2nd delivery

- ***The comments that you received during the 1st delivery and how you have addressed them (if you decided/managed to address them). You should justify your decisions.***

The main criticism of the previous project presentation concerned the use of the mesh network and the Telosb device as a sensor node. I received this criticism for the main reason of the Telosb programming difficulties. I also got the critique regarding the limited range of the Zigbee network, which is around a couple of hundred meters. This would mean that the Telosb closest to the fixed station is no more than a couple of hundred meters away and this can in some cases be too restrictive.

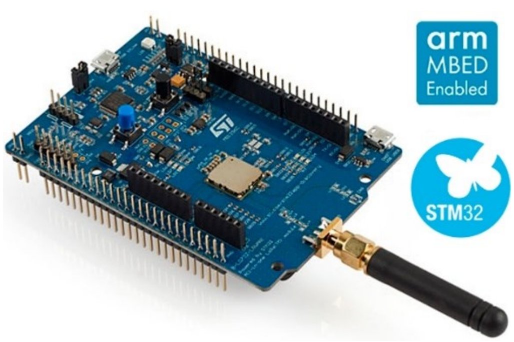
The advantage that the Zigbee protocol would have had instead would have been a certain autonomy and choral functioning as, given the conformation of some aquaculture facilities, they are mostly adjacent at distances of a few hundred meters and require communication between them. (e.g. oxygen, ph, pollution level data) can be self-regulated by exchanging local information between nodes and delegating certain functions to the single node edge instead of to the cloud. Furthermore, an advantage is the complete free of charge of this network, and greater security from external geographic attacks.

It was finally decided to opt for the LoRaWAN network, purchasing b-l072z-lrwan1 as a LoRaWan module, and iC880A-SPI concentrator board as a Gateway. The choice of the Gateway was made as it ensures that we can be connected to the network in any location, since the Things Network gateways can be scarce on the coast.

- ***Changes made to your concept, architecture and evaluation plan since the 1st delivery. You should justify the need for these changes.***

A)

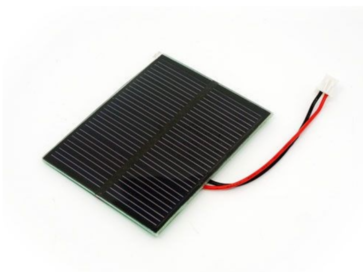
As stated before the architecture has changed since it is not used anymore the TelosB , but the sensor and actuators connected to the main board. In order to use LoRaWAN we have adopted **b-l072z-lrwan** as a final board.



B)

In order to harvest energy it has been envisaged a fotovoltaic device. This Small Solar Panel

55x70mm 0.5W is made of single-crystal material that performs high solar energy transformation efficiency. It has a fine resin surface and sturdy back suitable for outdoor environments.



It can be buyed here https://www.seeedstudio.com/0-5W-Solar-Panel-55x70.html

C)

In order to store the energy harvested with the solar panel this is a very small, extremely lightweight battery based on Lithium Ion chemistry, with the highest energy density currently in production has been chosen. Each cell outputs a nominal 3.7V at 400mAh! Comes terminated with a standard 2-pin JST-PH connector with 2mm spacing between pins. These batteries require special charging; do not attempt to charge these with anything but a specialized Lithium Polymer charger.

It can be buyed here
https://www.sparkfun.com/products/13851

D) iC880A-SPI

This device, in combination with an embedded Linux board like Raspberry Pi, Beagle Bone, Banana Pi and the HAL software from Github a complete LoRaWAN® gateway can be set up easily.

This component of the architecture is not strictly necessary, as it is often possible to connect directly to The Things Network. Only in cases where a node of this network is unavailable, will it be necessary to have a concentrator. For this reason the concentrator in question will be purchased and partially tested.

It can be buyed here https://shop.imst.de/wireless-modules/lora-products/8/ic880a-spi-lorawan-concentrator-868-mhz?c=12.

Here you can find more info on further devices needed Home · ttn-zh/ic880a-gateway Wiki (github.com)

E)

The Smart Buoy

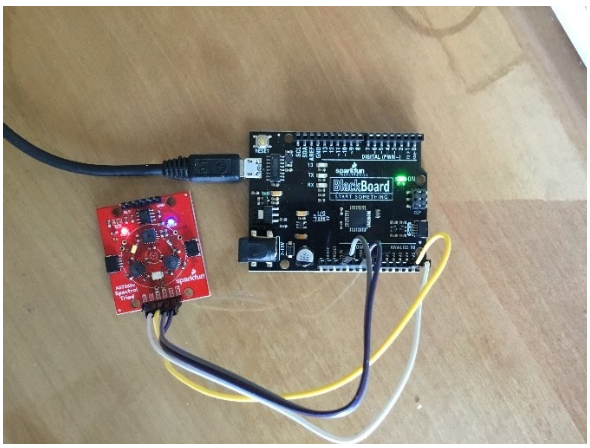
A construction project for a buoy which is used to install all the devices is reported on the site referred to in the link below.

https://www.hackster.io/amerch92/lora-weather-buoy-9e739b

-
-
- ***Brief presentation of the technical work done since the 1st delivery.***
-

The work starting from the first delivery involved the development of an application for RIOT-OS that reads and displays the data of the spectrometer. I focused on the I2C interface mode that can be activated via menu commands from the shell command line. The C program interfacing with I2C **is not complete.**

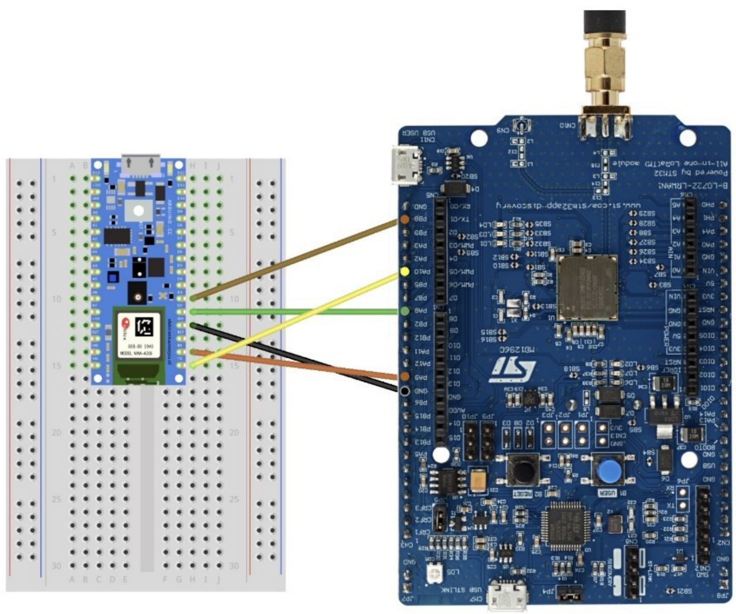
The readings of the spectrometer were however made possible through the libraries available for Arduino. So this last device was connected to the sensor and repeated scanning tests of water samples were made.



Therefore, the use of arduino (Sparkfun BlackBoard) was temporarily envisaged in order to use the existing libraries. This board will be connected with the b-l072z-lrwan which will carry out the communication functions of the network.

Arduino will be used as long as the application interface is not ready on Riot-Os.

In order to communicate between Arduino e b-l072z-lrwan1 please see https://github.com/2ni/lorawan modem/ .



- ***A brief list of the functionality that is still missing and will be done till the final delivery.***

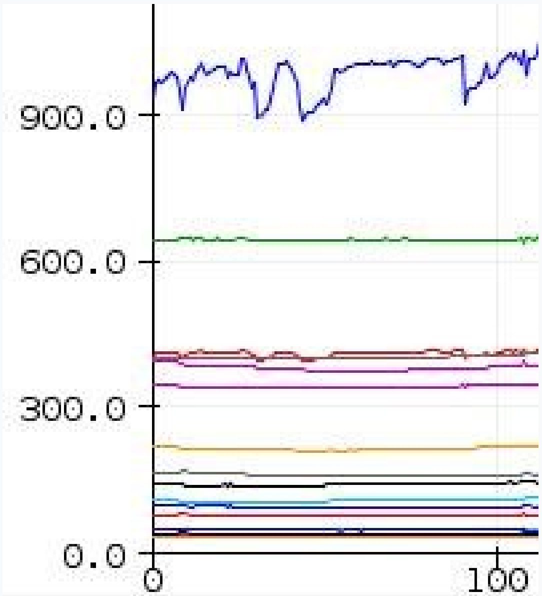
Missing features for final delivery is the completion of the data reading program in Riot -os. In fact, Arduino is currently used and then it is connected to the LoRaWan module to send data to the concentrator or Gateway. It is therefore necessary to implement the connection to The Things network and transfer the data to the Amazon Cloud where they can be viewed in a dashboard. This last module has already been realized in the first individual assignment and thus can be considered reusable.

- ***Brief presentation of the evaluation conducted since the 1st delivery.***

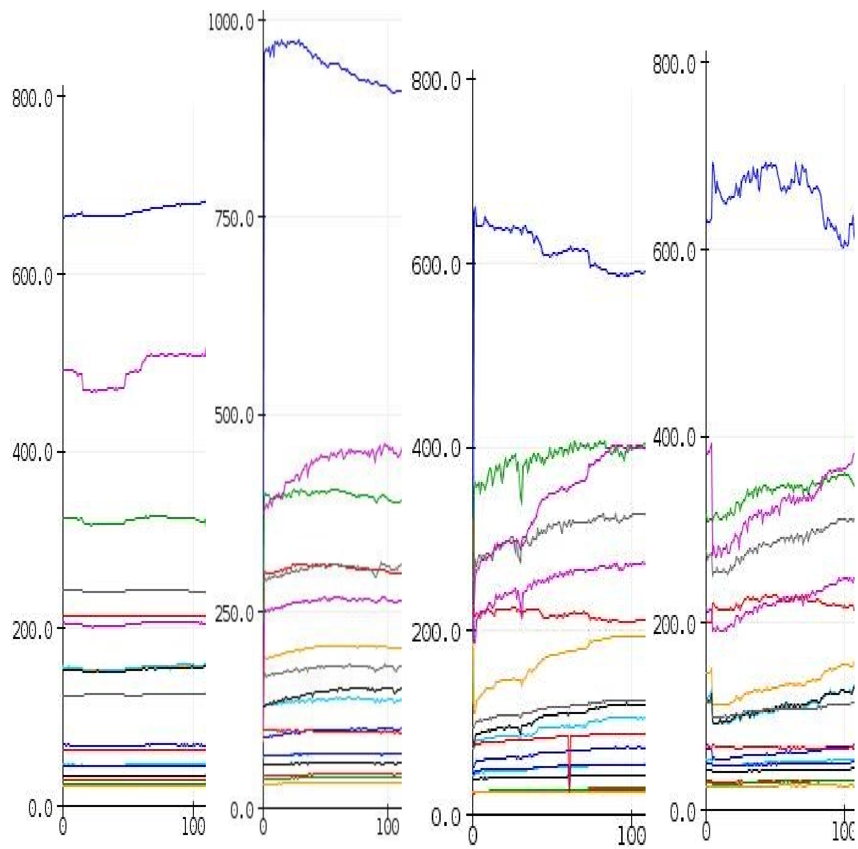
So far, experiments on the spectrometer have been carried out thanks to the Arduino libraries. The goal was to do feasibility tests on the spectrometer's ability to make real-time measurements directly in water. Obviously it is assumed that the spectrometer can be inserted directly into the water to measure the quantities of interest. For the sake of simplicity and testability the spectrometer was firstly used as in the laoratory. Instead of a quartzum cuvette the liquid was filled into a glass jar and the spectrometer was attached to the jar to take the measurements of the contained liquid. Another approximation was to subject water samples added with additives to test the ability of the spectrometer to identify unknown samples in the composition and percentage of presence of substances harmful to the life and development of fish and the surrounding environment.

To simulate variations in water salinity, salted was added to the sample. In order to simulate variations in Ph, sodium bicarbonate or vinagar was added to the sample. In order to simulate pollution oil was added to the sample.

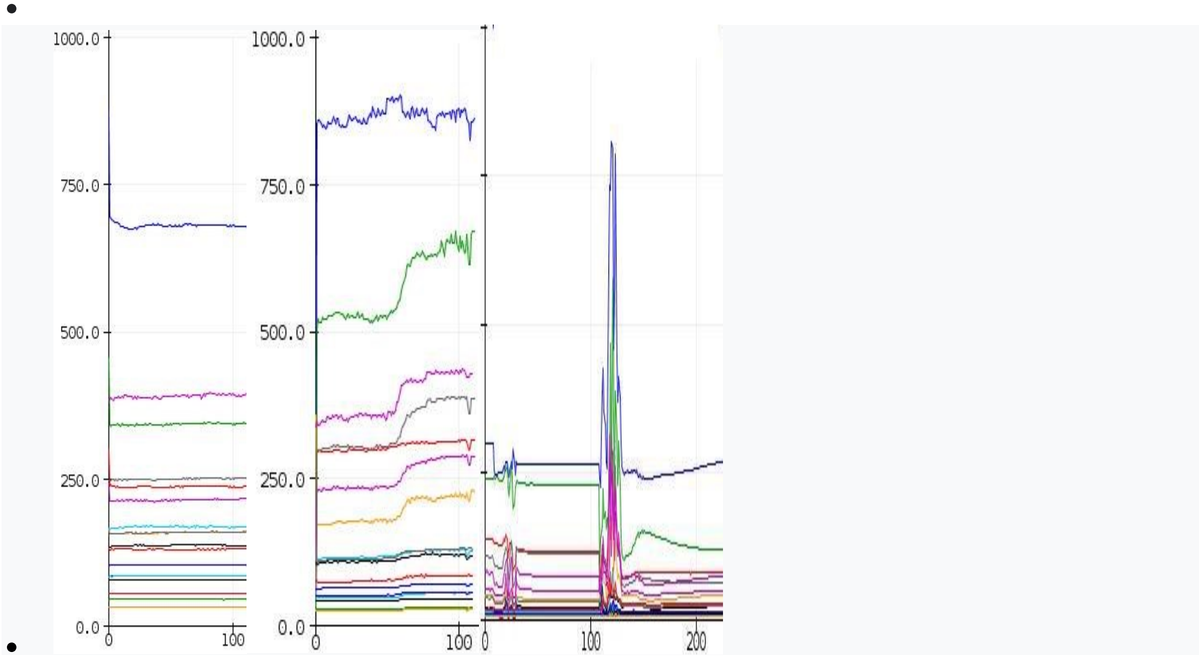
Some of the experiments carried out are shown below.



- Figure 8 Sample of pure drinkable water



- Figure 9 water with salt added
- Figure 10 water with dejections added
- Figure 11 water with oil, dejections, salt added
- Figure 12 water with oil, dejections added



- Figure 13 water with oil added
- Figure 14 water with salt, dejections, bicarbonate added
- Figure 15 with bicarbonate added during measurement

It would be desirable to complete the data collection and the construction of the neural network model for the recognition of unknown water samples but it is believed that this is beyond the scope of this project.
A functionality that could be tested, even if at a later stage of the project, outside of this course, is the compatibility with the functions of Artemis (Machine Learning module) and Tensorflow.

https://www.sparkfun.com/artemis

https://www.sparkfun.com/products/15444

https://www.amazon.com/SparkFun-Development-Connector-Microphone-TenserFlow/dp/B07YCVB6LL

Some Comments and Considerations on the work to be done

Since the spectrometer sensor was initially set up with an STM32F401 controller the 1st thing you need to do is write the functions needed to communicate. From the datasheet you know that there are only 4 registers - the device address, status, read, and write. To read or write you need to read and write the virtual registers and provide a "virtual register" address to get to the register you want. This is rather uncommon among similar devices. There is some example code in the data sheet, which is helpful. So if you are not using this with an Arduino you need to download the AS726X header and implementation files (.h and .ccp) from github provided. The library In c++ can be written by using these templates.

https://github.com/sparkfun/SparkFun_AS726X_Arduino_Library/tree/master/src

They should be used these as a guide to write the functions.