Internship Report

Development of a real time water quality monitoring device and visualisation dashboard

Teddy Babel

BUT in Computer science

BUT2 – 2023

17/04/2023 – 16/06/2023

For the attention of :

- Dr. Andrew Donnellan, Internship supervisor
- Isabelle Clavel, Educational tutor

Abstract

Water pollution poses numerous dangers to both the environment and human populations relying on these water bodies. First, chemical contaminants discharged into rivers from industrial and agricultural activities, as well as improper waste disposal can have **devastating consequences**. Heavy metals, pesticides, and toxic chemicals can accumulate in sediments and enter the food chain, leading to bioaccumulation and bio magnification.

These pollutants can have **long-term effects on aquatic organisms**, disrupting their reproductive systems, impairing growth and development, and even causing population declines. Moreover, the discharge of untreated sewage and wastewater into rivers can introduce **harmful pathogens**, increasing the risk of **waterborne diseases**.

Communities living near polluted rivers are at a higher risk of contracting illnesses such as **diarrhea, cholera, and hepatitis**. Additionally, excessive nutrient in water can lead to eutrophication, where algal blooms deplete oxygen levels in the water, leading to marine life death and the collapse of aquatic ecosystems.

This is where our device helps, we aim to create **an Internet of Things** based device that will **monitor the water quality** based on multiples factors present in the water, and have a real time visualisation dashboard that will display all the data received from the devices.

With this device we aim to create an **easy to make and affordable** way to monitor the water quality of the stream next to TU Dublin at first for test and then in the rivers around the parks of Tallaght.

Table of contents

Introduction

Studies show that the **deterioration of the water quality** in the world affects **everything on the planet** in many ways: Destruction of biodiversity, Contamination of the food chain, lack of potable water, diseases and infant mortality.

The World Health Organization (WHO) already estimates that about **2 billion people have no option but to drink water contaminated** by excrement, exposing them to **diseases**, the United Nations (UN) says that billions of people simply **don't have access to any clean water** to drink and that diseases from polluted water like diarrhoeal diseases cause the **death of about 1000 children every day** in the world.

There are **multiple factors** to take in consideration so that we can evaluate the water quality of rivers given by the **Environmental Protection Agency** (EPA) such as: pH level, sedimentation, dissolved oxygen, Aquatic Weeds or Algae or total dissolved gas.

Half of the rivers in Ireland are categorized as "**less than good**" on the scale of the European Union Water Framework Directive (WFD) and **18.5%** are classified as "**poor**" or "**bad**".

Therefore, monitoring the water quality of our rivers to ensure that **it's good enough to purify and drink** without any problems and understanding any problems that may arise is an **extremely valuable information** so that we can act upon it.

That is why this project will start by monitoring a **close by river** that is connected to other rivers then we will expand the monitoring to the rivers in the parks around Tallaght with in mind the goal of acting to help **better and maintain** the water quality.

To evaluate the water quality, we will use seven factors: **water temperature, pH, total dissolved solids, turbidity, electrical conductivity, water flow and dissolved oxygen**.

To give a better context for this project we'll first see a presentation of the organisation, then, we will define the objectives of this project by analysing the needs and means, after that we will look at the technical achievement of this project, the methodology, the progress and the results.

Finally, we will conclude with a technical, human and personal assessment.

1. Presentation of the organisation

1.1. The history of TU Dublin

My internship is at the Tallaght campus of Technological University Dublin (TU Dublin). As of January 1st, 2019, TU Dublin became Ireland's first Technological University, combining three previous organizations: Dublin Institute of Technology, Institute of Technology Blanchardstown, and Institute of Technology Tallaght (ITT).

My internship is specifically located at the **Tallaght campus**, which was one of the three founding institutions. Established in 1992, it was a third-level institution located in Dublin's largest suburb, offering degrees and postgraduate courses in fields such as accountancy, business, computing, engineering, humanities, and science. It also provided adult education courses before being merged with the other institutions.

TU Dublin is a university that brings together arts, sciences, business, and technology, providing paths to obtain a degree ranging from apprenticeship to PhD. With a student population of 30,000 from over 140 countries spread across its campuses in the three biggest population centres in the greater Dublin Region, TU Dublin is the destination of choice for higher education.

ITT DUBLIN
Institute of Technology Tallaght
Institiúid Teicneolaíochta Tamhlacht

Figure 1 : Old logo of ITT Dublin

OLLSCOIL TEICNEOLAÍOCHTA BHAILE ÁTHA CLIATH
TU DUBLIN
TECHNOLOGICAL UNIVERSITY DUBLIN

Figure 2 : Logo of TU Dublin

1.2. Description of the establishment

Located in the largest settlement and county town of South Dublin, Ireland, and the largest satellite town of Dublin, TU Dublin's establishment is situated between Belgard Road, Old Blessington Road, and Greenhills Road, on the southwest side of Tallaght. With a total area of 18 hectares, the campus caters to over 5,000 students and is well-equipped with modern facilities such as lecture theatres with the latest audio-visual equipment, science, electronic, and computer labs, state-of-the-art culinary kitchens, a restaurant, library, canteen, coffee shop, Students Union, automated banking facilities, and even a soccer pitch.

Moreover, the campus provides ample free parking space for both cars and bikes.

1.3. My place in the organisation

From my first day after coming to the TU Dublin Tallaght campus in room 221 on the second floor which is the school of engineering floor. I was greeted by Dr Andrew Donnellan my internship supervisor and Mr. John Fox the project supervisor and I am now in the **IOT research group** under the research department.

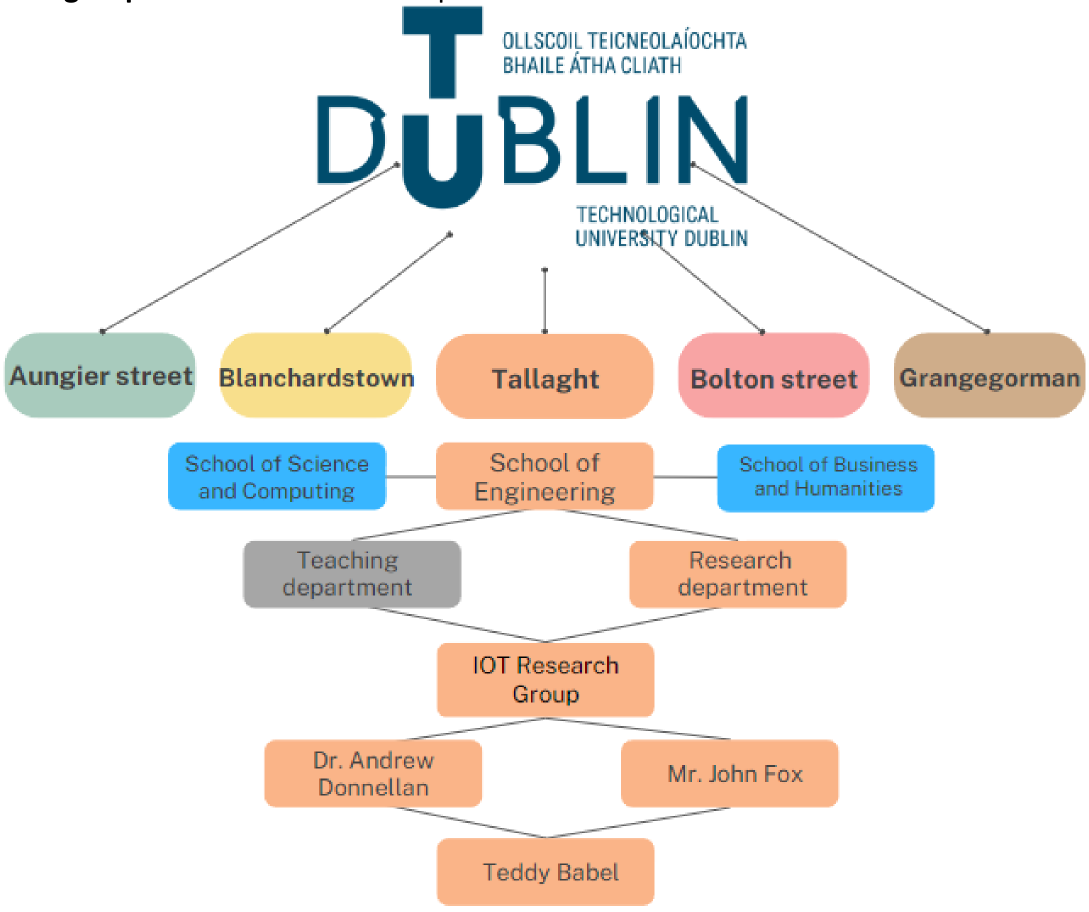


Figure 3 : Organisational chart of TU Dublin

2. Analysis of needs and means

2.1. Analysis of the demand / existing situation

The goal of this project is to make an **autonomous** Internet of Things (IoT) **based water quality monitoring** device using LoRaWAN technology.

To create that device, we have multiple **demands** that need to be looked at, first, the device itself it needs to be able to **gather and send data** from the sensors that it possesses as well as turn on and off by itself.

Next, the housing needs to be **waterproof** and **floating**, there must also be a **dashboard** that shows data in real time.

In conclusion, the device needs to be able to collect and send data from its underwater sensors to a LoRaWAN gateway and we must be able to see the incoming data from a dashboard.

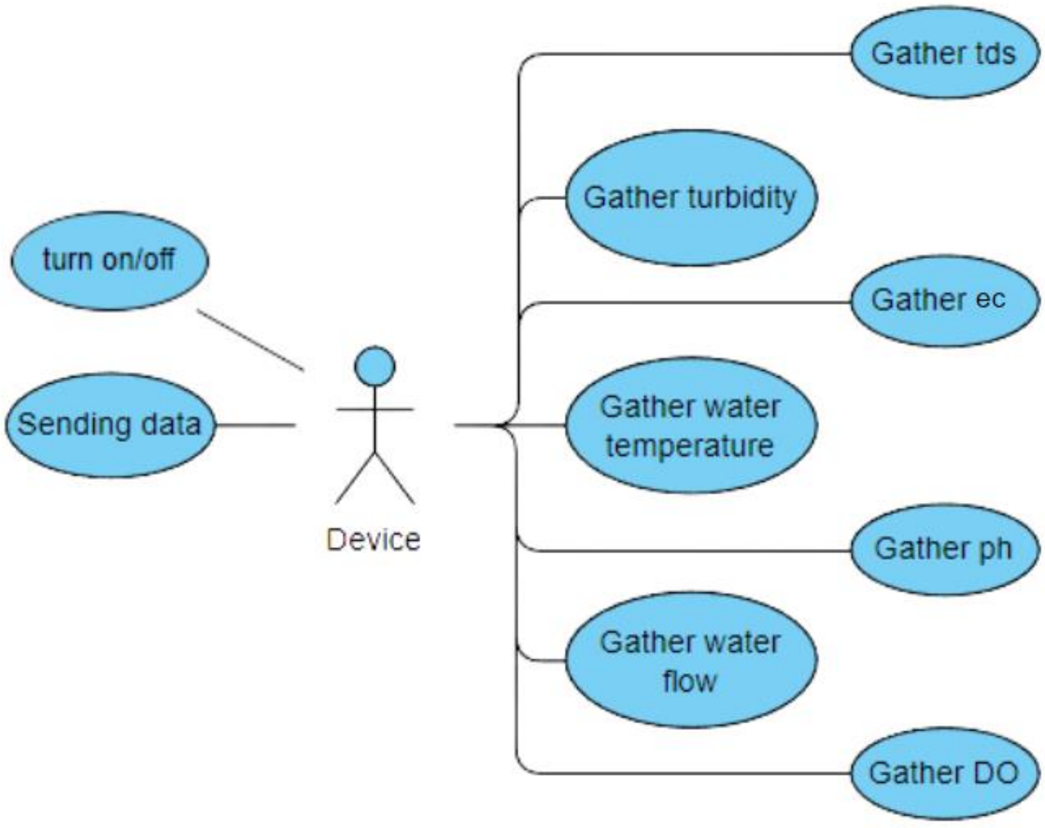


Figure 4 : Use case diagram for the device

We were not the first to work on this water quality monitoring project, in fact, the project from last year's team was nearly finished but due to lack of time the housing for the device could not be completed, and therefore, the device could not be tested nor finished.

Pictured below is the unfinished housing for the device that was made by the team of interns from last year.

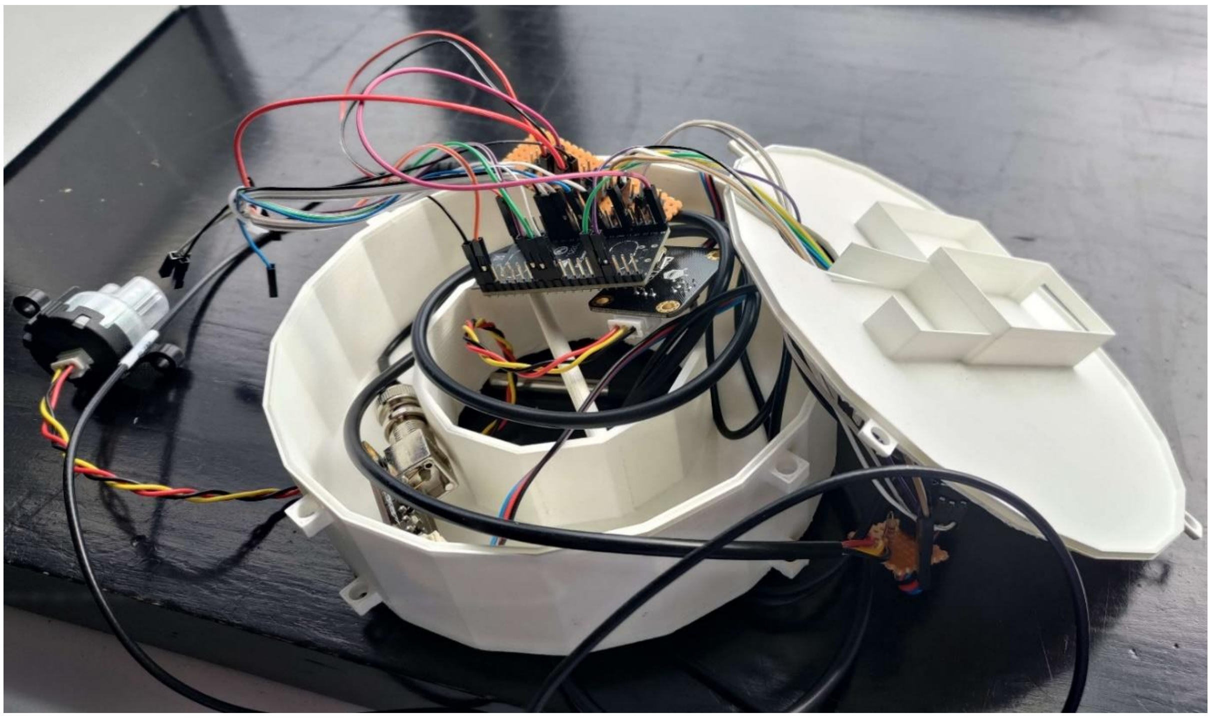


Figure 5 : unfinished housing from last year project

Thanks to this **existing project**, we had a rough idea of what could pose us problems in this year's project and what would be easier to do first or last.

We now knew that the code for the sensors itself was not a problem as it could easily be found online within forums or the seller's help, since there were no test of the device we also didn't have any real data to show for a dashboard so we could not conclude and estimate how hard its creation would be, but the determining factor of the project's failure was the housing, it was done last after getting everything else to work, which proved to be the wrong decision as there was not enough time left to design and print a fully working device and then test it and gather data from the stream going around the campus.

2.2. Defining the constraints and the goals

We decided to **start over from scratch** because most of the project was unusable for what we were planning on doing. Our project will focus on the creation of a real time water quality monitoring device, and for that we need to take into considerations a lot of things : first of all, the material we are going to use for the housing of the electronics, it needs to be water proof, the housing also need to be **impermeable and watertight** so as to not let any water come in from the gaps that there could be on the housing, the housing needs to be big enough to accommodate multiple sensors of different sizes, it also need to be able to float on the water without sinking.

Then, we know the project will be coded in Arduino so its own language which is a variant of C++ needs to be learned, depending on which Arduino board we use there may be memory storage issues, since some Arduino boards have low memory space.

After that, there are the sensors which are one of the **most important** part of the device since they are the ones who collects the necessary data in real time, careful considerations must be made in order to determine which sensors will be used, such as: compatibility with the Arduino boards we will be using, how useful is it in determining whether or not a river’s water have good or poor quality, how long will it be able to last while continuously submerged under water, how much power does it need to collect data and how long will it stay activated and deactivated.

Next, there is the battery that also poses problems since we want to be able to leave the device without changing it for long periods of time, we need to have a battery that can last long or that is rechargeable without having to take the device off the water and changing the battery manually.

Furthermore, we also have to choose how to collect the data received from the sensors, we cannot retrieve the data manually from an SD card for example because we want the device to be able to completely stand alone in the water without human intervention.

Finally, there will be a dashboard that need to receive and show the data that we collected.

The end goal of this project is to have a working water quality monitoring device that will in real time send data which will then be displayed on a dashboard to monitor the different water quality factors.

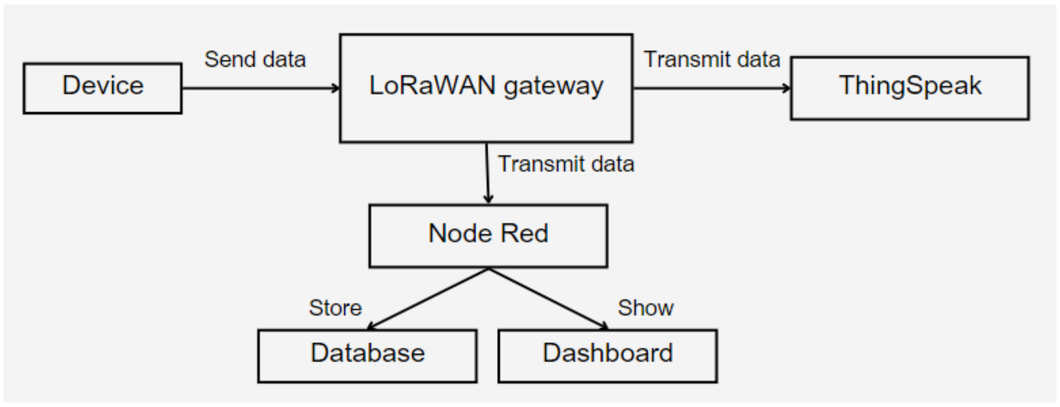


Figure 6 : Infrastructure goal

2.3. Tools and technologies used

2.3.1. ESP32 and Arduino Pro mini

For this project we used two different boards: first, we used an Arduino pro mini to test out the codes and the functionality of every sensors, then, for the actual device that would be the final version we used an ESP32.

There are multiple reasons for the choice of these two boards: to start with, both boards are cost effective for what they propose, they both have Arduino compatibility and consume little power enabling us to have longer battery life.

We first used the Arduino pro mini as a test board because it is small, quick, and efficient but in comparison of the ESP32 it's less suitable for the actual project since the ESP32 can process much more information, it has a bigger memory storage and many more I/O pins which allows for more freedom on how many sensors would be used on a single device.

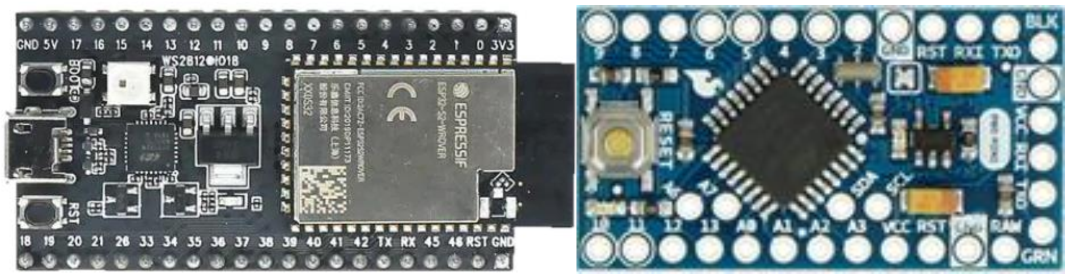


Figure 7 : ESP32 and Arduino Pro Mini

2.3.2 Arduino IDE

To enable communication between the boards and other components, an integrated development environment (IDE) is necessary. The IDE encompasses a graphical interface containing a text editor and all the essential tools for programming tasks. It lets you write your program, save it, compile it, verify it, and transfer it to the board.

Arduino provides a free, open-source IDE software called "Arduino IDE" specifically designed for Arduino microcontrollers. It has been developed to provide a visual, user-friendly, and comprehensive programming experience for Arduino boards. It is particularly user-friendly, making it ideal for beginners, and includes a wide range of sample sketches to help you get started.

Additionally, the Arduino IDE offers a serial monitor that facilitates bidirectional communication between your computer and the board through the USB port. This allows you to send messages from your computer to the board and receive messages from it.

Figure 8 : Arduino IDE logo

2.3.3. LoRaWAN

To be able to receive the data we decided to use LoRaWAN, which stands for Long Range Wide Area Network, is a wireless communication protocol designed specifically for Internet of Things (IoT) applications. It utilizes the LoRa modulation technology, enabling long-range communication with low power consumption.

It comprises three main components: end devices, gateways, and network servers. End devices are the sensors or devices that collect data. Gateways serve as the intermediaries, receiving data from the end devices and transmitting it to the network servers.

LoRaWAN utilizes chirp spread spectrum modulation, operating in the sub-GHz frequency bands. This modulation technique allows for long-range communication, even in environments with significant obstacles or interference. It achieves this while consuming minimal power, making it suitable for battery-powered or energy-efficient IoT devices.

Since LoRa technology operates in the license-free ISM (Industrial, Scientific, and Medical) bands, there is no need for complex infrastructure and we can establish our own private LoRaWAN network.

Those properties make LoRaWAN the most suitable way for us to receive data without manually interfering with the device.

Figure 9 : LoRaWAN logo

2.3.4. ThingSpeak

For this project we decided to utilize the ThingSpeak cloud computing platform, which is an open-source application and API for IoT. It enables the storage and retrieval of data from sensors using the HTTP protocol. ThingSpeak facilitates the development of various applications, such as sensor-logging and location-tracking, and even allows for the creation of a social network of connected devices with status updates. By sending data from sensors to ThingSpeak at regular intervals, the platform automatically generates, stores, and displays data trends without manual intervention.

Figure 10 : ThingSpeak logo

2.3.5. Node-red

We decided to make use of node-red as well in our project as it a web browser-based, open-source programming tool that works almost entirely by visuals and not lines of code, it let us build many different programs and application by connecting pre-built nodes together. It provides a flow-based programming approach, where you define a series of nodes that represent different functions or data sources, and connect them to create the desired outcome.

There is also a lot of documentation, help and packages that you can install to personalise your flows.

Figure 11 : Node-red logo

2.3.6. MySQL Workbench

MySQL Workbench is a graphical tool designed to interact with MySQL databases. It provides a visual interface that allows you to manage and administer MySQL databases efficiently.

It does that by being mostly controlled with a graphical interface and giving as much freedom as possible to use MySQL commands without needing to know or learn the MySQL language, but still allows the use of queries to make more advanced commands in the database.

Figure 12 : MySQL Workbench logo

2.3.7. Sx1272 LoRa module

To be able to send the data received from the sensors to our LoRaWAN gateway we need an end device capable of sending packets, we chose to use the sx1272 because it is able of long-range communication up to 5km in urban areas and 50km in outdoors without obstacles, it consume little power on each activation to send its data, it is capable of great signal penetration allowing data to pass through uncorrupted even if there are multiples buildings in the way of the signal.

Being manufactured by Semtech the sx1272 is widely supported by many libraries, documentation, and community resources and it is not an expensive module allowing us to have multiple devices.

Those are the reasons we decided to use the sx1272 for our project.

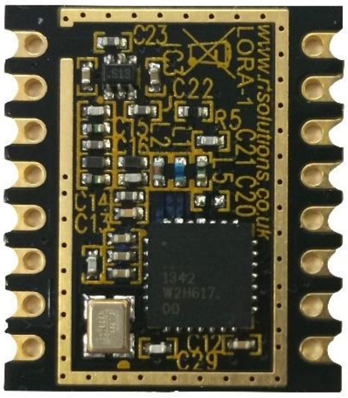


Figure 13 : sx1272 microchip with board

2.3.8. Sensors

While searching for which sensors to use we looked at multiple articles from the EPA and other sources to find what kind and how many sensors we would have we ended up using seven different factors: pH, turbidity, electrical conductivity, total dissolved solids, water temperature, water flow and dissolved oxygen.

We chose those factors because they give a quick and understandable value for us to know whether there is a problem with the quality of the water or not.

We also looked if we already had some of the sensors from last year's project that could be reused.

2.3.8.1. Gravity: Analog pH meter V2

PH levels are a measure of acidity or alkalinity in a solution. The pH scale ranges from 0 to 14, with 7 considered neutral. A pH value below 7 indicates acidity, while values above 7 indicate alkalinity, the lower the pH value, the more acidic the solution, and the higher the pH value, the more alkaline it is.

It is critical to monitor pH levels while assessing water quality because different organisms and processes thrive under specific pH conditions, for instance, aquatic plants and animals have preferred pH ranges for optimal growth and survival.

Monitoring pH levels in bodies of water helps identify potential environmental issues and can guide actions to mitigate them which is one of the goals of our project.

The sensor works by comparing the electric potential of a pH-sensitive system to the potential of a stable reference system.

The sensing system uses a pH-sensitive glass bulb which changes voltage proportionally to the concentration of hydrogen ions. A sensing electrode measures the potential of the glass bulb.

The difference between the stable reference and the sample will then allow the sensor to know the PH and will send us the data

We decided to use the DFRobot pH sensor because like many other DFRobot products and the Gravity series it is compatible with Arduino, it is also very accurate with its measurements, it’s capable of working from 3.0V to 5.0V which make it usable for a wide range of boards, it is also quite durable and is made to last long although being continuously submerged under water will reduce its life expectancy it still can last from six months to a year.

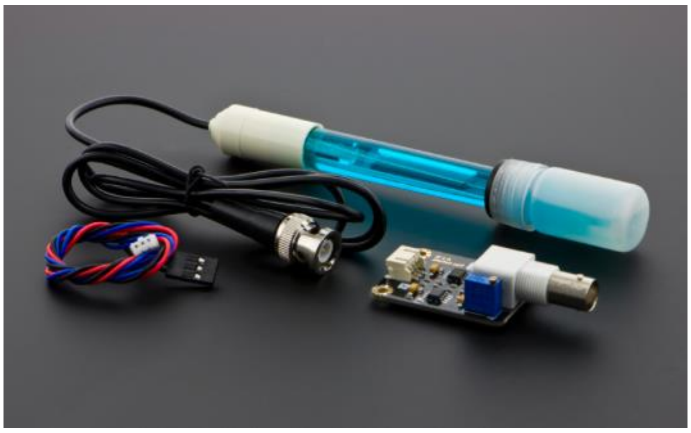


Figure 14 : Gravity: Analog pH meter V2

2.3.8.2. Gravity: Analog turbidity Sensor

The turbidity in water is the amount of haziness of a liquid caused by suspended particles. These particles can include sediment, clay, silt, organic matter, or microscopic organisms. Turbidity is an important parameter in water quality analysis as it provides information about the clarity and cleanliness of water.

Turbidity measurements are used to assess the health of aquatic ecosystems, high turbidity levels can reduce sunlight penetration, affecting photosynthesis and the growth of aquatic plants, it can also impact the survival of fish and other aquatic organisms by reducing their visibility and modifying their feeding patterns.

The turbidity sensor is able to detect suspended particles in water by measuring the light transmittance and scattering rate which changes with the amount of total suspended solids (TSS) in water. As the TTS increases, the liquid turbidity level increases.

The probe works by sending a light beam into the water to be tested. This light will then be scattered by any suspended particles. A light detector is placed at a 90-degree angle to the light source, and detects the amount of light that is reflected back at it, which will in turn allow the sensor to know the turbidity level.

For the turbidity sensor we chose the Gravity series analog turbidity sensor because of its accuracy, cost-effectiveness and long life expectancy from 1.5 years up to 3 years.

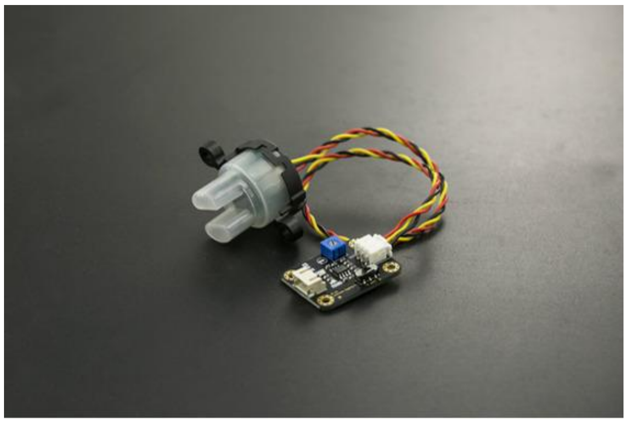


Figure 15 : Gravity: Analog turbidity Sensor

2.3.8.3. Gravity: Analog electrical conductivity meter

Water electrical conductivity refers to the ability of water to conduct an electric current, it measures the concentration of dissolved ions or salts in the water. River water has some conductivity around 50 to 1500 µS/cm (microsiemens per centimetre) because of the dissolved ions. However, when salts or other dissolved substances are more present in the water, like sea water, they dissociate into ions, increasing the conductivity.

Electrical conductivity is an essential parameter in water quality monitoring, as it can be used to assess the salinity of water which is important, for example different plants have different salt tolerance levels, so measuring conductivity helps determine the good growth of plants.

The conductivity sensor will emit an electrical charge via its conductivity probe, which will be submerged in the water being tested, if there is an increase or decrease in the number of dissolved ions, it will result in an increase or decrease in the electrical charge.

The conductivity sensor will measure this charge, and provide us with the water's conductivity.

For this sensor we also took a Gravity series DFRobot sensor as they often make the most accurate, affordable and durable consumer-grade sensors while also being Arduino compatible with their own libraries for their sensors.

For the electrical conductivity sensor, we need the DFRobot_EC library which is not on the library manager of the Arduino IDE and can only be found on GitHub>

Figure 16 : Gravity: Analog electrical conductivity meter

2.3.8.4. Gravity: Analog TDS Sensor

Total Dissolved Solids (TDS) is a measure of the combined content of all inorganic and organic substances present in water. It represents the total concentration of dissolved substances that are typically found in rivers, such as minerals, salts, metals, and organic compounds.

excessive TDS can have ecological impacts, affecting aquatic organisms' health and biodiversity. Additionally, high TDS levels can pose challenges for water treatment processes, as the dissolved substances may interfere with the efficiency of filtration systems and require more extensive treatment measures.

The tds sensor acts like an electrical conductivity sensor by sending an electrical charge between its two electrodes, it will then convert the conductivity reading it received with a formula that allows the sensor to estimate the amount of total dissolved solids in the water.

To evaluate TDS levels in water we decided to take the Gravity series DFRobot TDS sensor because like most water sensors of this series it is very accurate, affordable, durable and Arduino compatible with its own library the GravityTDS library which needs to be retrieved from GitHub as you cannot find it in the Arduino IDE's library manager.

Figure 17 : Gravity: Analog TDS Sensor

2.3.8.5. DS18B20 Water temperature sensor

Water temperature plays an important role in almost all USGS water science. Water temperature exerts a major influence on biological activity and growth, has an effect on water chemistry, can influence water quantity measurements, and governs the kinds of organisms that live in water bodies.

The DS18B20 is a One-Wire temperature sensor manufactured by Dallas Semiconductor, so it only needs one digital pin to communicate with the Arduino board.

However, to work it needs two specific libraries found on the library manager: the OneWire library and the DallasTemperature library.

To work the DS18B20 will also need a resistor connected to it, thanks to the One-Wire protocol the DS18B20 have the ability to operate without an external power supply, power is instead supplied through the One-Wire pullup resistor through the DQ pin when the bus is high.

The high bus signal also charges an internal capacitor, which then supplies power to the device when the bus is low, this method of deriving power from the One-Wire bus is referred to as "parasite power".

We chose the DS18B20 water temperature sensor because of the capability of the One-Wire protocol, the accuracy of the probe, it's long life even submerged, its reliability and its cheap cost.

Figure 18 : DS18B20 Water temperature sensor

2.3.8.6. Gravity: Digital Water Flow Sensor – 1/8”

Water flow is an important factor to consider when monitoring river water quality because it plays a crucial role in determining the overall health and characteristics of a river ecosystem because it affects multiple other factors like diluting and transporting the pollutants downstream, good water flow helps the exchange of gases, in particular oxygen, so bad water flow can impact oxygen levels, it also has an effect on the water’s temperature and many other factors, that’s why it’s important to monitor it.

To monitor the water flow we decided to use the Gravity series digital water flow sensor, this sensor consists of a plastic valve body, flow rotor and hall effect sensor, when liquid flows through the sensor, a magnetic rotor will rotate and the rate of rotation will vary with the rate of flow. The hall effect sensor will then output a pulse width signal.

We chose this sensor because series it is very accurate, affordable, durable and it is Arduino compatible without needing any additional library to install or use.

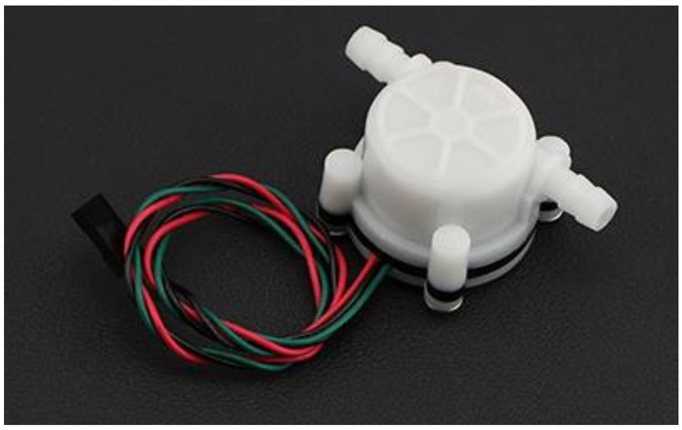


Figure 19 : Gravity: Digital Water Flow Sensor – 1/8"

2.3.8.7. Gravity: Analog Dissolved Oxygen Sensor

Good water quality is very important to aquatic organisms. Dissolved oxygen is one of the most important parameters to reflect water quality. Low dissolved oxygen will lead to problems for any aquatic life from microorganisms to the biggest marine life as it will slow down the nutrient cycling and may make it harder to breathe for the aquatic fauna and flora which can only lead to poor water quality.

The dissolved oxygen (DO) sensor consists of an electrochemical cell that contains two electrodes: an anode and a cathode immersed in an electrolyte solution, as well as a gas-permeable membrane, this membrane allows dissolved oxygen from the water to diffuse into the sensor while preventing the direct contact of the electrolyte with the water sample.

There will then be a chemical reaction which will then generate an electrical current proportional to the concentration of dissolved oxygen in the water.

We decided to use the Gravity series analog DO sensor because although it is expensive it is extremely accurate plus the probe is galvanic, does not need polarization time, and stays available at any time. The filling solution and membrane cap are replaceable, leading to the low maintenance cost. The signal converter board is plug-and-play and works from 3.3V - 5V which makes it compatible with Arduino boards.

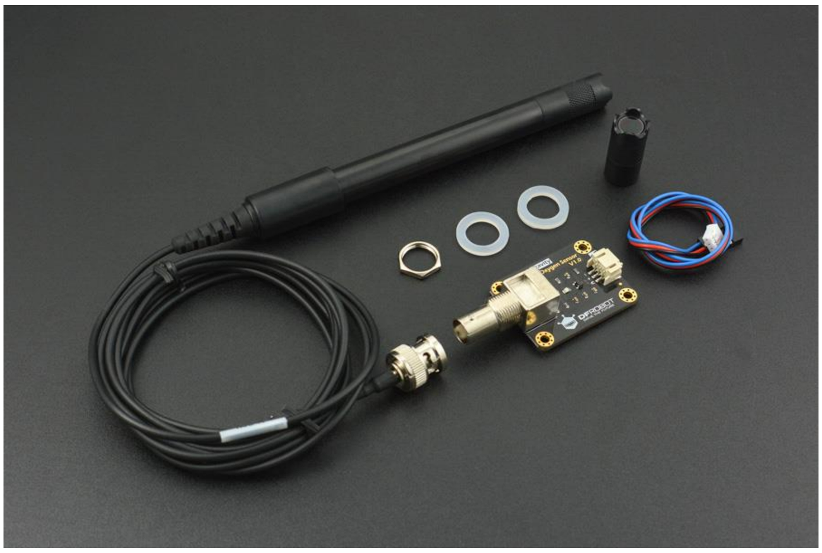


Figure 20 : Gravity: Analog Dissolved Oxygen Sensor

2.3.9. Solidworks

In this project we needed to make our own housing therefore we needed a 3D modelling software to plan and create the housing for our device, for this task we decided that the best software to use would be Solidworks.

Solidworks uses a parametric modelling approach, so that users can easily modify designs by changing parameters, with the software automatically updating related features.

It provides a sketching environment for creating 2D profiles that serve as the foundation for 3D models, users can create complex parts using features like extrusions, revolves, sweeps, and lofts, and add details such as fillets, chamfers, and holes.

Assembly modelling allows for the creation of assemblies by defining relationships and constraints between components.

We decided to use Solidworks for multiple different reasons and advantages: first, Solidworks is taught to the students here at TU Dublin, so it is already installed on the school computers, it’s a very easy to learn program thanks to its beginner friendly user interface and community help, while the learning curve is steep the level of details and advanced techniques needed for our project’s housing is low and can be learned quickly enough, Solidworks allows us to precisely measure our creation so as to be sure that everything will work as intended once printed.

So overall Solidworks is a great and widely used 3D modelling software which will allow us to easily and quickly learn, plan and make a 3D housing for our device that will be ready to be printed.

Figure 21 : Solidworks logo

2.3.10. Ultimaker Cura

After building our housing in 3D with Solidworks we need to print it with a 3d printer, we already have a 3d printer at our disposition, the Ultimaker 2+ which comes with its own software to make it work: Ultimaker Cura.

UltiMaker Cura is a 3D printing software, it will take a STL file 3D model and "slice" it into multiple layers and save the information of each layer to then give to the 3D printer which will print one layer at a time, support can also be added to maintain the stability of the print if there are places floating above the ground with nothing underneath.

Most prints will take multiple hours if not entire days to finish while requiring a lot of printing material but it is overall the easiest and cheapest way to build our own housing for the device of this project.

UltiMaker Cura is a free, easy-to-use 3D printing software, which allows the use of non-Ultimaker 3d printers and offers more than 400 different settings to be able to have the best slicing and results.

We decided to use Ultimaker Cura for all of its advantages and the fact that we already have an Ultimaker 3D printer which would work best with Ultimaker Cura.

Figure 22 : Ultimaker Cura logo

3. Technical realisation

3.1. Methodology

In this project **understanding what we were going to do and the methods that we would use to create and finish this project was important**, therefore the first week of my internship was dedicated to **research** on different subjects with the help of online documentation, research papers, school reports so that I could better understand every needs of the project and the stakes present.

For the next weeks we can start experimenting and testing by creating test code for every one of the sensors using the Arduino pro mini while seeing the possibility and features the boards create and possesses.

Then, we will start working on transmitting the data received from sensors to our LoRaWAN gateway in the form of a packet using the sx1272 a LoRa transreceiver.

After that, we will create a node red infrastructure that can receive and show the data coming to our gateway whether with debug nodes or other nodes such as dashboard nodes.

With this and a good amount of time and progress made on the project we should start working on our internship reports which will be regularly updated with the work done for each week.

While also writing the report, we will start to learn how to use Solidworks to be able to make 3D prototypes model and to familiarize ourselves with the software so that we may be able to print a good finished housing and have a fully functional device in the end.

The next few weeks should be spent between Solidworks and coding to optimize and ensure good quality as well as implementing any new ideas that would come up while working.

The week after that we will print and test a full device, while waiting for prints or for the incoming data we will also work on the dashboard and experiments between different ways of implementing the dashboard such as using node-red, ThingSpeak or plotly dash.

In this week we also have to finish and send our internship reports as well as prepare for the presentation.

Finally, the last week will mostly be about testing and fine tuning the dashboard with the data we receive but there will also be the final presentation for both TU Dublin and the IUT of Blagnac.

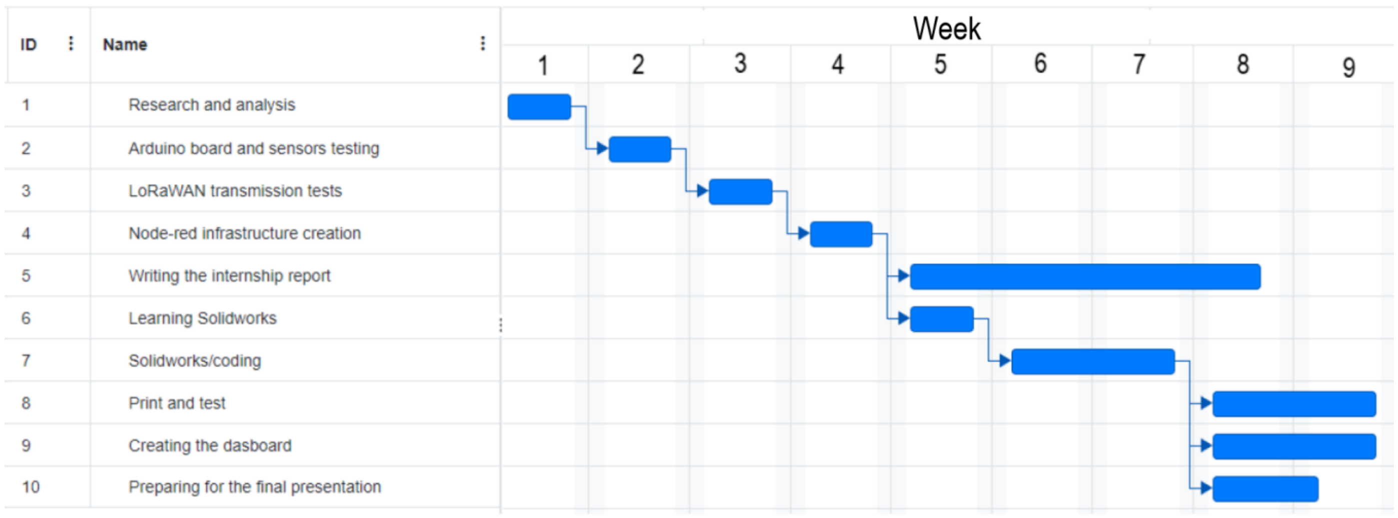


Figure 23 : Provisional Gantt chart

3.2. Progress

Now, we will see how the **project progressed** over the nine weeks of my internship, by taking a closer look at what happened every week, what setbacks we had and what went easier than expected, **the technical difficulties and solutions adopted**.

The first week was dedicated to learning all about the project, on this week we learned about LoRaWAN and all its fundamentals like the infrastructure with the end devices and the gateways, we also learned about the different spreading factors and the duty cycle for IOT devices.

We also made a prototype for a dashboard on plotly dash when we had the time to make sure it would be possible and how hard it would be to implement into the project since plotly dash is a python library which is very different to our main code on Arduino.

The dashboard prototype also needed to be working on a raspberry pi and therefore needed to be an executable to be launched right after the raspberry pi boot up without manually opening the app up.

To make a plotly dash program into an executable we need more libraries : Selenium to be installed in the console with pip install selenium and making a dedicated function that allows selenium to open the desired browser on a specified IP and port, we need to import Timer from threading, as well as importing webbrowser and turning the debug mode off.

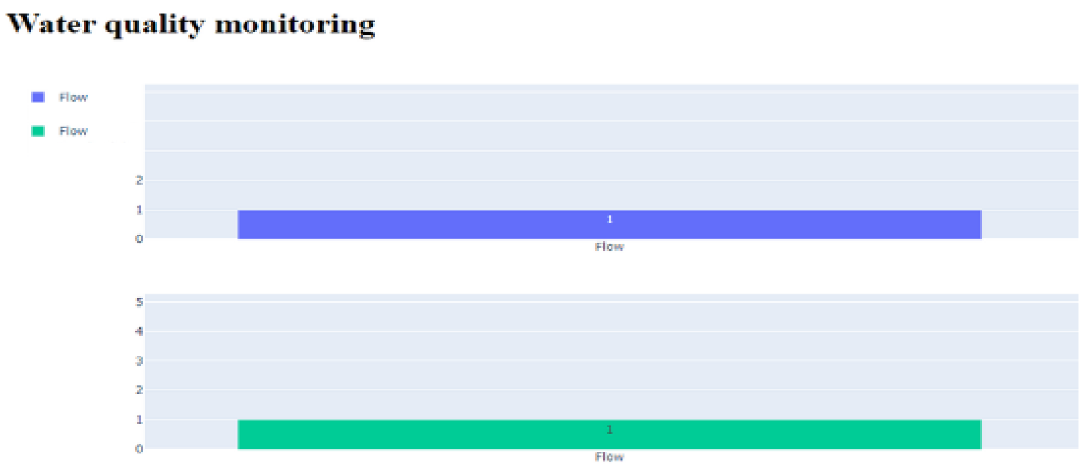


Figure 24 : Plotly dash dashboard prototype

The second and third week were focused upon learning how to use an Arduino board and wire it to every different sensor, since we never used Arduino before we had to learn both the language and how the boards work.

After learning the basics of Arduino, we quickly managed to make the sensors work because every one of them has extensive documentation, the LoRaWAN transreceiver took more time to understand how to make it sends packet because we had to find the board's mark and its datasheet as the sx1272 chip can be mounted on different kind of boards.

After making the good connections we made a quick node-red infrastructure with an MQTT and debug node to see if the LoRaWAN transreceiver worked as intended.

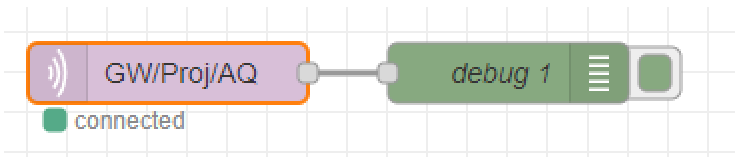
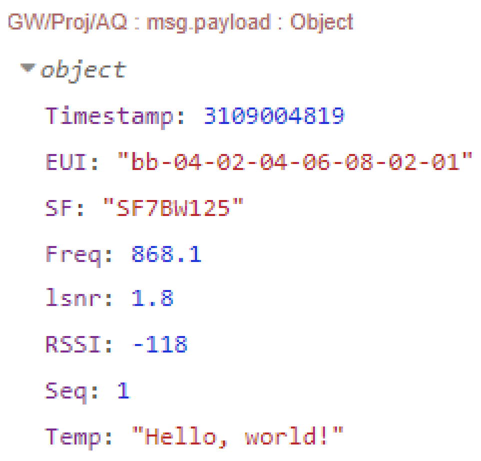


Figure 25 : Node-red debug nodes



```
GW/Proj/AQ : msg.payload : Object
▾object
  Timestamp: 3109004819
  EUI: "bb-04-02-04-06-08-02-01"
  SF: "SF7BW125"
  Freq: 868.1
  lsnr: 1.8
  RSSI: -118
  Seq: 1
  Temp: "Hello, world!"
```

Figure 26 : Node-red debug output

Then we decided to try and make a code that would allow us to change spreading factor without having to change the code and upload it directly to the board.

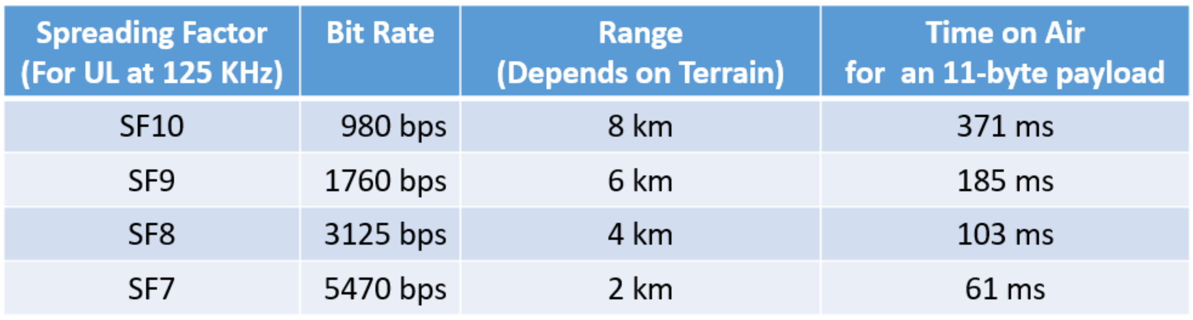
We decided to do that because spreading factors are a very important part of an IOT device and we will see why.

LoRa (which LoRaWAN is built on top of) is based on Chirp Spread Spectrum (CSS) technology, where chirps (also known as symbols) are the carrier of data.

The spreading factor controls the chirp rate, and thus controls the speed of data transmission, lower spreading factors mean faster chirps and therefore a higher data transmission rate.

For every increase in spreading factor, the chirp sweep rate is halved, and so the data transmission rate is halved.

This means that a lower spreading factors reduce the range of transmissions, because they reduce the processing gain and increase the bit rate and inversely a higher spreading factor will increase the range by increasing the processing gain and lowering the bit rate.



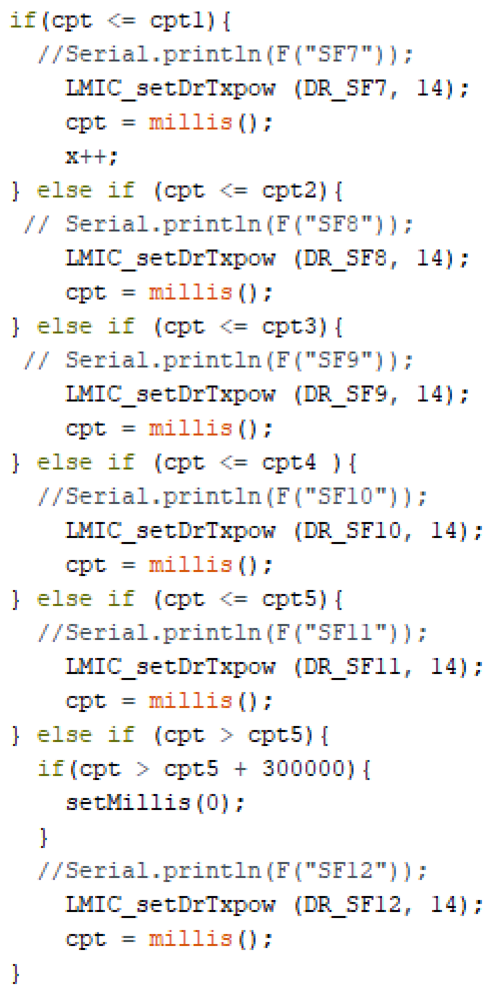
Spreading Factor (For UL at 125 KHz)	Bit Rate	Range (Depends on Terrain)	Time on Air for an 11-byte payload
SF10	980 bps	8 km	371 ms
SF9	1760 bps	6 km	185 ms
SF8	3125 bps	4 km	103 ms
SF7	5470 bps	2 km	61 ms

Figure 27 : Spreading Factor overview chart

This is why it is very important to choose the good spreading factor as too high of a factor we might not receive any packet and too much would be wasteful on the battery as it would keep the device on for longer periods of time.

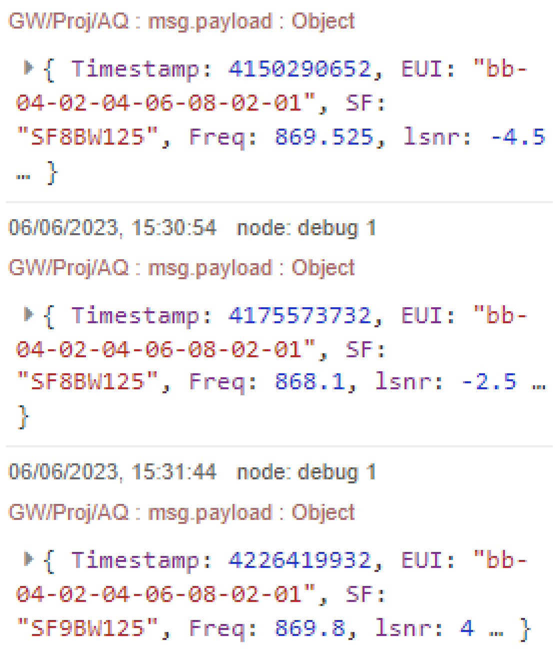
Those are the reason we decided to have an automatic change of spreading factor, we need to know what spreading factor we will use but changing the code after every test would take too long.

The code works by having a timer, every time it finishes sending a packet it will check the time in millisecond and check if a certain amount of time has passed, we chose 5 minutes or 300 000 milliseconds because it will send a satisfying number of packets, and we will be able to know how many packets we receive and lose.



```
if(cpt <= cpt1){
  //Serial.println(F("SF7"));
    LMIC_setDrTxpow (DR_SF7, 14);
    cpt = millis();
    x++;
} else if (cpt <= cpt2){
 // Serial.println(F("SF8"));
    LMIC_setDrTxpow (DR_SF8, 14);
    cpt = millis();
} else if (cpt <= cpt3){
 // Serial.println(F("SF9"));
    LMIC_setDrTxpow (DR_SF9, 14);
    cpt = millis();
} else if (cpt <= cpt4 ){
  //Serial.println(F("SF10"));
    LMIC_setDrTxpow (DR_SF10, 14);
    cpt = millis();
} else if (cpt <= cpt5){
  //Serial.println(F("SF11"));
    LMIC_setDrTxpow (DR_SF11, 14);
    cpt = millis();
} else if (cpt > cpt5){
  if(cpt > cpt5 + 300000){
    setMillis(0);
  }
  //Serial.println(F("SF12"));
    LMIC_setDrTxpow (DR_SF12, 14);
    cpt = millis();
}
```

Figure 28 : Arduino code to change spreading factor



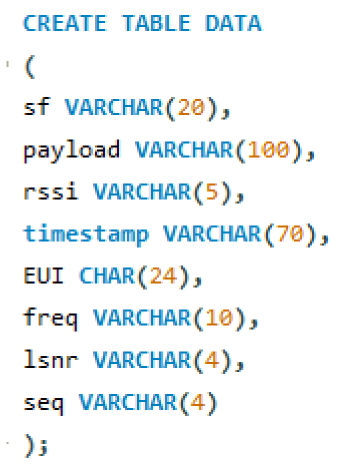
```
GW/Proj/AQ : msg.payload : Object

▸{ Timestamp: 4150290652, EUI: "bb-
04-02-04-06-08-02-01", SF:
"SF8BW125", Freq: 869.525, lsnr: -4.5
… }

06/06/2023, 15:30:54   node: debug 1
GW/Proj/AQ : msg.payload : Object

▸{ Timestamp: 4175573732, EUI: "bb-
04-02-04-06-08-02-01", SF:
"SF8BW125", Freq: 868.1, lsnr: -2.5 …
}

06/06/2023, 15:31:44   node: debug 1
GW/Proj/AQ : msg.payload : Object

▸{ Timestamp: 4226419932, EUI: "bb-
04-02-04-06-08-02-01", SF:
"SF9BW125", Freq: 869.8, lsnr: 4 … }
```

Figure 29 : Results on the debug

The following week was focused on node-red, thanks to the IOT classes in the IUT of Blagnac we had a rudimentary understanding of node-red and node-red dashboard, so we were able to make a working infrastructure quickly but our needs changed multiple times during the projects making the node-red infrastructure changing very often and taking longer to finish.

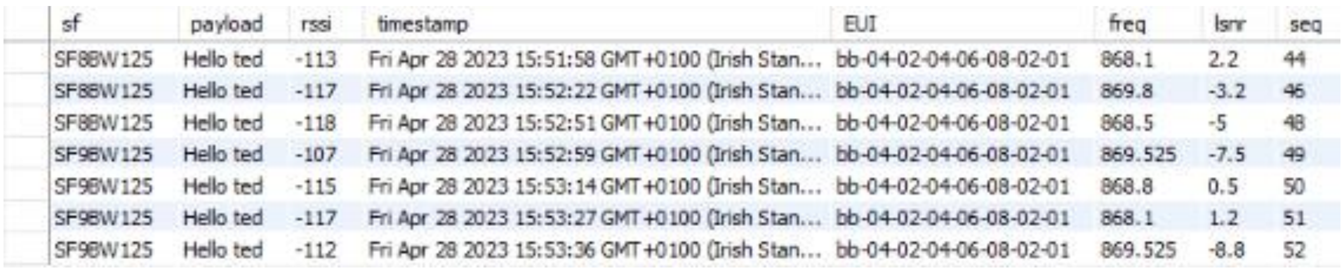
It was also decided that we would make a mySQL database to be able to keep the data we receive and have an easy access to it.

We quickly made a database using our knowledge of Oracle SQL learned in our degree after seeing the differences between it and mySQL.



```
CREATE TABLE DATA
(
sf VARCHAR(20),
payload VARCHAR(100),
rssi VARCHAR(5),
timestamp VARCHAR(70),
EUI CHAR(24),
freq VARCHAR(10),
lsnr VARCHAR(4),
seq VARCHAR(4)
);
```

Figure 30 : Database creation script



sf	payload	rssi	timestamp	EUI	freq	lsnr	seq
SF8BW125	Hello ted	-113	Fri Apr 28 2023 15:51:58 GMT+0100 (Irish Stan...	bb-04-02-04-06-08-02-01	868.1	2.2	44
SF8BW125	Hello ted	-117	Fri Apr 28 2023 15:52:22 GMT+0100 (Irish Stan...	bb-04-02-04-06-08-02-01	869.8	-3.2	46
SF8BW125	Hello ted	-118	Fri Apr 28 2023 15:52:51 GMT+0100 (Irish Stan...	bb-04-02-04-06-08-02-01	868.5	-5	48
SF9BW125	Hello ted	-107	Fri Apr 28 2023 15:52:59 GMT+0100 (Irish Stan...	bb-04-02-04-06-08-02-01	869.525	-7.5	49
SF9BW125	Hello ted	-115	Fri Apr 28 2023 15:53:14 GMT+0100 (Irish Stan...	bb-04-02-04-06-08-02-01	868.8	0.5	50
SF9BW125	Hello ted	-117	Fri Apr 28 2023 15:53:27 GMT+0100 (Irish Stan...	bb-04-02-04-06-08-02-01	868.1	1.2	51
SF9BW125	Hello ted	-112	Fri Apr 28 2023 15:53:36 GMT+0100 (Irish Stan...	bb-04-02-04-06-08-02-01	869.525	-8.8	52

Figure 31 : MySQL database

We continued on the fifth week, by starting on the internship report and opening and learning Solidworks for the first time, although Solidworks offer an intuitive user interface and possesses extensive documentation in form of forums, Q&A or videos, it is pretty confusing working in three dimensions for the first times.

Even after beginning to understand the basics, most of the time spent on Solidworks was trying to find a design for the housing.

We went through many ideas and designs for the housing some not even making it into a Solidworks model, but since this week was mostly focused on learning how to use Solidworks and starting the internship report we were fine with not having completely mastered the software and we were happy with the first few design prototypes.

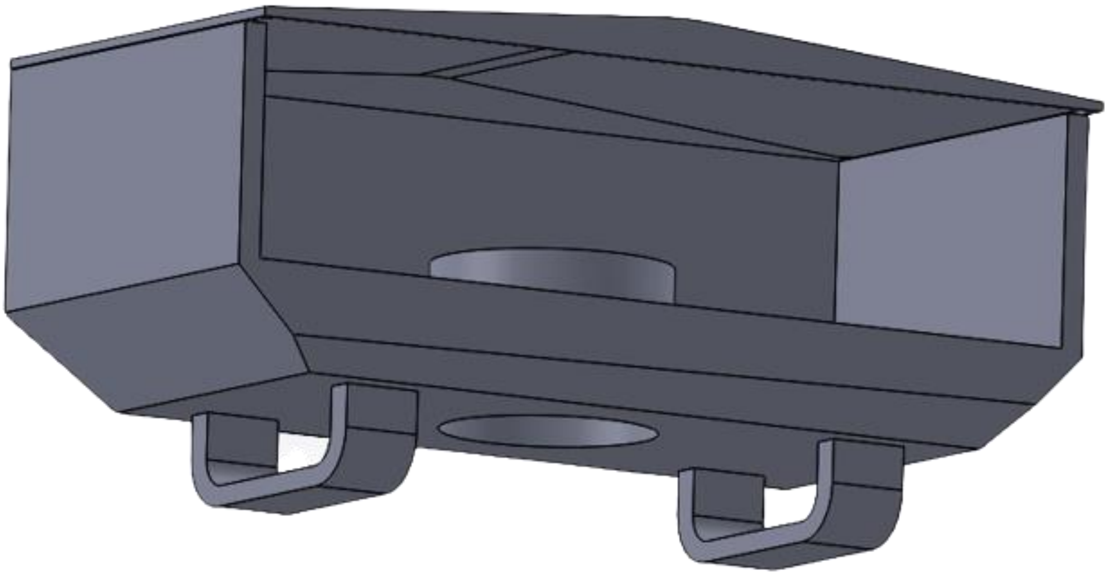


Figure 32 : One of the first full design of the housing (one wall has been made transparent to see the inside)

The next two weeks were focused on advancing our solidworks skills and finishing coding for every device that will house different sensors.

We found while working on the code that with the memory space available on the arduino pro mini board and the size of the sensors it would be best to make three devices each of them having a specific set of sensors.

Since the coding part was finished quickly, we decided to continue working on node-red : we wanted to be able to know easily from which device the data we received came, the easiest way to tell the devices apart was to use their device Extended Unique Identifier (DevEUI) which as the name indicates is an unique id given to every device, we decided to create a new table for every new DevEui sending packets on our LoRaWAN gateway but we didn't want to manually add a new table for every device so we used the function nodes of node-red which allows us to create javascript code which will let us make an SQL query in the form of a string that we can transmit thanks to the mySQL node from the node-red mySQL package that we need to install if we want create a connection between node-red and mySQL, thanks to that we were able to completely automate the creation and filling of the database.

We also created a prototype for the dashboard with the node-red dashboard package.

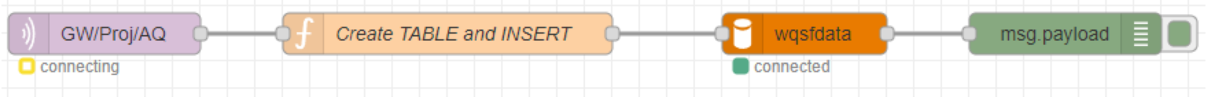
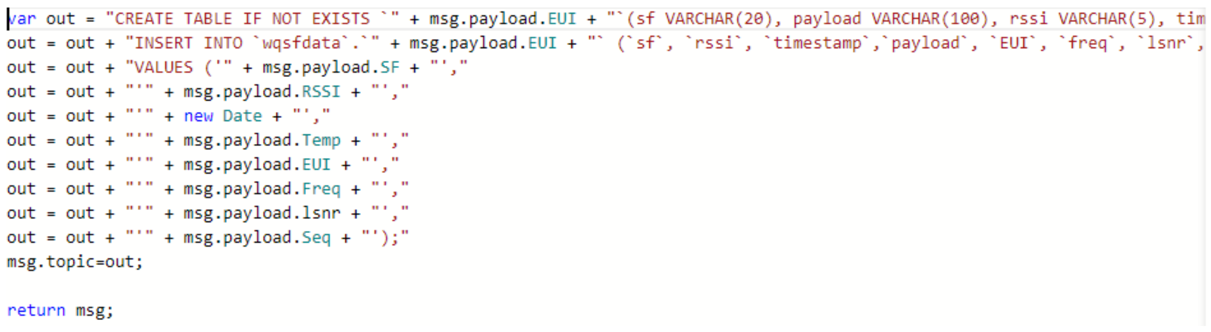
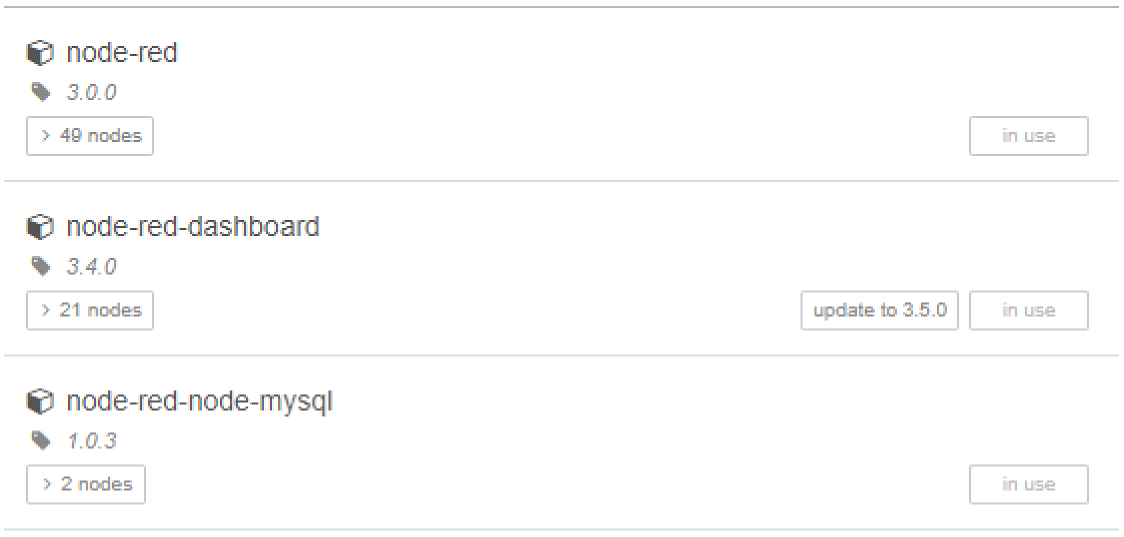


Figure 33 : node-red SQL infrastructure



```
var out = "CREATE TABLE IF NOT EXISTS `" + msg.payload.EUI + "`(sf VARCHAR(20), payload VARCHAR(100), rssi VARCHAR(5), tim
out = out + "INSERT INTO `wqsfdata`.`" + msg.payload.EUI + "` (`sf`, `rssi`, `timestamp`,`payload`, `EUI`, `freq`, `lsnr`,
out = out + "VALUES ('" + msg.payload.SF + "',"
out = out + "'" + msg.payload.RSSI + "',"
out = out + "'" + new Date + "',"
out = out + "'" + msg.payload.Temp + "',"
out = out + "'" + msg.payload.EUI + "',"
out = out + "'" + msg.payload.Freq + "',"
out = out + "'" + msg.payload.lsnr + "',"
out = out + "'" + msg.payload.Seq + "');"
msg.topic=out;

return msg;
```

Figure 34 : Function node javascript code



node-red
3.0.0
› 49 nodes
in use

node-red-dashboard
3.4.0
› 21 nodes
update to 3.5.0
in use

node-red-node-mysql
1.0.3
› 2 nodes
in use

Figure 35 : Node-red packages manager

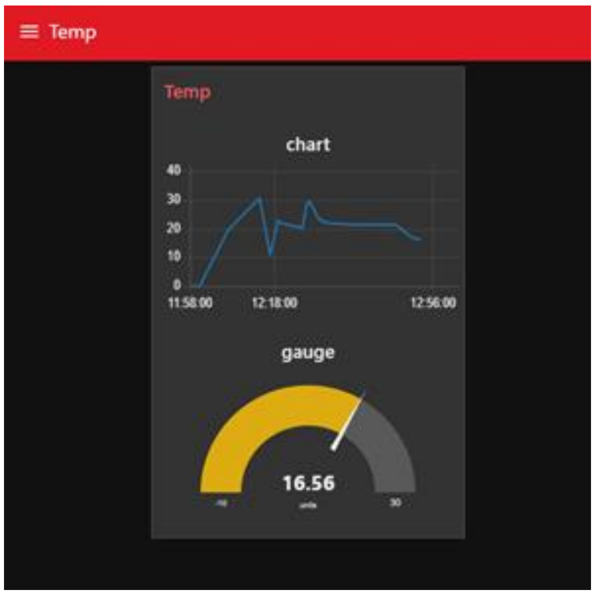


Figure 36 : Dashboard prototype

While we were sometimes working on node-red, most of the time of those weeks was spent working on Solidworks, we went through many more ideas and designs and in doing so we kept refining our skills and our understanding of the software.

After many measurements of sensors and other accessories that would go on and in the housing of the device, we had a working prototype that we could use to print and see any problems that may have been forgotten or overlooked on this first version.

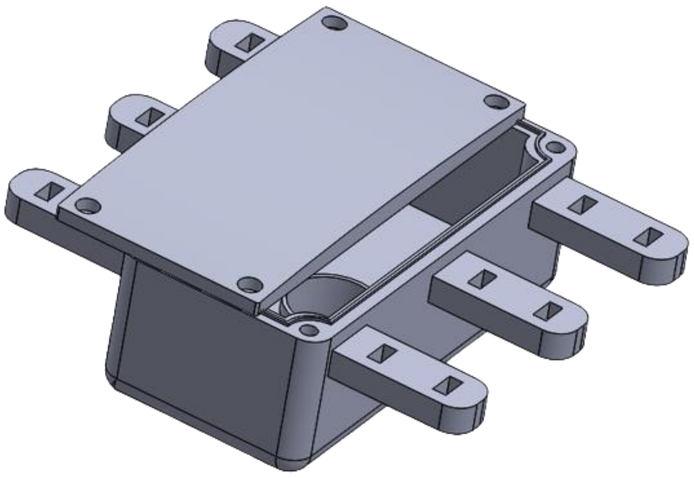


Figure 37 : Prototype 3D model on Solidworks

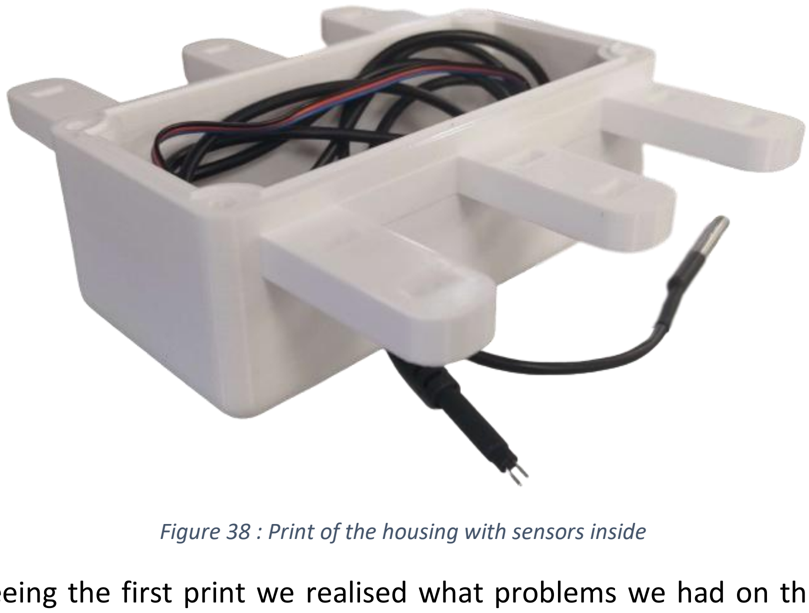


Figure 38 : Print of the housing with sensors inside

After seeing the first print we realised what problems we had on this prototype, it wasn't tall enough for some of the sensors when covered with the lid, the sensors could not be secured to the housing and may risk falling out of the box, there was no real way to put solar panels on it and no way to connect the solar panels to the inside of the housing.

Therefore, we continued working on the housing and arrived at a new design that added fixes to the problems we had, we also made a new cover for our solar panels, the sp 3-37, a flexible, very small, weather resistant solar panels which makes it possible to hold 6 solar panels and then connect them to the inside of the housing to recharge the battery.

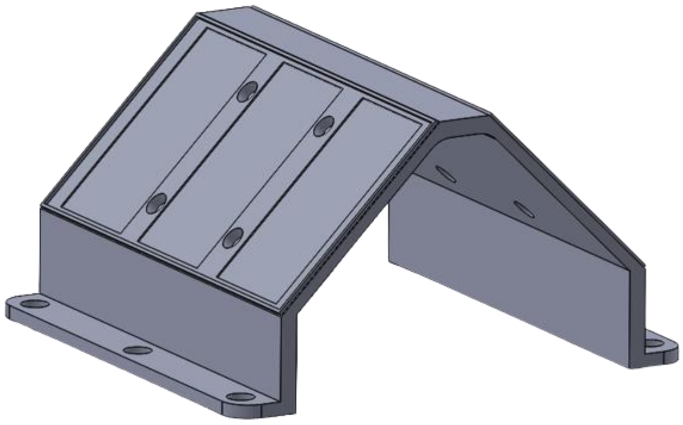


Figure 39 : Cover for solar panels 3D model

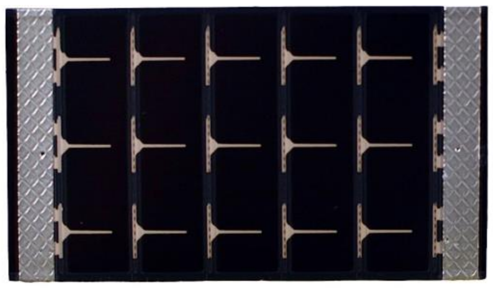


Figure 40 : Sp 3-37 solar panel

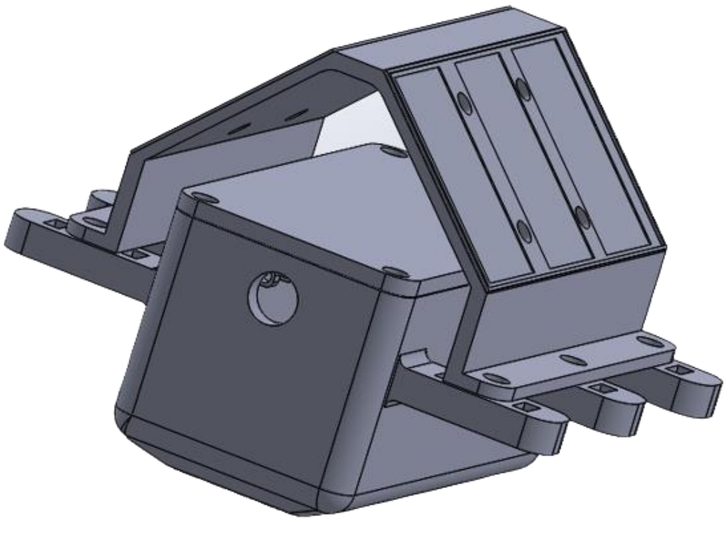


Figure 41 : 3D Model of the full housing

The eight week is focused on printing, testing and perfecting the housing and the device to see if everything works out well, we printed and are testing the device, but there are still problems to fix like the height of the box that is not enough to accommodate the battery.

It is also the week where we finish and give our internship reports back to our university tutor, so this report may not contain the final results of our project as we were not able to fully test out the device before the internship report deadline.

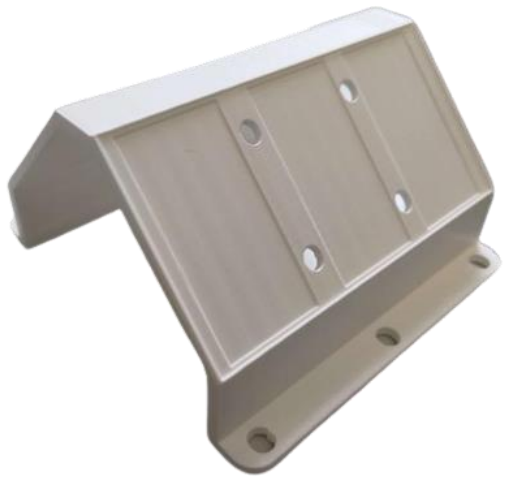


Figure 42 : Print of the cover for solar panels

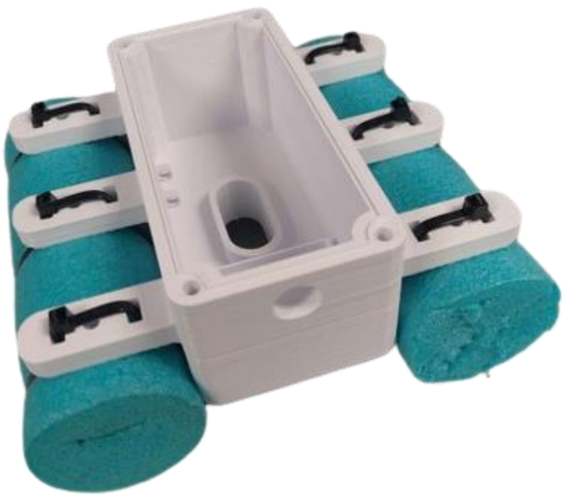


Figure 43 : Print of the new design with pool noodle as floatation device

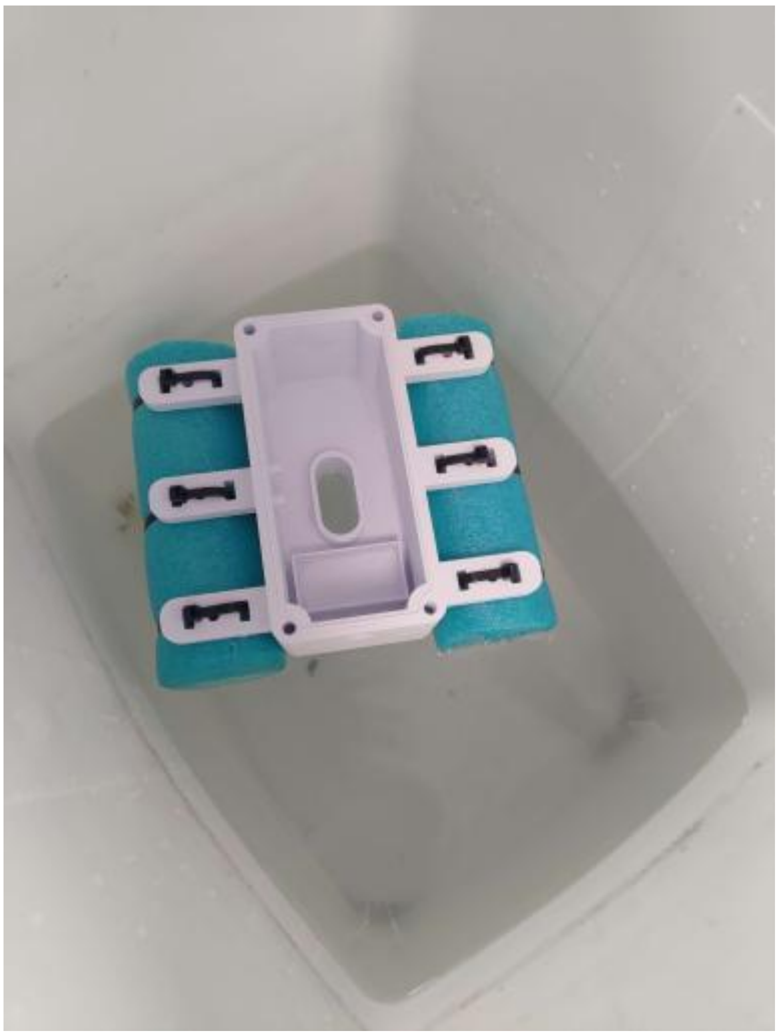


Figure 44 : Housing floating on top of water in a bucket

Finally, if everything goes well with the device tests we should start working on ThingSpeak and finish the dashboard on the ninth and final week.

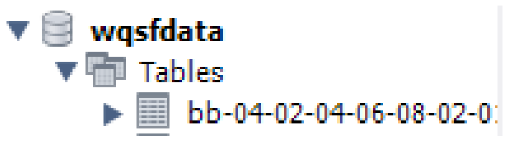
3.3. Results

Now, we are going to see the **results of this project while keeping in mind that the project does extend a week after the delivery of the internship report** and changes may happen to the results in between this time frame.

Overall the **project went really well**, especially considering working alone on the water quality monitoring device and learning many new technologies over the course of only two months.

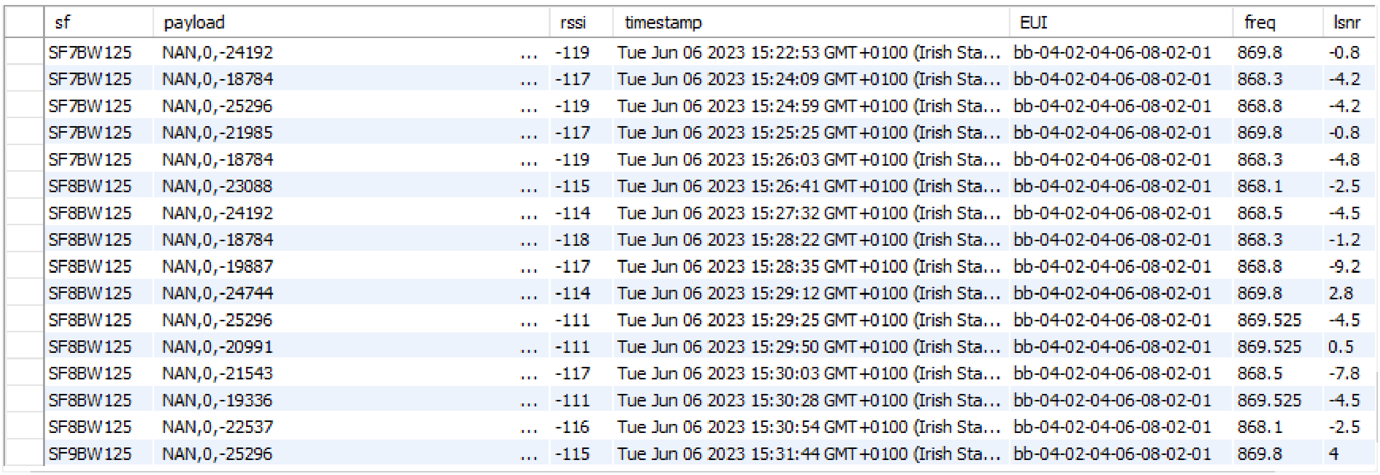
Most objectives have been met, **around 80% of the project is finished** and the remaining tasks are being worked on in the last week.

To begin, we are going to take a look at our MySQL database.



wqsfdata
Tables
bb-04-02-04-06-08-02-0

Figure 45 : Database and tables



sf	payload	rssi	timestamp	EUI	freq	lsnr
SF7BW125	NAN,0,-24192 ...	-119	Tue Jun 06 2023 15:22:53 GMT+0100 (Irish Sta...	bb-04-02-04-06-08-02-01	869.8	-0.8
SF7BW125	NAN,0,-18784 ...	-117	Tue Jun 06 2023 15:24:09 GMT+0100 (Irish Sta...	bb-04-02-04-06-08-02-01	868.3	-4.2
SF7BW125	NAN,0,-25296 ...	-119	Tue Jun 06 2023 15:24:59 GMT+0100 (Irish Sta...	bb-04-02-04-06-08-02-01	868.8	-4.2
SF7BW125	NAN,0,-21985 ...	-117	Tue Jun 06 2023 15:25:25 GMT+0100 (Irish Sta...	bb-04-02-04-06-08-02-01	869.8	-0.8
SF7BW125	NAN,0,-18784 ...	-119	Tue Jun 06 2023 15:26:03 GMT+0100 (Irish Sta...	bb-04-02-04-06-08-02-01	868.3	-4.8
SF8BW125	NAN,0,-23088 ...	-115	Tue Jun 06 2023 15:26:41 GMT+0100 (Irish Sta...	bb-04-02-04-06-08-02-01	868.1	-2.5
SF8BW125	NAN,0,-24192 ...	-114	Tue Jun 06 2023 15:27:32 GMT+0100 (Irish Sta...	bb-04-02-04-06-08-02-01	868.5	-4.5
SF8BW125	NAN,0,-18784 ...	-118	Tue Jun 06 2023 15:28:22 GMT+0100 (Irish Sta...	bb-04-02-04-06-08-02-01	868.3	-1.2
SF8BW125	NAN,0,-19887 ...	-117	Tue Jun 06 2023 15:28:35 GMT+0100 (Irish Sta...	bb-04-02-04-06-08-02-01	868.8	-9.2
SF8BW125	NAN,0,-24744 ...	-114	Tue Jun 06 2023 15:29:12 GMT+0100 (Irish Sta...	bb-04-02-04-06-08-02-01	869.8	2.8
SF8BW125	NAN,0,-25296 ...	-111	Tue Jun 06 2023 15:29:25 GMT+0100 (Irish Sta...	bb-04-02-04-06-08-02-01	869.525	-4.5
SF8BW125	NAN,0,-20991 ...	-111	Tue Jun 06 2023 15:29:50 GMT+0100 (Irish Sta...	bb-04-02-04-06-08-02-01	869.525	0.5
SF8BW125	NAN,0,-21543 ...	-117	Tue Jun 06 2023 15:30:03 GMT+0100 (Irish Sta...	bb-04-02-04-06-08-02-01	868.5	-7.8
SF8BW125	NAN,0,-19336 ...	-111	Tue Jun 06 2023 15:30:28 GMT+0100 (Irish Sta...	bb-04-02-04-06-08-02-01	869.525	-4.5
SF8BW125	NAN,0,-22537 ...	-116	Tue Jun 06 2023 15:30:54 GMT+0100 (Irish Sta...	bb-04-02-04-06-08-02-01	868.1	-2.5
SF9BW125	NAN,0,-25296 ...	-115	Tue Jun 06 2023 15:31:44 GMT+0100 (Irish Sta...	bb-04-02-04-06-08-02-01	869.8	4

Figure 46 : Data stored in the database received from the LoRaWAN gateway

As we can see, thanks to our node-red JavaScript to MySQL script we are able to create a new table with the name set for the DevEUI of the device sending data.

This will let us easily check the data from any device in any place with much more information and precision than what is given on the dashboard.

Next, we will see the results that we obtained on node-red for receiving the data both on the dashboard and on the MySQL database.

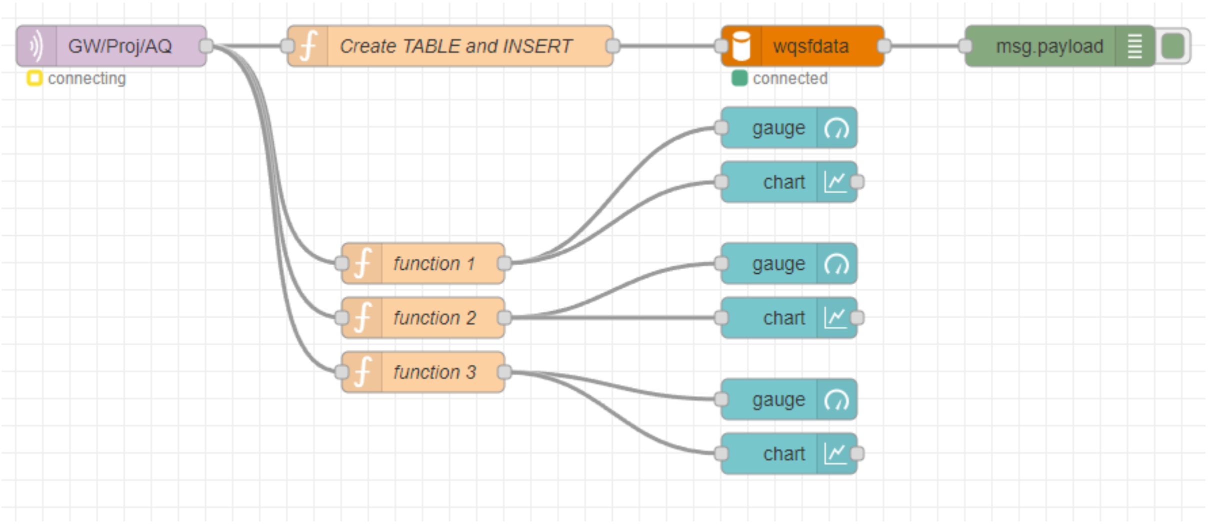
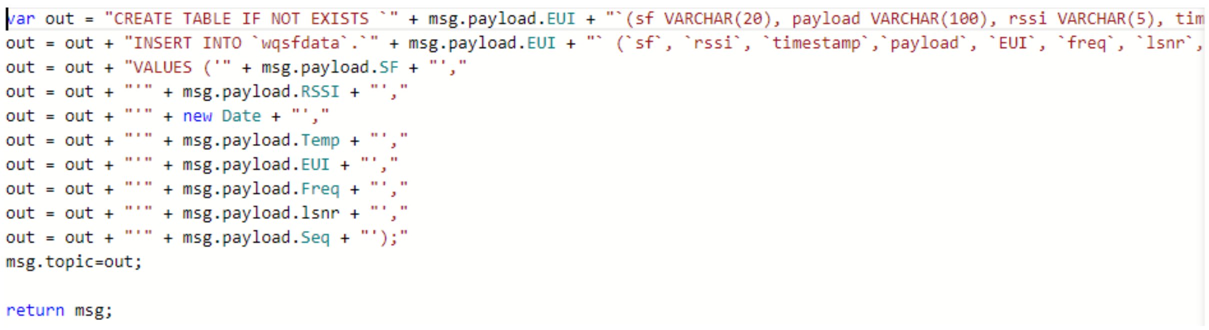


Figure 47 : Node-red infrastructure

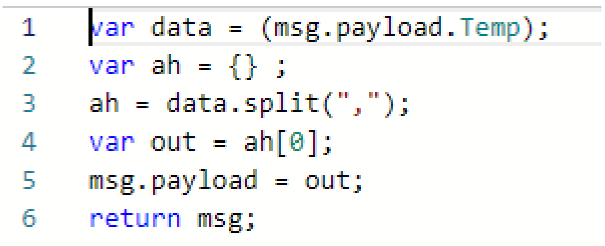
On this image we see the node-red infrastructure separated in two, after receiving the data from the MQTT in node with the right topic the nodes are separated between MySQL and the dashboard, we will first examine the function node for the MySQL.



```
var out = "CREATE TABLE IF NOT EXISTS `" + msg.payload.EUI + "`(sf VARCHAR(20), payload VARCHAR(100), rssi VARCHAR(5), tim
out = out + "INSERT INTO `wqsfdata`.`" + msg.payload.EUI + "` (`sf`, `rssi`, `timestamp`,`payload`, `EUI`, `freq`, `lsnr`,
out = out + "VALUES ('" + msg.payload.SF + "',"
out = out + "'" + msg.payload.RSSI + "',"
out = out + "'" + new Date + "',"
out = out + "'" + msg.payload.Temp + "',"
out = out + "'" + msg.payload.EUI + "',"
out = out + "'" + msg.payload.Freq + "',"
out = out + "'" + msg.payload.lsnr + "',"
out = out + "'" + msg.payload.Seq + "');"
msg.topic=out;

return msg;
```

Figure 48 : Node-red function node for MySQL

This script within the function node first create if it doesn't already exist a table with the name being the DevEUI, it will then proceed to fill the database with the information received before sending it to the MySQL node which will take the String that is in a form of a MySQL query and execute it.



```
var data = (msg.payload.Temp);
var ah = {} ;
ah = data.split(",");
var out = ah[0];
msg.payload = out;
return msg;
```

Figure 49 : Dashboard function nodes

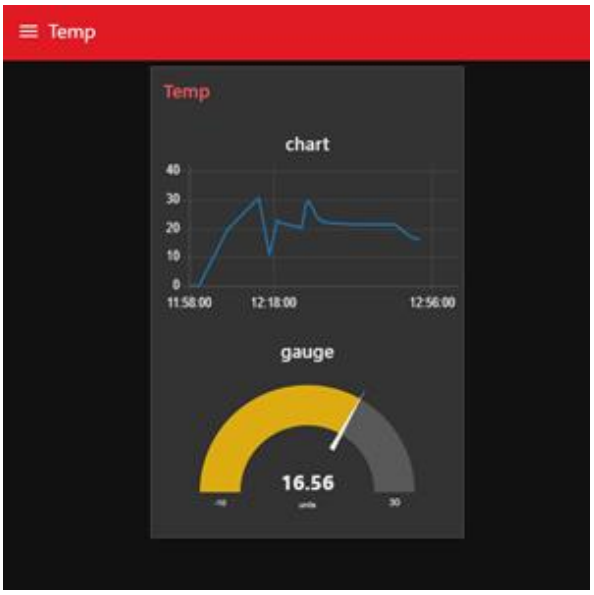
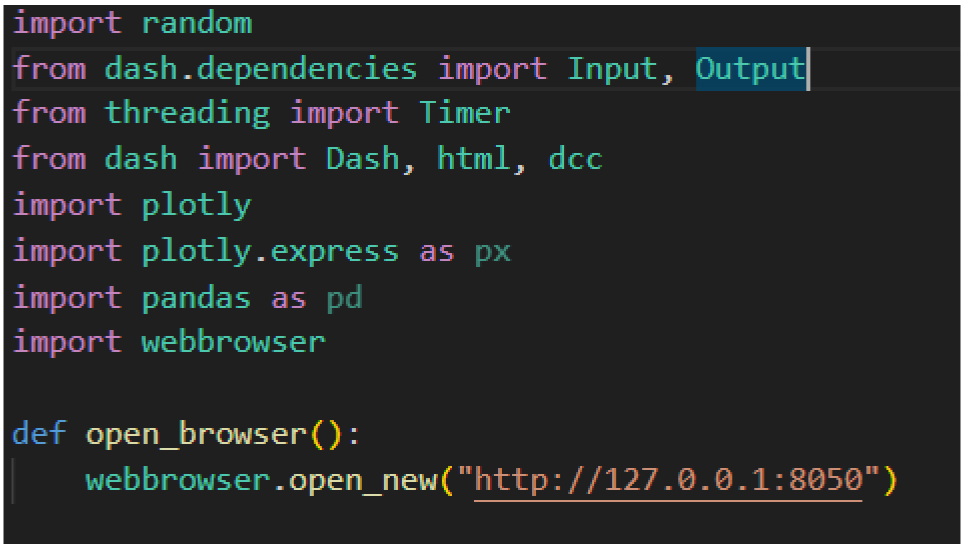


Figure 50 : Node-red temperature dashboard

Next, we will see the results from our code.

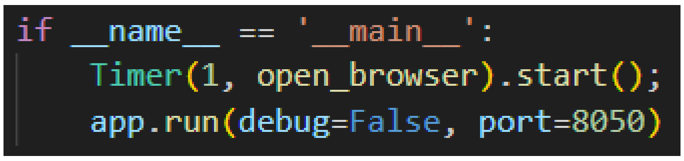
Although coding was quite scarce within this project due to the well documented sensors that already had working example codes, we still made and needed codes for different usages.

At the start of the project, we wanted to have a dashboard accessible from a raspberry pi, we decided to make it in python as it is compatible with the raspberry pi and contains useful libraries to create dashboards like the one we used plotly dash.



```python
import random
from dash.dependencies import Input, Output
from threading import Timer
from dash import Dash, html, dcc
import plotly
import plotly.express as px
import pandas as pd
import webbrowser

def open_browser():
    webbrowser.open_new("http://127.0.0.1:8050")
```

Figure 51 : Python script libraries and link to the created dashboard

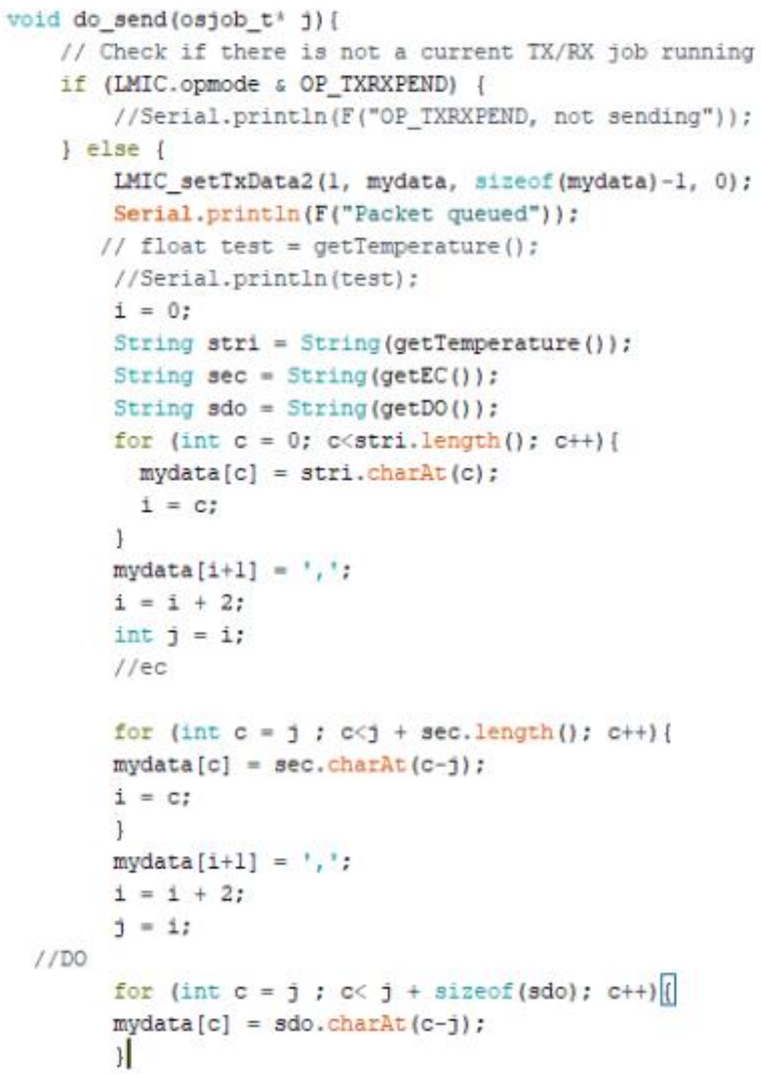


```python
if __name__ == '__main__':
    Timer(1, open_browser).start();
    app.run(debug=False, port=8050)
```

Figure 52 : Python script that open the browser to the dashboard and create the dashboard with plotly dash

The whole python scrip will create the dashboard using plotly dash, the part of the script pictured above handle opening the browser and the address where the dashboard appears after the script has been executed and will keep the data updated in real time.

After that, we also created Arduino codes, three codes that will handle the sensors from different devices but those were easy to create with the existing documentation.

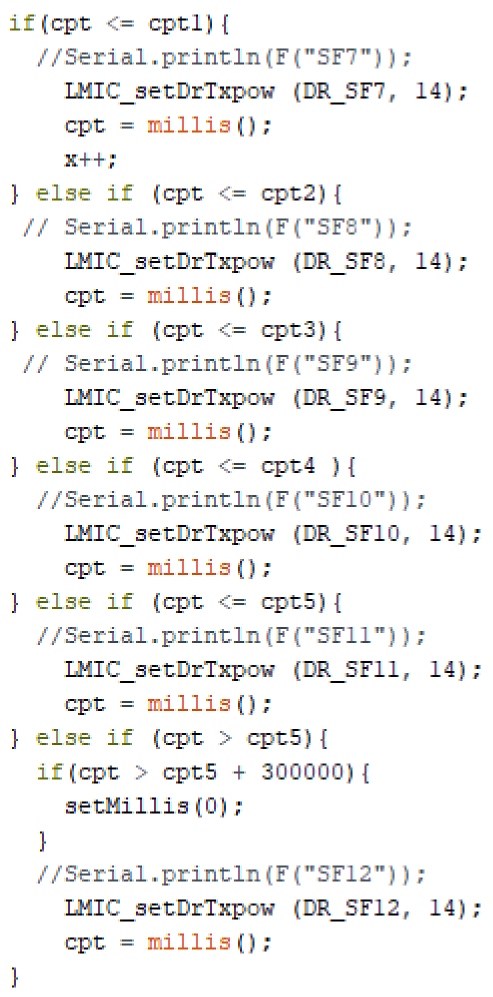


```
void do_send(osjob_t* j){
    // Check if there is not a current TX/RX job running
    if (LMIC.opmode & OP_TXRXPEND) {
        //Serial.println(F("OP_TXRXPEND, not sending"));
    } else {
        LMIC_setTxData2(1, mydata, sizeof(mydata)-1, 0);
        Serial.println(F("Packet queued"));
       // float test = getTemperature();
        //Serial.println(test);
        i = 0;
        String stri = String(getTemperature());
        String sec = String(getEC());
        String sdo = String(getDO());
        for (int c = 0; c<stri.length(); c++){
          mydata[c] = stri.charAt(c);
          i = c;
        }
        mydata[i+1] = ',';
        i = i + 2;
        int j = i;
        //ec

        for (int c = j ; c<j + sec.length(); c++){
        mydata[c] = sec.charAt(c-j);
        i = c;
        }
        mydata[i+1] = ',';
        i = i + 2;
        j = i;
  //DO
        for (int c = j ; c< j + sizeof(sdo); c++){
        mydata[c] = sdo.charAt(c-j);
        }
```

Figure 53 : Arduino code that create and send the packet

Of course, we still needed make changes to be able to transfer the data we receive in the form of a packet through LoRaWAN, the method we used was to get the sensor data in the form of a String and then we had to add every character of the string individually to not cause problems with the “mydata” variable, we then separate every sensor data by a comma.

The most important code we created was the one that cycled through all six spreading factors in intervals of five minutes for reasons we explained in the progress part of the report.



```
if(cpt <= cpt1){
  //Serial.println(F("SF7"));
    LMIC_setDrTxpow (DR_SF7, 14);
    cpt = millis();
    x++;
} else if (cpt <= cpt2){
 // Serial.println(F("SF8"));
    LMIC_setDrTxpow (DR_SF8, 14);
    cpt = millis();
} else if (cpt <= cpt3){
 // Serial.println(F("SF9"));
    LMIC_setDrTxpow (DR_SF9, 14);
    cpt = millis();
} else if (cpt <= cpt4 ){
  //Serial.println(F("SF10"));
    LMIC_setDrTxpow (DR_SF10, 14);
    cpt = millis();
} else if (cpt <= cpt5){
  //Serial.println(F("SF11"));
    LMIC_setDrTxpow (DR_SF11, 14);
    cpt = millis();
} else if (cpt > cpt5){
  if(cpt > cpt5 + 300000){
    setMillis(0);
  }
  //Serial.println(F("SF12"));
    LMIC_setDrTxpow (DR_SF12, 14);
    cpt = millis();
}
```

Figure 54 : Arduino code that cycles through spreading factors

Finally, there is the housing of our device, but due to **time constraints** we were not able to fully print, build and test it before the deadline of the internship report.

Although, we weren't able to test the device in time for the results in the internship report we know that our sensors work, and we were able to at least confirm that the housing would float on water with all of the electronics inside it, while still being waterproof thanks to the print material we're using, the polypropylene which is a special waterproof material that we decided to use instead of polylactic acid (PLA) which is water resistant but not made to handle being constantly exposed to water.

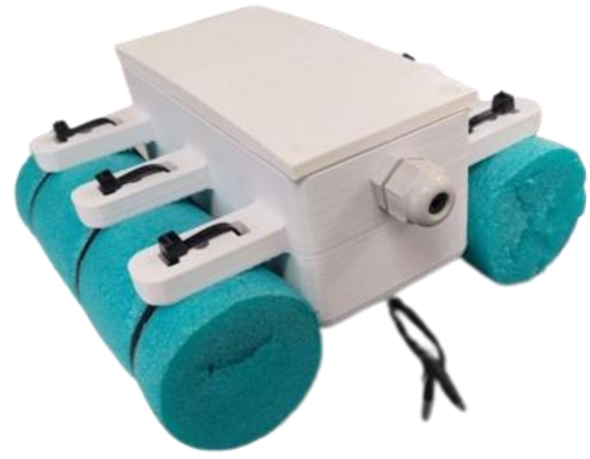


Figure 55 : Full device with tds and water temperature sensor

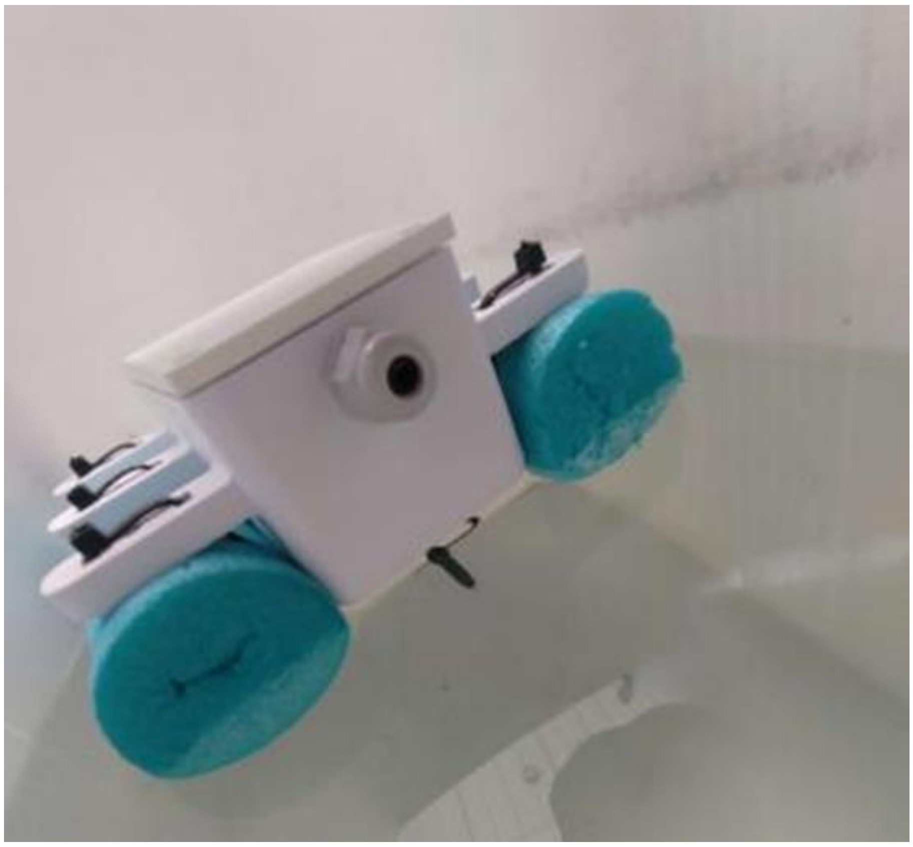


Figure 56 : Full device in water with sensors submerged

4. Assessments

4.1. Technical assessments

The technical assessment of this project is quite good as **most of the project was done by the time of the report**, we have a database with tables of different devices, a node-red infrastructure connected to our MySQL database, we possess prototypes for dashboard in multiple languages and web sites, we have **more than 800 lines of code for Arduino** and multiple Arduino boards wired to sensors.

Finally, we printed at least **one full housing for our device** in water proof polypropylene and are waiting to test it.

What remains of the project that is **unfinished is being worked on** : we aim to make a more sophisticated dashboard using either node-red or ThingSpeak, we also want to create a way to do data analysis with ThingSpeak.

We still have not been able to test our device at the time of the report but we are making good progress on it and we should have tests and results by the final week.

There is also, the solar panels which even if we made structures to accommodate them we have not tested their effectiveness and usefulness to our project yet, at the moment they are still being considered but they're left for last in our priorities as we want the device to be fully functional on battery alone first and then we will look at recharging the battery with solar panels.

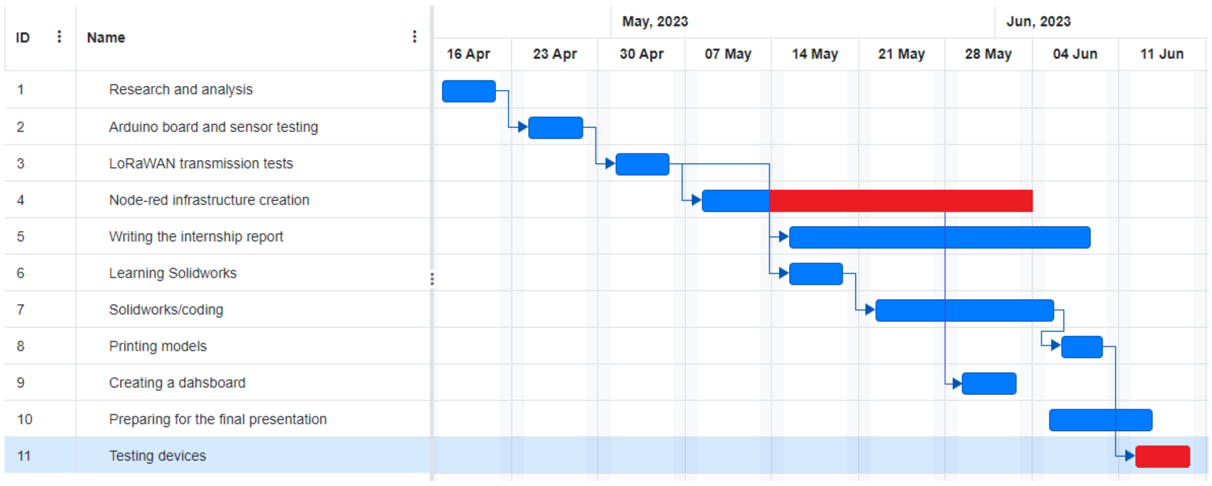
In the end our **project was useful to TU Dublin** as the housing is finished even if it can still be improved, the code we provided is also very useful and makes it easy to know which spreading factor and transmission power we want to use.

The sensors have been found, the database is working and our node-red infrastructure allows us to see the results.

When we will leave and when the next team will pick this project up **they won't have to start from scratch again**, they will be able to **improve** on what is already there and focus on what we couldn't finish in time.

4.2. Human assessments

In this internship I was **alone working on the water quality monitoring device**, with help sometimes from the air quality monitoring team and my project supervisor Mr. John Fox, most of the communication between us was made directly on site in lab 221 where we worked for the entirety of the internship, sometimes we also used outlook to communicate files such as STL files that are the files containing printable 3D models.



ID	Name
1	Research and analysis
2	Arduino board and sensor testing
3	LoRaWAN transmission tests
4	Node-red infrastructure creation
5	Writing the internship report
6	Learning Solidworks
7	Solidworks/coding
8	Printing models
9	Creating a dahsboard
10	Preparing for the final presentation
11	Testing devices

Figure 57 : Actual Gantt chart

As we can see on the actual Gantt chart most of the project's **planification was respected,** the node-red infrastructure creation took much more time than we thought it would, because while making the basic infrastructure was finished in time, we continuously wanted to **expand upon it** adding new nodes functions and tests that resulted an increase of the time it took to finish it.

Then the testing of the device which is still waiting to be done on the next and final week was pushed back due to problems with the prototypes prints that would need modification before testing.

4.3. Personal assessments

For me this internship was a **great experience** I was able to use some of my knowledge from the BUT like our lessons of IOT and **improve my skills** about the Internet of Things and the field of electronics with the use of electronic devices and circuits diagrams, but what was even more interesting was to learn a lot of new subjects that I did not know at all before such as Arduino and 3D modelling.

This internship also immensely helped me to perfect my English knowledge and pronunciation which had to be technical and fluid in order to make communication with my supervisors clear.

The internship also helped me to cement my choices on my degree, because even though the project was very interesting and enjoyable, I am not in my element doing IOT, 3D modelling or electronics like I am when **I am coding applications** and such in my computer science degree.

5. Conclusion

The goal of this project was to create a water quality monitoring device and a visualisation dashboard, to this end we had to first learn about the stakes of the project and the technologies that will be used, we then had to make all the Arduino board wiring, coding the sensors and the LoRa transreceiver.

We made **node-red and MySQL infrastructures**, to communicate, show and store all the data we would receive.

And finally, we learned all about Solidworks so that we could **design, create and print our own housing for the device**, we were able to create a functioning housing with a prototype for a bracket that can hold solar panels.

If this was to **be done again and for future work** references on this project I would consider making the housing a priority after research as it is the hardest and longest part due to the time to wait in between prints and learning how to make am efficient design, more focus should also be put on the solar panels, because as of now they are not a being tested or experimented with even though it would be a great help to the end goal of this project, the selected team should also be more specialised in IOT as I only had a few introductory lessons before starting this project which led to a lot of time being spent on researching the bases of IOT.

There is also always the possibility to add new sensors and to update the dashboard to have a more accurate depiction of the quality of the river's water.

Participating in this internship has been an **incredibly enriching experience**.

Taking part in a journey alone into a foreign country and applying the skills I gained during my studies at the IUT of Blagnac as well as learning new skills was truly rewarding.

Although it was very different to a regular internship, I enjoyed working on a project from scratch to completion.

Beyond work, the Irish culture is truly interesting and captivating, and my two-month immersion in this society will forever hold a special place in my memories.

6. Illustration table

7. Glossary

IDE : integrated development environment.
Internet of Things (IoT) : The interconnection via the internet of computing devices embedded in everyday objects, enabling them to send and receive data.
Library : Libraries in programming languages are collections of prewritten code that users can use to optimise tasks
SD card : A Secure Digital (SD) card is a tiny flash memory card designed for high-capacity memory and various portable devices.
Dashboard : A dashboard is a way of displaying various types of visual data in one place.
Microcontroller :A microcontroller is a compact integrated circuit designed to govern a specific operation in an embedded system.
Debug : The process of identifying and removing errors from computer hardware or software by seeing the internal results.
Bit rate : Describes the rate at which bits are transferred from one location to another. It measures how much data is transmitted in a given amount of time.
Chirp : A chirp is a tone in which frequency changes with time.

8. Bibliography/Webography

https://www.iberdrola.com/sustainability/water-pollution

https://www.epa.ie/our-services/monitoring--assessment/freshwater--marine/rivers/

https://www.semtech.fr/products/wireless-rf/lora-connect/sx1272

https://www.dfrobot.com/product-1782.html

https://www.dfrobot.com/product-1394.html

https://www.dfrobot.com/product-1123.html

https://github.com/DFRobot/DFRobot_EC

https://www.dfrobot.com/product-1662.html

https://github.com/DFRobot/GravityTDS

https://www.dfrobot.com/product-1531.html

https://www.thethingsnetwork.org/docs/lorawan/

https://www.epa.ie/our-services/monitoring--assessment/freshwater--marine/conditions-of-our-water/

9. Annexes

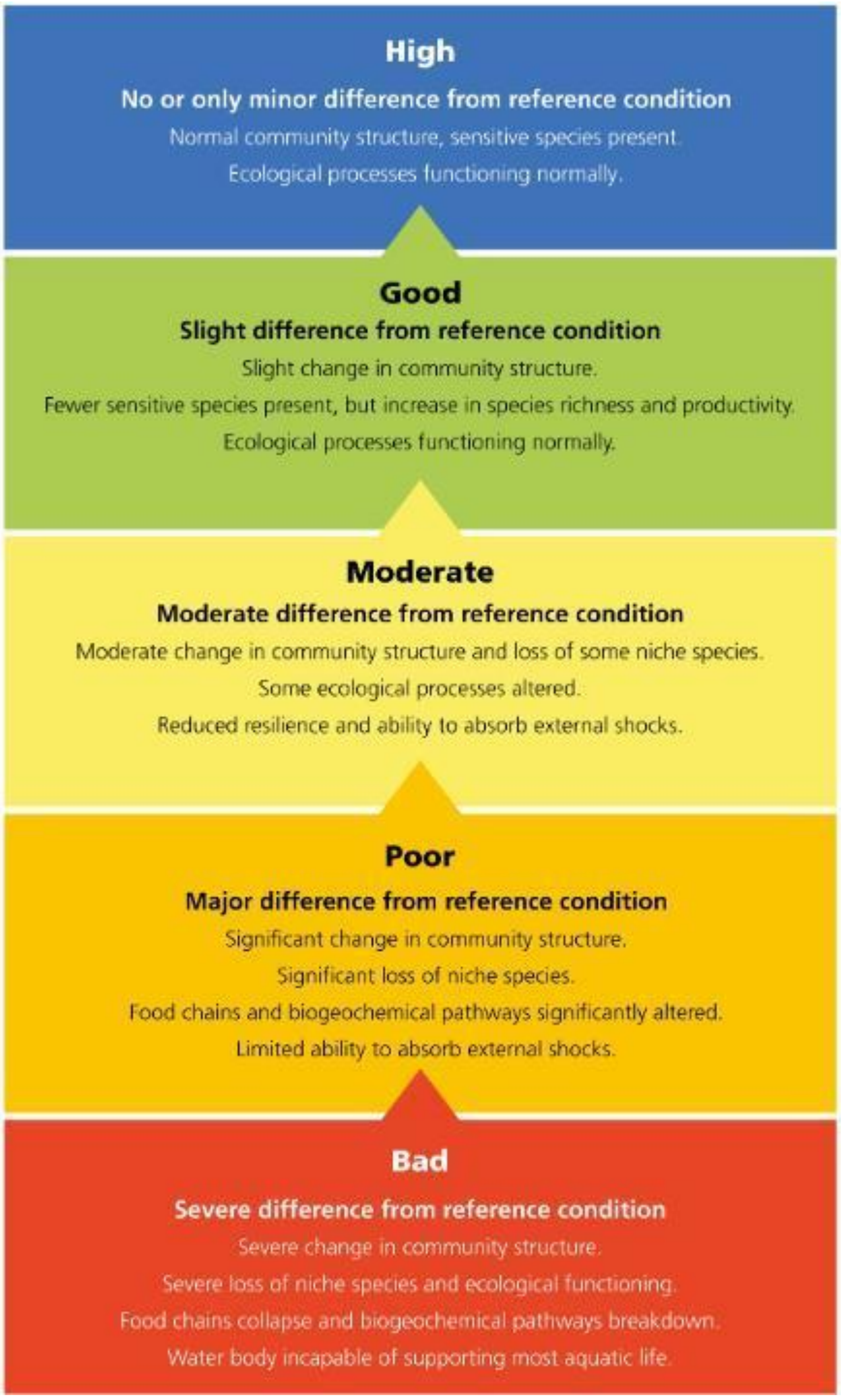

High
No or only minor difference from reference condition
Normal community structure, sensitive species present.
Ecological processes functioning normally.
Good
Slight difference from reference condition
Slight change in community structure.
Fewer sensitive species present, but increase in species richness and productivity.
Ecological processes functioning normally.
Moderate
Moderate difference from reference condition
Moderate change in community structure and loss of some niche species.
Some ecological processes altered.
Reduced resilience and ability to absorb external shocks.
Poor
Major difference from reference condition
Significant change in community structure.
Significant loss of niche species.
Food chains and biogeochemical pathways significantly altered.
Limited ability to absorb external shocks.
Bad
Severe difference from reference condition
Severe change in community structure.
Severe loss of niche species and ecological functioning.
Food chains collapse and biogeochemical pathways breakdown.
Water body incapable of supporting most aquatic life.

Figure 58 : WFD five quality classes of water surfaces

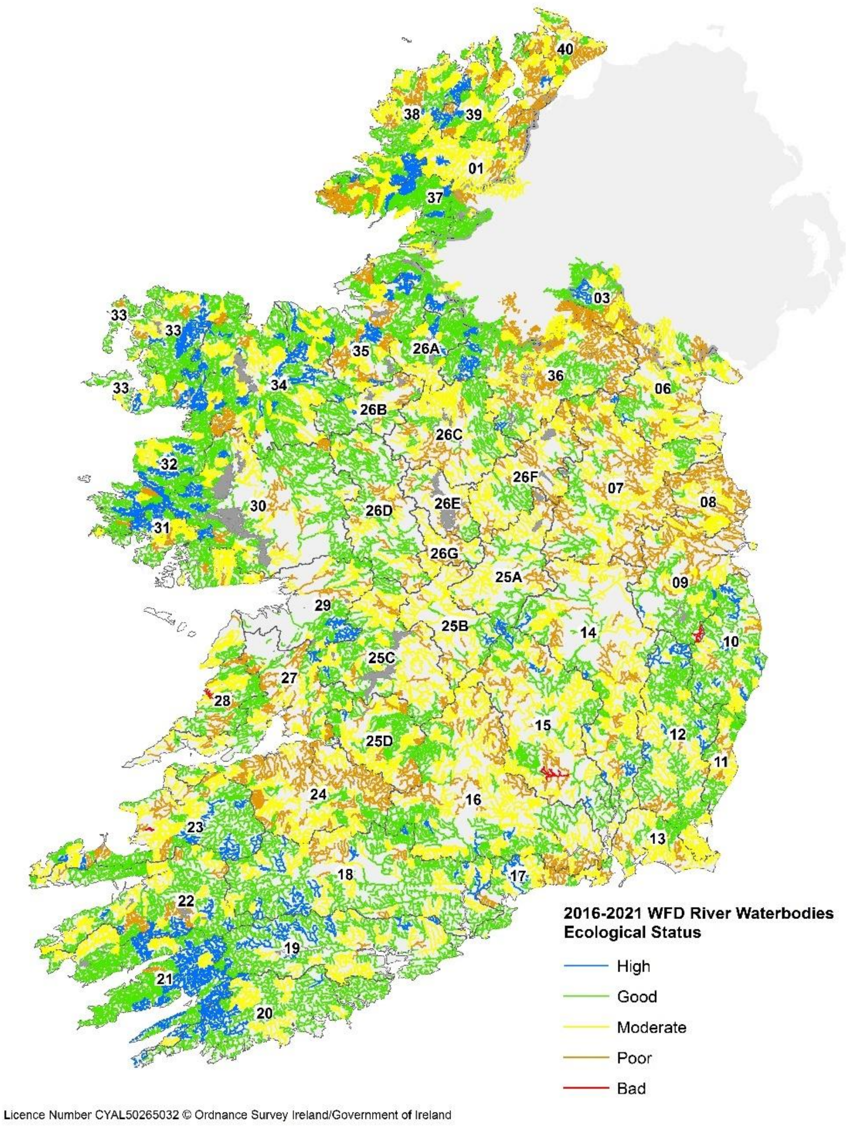

2016-2021 WFD River Waterbodies Ecological Status
High
Good
Moderate
Poor
Bad
Licence Number CYAL50265032 © Ordnance Survey Ireland/Government of Ireland

Figure 59 : The EPA's ecological status of rivers in Ireland from 2016 to 2021

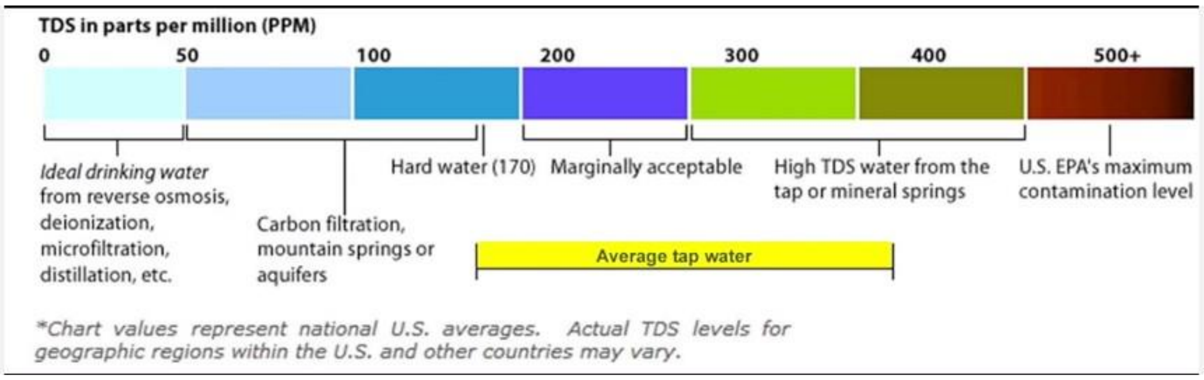


Figure 60 : TDS chart

10. Acknowledgment

First, I would like to express my deepest thanks to the Technological University Dublin organisation as a whole for giving me the opportunity to an internship in Ireland which is an amazing and eye-opening experience that allowed me to have a unique look into the professional world and my passion for programming as well as helping me perfect my language skills.

I would also like to express my thanks to Dr. Andrew Donnellan for entrusting me in the realisation of this project and for his guidance.
I would like to express my sincere gratitude to Mr. John Fox for guiding me with his all throughout the project and his supervision on the decisions to make, which greatly helped me to carry out this project.

And finally i express my deepest thanks to Ms. Isabelle Clavel, who offered me this chance to do an internship abroad that allowed me to get out of my comfort zone. I would like to thank the teachers of the IUT Blagnac who allowed me to acquire fundamental skills in computer science during my first two years thanks to their courses.