Master's Thesis

A LoRa-based Framework with applications to Smart Buildings

Marcos Torres Padín

March, 2020

Examiner: Prof. Dr.-Ing. Sergio Lucia

Technische Universität Berlin
Department of Electrical Engineering and Computer Science

Chair of Internet of Things for Smart Buildings

Abstract

Modern communication technologies allow people to access and monitor anything from anywhere using networks like LoRaWAN that is often described as one of the Internet of Things (IoT).

For this reason, in this project the LoRa technology will be used as a means of communication integrated with the Lopy4 commercial devices with the need to obtain data on the environment of a building to adjust theoretical models were created to predict the indoor behaviour of the building.

This work is implemented by placing a device in a room for the collection, analysis and verification of data from the sensors and send them through LoRa to The Things Network platform. Then, this data is processed together with the forecast weather in a website designed in Angular. In addition, the energy harvesting mechanism will be applied and analysed using a solar panel.

The goal of this field work is to verify that the data obtained from the sensors is reliable and accurate. Then, analyse why these values are affected by factors such HVAC systems (Heating, Ventilating, Air Conditioned). In addition to the data collected, this research also tries to reduce the energy consumption of the device as well as analyse the consumption in the different phases to estimate the useful life or increase it by applying the energy harvesting mechanism.

The results collected by the sensors are consistent with the reality and the manufacturer's specifications. However, the mechanism of energy harvesting has not been favourable because the solar charger consumes a significant amount of energy when it is sunless.

Acknowledgement

I want to express my sincere gratitude to all the people who walked with me and supported me during the training, both professionally and personally in Berlin, Barcelona, and Galicia.

I also thank both Polytechnic University of Barcelona and TU Berlin for providing me the opportunity to carry out the Erasmus program, to take part in the training and gain experience abroad. I also thank Professor Sergio Lucia for having accepted the direction of this work and his constant support and guidance.

The completion of this project could not have been achieved without the support of my friends especially Michelle, Jean, Shinto, Bruno, Waheed and Sebastian for turning these months into a few days. I also thank my family members were always with me all throughout the entire period.

Overall, thanks for this experience and great encouragement that have been invaluable.

List of Figures

List of Tables

Abbreviations

3GPP	3rd Generation Partnership Project
ABP	Activation By Personalization
ACK	Acknowledgement
AES	Advanced Encryption Standard
API	Application Programming Interface
AppSkey	Application Server Key
CORS	Cross-Origin resource Sharing
CSS	Chirp Spread Spectrum
DHI	Diffuse Horizontal Irradiation
DNI	Direct Normal Irradiance
DoS	Denial-of-Service
ETSI	European Telecommunications Standards Institure
GHI	Global Horizontal Irradiation
HTML	Hypertext Markup Language
HVAC	Heating, Ventilating, Air Conditioned
IBM	International Business Machines
IoT	Internet of Things
ISM	Industrial, Scientific and Medical
LoRa	Long-Range
LPN	Low-Power Network
LPWAN	Low-Power Wide-Area Network
MAC	Media Acces Control
MVC	Model-View-Controller
OSI	Open System Interconnection
OTAA	Over-The-Air Activation
NwkSKey	Network Server Key
P2P	Point Two Point
PHY	Physical
SF	Spread Factor
SPA	Single Page Application
TDOA	Time Difference of Arrival
TTN	The Things Network

Contents

1 Introduction

1.1 Background and motivation

In Europe, the existing buildings account for approximately 40% of the energy consumed and 35% of carbon dioxide emissions [3]. Some of this energy is wasted because most of the buildings are old, inefficient and does not have proper energy management system. Applying few techniques such as insulation of the external walls, using double-glazed windows, installation of solar panels [4], it is estimated that the efficient energy measures available could save up to 28%. But the problem of renovating an old building is not so simple and often costs high.

If we do not want to make a high investment, a first step would be to install certain devices to measure different parameters such as temperature, humidity and light in order to improve and have a more complete study of the house. This type of device can use low power wide area network (LPWAN) which is gaining popularity because it's low power, long range and low cost. This type of technology can cover distances up to 40 km with high energy efficiency suitable for placing the device in environments where there is no nearby power outlets or the need to deploy cables or to modify the infrastructure. Within the LPWAN, it has beheaded for using LoRa technology as a means of communication in this project.

This work arises from the need to obtain real data from a research conducted by Danny Nowka in his thesis "Automatic Thermal Modeling of City Districts Based on Open Data for Energy-Optimal Predictive Control" where the public information that the Berlin administration has about the structures of the buildings, theoretical models were created to predict the indoor behaviour of the building.

The next step after the aforementioned investigation is to gather and verify that the data collected by the sensors are correct using

LoRa technology. Together with the previous data, a website will be designed in Angular to graphically display the data collected as well as those of the forecast weather. Also, to analyse and check the consumption of the device in the different phases of operation to estimate the useful life as well as apply energy harvesting techniques to prolong autonomy.

1.2 Objetives

As previously mentioned, technology and data transmission using wireless sensors have shown to be of great interest, since they facilitate and improve efficiency compared to other technologies used in the past. This document is intended to use one of those technologies, LoRa.

Based on this technology have been defined a series of specific objectives that will allow reaching the final goal:

1. Installation of pycom devices in a room with temperature, humidity and light sensors, integrating LoRa as a communication medium.

2. Programming in mycrophyton of the sensors equipped in the pycom as well as the configuration of the Lora communication.

3. Transmission of data to the Lorawan network via The Things Networks platform every 30 minutes.

4. Implementation of a website in Angular to represent the data obtained combined with the forecast weather.

5. To evaluate the behavior of the data faced with certain interactions and that are coherent with the reality.

6. Optimize the code of Pycom devices to increase the lifespan.

7. Implement energy harvesting with a solar panel and analyse how much you can increased the lifespan of the device.

1.3 Scope and structure

Based on the above ideas, the main part of this thesis is to build a node with commercial devices that are able to gather sensor readings and send them through LoRa technology as well as integrating the energy harvesting mechanism with the use of a solar panel and a solar charger to increase the shelf life. It is not the goal of this thesis to create a specific hardware where it integrates all the features mentioned above or an in-depth analysis of the deployment of LoRa communication technology.

According to this, a field study was carried out and framed in the following way:

- Chapter 2: Literary review on Comparison of LoRa with other technological communication devices such as WiFi, Bluetooth, etc.
- Chapter 3: It includes the theoretical framework where the theoretical bases are presented to that support it.
- Chapter 4: The methods and procedures adopted for the development of the project, detailing the calculations and artefacts used.
- Chapter 5: The results obtained during the investigation and development time from the aforementioned procedures.
- Chapter 6: The conclusions and recommendations derived from this fieldwork.
- Chapter 7: The cost of the devices as well as the labour cost and maintenance.
- Chapter 8: Possible future lines to follow in the next projects.

Finally, a User Manual is prepared for the configuration and to the handle different environments used in this project to work properly.

2 State of the art

In a rapidly progressing world of technology, where advancements are unstoppable, the internet of things (IoT) emerges as a way of connecting different devices in order to make our lives easier and more efficient [5].

Nowadays, you can find many practical applications of this technology in agriculture, tracking, security, smart metering, smart buildings and a lot more. IoT applications have specific advantages such as long range, low data rate, low energy consumption and low cost, so technologies such as Zigbee, Bluetooth, Wifi or cellular communications do not fulfil IoT's requirements. Therefore, there is a new wireless communication called LPWAN that fulfill IoT specifications and within them technologies like Sigfox, LoRa among others.

In recent years, different studies have been carried out on LPWAN technology in which they have been tested under various conditions and applications. Some of the contributions concerning this technology are explained below.

A study conducted in the city of Oulu, Finland using a fixed base and a mobile node, differentiating between urban environment and open areas. As a result, in urban environments using a car, up to 85% of packages are received correctly over distances of up to 5 km. Between 5-10 km, 67% and between 10-15 km only 26%. While in clear areas using a boat, 69% of packages are received correctly over distances of up to 15 km, and between 15-30 km, 62%, being the maximum distance where a connection could be established [6].

Another study carried out in Prague was about two experiments conducted in a reinforced concrete building. The first placing the receiver in the basement and the second on the roof. Both experiments managed to cover the entire building, but the one on the roof has managed to cover the adjacent buildings [7].

With the experiments analysed in [6],[7], it has been shown that they have a good coverage area, ideal for applications such as smart agriculture [8]. Analysing natural changes such as temperature, humidity, light, wind speed and so on, it can cause poor growth of the crop or diseases, reducing crop yield. Therefore, this study uses an environmental sensor to collect and monitor changes in the physical world and act on them to improve the productivity of farmers.

As a result of these studies, it is concluded that the LPWAN technology encompasses many still unknown possibilities, thanks to its characteristics such as long-range, low-power and communication service platforms for monitoring and data collection.

3 Theoretical concepts and technologies used

3.1 LPWAN

Power Wide Area Network (LPWAN) or Low power network (LPN) is a wireless data transport protocol that is growing popularity in Internet of Things (IoT) implementation because of its low-power, wide-area, and low-cost communication features.

The three main attributes that match with IoT requirements are [1]:

- **Long-range**: providing communications up to 10-40 km in rural areas and 1-5 km in urban areas. It can also allow reliable communications both inside and underground locations that were previously infeasible.
- **Low-power**: Highly efficient power consumption, which transceivers can operate with small and inexpensive batteries for more than 10 years.
- **Low cost**: regulate non-constant transport in a tiny amount of data.

These features also differ from other wireless technologies such as Wi-Fi, Bluetooth, 3GPP, and Zigbee as shown in Fig. 3.1. As seen in the graph, the same technology cannot cover all the IoT requirements. LPWAN technology covers the needs of those applications that require low data transmission, fewer times a day over long distances.

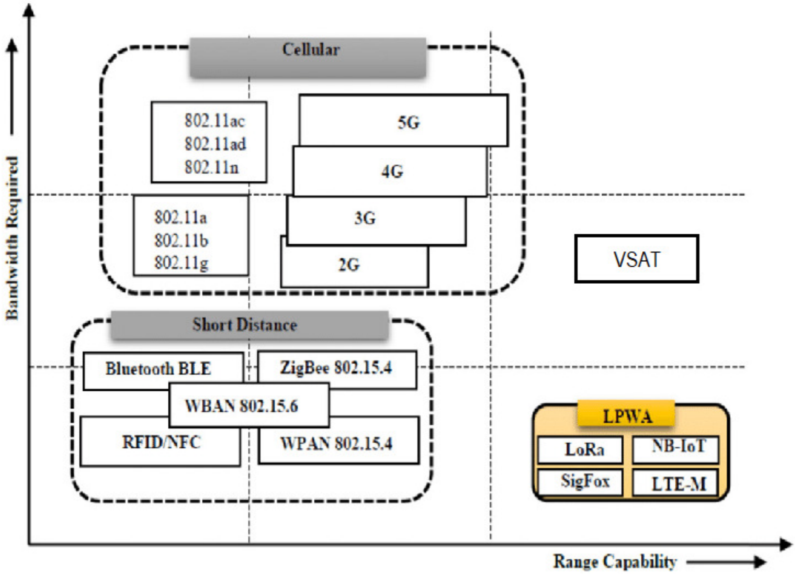

Figure 3.1: Bandwidth vs range capability of Wireless communications technologies: LPWAN [1].

Another interesting point is LPWAN technologies have emerged in the licensed and unlicensed frequency bandwidth. LoRa is one among them.

LoRa (Long-Range) was developed and acquired by Semtech. It is at level 1 of OSI Model where physical layer which modulates the signals in the unlicensed ISM (Industrial, Scientific, and Medical) band, which represents a spectrum that is internationally reserved for non-commercial use associated with the industry, science, and medical services. Thus, depending on the country, LoRa usually operates in the 433MHz, 868 MHz and 915 MHz bands. The communication P2P (Point-to-Point) is provided by the Chirp Spread Spectrum (CSS)[9], where the signal uses more width of needed, giving the communication superior robustness against noise. The Table 3.1 shows the main characteristics of LoRa.

LoRaWAN	
Modulation	CSS
Frequency	Unlicensed ISM bands (868 MHz in Europe, 915 MHz in North America, and 433 MHz in Asia)
Bandwidth	250 kHz and 125 kHz
Maximum data rate	50 kbps
Bidirectional	Yes / Half-duplex
Maximum messages/day	Unlimited
Maximum payload length	243 bytes
Range	5 km (urban), 20 km (rural)
Interference immunity	Very high
Authentication and encryption	Yes (AES 128b)
Adaptive data rate	Yes
Handover	End-devices do not join a single base station
Localization	Yes (TDOA)
Allow private network	Yes
Standardization	LoRa-Alliance

Table 3.1: LoRa overview.

LoRa is a type of P2P communication without requiring an intermediate device to manage it. But if you want to have global visibility where the devices connect to the network in the form of Mesh, the LoRaWAN protocol will be needed. In the protocol you can make configurations such as speed, channel, bandwidth, Spread Factor (SF), among others defined by the LoRa Alliance.

The LoRa Alliance is an open non-profit organization that defines and implements the LoRaWAN protocol (Level 2 OSI Model or MAC Layer) as an open global standard. You can find tech giants like Cisco, IBM, Alibaba, The Things Networks Foundation between others in the Alliance.

LoRa alliance defines a start network under the LoRA MAC Layer, where LoRaWAN ensures that the nodes transmit their data to the gateways and these to the server through an end-to-end encrypted connection. Fig. 3.2.

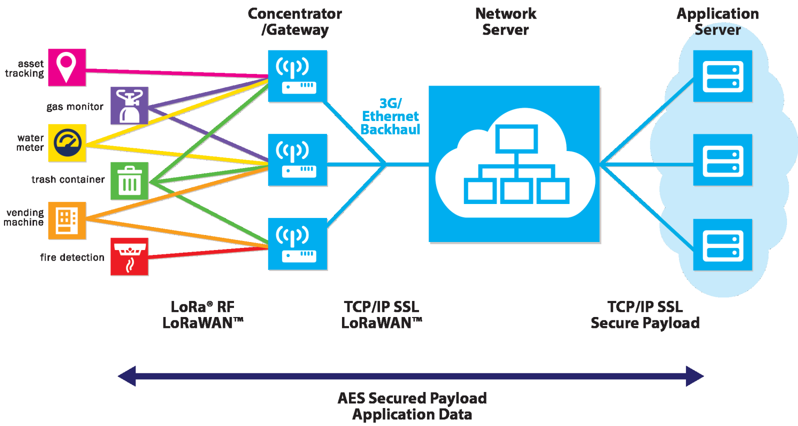

Figure 3.2: LoRaWAN network structure. (source SEMTECH)

The final devices may have different requirements depending on the application and in order to optimize this technology to the different odds of LoRaWAN devices, they are classified into different classes:

- Class A: Suitable for Low power and must be compatible with all LoRaWAN devices. They are asynchronous because it only enters the listening mode after sending data to the gateway.
- Class B: The devices have the reception windows based on predetermined times with the gateway.
- Class C: This class offers the lowest energy savings power because it is always in listening mode so that a low latency bidirectional communication channel is offered.

Fig. 3.3 shows how LoRa (PHY) and LoRaWAN (MAC) are organized in the OSI stack [2].

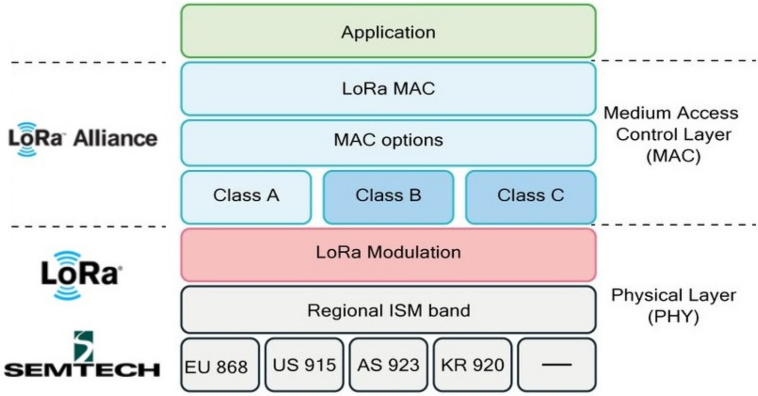

Figure 3.3: The LoRaWAN (MAC) protocol stack implemented on top of LoRa modulation (PHY) [2].

3.2 Things Networks

As Previously explained, LoRa is the LoRaWAN network modulation technology that connects to a Gateway to join the network. It allows small connected IoT devices to exchange tiny amounts of data at low speed with a long reach and low power consumption as well as the benefits of the protocol that is end-to-end encryption. To manage this whole process, The Things Network (TTN) platform emerges.

TTN is a global network of shared, user-managed and open-source IoT data [10]. It was first implemented in Amsterdam in 2015 by the founders Wienke Giezeman and Johan Stokking, providing LoRaWAN coverage and some gateways, to the city. Nowadays, the TTN has more than 10806 gateways deployed in 149 countries, in addition to being a member of the LoRa Alliance [11].

Each of the components of the LoRaWAN architecture will be detailed below from the TTN point of view, as summarized in Fig. 3.4:

- **End Devices, Node**: A device consisting of a microcontroller, a LoRa transceiver and one or more sensors connected to a battery or a power supply depending on the location and task. Also, each node has a unique identifier that will be used to register in TTN network called DevEUI. Within this branch, there are several commercial devices [12], [13] but we will focus more on LoPy4, explained in Section 3.3.
- **Gateway**: It is a device that acts as bridge between nodes and TTN. It receives broadcast messages from End Devices and sends messages back to End Devices. Useful, since a single gateway can serve thousands of devices. The gateways are usually public, and use networks like WiFi, Ethernet or Cellular to connect to TTN, but there are other several options to setup a gateway, even TTN sells a low-cost one called: The Things Indoor Gateway [14].
- **Network or Network Server**: It is responsible for routing data between nodes and applications and vice versa. For communication to be possible, the Nodes must be set up from TTN Console as well as the device sign-up using the DevEUI and the type of ABP (Activation-By-Personalisation) or OTAA (Over-The-Air-Activation) connection ensuring the operation of the network and keeping its security.

  The difference between both connections is that ABP does not need DevEUI, AppEUI or an AppKey. Instead, the NwkSkey and AppSkey keys are preprogrammed on the device and the device is previously registered on the network, so you do not need to perform a binding procedure. The disadvantage of this method is that if the attackers obtain the encryption key, they can launch Denial-of-Service (DoS) attacks against the network server and the devices as well as send fake data to the network. For that reason, OTAA configures an AES-128 root key called AppKey that is unique to each device and is used to sign the messages exchanged when a device joins a network.
- **Applications**: It is a software that allows you to manage and create the nodes by running on a server and being able to access it through an API to your final application (Angular explained

in Section 3.4). API (Application programming interface) is a set of functions that offers a certain library for use in other software.

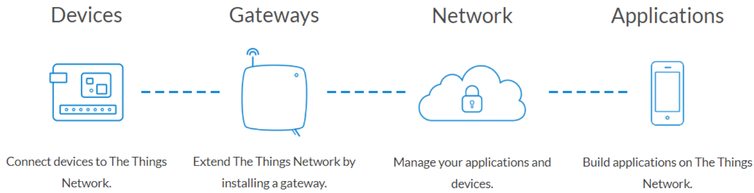

Figure 3.4: Overview TTN.

Lastly, in the network of the public community of The Things Network, they use a Fair Access Policy that limits the uplink connection time to 30 seconds per day and per node. And for downlink messages to 10 messages per day and per node. If you use a private network, these limits do not apply, but you must still comply with government limits and LoRaWAN that the duty cycle of the devices must not exceed 1% according to ETSI EN300.220 [15].

3.3 Energy harvesting and Lopy4

With the emergence of low power devices in recent years suitable for IoT with multiple sensors such as temperature, humidity and light, they have certain disadvantages that sometimes owing to the environment it is almost impossible to supply power to the device due to lack of a power outlet. Despite these constraints, the devices remain functional and fulfil their requirements for months or even years. Therefore,it is necessary to prolong the life time by applying different energy minimization techniques or providing energy harvesting mechanisms that are becoming more and more important [16], [17].

The energy harvesting mechanism is the process that captures small amounts of energy from external sources such as light, wind, humidity, heat and rain in order to improve the efficiency or taking advantage of residual heat of the heating system for example, enable new technologies, free maintenance and reduce carbon footprint.

In addition, to conserve energy, the node may go into sleep mode by disabling all its subsystems when it has no task to perform. In this study, the node is composed by a solar panel using the energy harvesting mechanism as well as Lopy4 and Pysense shown in the Fig. 3.5.

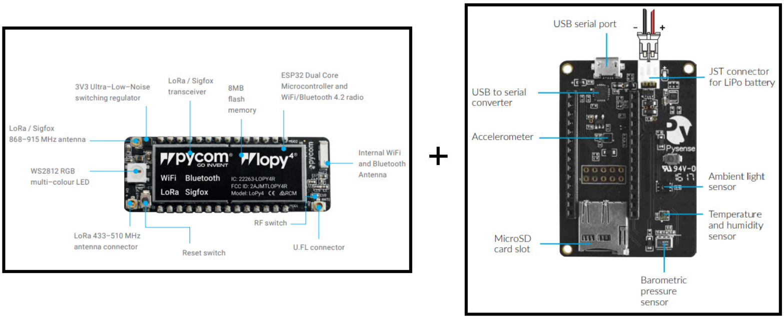

Figure 3.5: Lopy and pysense device (source Pycom datasheet).

The LoPy4 is a quadruple bearer MicroPython enabled development board (Sigfox, LoRa, Bluetooth, WiFi) to create and connect your things everywhere and fast, perfect for IoT. The best combination of implementation and access to LPWAN networks that are being deployed worldwide. However, the LoPy does not have sensors, no battery charger, nor it can be connected to the laptop via USB, so the Pysense provides the functionalities mentioned above.

Focusing on energy saving when the device applies techniques such as sleep mode, the lopy4 device consumes approximately 18.5uA [18] and Pysense around 8uA [19].

3.4 Angular

Angular is a free and open-source framework produced by Google to facilitate and optimize the creation of SPA (Single Page Applications) web and developed in TypeScript, a superset of JavaScript which essentially adds static types and class-based objects. Another purpose is the complete separation between the front-end and the back-end and a component-based architecture and good practices [20].

The main features of Angular are set out below Fig. 3.6:

- **Consistent coding**: It implies that any angular programmer can see the code written by another programmer, avoid delays in the project splitting the work among the different members, it is easy to replace a programmer or resume a Project.
- **Modular structure**: Angular organizes the code in modules that can be components, services, directives or pipes. Also, when structured into components, it makes it reusable, easy maintenance and simplified unit test.
- **Simplified MVC (Model-View-Controller) architecture**: it is a style of software architecture that separates the data of an application, the user interface and the control logic into three different components.

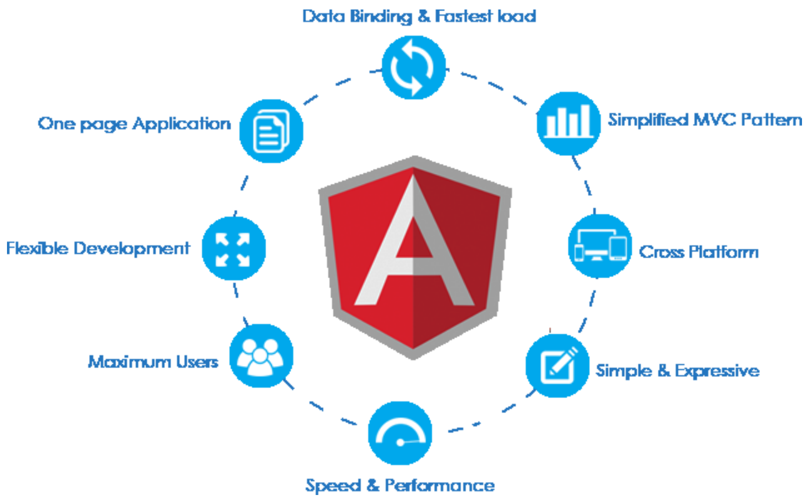

Figure 3.6: Overview Angular.

Once basic principles have been introduced, now we are totally sure about what strategy we want, how it should work and what we want to obtain using Lopy4 as a Node, TTN as LoRaWAN network server and Angular as final application to show the results.

4 Methods and procedures

Fig. 4.1 displays the main components of the whole system. As it depicts it can be divided into 3 different parts. The first one has the objective of gathering temperature, humidity, light, and battery status data to be transmitted by LoRa. The second one aims to receive the data sent by the node (Lopy4) to the nearest gateway using the LoRaWAN network deployed by the TTN platform and finally, the third one has the functionality to process the data collected by the nodes and compare them with the weather forecast in a graphic way.

Figure 4.1: Technologies used to process data, perform LoRa communication as well as energy saving methods.

The following Fig. 4.2 shows in detail the flow diagram of the entire system that will be explained in the sections below.

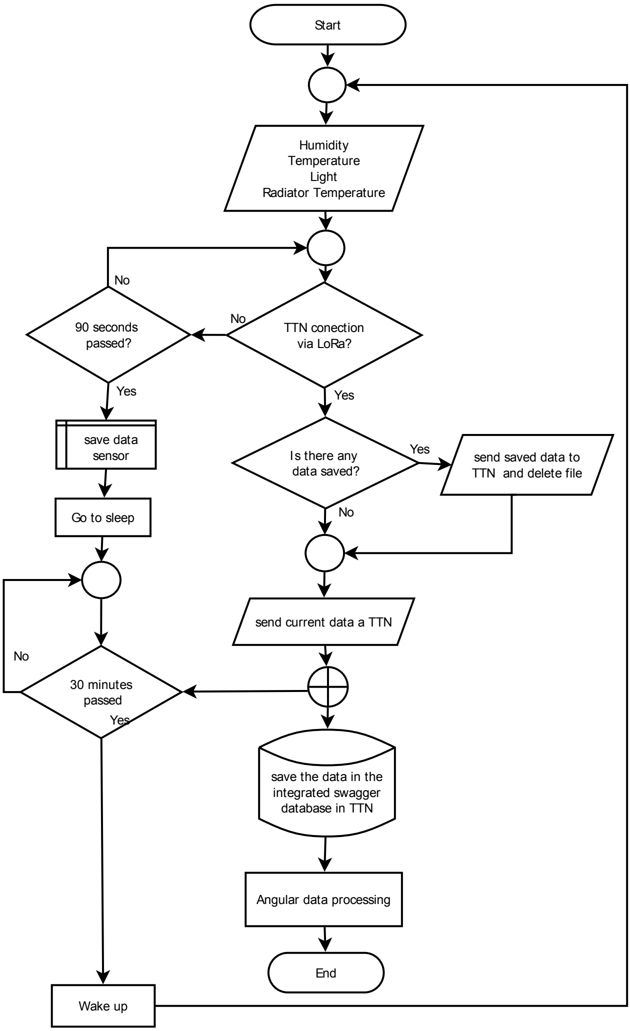

Figure 4.2: Flow diagram of the entire system.

4.1 Data collection

Before getting into detail, lets look into a brief description of the device. Fig. 4.3 shows four components. We will focus on the module that integrates with the Pycom called Pysense. This little fellow contains several sensors such as determining the ambient light, humidity and temperature (-45ºC and 85ºC) [19] of the room as well as the state of the battery, among others.

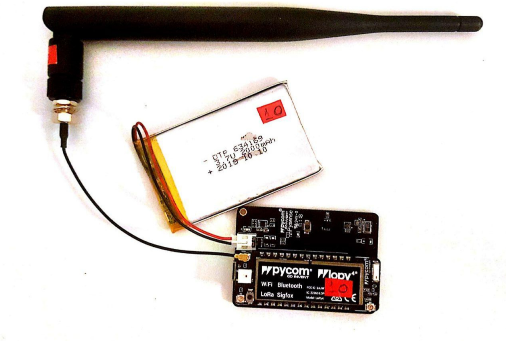

Figure 4.3: Node elements.

To have better reliability of temperature measurements and their variance, an extra sensor has been added to the device. The DS18B20 sensor can measure temperatures between -55ºC and 125ºC [21]. It is a very wide range; however, it varies depending on the range. Temperatures between -10ºC and 85ºC we can have ±0.5ºC. For the rest, the error is ±2ºC.

Before installing, you have to always put a pull-up resistor with the DQ pin. The reason for this resistance is for the electronic system to control the communication bus. The value will depend on the length of the cable. As a rule, a 4.7kΩ is always used, since the cable is less than 5 m long. The placement of the sensor on the Pysense board is reflected in Fig. 4.4

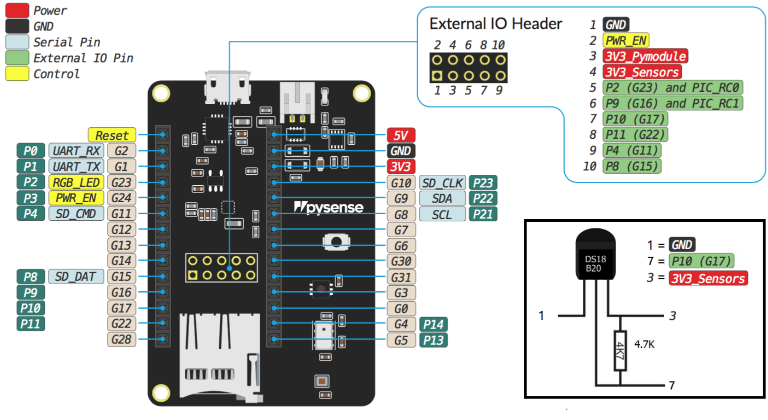

Figure 4.4: Outline how to configure the DS18B20 sensor on the pysense board.

Another interesting fact would be the weather forecast to observe and compare how it influences the readings taken by the node sensors. Therefore, the forecast capture is done on the application side (Angular) through an API. In our study, the most relevant data that can influence the behavior of node readings are the humidity and temperature taken from the Meteostat API as well as solar radiation through the Solcast API.

4.2 Communication and data processing

Once it is clear that data are the most relevant to the system, communication to the TTN platform will proceed through LoRaWAN technology. For communication to be possible, you will need to get the **app_ eui** and **app_ key**, registering the node and choosing the type of encryption on the TTN console. The registration process of the node is found in the user manual in the Appendix A.

Now, with the authentication parameters, the device can send the data to TTN. As shown in Fig. 4.2, After every 30 minutes the device wakes up from sleep mode, tries to connect to the LoRaWAN network of TTN. After 90 seconds, if the connection have not yet established, it saves the data and waits for the next successful connection and can send the stored data as well as the current one.

This does not guarantee the message reaches TTN since LoRAWAN has certain limitations. The main one is that we cannot transmit more than 1% of the time, that is, on average there are 30 seconds of uplink in the air, per day, per node. Another limitation is that there is a maximum of 10 downlink messages per day, including confirmed uplink ACKs. Extrapolating to this study, there may be a successful connection, but not receive the package, since if we send packages every 30 min, the ACK can not be implemented to confirm that the packages were correctly received to TTN. The next question would be how we encode and decode the package.

On the encoding side (node), the values collected from the sensors, $float$ type, have been converted to type int and multiplied by 100 to better process the data and limit to two decimals in the decoding as shown in the Fig. 4.5 below.

```
byte_value = bytes(struct.pack("hhhhh", int(tempe*100),int(wet*100),light[0],
int(temp_heater*100),int(battery_voltage*100))) # convert float to int
```

Figure 4.5: Encoding data to bytes for transmission with Lora protocols.

The $struct.pack(format, v1, v2, \dots)$ function is a module capable of performing conversions between python and C. So it returns a string containing the values $v1, v2..$ that are packaged according to the given format. In this case, the letter h represents the short format in C.

The $bytes()$ function returns an immutable byte object initializing with the given size and data.

On the decoding side (TTN), we receive the data as a byte object through the corresponding port. As shown in Fig. 4.6, 8-bit shifts are made to obtain the data and then divided by 100 to convert it to float.

```
function Decoder(bytes, port) {
 var temperature = (bytes[1] << 8) | bytes[0];
 var humidity = (bytes[3] << 8) | bytes[2];
 var light = (bytes[5] << 8) | bytes[4];
 var heater = (bytes[7] << 8) | bytes[6];
 var battery = (bytes[9] << 8) | bytes[8];
 return{
  temperature: temperature/100,
  humidity: humidity/100,
  light: light,
  heater: heater/100,
  battery: battery/100
 }
```

Figure 4.6: Decoding data on The Things Network platform

With data in TTN, the platform is full of integrations with external IoT platforms to synchronize the device register and uplink and downlink data. Therefore, you don't need to write code or use TTN console. The external platform to be used will be Swagger, where the data will be stored for seven days, accessing them through an API.

The data of the nodes available in the API as well as the forecast weather APIs, the application is created in Angular to show the results obtained and draw conclusions as displayed in the Fig. 4.7.

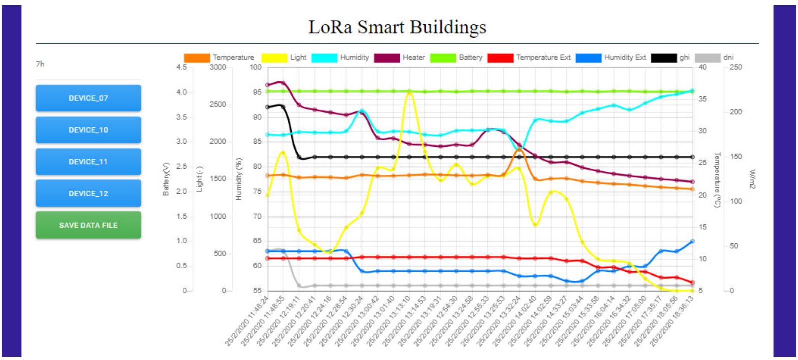

Figure 4.7: Angular application where the final data is displayed graphically.

4.3 Consumption analysis

As seen in the previous Fig. 4.7, the graph reflects the state of the battery. This is a fact to take into account when knowing the status as well as the battery change. The Fig. 4.8 shows the type of battery and the Table 4.1 its characteristics.

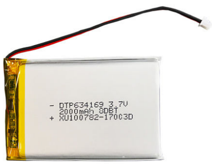

Figure 4.8: Battery.

Battery	
Model NO.	DTP634169
Rechargeable	Yes
Battery connector	Jst 2p/Molex
Nominal Capacity	2000mAh
Nominal Voltage	3.7 V
Cell Type	Lithium-Polymer Battery
size	6.5x41.5x71 mm

Table 4.1: Battery features.

Known the battery characteristics, the average current consumption of the device during a cycle will be calculated, being one cycle about 30 minutes $ON + OFF(Sleep\ mode)$.

To find the average of the current in a cycle, the average value theorem for integrals can be applied as shown in the Fig. 4.9. If a function is continuous in a closed interval $[a, b]$, there is a point c inside the interval such that (4.1)

$$f_{avg}(c) = \frac{1}{b-a}\int_a^b f(x)dx. \tag{4.1}$$

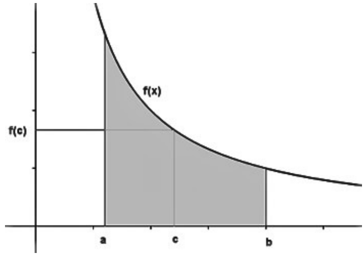

Figure 4.9: Graphic representation of the average value within the closed interval.

In this study, the INA219 DC current sensor and an Arduino due were used to measure the current consumption of the node in the different operating phases. The INA219 sensor Fig. 4.10 measures the voltage through the shunt resistor of 0.1Ω. This means that it can measure up to ±3.2 Amps with a resolution range of 0.8mA. To obtain the measurements, the Arduino due Fig. 4.11 will be configured and the INA219 chip will be connected to the power and the I2C bus.

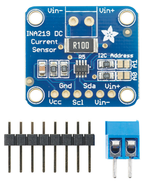

Figure 4.10: INA219 sensor.

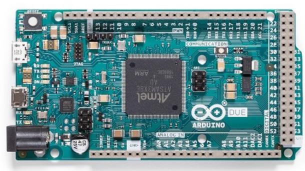

Figure 4.11: Arduino board.

The disadvantage of the Arduino is that it cannot save the captured sensor data locally. It is necessary to install software called CoolTerm on the computer to create this .txt file. The data obtained is dumped into Excel and then processed in Matlab.

With this data, we can analyze the node consumption and compare with the data provided in the manufacturer's datasheet shown in Table 4.2 as well as the average current value.

Mode	**Average**	**Units**
Idle (no radios)	35.4	mA
LoRa Transmit	108	mA
Deep Sleep	18,5	μ A

Table 4.2: Consumption according to supplier datasheet.

With the average value of the current, the battery life is calculated according to the nominal capacity and the current. The estimated result may differ from the real value since the actual result vary according to the condition of the battery, the age, the temperature, the discharge rate and other factors hence the factor 0.7 of the formula (4.2).

$$Estimated\ battery\ life = \left(\frac{Battery\ capacity\ (mAh)}{device\ current\ (mAh)} \right) Factor. \tag{4.2}$$

The steps taken so far served to know the consumption of the device as well as the estimated device lifetime. The next step is to prolong the life of the device by applying energy harvesting mechanisms.

The mechanism applied will be a solar panel with its corresponding solar manager where it will be analyzed how much extra energy can be generated and if it is possible to apply this method in a closed environment, in this case, a room. Below are the components to be used in this study Fig. 4.12 , Fig. 4.13 as well as their main characteristics Table 4.3 , Table 4.4.

Figure 4.12: Solar Panel.

Solar Panel	
Model NO.	CNC110X60-6
Maximum Power Voltage	6 V
MAximum Power Current	167 mA
Power peak	1 W
Open-circuit Voltage	7.3 V
Solar cell	Polycrystalline
size	60x110x3 mm

Table 4.3: Solar Panel features.

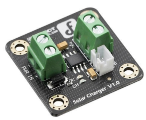

Figure 4.13: Solar Manager.

Solar Manager	
Model	DFR0264
Input PWR	4.4 to 6 V
Max Current	500 mA
VOUT	5V / 1 A
Max. Quiscent current	-
size	33x33x12 mm

Table 4.4: Solar Manager features.

Once the devices are known, the energy that can be produced with this solar panel is estimated through the solar irradiation data provided by the European Commission [22].

The parameter GHI (Global Horizontal Irradiation) is chosen, which is the sum of direct normal irradiation (DNI) and diffuse horizontal irradiation (DHI)

$$GHI = DNI + DHI$$

shown in Fig. 4.14. It is a measure of the amount of solar energy it receives in a specific place. This irradiation varies throughout the year as well as the day depending on the season, the position of the sun in the sky and the weather.

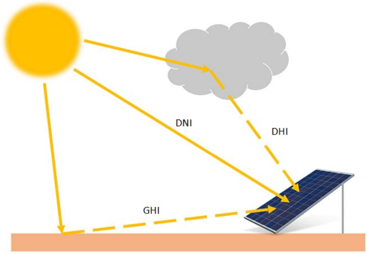

Figure 4.14: behaviour of solar irradiation in front of a Solar Panel.

Now, with the defined irradiance and the chosen parameter, the data for Tierpark, Berlin location from 2005 to 2016 are downloaded, as shown in the following Table 4.5.

Global Horizontal Irradiation													
Month	**2005**	**2006**	**2007**	**2008**	**2009**	**2010**	**2011**	**2012**	**2013**	**2014**	**2015**	**2016**	**Average**
January	16,1	24,16	18,56	17,3	19,28	21,05	20,44	20,35	16,55	19,34	17,91	18,02	19,09
February	29,95	25,1	33,08	41,48	26,11	33,42	41,94	39,52	22,81	44,52	46,41	36,72	35,09
March	71,22	61,97	88,47	75,31	63,04	75,17	93,84	85,97	67,28	88,1	77,15	62,09	75,8
April	140,39	103,12	149,43	107,68	162,43	132,57	135,6	121,8	106,59	125,5	125	120,67	127,57
May	144,93	156,73	164,97	183,2	160,88	104,41	180,51	164,72	123,41	139,92	157,35	174,8	154,65
June	172,27	176,47	160,74	182,54	146,55	180,85	177,12	144,13	161,7	159,93	164,07	174,67	166,75
July	143,89	202,99	143,23	148,3	160,98	186,07	131,63	155,08	176,19	174,27	167,85	155,19	162,14
August	140,42	122,45	136,19	115,74	155,65	117,92	131,86	144,63	148,03	140,86	167,5	143,45	138,73
September	108,9	112,98	89,16	82,64	110,9	85,9	103,06	105,12	76,45	92,73	98,42	117,55	98,65
October	72,36	59,92	54	49,95	43,46	61,42	65,29	59,12	50,95	53,53	56,53	40,39	55,58
November	27,3	27,09	21,27	18,85	25,67	20,31	37,3	21,94	19,7	24,18	24,12	26,55	24,52
December	13,62	15,83	14,47	14,19	13,73	16,85	16,09	15,09	16,21	14,77	20	18	15,74

Table 4.5: Global Horizontal Irradiation $[Kwh/m^2]$ data monthly between 2005 and 2016.

The graph in Fig. 4.15 shows the average monthly irradiance of the last 11 years, which corresponds to the last column of the Table 4.5.

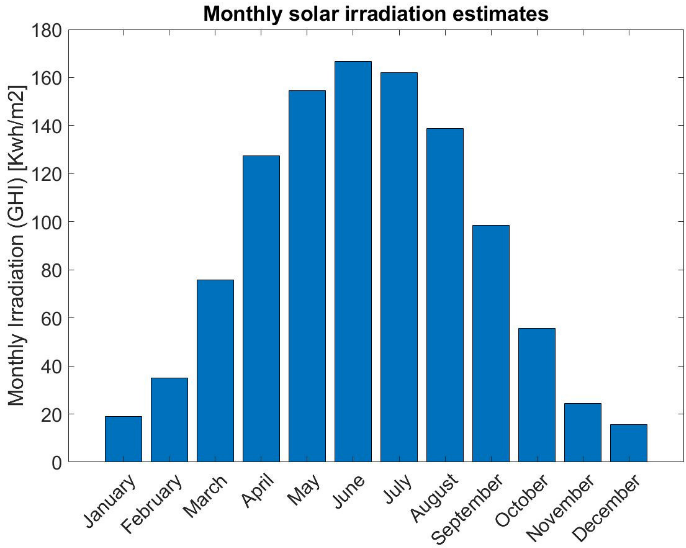

Figure 4.15: Average Global Horizontal Irradiation between 2005 and 2016.

With the data obtained, the theoretical calculation of the energy produced by our solar panel for the month of February will be made with the following formula (4.3) :

$$E = A\, r\, H\, PR, \tag{4.3}$$

where:

E = Energy (kWh).
A = Total solar panel Area (m^2).
r = solar panel yield or efficiency(%), $r = \left(\frac{Power\, peak\, of\, solar\, panel}{Area}\right)$.
H = Monthly average solar radiation (shadings not included).
PR = Performance ratio, coefficient for losses (range between 0.5 and 0.9, default value = 0.75).

And with the characteristics of the solar panel, it is known that the energy produced in a month by our system is of

$$E = 0.0066 * 15.15 * 35.09 * 0.75 = 0.0263\ kWh/m^2/month.$$

5 Results

Once the procedures and the methods used have been explained, the results obtained will be analyzed and examined if they are consistent with the theory. In addition, all tests have been performed in the Lichtenberg district close to Tierpark, Berlin.

5.1 Pycom consumption and battery lifespan

The Fig. 5.1 shows the current consumption every 3 milliseconds in a node duty cycle. The work cycle is formed by the sleeping mode that consumes 375 uV as seen in the data tips for 30 min and an active mode that is responsible for data collection and transmission.

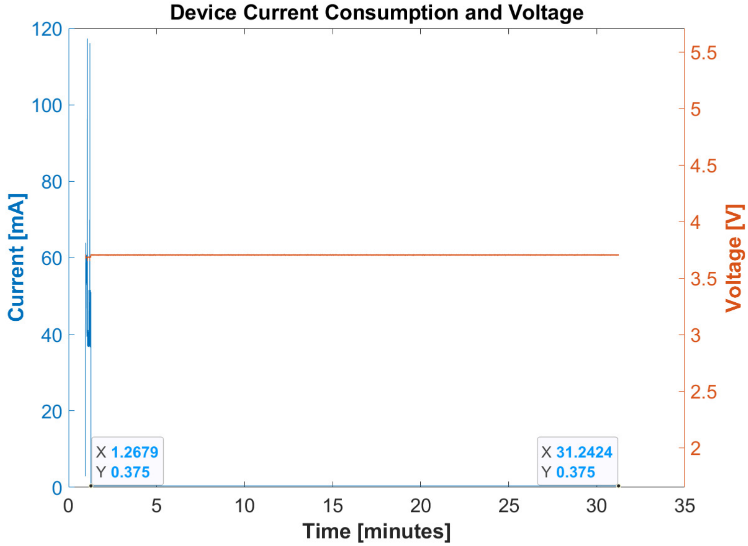

Figure 5.1: Duty cycle of the node. ON and OFF (sleep mode).

The Fig. 5.2 decapitates the ON mode of the node cycle which lasts 18 seconds although this value can be increased up to 108 seconds if it fails to connect to TTN. The figure shows different phases of consumption: The first is the library load when you wake up. The second phase tries to connect to TTN and it can last from 2.5 seconds to 90 seconds. The third is the transmission of data and finally, the last indicates that the data has been sent.

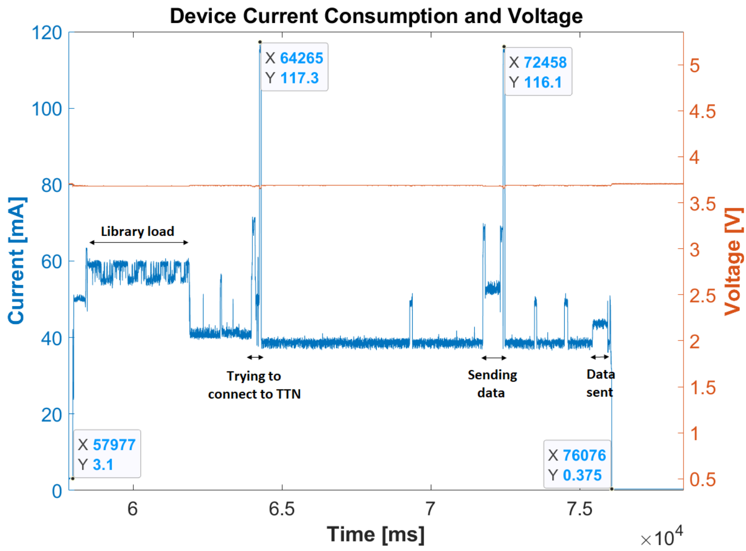

Figure 5.2: Device ON.

The Fig. 5.3 shows two consumption behaviours by zooming in the sending data section. The first is due to an LED indicating that the connection was established and lasts 0.5 seconds. The second consumption is the transmission of the data for 57 milliseconds.

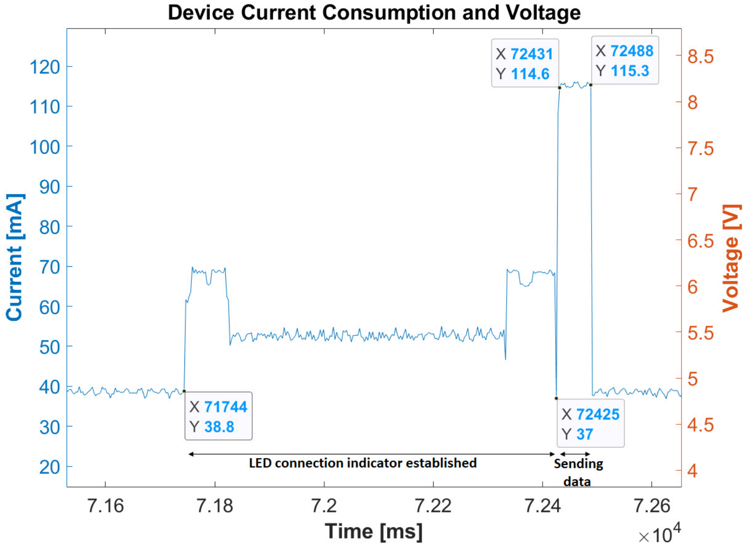

Figure 5.3: Sending data.

Calculating the average value of the current consumption in the different operation phases of the node, it is concluded that the sleep mode is much higher than that indicated in the pycom datasheet as shown in the following Table 5.1 .

Mode	Theoretical Avg.	Real Avg.
Idle (no radios)	35.4 mA	44 mA
LoRa Transmit	108 mA	115 mA
Deep Sleep	18,5 μA	375 μA

Table 5.1: Comparison of practical and theoretical consumption of the node.

Once the device's behavior is analyzed, the lifetime of the node will be calculated. Taking into account the characteristics of the battery reflected in Table 4.1 (2000mAh) and applying equation(4.1) the average current value in a cycle, the average consumption of the node is 0.7 mA. With these two data, the lifespan of the node can be calculated using equation (4.2) obtaining a value of 71 days as shown in the following Table 5.2 .

Battery Lifespan	
Battery capacity (mAh)	2000
Consumption rate	0.7
Avg Current Lopy(mA)	0.81
Estimated hours	1717
Estimated days	**71.54**

Table 5.2: Estimation of battery lifespan.

The results shown so far uses its own gateway (the Things Indoor GAteway TTIG), since if the ones deployed by The thinks networks are used, the connection attempt could be failed or could be extended up to 90 seconds, consuming more energy and reducing the useful life of the node at only 30 days. In addition, many times the data did not reach the receiver even though a connection was established.

5.2 Solar Panel energy

Applying the energy harvesting mechanism we have obtained the following results.

The solar panel and the solar charger is located inside the room with a window facing East. The amount of energy generated will vary depending on the season as well as the weather.

In our study, the energy produced on two different days was analyzed. The first day was on March 10, 2020 from 6:15 to 18:15 was monitored with a reading interval of 30 seconds on a cloudy day with 7°C displayed Fig. 5.4 with its estimates of lifespan battery reflected in Table 5.3

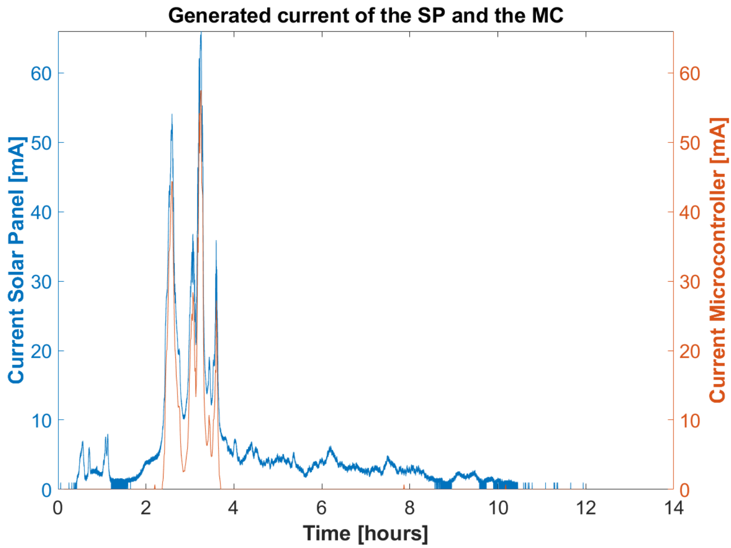

Figure 5.4: Current generated by the solar panel and Solar charger on March 10, 2020 from 6:15 to 18:15 on a cloudy day.

Increase battery lifespan	
Battery Voltage	3.7
consumption rate	0.7
Lopy Avg Current(mA)	0.81
Avg current Solar charger	1.967
Avg current Solar charger (1 day)	0.981
Avg power mW (1 day)	3.631
Energy (Wh\day)	0.0036
Incre battery capacity (mAh)	0.9814
Lifespan battery (hours)	**0.848**

Table 5.3: Daily useful life estimate for March 10.

and the second day was on March 14, 2020 from 6:15 to 18:15 was monitored with a reading interval of 30 seconds on a sunny day with 13ºC displayed Fig. 5.5 with its estimates reflected in Table 5.4.

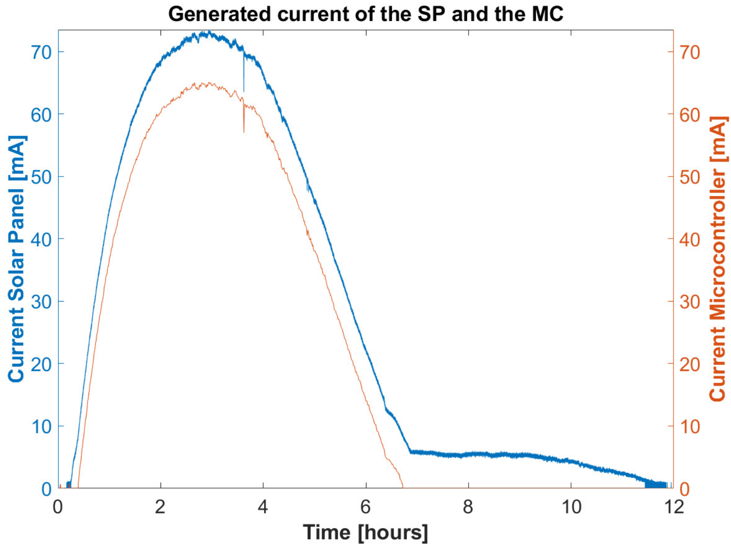

Figure 5.5: Current generated by the solar panel and Solar charger on March 14, 2020 from 6:15 to 18:15 on a sunny day.

Increase battery lifespan	
Battery Voltage	3.7
consumption rate	0.7
Lopy Avg Current(mA)	0.81
Avg current Solar charger	22.416
Avg current Solar charger (1 day)	11.205
Avg power mW (1 day)	41.46
Energy (Wh\day)	0.0414
Incre battery capacity (mAh)	11.205
Lifespan battery (hours)	**9.683**

Table 5.4: Daily useful life estimate for March 14.

The graphs show the current generated by the solar panel (blue line) and the current supplied to the battery after the solar charger (orange line). By analysing both lines, it is concluded that you have to overcome a certain threshold of current generated to charge the battery that is around 8 mA. Otherwise the solar charger consumes on average 5 μA.

In both tables mentioned above, the first two parameters *Battery Voltage* and *consumption rate* are constant and the third one *Lopy Avg Current (mA)* is 0.81 mA. This last data is obtained from the analysis of the consumption of the device reflected in Table 5.2 of Section 5.1 that will be necessary to calculate how much longer the useful life of the node can be extended by applying the energy harvesting mechanism.

To calculate the total lifespan of the device, the sum of the geometric series will be applied as shown below (5.1):

$$\sum_{n=1}^{N} a_1 r^{n-1}, \tag{5.1}$$

where:
$a1$ = Useful life of the node without energy harvesting(days) that in this study is 71.54 days as reflected in Table 5.2.
$r = \frac{Useful\, life\, gained\, in\, hours}{24\, hours}$, and
N = Number of consecutive days with good harvest.
If N tends to infinity, the geometric series converges to (5.2)

$$N_{days} = \frac{a_1}{1-r}. \tag{5.2}$$

Therefore, the device will last 74.16 days for cloudy weather with the Table 5.3 and 119.92 days for sunny day with the Table 5.4 applying the (5.2).

The results shown are with very bad weather conditions or the opposite. If both values *Lifespan battery (hours)* reflected in the tables are averaged

$$Lifespan\, battery_{avg} = \frac{9.683 + 0.848}{2} = 5.265,$$

the total lifespan of the device would be 91.64 days.

5.3 Data analysis

The data sensors gathered have been checked with an indoor thermometer to fulfil the manufacturer's specifications of the devices explained in Section 4.1.

Once the data is verified, the next procedure is to analyse the behaviour of variation in the data due to various factors such as heater, number of people, windows and so on. The following Fig. 5.6 Fig. 5.7 show the data taken for a week from February 28 to March 6.

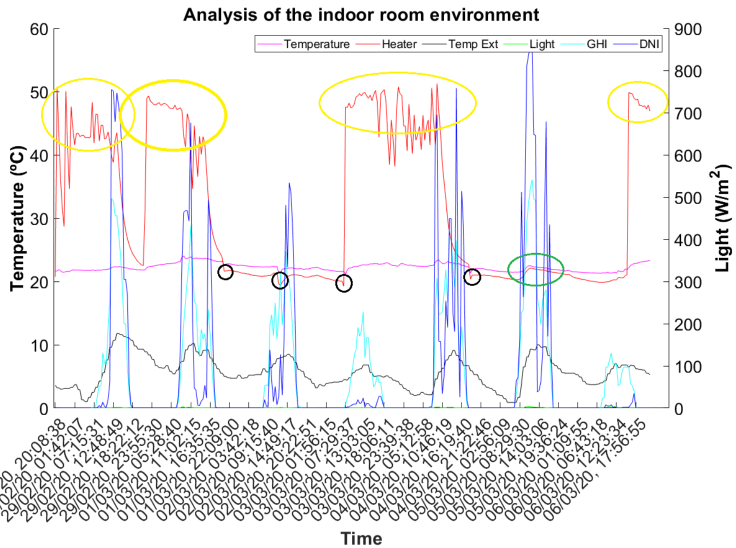

Figure 5.6: Temperature and solar radiation analyzed from February 28 to March 6 in an indoor room.

The Fig. 5.6 displays temperatures from different scenarios as well as the solar irradiance that it explained below:

- The multiple blue peaks (GHI and DNI) represents the solar irradiance that match with the seven days analysed. The first three peaks shows partially cloudy day, the next day was mostly cloudy following two sunny days and finally in the last peak we can see a totally rainy day. An slight increase in the room temperature is noted during the sunny days highlighted by green circle.
- The four yellow circles reflects the temperature of the heater when it is active which also led to an increase in the room temperature.
- A drop in temperature is seen while the windows were opened for a short period of time (20 min) highlighted by black circles.

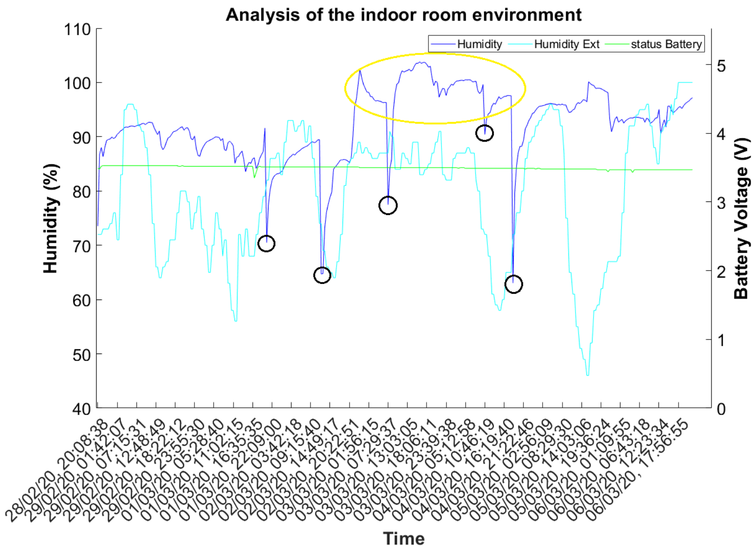

Figure 5.7: Humidity and battery status analyzed from February 28 to March 6 in an indoor room.

The Fig. 5.7 displays the humidity as well as the battery status of the device.

- As mentioned before, when the windows were opened for a short period of time, not only the temperature were dropped but also the humidity pointed by black circles.
- Finally, a major fluctuation seen in humidity highlighted by yellow circle is because wet clothes were kept near the heater during the day which led to an increase in humidity.

6 Conclusion

In this project, the behaviour of a room has been analysed through sensors such as temperature, humidity and light from low-consumption commercial devices with LoRa technology through the TTN platform and use it as a future line in the efficient management of energy consumption in home or community.

During this work it has been seen that Lopy4 and pysense have good accuracy when gathering sensor data every 30 min but it does not fulfil with the manufacturer's specification requirements of sleep mode. In addition, an own gateway was installed instead of using those deployed by TTN, since it increased the lifespan of the node and almost no data was lost in the transmission by not implementing ACK in LoRa.

Together with the node, a solar panel is added to implement the energy harvesting mechanism to improve the life of the system showing that the solar charger only consumes 5μA. To Apply energy harvesting, a number of factors must be taken into account such as location, orientation, as well as seasons and weather. During a sunny day we can recharge up to 4 hours per day but during a cloudy day almost an hour.

From the analysis of the data gathered we can see how different scenarios such as opening windows, turning the heater or drying the clothes inside the room impact the real data. The following Table 6.1 shows a comparison of the results obtained with the desired or theoretical ones.

6 Conclusion

Features	Theoretical	Real
Node		
Idle (no radios)	35.4 mA	44 mA
LoRa Transmit	108 mA	115 mA
Deep Sleep	18,5 μ A	375 μ A
Temperature sensor accuracy	[-10ºC,85ºC] $\pm$0.5ºC Others $\pm$2ºC	Meet requirements
Price	Costly	Costly
Gateway		
Using an own gateway improves the useful life and the data received		
Energy harvesting		
DFR0264	Consumes up 5 μA in standby mode and increase the battery lifespan between 0 to 4 hours in winter depending on the weather	

Table 6.1: Comparison of the results obtained with the expected.

7 Budget

The cost of the project will take into account the price of the material, labor and maintenance.

First,the total unitary cost without taxes of the whole system and of the measurement components is 198.10 € as disclosed in the Table 7.1.

Components	units	Unitary cost	VAT (Germany 19%)	Unitary Cost + Taxes	Total Cost
					Node / device components
Lopy4	1	34,95 €	6,64 €	41,59 €	41,59 €
Pysense	1	24,95 €	4,74 €	29,69 €	29,69 €
LoRa/Sigfox Antenna	1	9,00 €	1,71 €	10,71 €	10,71 €
Battery	1	4,41 €	0,84 €	5,25 €	5,25 €
Solar Panel	1	5,21 €	0,99 €	6,20 €	6,20 €
Solar Charger DFR0264	1	6,42 €	1,28 €	7,70 €	7,70 €
Temperature sensor DS18B20	1	3,03 €	0,58 €	3,61 €	3,61 €
Total		**87,97 €**	**16,78 €**	**104,75 €**	**€ 104.75 €**
			Gateway		
The Things Indoor Gateway(TTIG)	1	79,50 €	15,10 €	94,61 €	94,61 €
Total		**79,50 €**	**15,10 €**	**94,61 €**	**94,61 €**
			Measurement components		
Wires	1	5,04 €	0,96 €	6,00 €	6,00 €
INA219 DC current	1	8,87 €	1,69 €	10,56 €	10,56 €
Arduino Duo	1	16,72 €	3,18 €	19,90 €	19,90 €
Total		**30,63 €**	**5,83 €**	**36,46 €**	**36,46 €**
Total Cost		**198,10 €**	**37,71 €**	**235,82 €**	**235,82 €**

Table 7.1: Budget of the project.

Second, the salary of the worker shall be taken in reference to the minimum interprofessional salary of Germany which is 1557 €/month. Therefore, the total wage of the worker during the 4 months has been 6228 €

Finally, the maintenance price would simply be the battery change that is 4.41 € without fees.

8 Future development

This Project has enough potential to develop new ways to save energy without making high investments in infrastructure. Some additional future guidelines to implement may be:

- Based on the background of a project prior to this one where as of the public information that the Berlin administration has about the structures of the buildings, theoretical models were created to predict the temperature. A future line would be the adjustment of theoretical models based on the data obtained in order to achieve a valid simulation system that can be applied to other buildings.
- Create an app where it alerts the user to enable energy saving actions based on weather conditions and the building design.
- Design an integrated device to improve functionalities such as sleep mode, energy harvesting to increase the life.

A User Manual

This project is based on three main components that are the configuration of the device, registration of the nodes in the TTN platform and representation of the data on the website. Therefore, a user manual can come in handy to know how to use the different environments to make the service to work as expected.

The attached software with this project allows the reader to access the code.

A.1 Node settings: Pycom

This section will help you get started using the Pycom device. There are various editors used for coding, but I recommend installing Atom text editor because it is simple and intuitive. To move forward, please follow the below steps:

1. Install the latest version of the Atom from the following link: https://atom.io/

2. Open the Atom editor and navigate to Help ⇒ **Welcome Guide** ⇒ **Install a Package** ⇒ **Open Installer**.

3. Search for **Pymark** and select the official Pycom Pymakr Plugin. You should see and click the Install button, as shown in the Fig. A.1. After the installation, you need to restart the Atom editor.

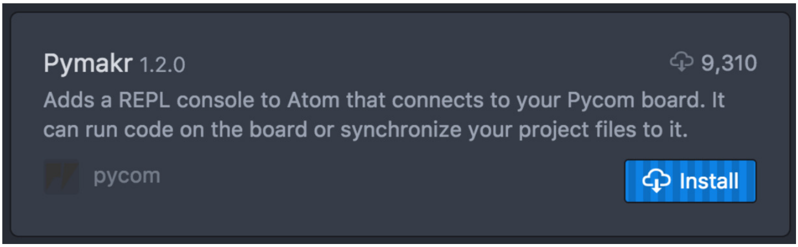

Figure A.1: Pycom Pymark Plugin.

You can connect your Pycom device to your computer via USB. You should ensure that the Plugin has correctly installed as you can see in the Fig. A.2 below.

Figure A.2: Atom terminal.

Now, you can use the code and their libraries attached to this project to run it on the Pycom. Some tips: if you are going to use LoRa communication, please connect the external antenna on the correct pin (in Europe 868MHz) before executing shown in the Fig. A.3.

Figure A.3: antenna placement at frequency 868MHz.

To know more about using Pycom, please visit the following link: https://docs.pycom.io/

A.2 Network settings: The Things Network

In order to use TTN, you have to go to The Things Network website and create an account. Once you have signed up, you can now register your Pycom module or your Gateway.

1. Sign up the Node

   To connect the device into the things network, you must first create an application for these devices that they belong to. Doing this, the network will send the device data to its desired locations.

   Select **Console** from your profile tab at the top of the screen, click **Applications** and then **add application**. Fill out the necessary details Fig. A.4

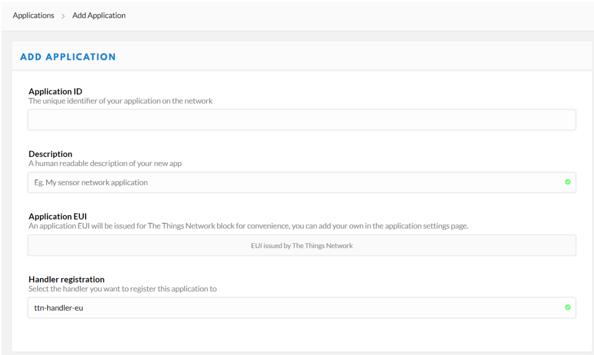

Figure A.4: Application registration on TTN.

The application ID entered should be unique and give a suitable description. Once it is done, navigate to the **Device** tab on the **Application** you have created and click the **Register Device** button and complete the details shown in the Fig. A.5.

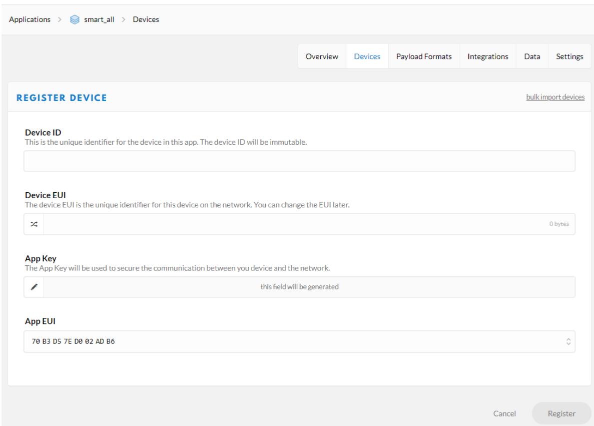

Figure A.5: Node registration on TTN.

To know the Device EUI, you should execute the next code on your pycom module:

$From\ network\ import\ LoRa$
$Import\ ubinascii$
$macadr = binascii.hexlify(LoRa(mode = LoRa.LORAWAN, region = LoRa.EU868).mac())$
$print('MAC : '.format(macadr))$

Once the device has been added, change the **Activation Method** to **OTAA**. This option can be found under the **Settings** tab. When the configuration is completed, the **App Key** and **App EUI** that you can see in Fig. A.5 need to be added in the Pycom code.

In order to use the data in Angular, you will have to configure the database that integrates TTN. Go to **Applications** ⇒**name of your application** ⇒ **Integration** ⇒ **add Integration** ⇒ **Data storage**.

2. Sign up the Gateway

   This step is not entirely necessary if you are using the antennas that implement The Things Networks in the city.

   In this study, I am using The Things Indoor Gateway because it has seen an improvement in the connection between the device and gateway as well as the battery lifespan. The following link shows how to configure this specific gateway:
   https://www.thethingsnetwork.org/docs/gateways/thethingsindoor/

   Select **Console** from your profile tab at the top of the screen, click **Gateways** and then **Register**. Fill out the necessary details shown in the Fig. A.6.

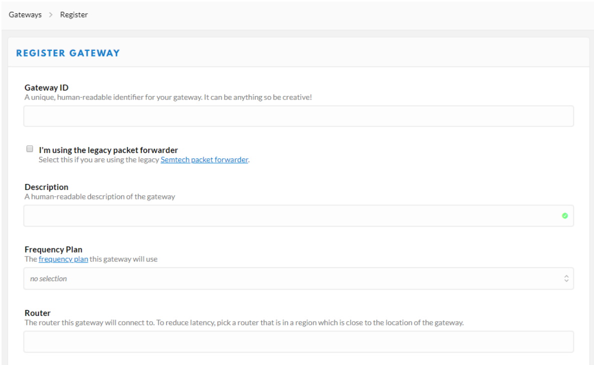

Figure A.6: gateway registration on TTN.

A.3 Application settings: Angular

To process the collected data, Angular will be used to facilitate and optimize the creation of a website. Before installing Angular, you need to install previously:

1. Install the latest version of Visual Studio Code from the following link: https://code.visualstudio.com/
2. Install the latest version of NodeJS from the following link:

   https://nodejs.org/es/
3. Open Visual Studio Code and write the following instruction in the console to know if NodeJS was installed correctly.

$$Node - v$$

4. To install angular run in the same terminal the following command is used:

$$npm\ install\ \breve{}g\ @angular/cli$$

   In this way, the latest stable version of Angular will be installed in your system.

Now, you can open the code folder attached to this project by clicking on **File** ⇒ **Open folder** shown in the Fig. A.7.

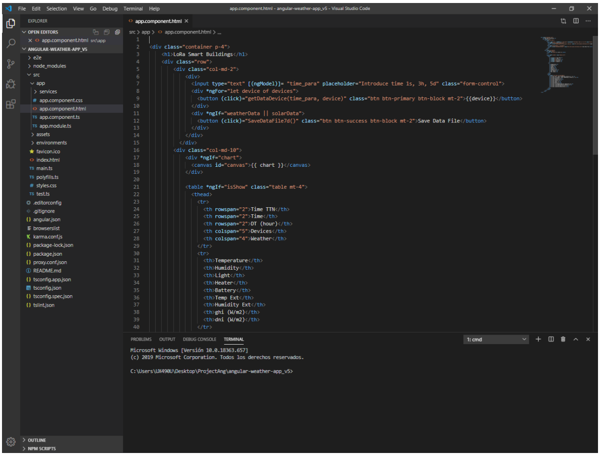

Figure A.7: view of the Visual Studio Code editor

Use the terminal to launch the local server for Angular with the following command:

$$npm\ start$$

and now we will be able to view it in the browser with the route **http://localhost:4200/**

Angular provides a way to create a new project with a fairly comfortable and practical structure. The structure is generally defined according to the complexity of the project, bigger, then more order requires.

I do not intend to enter into theoretical concepts, since there is very good information on the web that clearly explains these aspects, simply a help of how it is organized. Fig. A.8 represents how this project would be organized in a simple and easy to understand way.

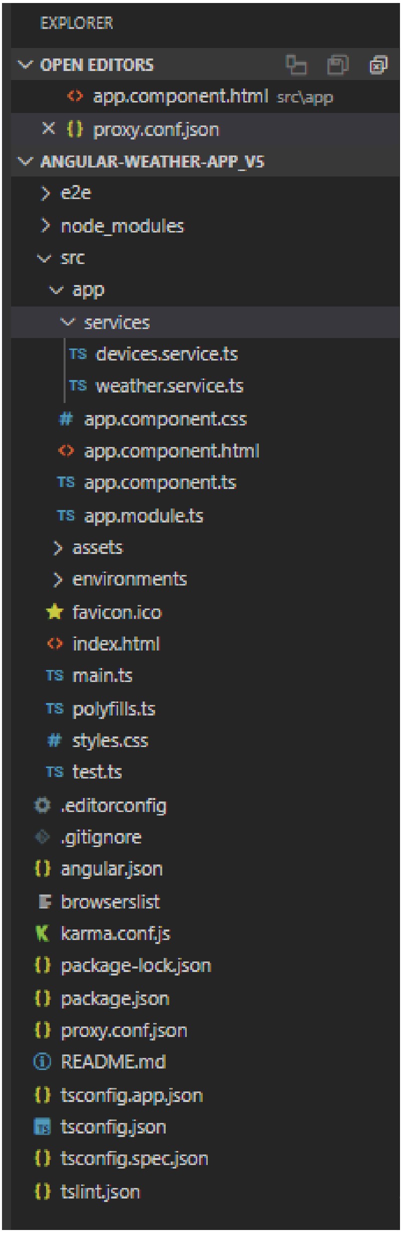

Figure A.8: Organization of a project in angular divided into components.

The **src** folder is the most interesting for the developer since that is where he will do all his work. Specifically, the **app** folder is where you will place the source code and organize it into various contents:

1. **app.module.ts file**: The purpose is to declare everything you believe in Angular. This file allows us to have an early compilation avoiding many import lines in its components.
2. **App.component.ts file**: It is the root component of the application that serves to control its behavior.
3. **App.component.html file**: Contains all the HTML code used by the component to display itself.
4. **Services folder**: It is created to have the logic of making calls to the API, in this case, the weather forecast API and TTN API.

Take into account the following information if an API is implemented.

You will have to change the TTN API in Angular to yours that you generated when you configured the integration to make it work in Appendix A.2.

Also, when a website is opened, loading data from third-party servers is strictly prohibited. However, if the administrators of both websites have agreed to work together, there is no reason to avoid sharing. In these cases, the so-called cross-origin resource sharing **CORS** regulates collaboration.

Running APIs to see the results, the browser showed an error message of type **CORS**. This happens because my local server does not have the pre-privileges to access the API and there was a cross request. One solution would be to modify an API parameter to allow cross-access of the initial URL where the front-end service is, but since the API server is not accessed, the solution was to create a proxy **proxy.conf.json file** in angular. Adding the API URL and the header authorization to skip the restriction and have access to the data decapitated in Fig. A.9.

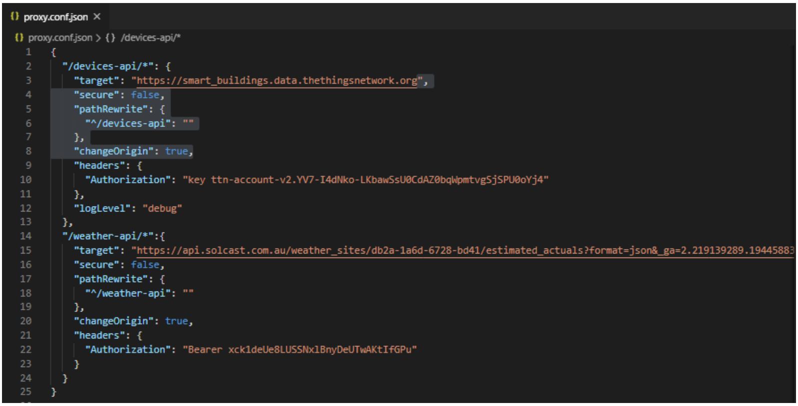

Figure A.9: Proxy configuration to solve CORS cross requests.

Bibliography

[1] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "A comparative study of lpwan technologies for large-scale iot deployment," *ICT express*, vol. 5, no. 1, pp. 1–7, 2019.

[2] D.-H. Kim, E.-K. Lee, and J. Kim, "Experiencing lora network establishment on a smart energy campus testbed," *Sustainability*, vol. 11, no. 7, p. 1917, 2019.

[3] R. Bull, N. Chang, and P. Fleming, "The use of building energy certificates to reduce energy consumption in european public buildings," *Energy and Buildings*, vol. 50, pp. 103–110, 2012.

[4] C. A. Balaras, A. G. Gaglia, E. Georgopoulou, S. Mirasgedis, Y. Sarafidis, and D. P. Lalas, "European residential buildings and empirical assessment of the hellenic building stock, energy consumption, emissions and potential energy savings," *Building and environment*, vol. 42, no. 3, pp. 1298–1314, 2007.

[5] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "A comparative study of lpwan technologies for large-scale iot deployment," *ICT express*, vol. 5, no. 1, pp. 1–7, 2019.

[6] 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)*, pp. 55–59, IEEE, 2015.

[7] L. Gregora, L. Vojtech, and M. Neruda, "Indoor signal propagation of lora technology," in *2016 17th International Conference on Mechatronics-Mechatronika (ME)*, pp. 1–4, IEEE, 2016.

[8] Y.-W. Ma and J.-L. Chen, "Toward intelligent agriculture service platform with lora-based wireless sensor network," in *2018 IEEE International Conference on Applied System Invention (ICASI)*, pp. 204–207, IEEE, 2018.

[9] SEMTECH, *AN1200.22,LoRa™ Modulation Basics*, May 2015.

[10] Community, "The things network." url: https://www.thethingsnetwork.org/docs/, 2020.

[11] S. supported LoRaWAN, "Lora alliance." url: https://lora-alliance.org/member-directory, 2015.

[12] S. authors, "Lorawan." url: https://lorawan.es/, 2020.

[13] Community, "Connect devices to the things network." url: https://www.thethingsnetwork.org/docs/devices, 2020.

[14] Community, "The things indoor gateway." url: https://www.thethingsnetwork.org/docs/gateways/thethingsin door/, 2020.

[15] ETSI, "Electromagnetic compatibility and radio spectrum matters (erm); short range devices (srd); radio equipment to be used in the 25 mhz to 1 000 mhz frequency range with power levels ranging up to 500 mw; part 1: Technical characteristics and test methods," ETSI EN 300 220-1 V2.4.1 REN/ERM-TG28-434, ETSI, 2020.

[16] F. K. Shaikh and S. Zeadally, "Energy harvesting in wireless sensor networks: A comprehensive review," *Renewable and Sustainable Energy Reviews*, vol. 55, pp. 1041–1054, 2016.

[17] L. Sigrist, A. Gomez, R. Lim, S. Lippuner, M. Leubin, and L. Thiele, "Measurement and validation of energy harvesting iot devices," in *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017*, pp. 1159–1164, IEEE, 2017.

[18] Community, "Lopy4 datasheet - pycom." url: https://docs.pycom.io/gitbook/assets/specsheets/Pycom_002_ Specsheets_LoPy4_v2.pdf/ , 2020.

[19] Community, “Pysense datasheet - pycom.” url: https://docs.pycom.io/gitbook/assets/pysense-specsheet.pdf, 2020.

[20] Google, “Angular framework.” url: https://angular.io, 2020.

[21] M. Integrated, “Ds18b20 sensor datasheet.” url: https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf, 2020.

[22] E. Commision, “Photovoltaic geographical information system.” url: https://re.jrc.ec.europa.eu/pvg_tools/en/tools.html, 2020.