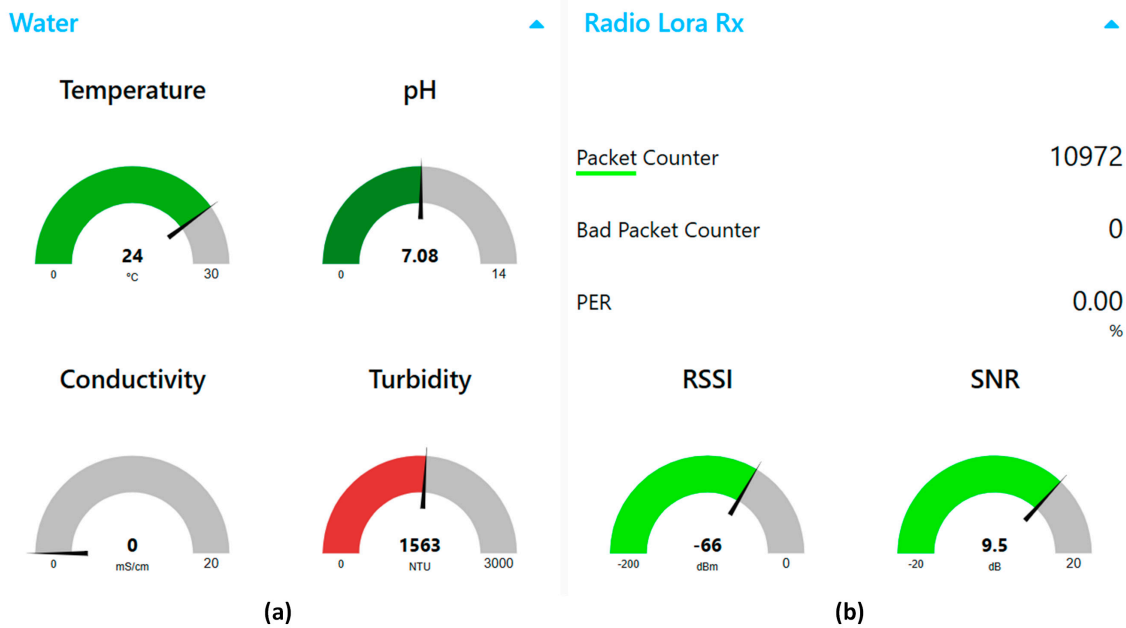


Figure 28. Dashboard, real time data page: (a) water data; (b) radio LoRa data.

Also, in this Node-RED flow, the data are saved in Comma-Separated Values (CSV) format, for historic view and future external processing. These files can be visualized and downloaded on the Historical page of the dashboard; see Figure 29.

File Browser

REFRESH
RESET
UP

Folder: /home/isel/P02/data_log

Hidden

File Name	Size	Created	Changed
river_20240626.csv	20305	2024-07-10 22:06:18	2024-07-10 22:06:18
river_20240709.csv	19927	2024-07-08 23:00:33	2024-07-09 22:56:57
river_20240710.csv	19569	2024-07-09 23:00:10	2024-07-10 22:56:48
river_20240711.csv	19957	2024-07-10 23:00:01	2024-07-11 22:56:57
river_20240712.csv	20096	2024-07-11 23:00:10	2024-07-12 22:57:15
river_20240713.csv	20112	2024-07-12 23:00:28	2024-07-13 22:57:25
river_20240714.csv	19999	2024-07-13 23:00:38	2024-07-14 22:57:30
river_20240715.csv	20246	2024-07-14 23:00:43	2024-07-15 22:57:31
river_20240716.csv	19543	2024-07-15 23:00:44	2024-07-16 22:57:30
river_20240717.csv	19794	2024-07-16 23:00:43	2024-07-17 22:57:31
river_20240718.csv	19634	2024-07-17 23:00:44	2024-07-18 22:57:46
river_20240719.csv	20334	2024-07-18 23:00:59	2024-07-19 22:57:56

Select a folder

OPEN
river_20240626.csv
DELETE
GRAPH

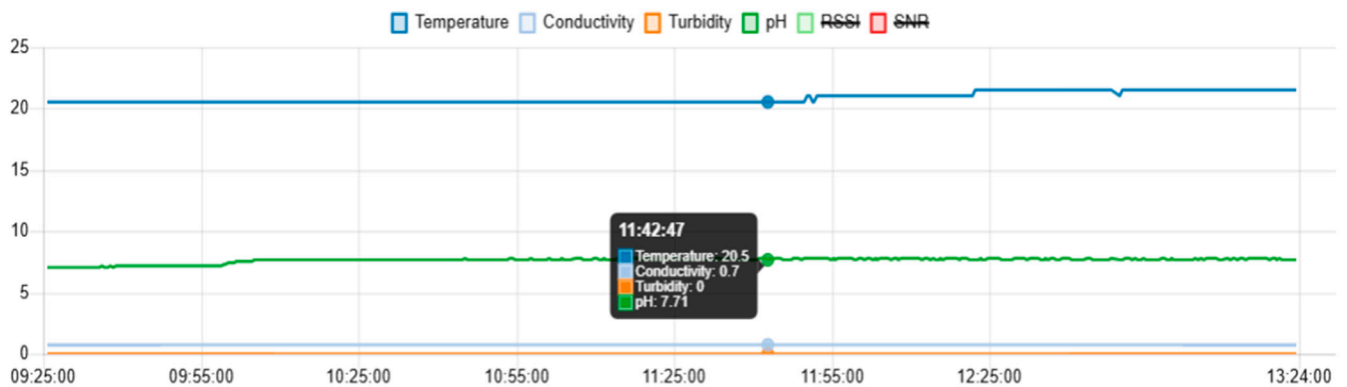


Figure 29. Dashboard, historical page, data and log files.

5. Results

In this section, energy consumption measurements for the IoT Node subsystem, LoRa radio coverage tests [35], and finally, a test in a real scenario on the Jamor River are presented.

5.1. Energy Consumption

As the IoT Node is to be operated with a battery, energy consumption measurements were carried out, with the Arduino in energy saving mode, defining the consumption in each phase of a work cycle of this subsystem: sensor reading (phase A), LoRa module transmitting (phase B) and Arduino in power down mode (phase C). See Figure 30.

And this leads to the definition of the IoT energy consumption in one hour (9):

$$Consumption_{(IoT\ Node)} = \frac{102 * T_A + 176 * T_B + 52 * T_C}{T_A + T_B + T_C} \text{ [mAh]} \quad (9)$$

where:

- T_A is phase A: sensors reading time [s];
- T_B is phase B: T_{packet} [s];
- T_C is phase C: Arduino powerDown (sleep) time [s].

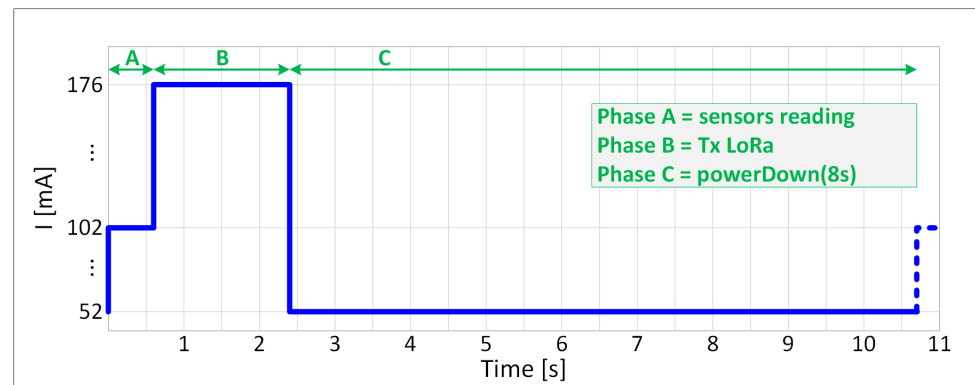


Figure 30. IoT Node, power measurements.

T_A is fixed (0.6 s), T_B is the LoRa T_{packet} and T_C is the powerDown time configured on the Arduino. It is thus possible to predict the duration of a given battery or calculate its capacity for a required duration of operation.

Phase C consumption could still be minimized by removing the signaling LEDs from this subsystem.

5.2. LoRa Radio Coverage

LoRa radio coverage tests were carried out and a maximum distance of 2.53 km was obtained, in an urban and Non-Line of Sight (NLOS) environment. For practical reasons, to facilitate this test, the IoT Node was fixed, and the Gateway was on the move; see Figure 31.

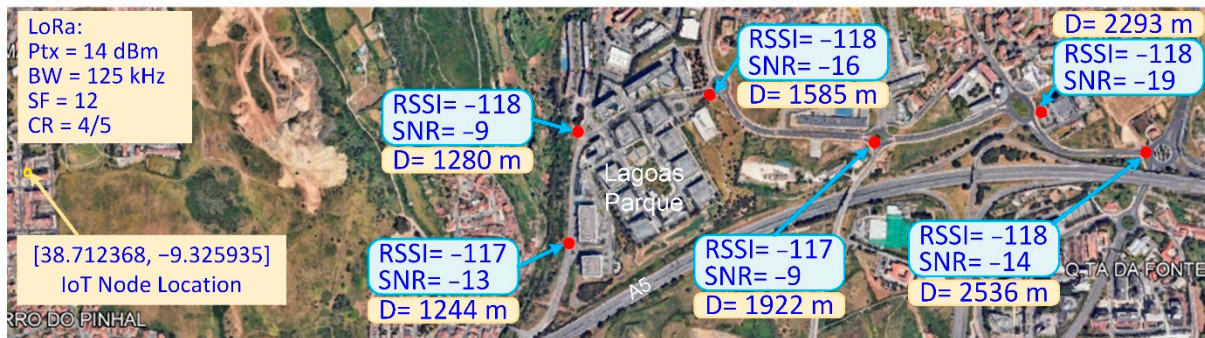


Figure 31. LoRa radio coverage test.

5.3. Monitoring a River (Real Scenario)

The real scenario testing site for this prototype was on the left bank of the Jamor River, in Carnaxide, Oeiras council, at the following coordinates: [38.721951, -9.249049]; see Figure 32.

The IoT Node was installed near the water line [36], close to the sensors, and the Gateway was installed near this location, on the nearby street, inside a car. Due to the topography of the location, there is a height difference between the two locations, which, although only 10 m away, influences the radio signal received in Gateway. The average Received Signal Strength Indication (RSSI) upon Gateway reception was -94 dBm, with an Effective Isotropic Radiated Power (EIRP) of approximately 14 dBm at the IoT Node, and with an average Signal-to-Noise Ratio (SNR) of 7.33 dB.

These tests were on 26 June 2024, and were conducted for almost 4 h in the morning, in the period between 9:25 am and 1:21 pm. In Table 2, a summary of the environmental conditions is shown.



Figure 32. River Jamor, test site: (a) openstreetmap location; (b) test site photo.

Table 2. Test site environmental conditions.

Air temperature	Around 25 °C
Weather conditions	Partly cloudy and light rain sometimes occurred
Water depth	Around one meter
Water speed	Very slow

The data were sent every 30 s, recorded and then exported to Excel for processing, and the analysis of results is presented below.

Figure 33a shows a graph with temperature variation, where a slight increase can be seen from 11:49, probably due to the influence of the sun. These values are normal for a location already close to the river mouth, as in this case.

In Figure 33b, the pH variation shows an average value of 7.7, except at the beginning, perhaps due to the sensor adaptation period. The pH is within normal values for water (6.5–8.5).

The average conductivity is 0.7 mS/cm and is below the maximum recommended value (2 mS/cm); the small variation is shown in Figure 33c.

The turbidity throughout the measurement time always showed values equal to zero NTU, as the water was transparent, as seen in Figure 33d.

For a comparison with the other water sources measured, see Table 3.

Table 3. Comparison with other water sources.

Water	Temp.	pH	Conductivity	Turbidity
River Jamor	21	7.7	0.7	0
Pub. Water supply (Cascais)	n.a.	6.8	0.12	0
Bottled Luso water	n.a.	6	0	0

And we can conclude, based on these parameters, that the water quality of the Jamor River is reasonable, but shows a slight presence of contaminants, due to the conductivity value of 0.7 mS/cm and some alkalinity due to the pH of 7.7.

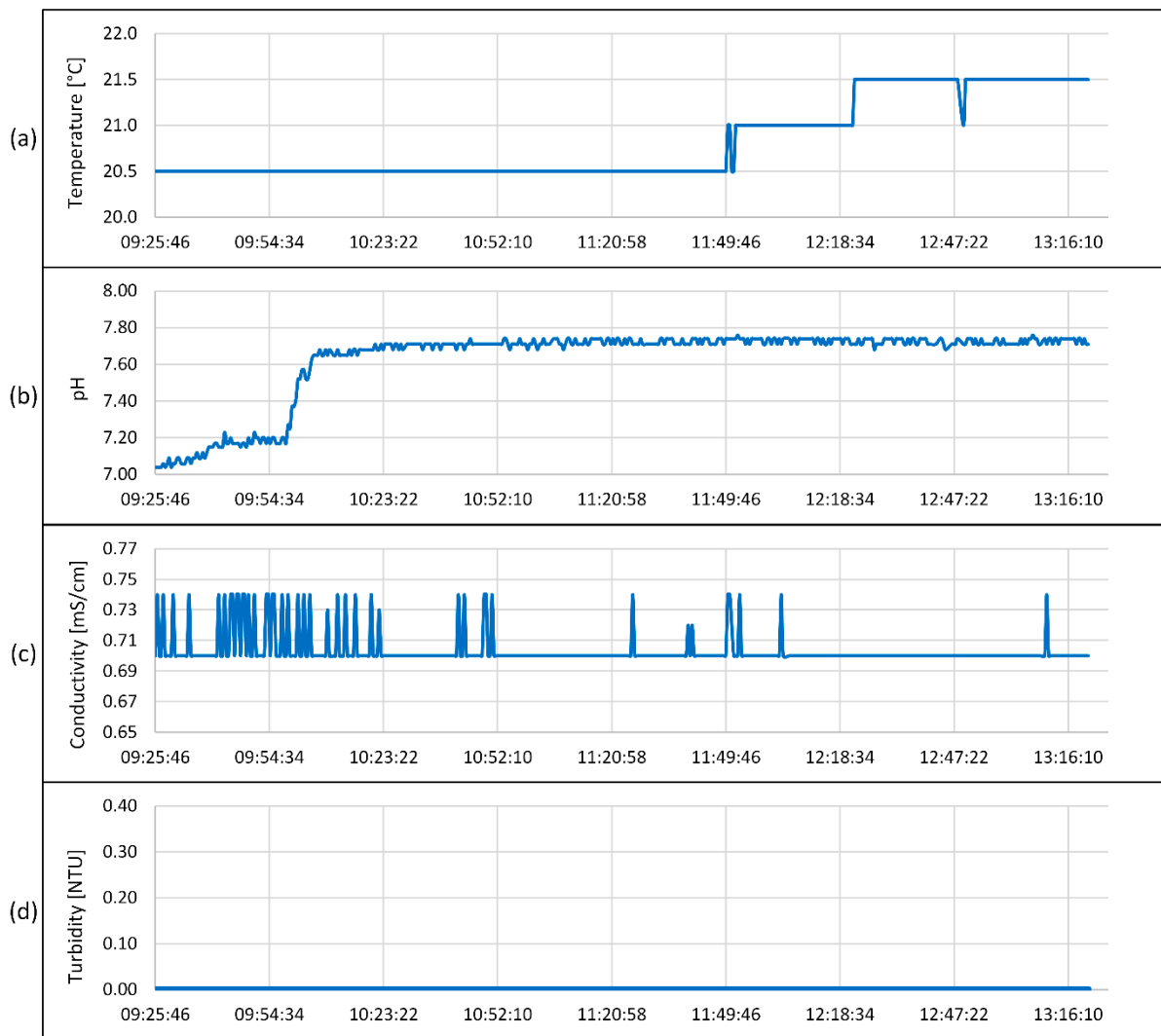


Figure 33. River water parameter variation: (a) temperature, (b) pH, (c) conductivity and (d) turbidity.

6. Conclusions and Future Work

The developed prototype offers a reliable and affordable solution, providing real-time river water parameters. This enables timely action to address any irregularities.

Other LPWAN networks were studied, but it was concluded that for this prototype, in a point-to-point connection, LoRa is the best option.

Using Node-RED, an application was developed to display all information in real time and record data in daily files for future reference. Additionally, with MQTT, these data can be published externally to the system, such as in cloud applications.

Radio coverage tests were conducted in an NLOS scenario within a suburban area. The results further validate the effectiveness of using LoRa technology in IoT applications.

Emphasis was placed on reducing energy consumption, particularly for the IoT Node. Consumption measurements were conducted to determine the optimal energy supply solution for this subsystem, considering any LoRa modulation configuration for future use.

A river was monitored for a few hours to demonstrate the prototype’s functionality in a real-world scenario.

One challenge is coverage, as riverbeds are typically located in valleys, which are obstructed and lack a direct line of sight to the reception Gateway.

Integrating this system into the LoRaWAN network would be a valuable future enhancement. Additionally, upgrading the sensors to those designed for continuous

use and incorporating additional sensors to measure parameters such as dissolved oxygen levels or dissolved organic matter would further improve the system.

Author Contributions: Conceptualization, L.M.P. and J.G.; methodology, L.M.P. and J.G.; software, L.M.P. and J.G.; validation, L.M.P. and J.G.; formal analysis, L.M.P. and J.G.; investigation, L.M.P. and J.G.; resources, L.M.P. and J.G.; data curation, L.M.P. and J.G.; writing—original draft preparation, L.M.P. and J.G.; writing—review and editing, L.M.P. and J.G.; visualization, L.M.P. and J.G.; supervision, L.M.P. and J.G.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Khan, M.A.I.; Hoque, M.A.; Ahmed, S. IoT-based System for Real-time Water Pollution Monitoring of Rivers. In Proceedings of the 2021 International Conference on Electronics, Communications and Information Technology (ICECIT), Khulna, Bangladesh, 14–16 September 2021.
2. YSI. Understanding pH in Water Quality. Available online: <https://www.ysi.com/parameters/ph> (accessed on 2 May 2024).
3. YSI. Conductivity of Water. Available online: <https://www.ysi.com/parameters/conductivity> (accessed on 4 May 2024).
4. YSI. What Is Turbidity? Available online: <https://www.ysi.com/parameters/turbidity> (accessed on 7 May 2024).
5. ORACLE. What Is IoT? Available online: <https://www.oracle.com/pt/internet-of-things/what-is-iot/> (accessed on 3 April 2024).
6. ANACOM. Utilização da Internet das Coisas. Relatório Utilização da Internet das Coisas, IoT. 2023, p. 7. Available online: <https://www.anacom.pt/render.jsp?contentId=1737942> (accessed on 2 October 2024).
7. Alnajjar, K.A.; Fahmy, M.M.A.; Ali, Y.M.; Ansari, S. Water Quality Monitoring System Based on the Internet of Things. In Proceedings of the IEEE 6th International Conference on Electrical, Electronics and System Engineering (ICEESE), Shah Alam, Malaysia, 29–30 August 2023; pp. 32–37.
8. Savel, V.; Rakluea, P.; Jangjing, T.; Kumkhet, B.; Mahatthanajatuphat, C.; Thaiwirot, W. IoT Based Water Quality Monitoring System Using Solar Powered and LoRaWAN. In Proceedings of the International Electrical Engineering Congress (iEECON), Khon Kaen, Thailand, 9–11 March 2022; pp. 1–4.
9. Cinterion, T. NB-IoT Explained: What Is It, and How Does It Work? 28 November 2022. Available online: <https://www.telit.com/blog/nb-iot-new-cellular-standard-means-business/> (accessed on 11 April 2024).
10. ROHDE & SCHWARZ. Narrowband Internet of Things. 2016. Available online: https://www.rohde-schwarz.com/us/applications/narrowband-internet-of-things-white-paper_230854-314242.html (accessed on 4 April 2024).
11. Sigfox. What Is Sigfox 0G Technology? Available online: <https://www.sigfox.com/what-is-sigfox/> (accessed on 11 April 2024).
12. Semtech. What Are LoRa and LoRaWAN? Available online: <https://www.thethingsnetwork.org/docs/lorawan/what-is-lorawan/> (accessed on 14 April 2024).
13. What Is LoRaWAN® Specification? Available online: <https://lora-alliance.org/about-lorawan/> (accessed on 14 April 2024).
14. Würth Elektronik. What Is a LoRaWAN® Gateway? Available online: <https://www.we-online.com/en/components/products/DAPHNIS-I> (accessed on 19 November 2024).
15. Kais, M.; Eddy, B.; Frederic, C.; Fernand, M. A comparative study of LPWAN technologies for large-scale IoT deployment. *ICT Express* **2019**, *5*, 1–7.
16. Kamoona, Z.; Ilyas, M. Investigating the Performance of LoRa Communication for Nominal LoRa and Interleaved Chirp Spreading LoRa. In Proceedings of the International Conference on Artificial Intelligence of Things (ICAIoT), Istanbul, Turkey, 29–30 December 2022.
17. SEMTECH. *LoRa Modulation Basics*. AN1200.22. 2015. Available online: <https://www.semtech.com/uploads/technology/LoRa/lora-and-lorawan.pdf> (accessed on 3 April 2024).
18. SEMTECH. *Designer's Guide*. AN1200.13. 2013. Available online: https://www.openhacks.com/uploadsproductos/loradesignguide_std.pdf (accessed on 23 April 2024).
19. LoRa Alliance Technical Committee Regional Parameters Workgroup. *LoRaWAN™ 1.1 Regional Parameters*; LoRa Alliance, Inc.: Fremont, CA, USA, 2017; p. 8. Available online: <https://lora-alliance.org/wp-content/uploads/2020/11/lorawan-regional-parameters-v1.1ra.pdf> (accessed on 23 April 2024).
20. Semtech. SX1276/77/78/79 Datasheet. Semtech, 2020. Available online: <https://www.semtech.com/products/wireless-rf/lora-connect/sx1276> (accessed on 5 April 2024).
21. Arduino. Arduino Mega ADK Rev3. Available online: <https://docs.arduino.cc/retired/boards/arduino-mega-adk-rev3/> (accessed on 12 March 2024).
22. OpenJS Foundation. Node-RED. OpenJS Foundation. Available online: <https://nodered.org/> (accessed on 12 March 2024).

23. Raspberry Pi 3 Model B+. Raspberry Pi. Available online: <https://www.raspberrypi.com/products/raspberry-pi-3-model-b-plus/> (accessed on 10 June 2024).
24. DFRobot. Waterproof DS18B20 Digital Temperature Sensor for Arduino. DFRobot. Available online: <https://www.dfrobot.com/product-689.html> (accessed on 12 March 2024).
25. Dallas, M. DS18B20. Analog Devices. Available online: <https://www.analog.com/en/products/ds18b20.html> (accessed on 12 March 2024).
26. Maxim, A.D. Overview of 1-Wire Technology and Its Use. Available online: <https://www.analog.com/en/resources/technical-articles/guide-to-1wire-communication.html> (accessed on 12 March 2024).
27. DFRobot. Gravity: Analog pH Sensor/Meter Kit V2. DFRobot. Available online: <https://www.dfrobot.com/product-1782.html> (accessed on 12 March 2024).
28. DFRobot. Gravity: Analog Electrical Conductivity Sensor. DFRobot. Available online: <https://www.dfrobot.com/product-1123.html> (accessed on 12 March 2024).
29. Seed Studio. Grove—Turbidity Sensor (Meter) V1.0. Seed Studio. Available online: <https://www.seeedstudio.com/Grove-Turbidity-Sensor-p-4399.html> (accessed on 12 March 2024).
30. TXB0108 8-Bit Bidirectional Voltage-Level Shifter. Texas Instruments. Available online: <https://www.ti.com/product/TXB0108> (accessed on 10 June 2024).
31. Python Software Foundation. Python. Python Software Foundation. Available online: <https://www.python.org/> (accessed on 12 March 2024).
32. Eclipse Foundation. Eclipse Mosquitto—An Open Source MQTT Broker. Eclipse Foundation. Available online: <https://mosquitto.org/> (accessed on 12 March 2024).
33. Eclipse Paho MQTT Python Client Library. Eclipse. Available online: <https://eclipse.dev/paho/index.php?page=clients/python/index.php> (accessed on 22 March 2024).
34. Pires, L. Redes de Sensores sem Fios: Aplicações para Supervisão e Monitorização de Infraestruturas. Revista Manutenção (Portuguese Magazine) No. 135. 2017. pp. 66–68. Available online: <https://www.revistamanutencao.pt/artigo-cientifico/redes-de-sensores-sem-fios-aplicacoes-para-supervisao-e-monitorizacao-de-infraestruturas/> (accessed on 27 February 2024).
35. Incibe. The LoRaWAN Protocol. 15 June 2023. Available online: <https://www.incibe.es/en/incibe-cert/blog/lorawan-and-its-contribution-iiot> (accessed on 18 May 2024).
36. YSI. Temperature. Available online: <https://www.ysi.com/parameters/temperature> (accessed on 18 May 2024).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.