

User guide

connected beehive

DELAS Léo | DOS SANTOS Damien | DURAND DE GEVIGNEY Basile | TAN Caroline

EI2I 4 II – 19/01/2022

We are very proud to present you our user manual to implement our project of connected beehive.

You will be able to find all documents, sources codes and other useful files of the project on the following GitHub:

<https://github.com/Ithoh/OpenRucheDLCB>

THANKS, AND ENJOY !!!!

Summary

[I – Required components 3](#_Toc93486301)

[II – PCB and connecting everything 4](#_Toc93486302)

[III – Code 6](#_Toc93486303)

[IV – Sigfox connection 9](#_Toc93486304)

[V – Ubidots 10](#_Toc93486305)

[VI – Installing the system on the beehive 12](#_Toc93486306)

[Conclusion 14](#_Toc93486307)

# I – Required components

For this project, you need to have these components:

|  |  |
| --- | --- |
| Solar cell of 5,5 V/360 mA and dimensions of 180 x 80 m*m* |  |
| Power adapter card (LiPo Rider Pro) |  |
| Arduino MKR FOX 1200 board ABX00014 with SigFox interface |  |
| Li-Ion battery 3,7V 1050 mAh |  |
| Multiple temperature sensors |  |
| Weight sensor: strain gauge and HX711 |  |
| Temperature and humidity sensor DHT22 |  |

# II – PCB and connecting everything

For our project we have decided to design a PCB to connect all our components, for this we used the software EasyEDA, which is a free software that allows you to do some PCB design and also some electrical schematics and others functions that we haven’t used in this project.

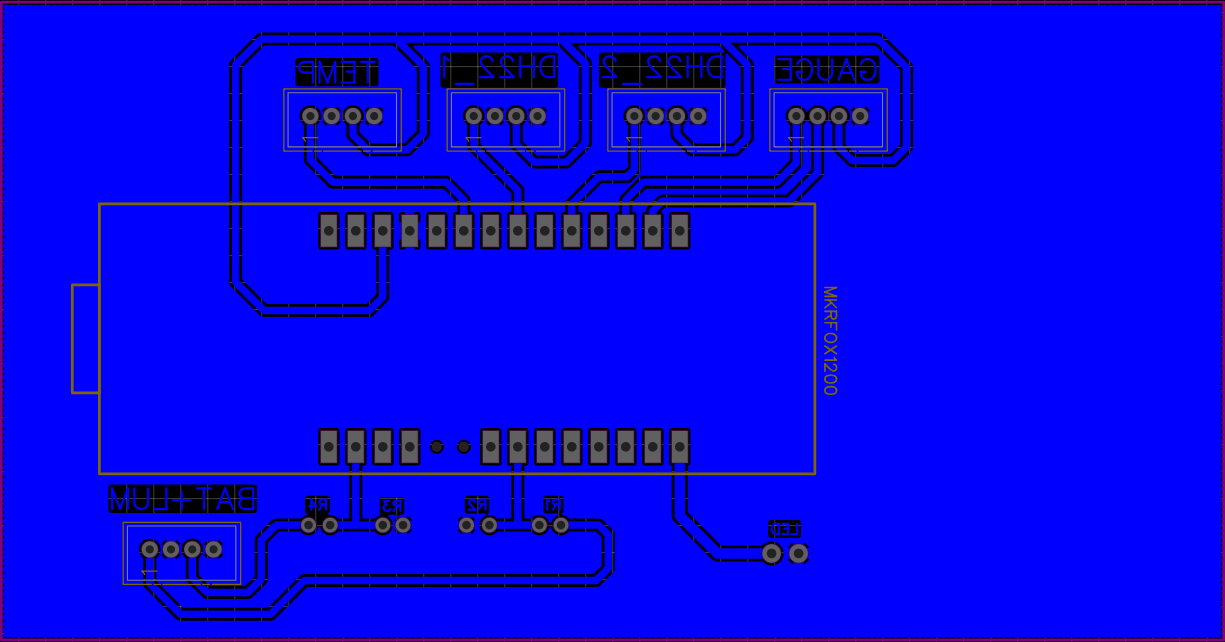
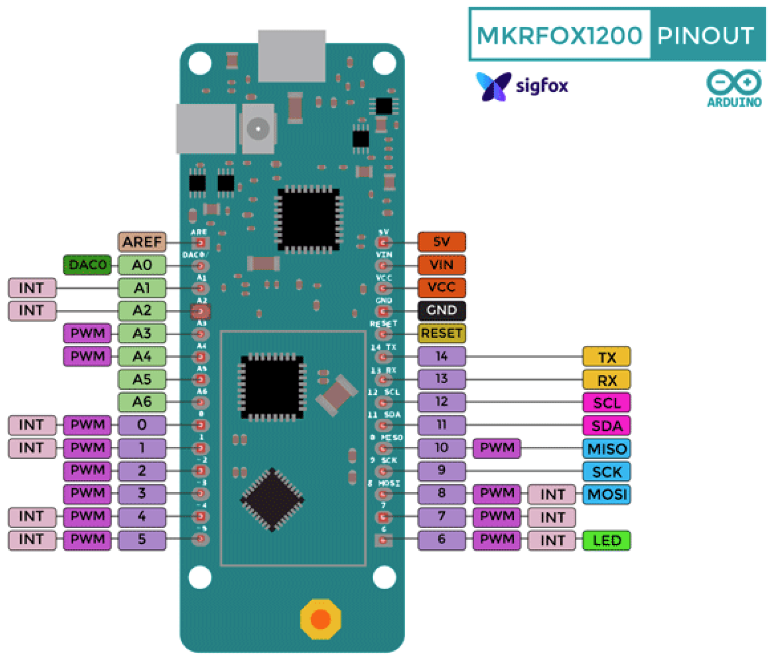
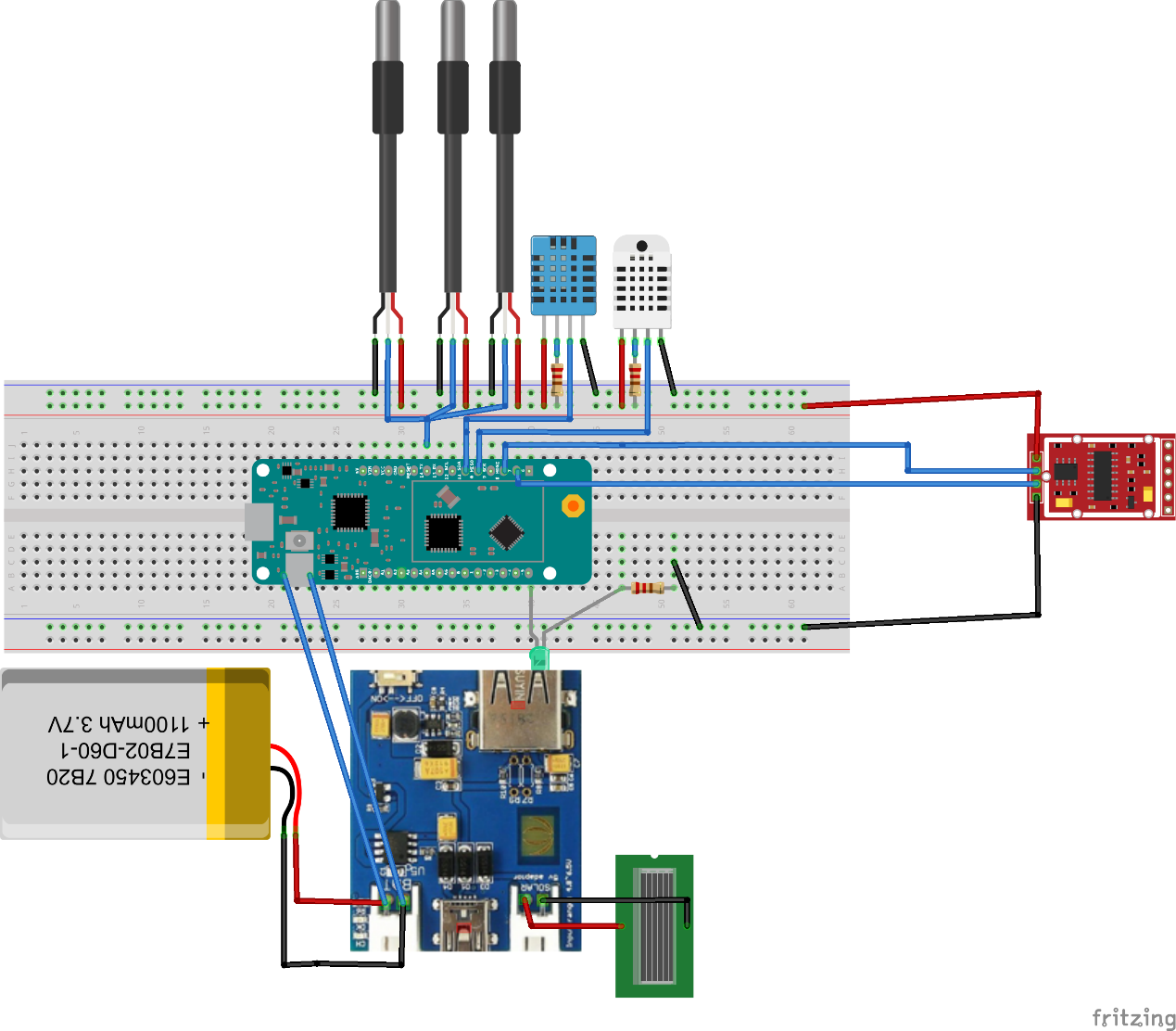


Figure 1 - PCB of the project

Everything is written on the PCB on how to link everything. It is purposely reversed so the print is not reversed and can be easily read.  
The way we put our Arduino board here, is by using some female pin header to easily swap out the board if needed while maintaining some robustness.  
The pinout of the maker fox is as follow:

Here is a schematic on how our project was implemented on a breadboard for tests:



You can easily find our PCB files here: <https://github.com/Ithoh/OpenRucheDLCB/tree/main/Ressources/PCB_Schematic>

# 

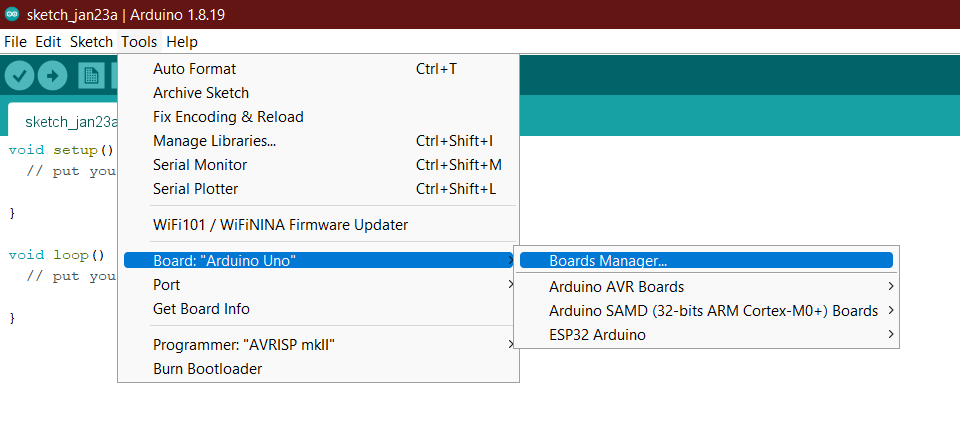
# III – Code

In this project we will use Arduino IDE to program the MKRFOX board.

You need to install the Arduino IDE and download the codes from our GitHub: <https://github.com/Ithoh/OpenRucheDLCB/tree/main/Codes>

Before uploading our code, you need to prepare your IDE installing all needed dependencies.

First of all you need to access to the boards manager via:   
Tools->Board: “…” ->Boards Manager.



And searching “MKRFOX”, you will be able to install the following add-on:

Une image contenant texte

Description générée automatiquement

Be sure to install “1.8.11” Version, we encountered some troubles using later version.

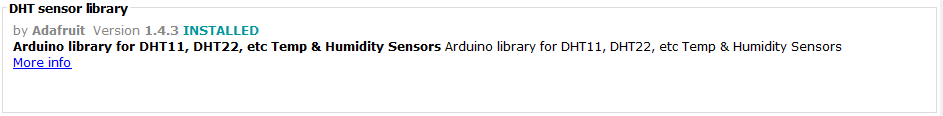
After that you need to install all following libraries used for sensors:

Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

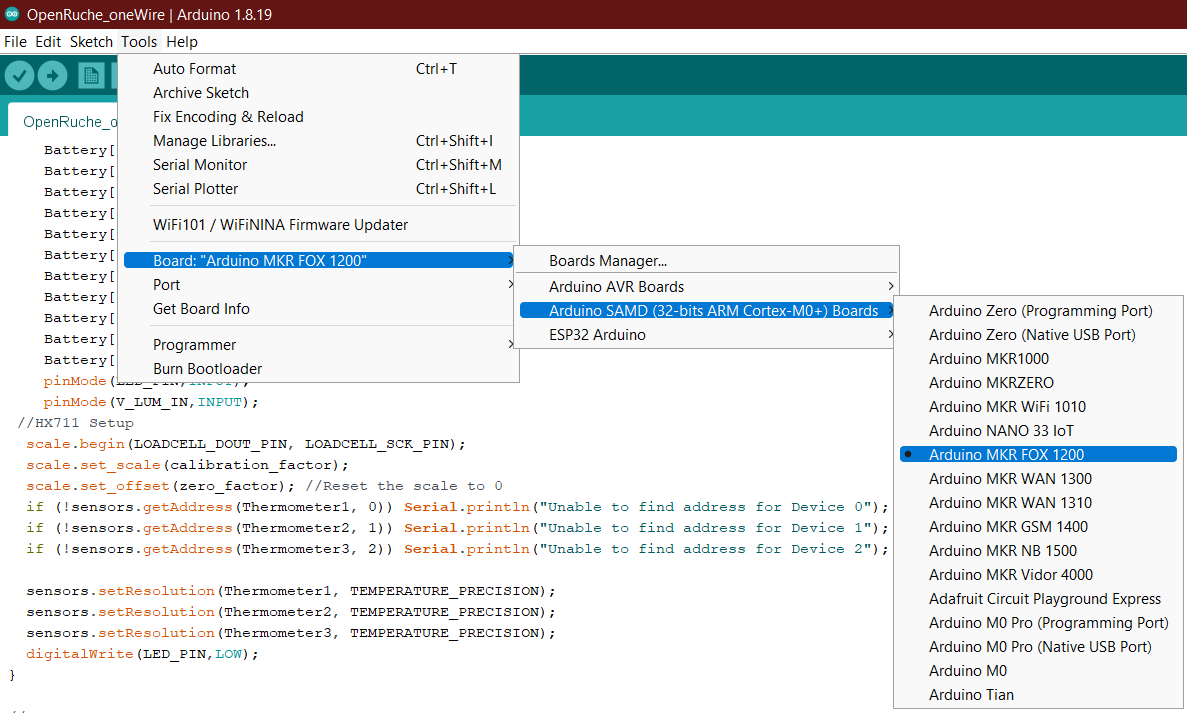


Une image contenant texte

Description générée automatiquement

If the software asks to install dependencies, do it, it is needed to the project.

Now, you will be able to compile the project without issues, to do it choose the right board in Tools->Board button as showed in the picture bellow:



And then select the port (the name of the board should be shown beside “COMx”

Our code:

Our codes are divided between the main code that are needed for the project:

<https://github.com/Ithoh/OpenRucheDLCB/tree/main/Codes/Programmes_principaux/OpenRuche_oneWire>

And the test codes to check whether the sensors are working or not :

Une image contenant texte

Description générée automatiquement

At first, you should test the sensors through compilation and then upload the code on the Arduino. Do not forget to select the right Arduino and the right port.

After checking that everything is working correctly, you can compile and upload the main code on the Arduino.

You can find our project on our github, the only things you could have to do it to change port of your sensors as you plug yours on the board.

<https://github.com/Ithoh/OpenRucheDLCB/tree/main/Codes>

# IV – Sigfox connection

To access the data that is recovered, you need to log in your account Backend Sigfox with your login. We recommend using a Yopmail account if the use of the beehive is for a school project, otherwise your regular mail is totally fine.

If it is for school purposes, the name of the mail should be the one on the Arduino and the password should be something easy to remember like below :

|  |  |
| --- | --- |
| Mail | 001d80XX@yopmail.com |
| Password | Sigfox21# |

On the website, you can check the messages that are sent every ten minutes by clicking on: Device, then Messages.

You should be able to see this after following these steps:

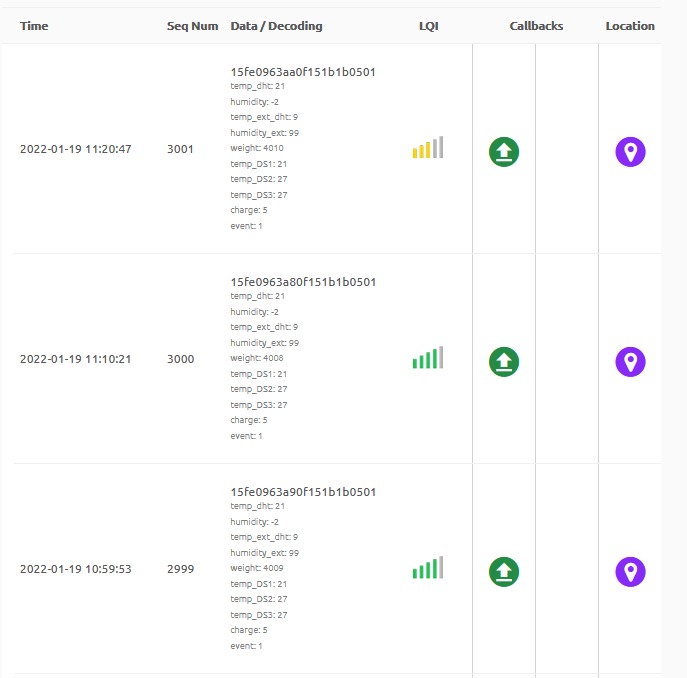


Figure 2 - Sigfox messages

If you want to know more about how Sigfox can read these messages, it is thanks to the parameters of the callback which you can access via Device Type, Callbacks and editing the only one that is there:

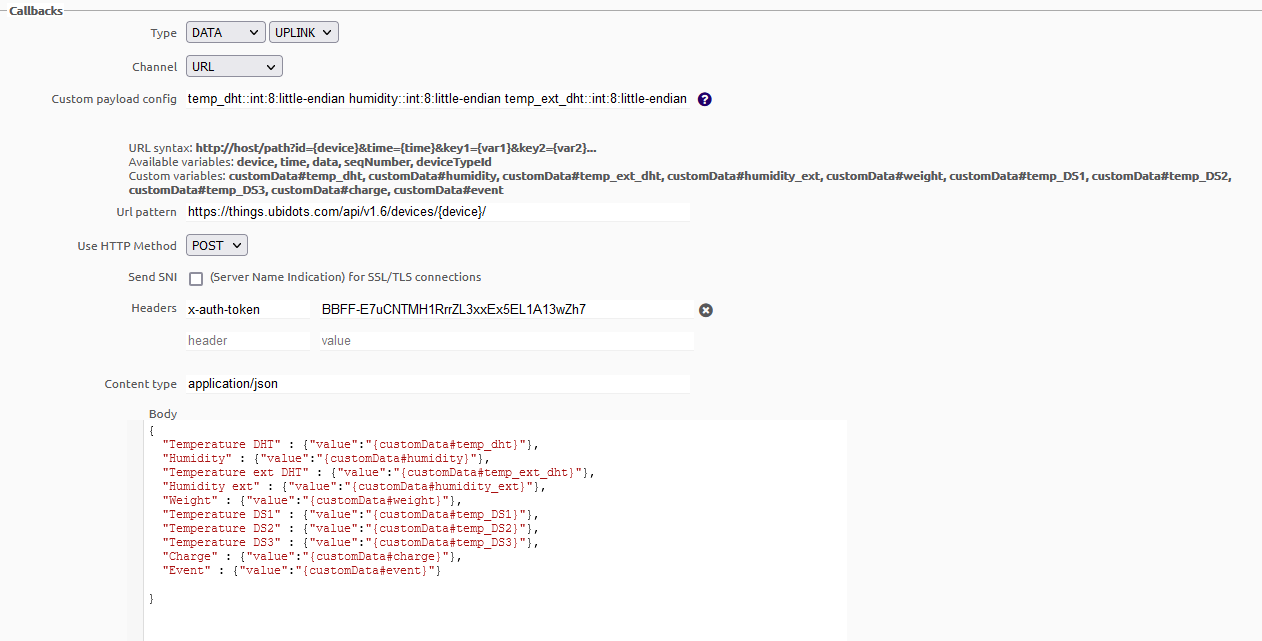


Figure 3 - Screenshot of the callbacks

# V – Ubidots

First, you need to create an account on Ubidots, we recommend to use the same name on that website to not lose the login. For this project, we have decided to use the same password on Sigfox and Ubidots.

To connect to this website, the logins are :

|  |  |
| --- | --- |
| Login | 001d80XX |
| Password | Sigfox21# |

After that, you need to add your device, which should be named like the login if it is your school project. In our case, we selected “Blank device” on Ubidots.

When this is done, you can access your device and its token. This is very important because it will be the link between Sigfox and Ubidots so it is unique and should be kept secret.

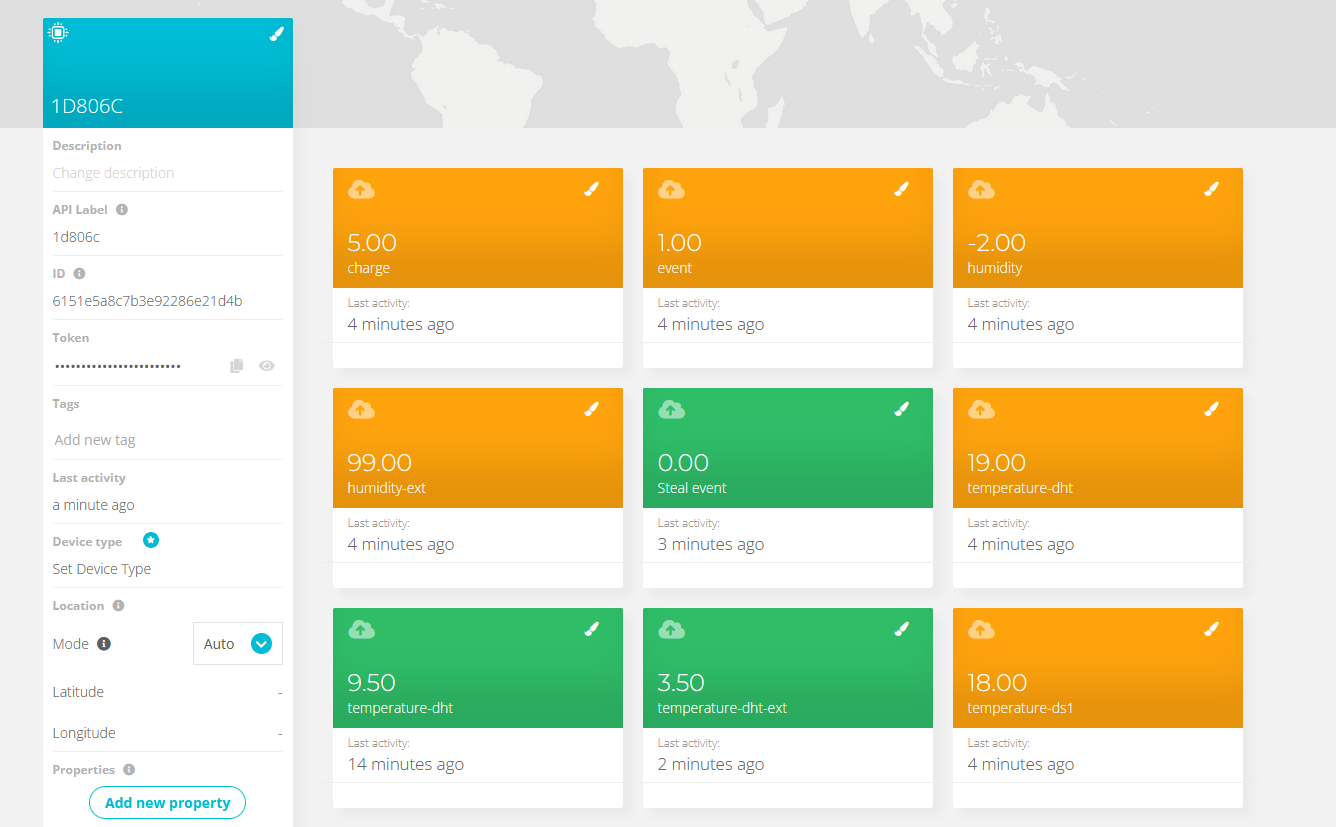


Figure 4 - Exemple of our device on Ubidots

On this screenshot, you can see the different variables created in figure 3. You need to add them by clicking on “Add variable” and then on “Raw” :

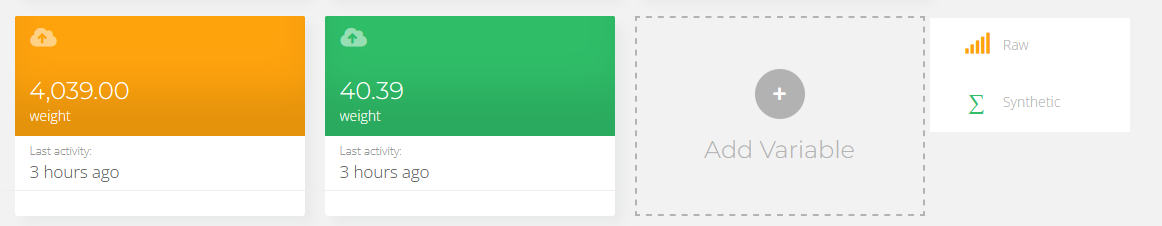


Figure 5 - Screenshot of Ubidots

After that, you need to change the name for it to correspond to the ones gave in the callbacks made in Sigfox.

And after configurating all variables, you can see the inputs sent from the Arduino to Sigfox and then to Ubidots.

Thanks to this website, you can see charts of the temperature and humidity, the battery level and the weight of the beehive:



Figure 6 - Information of the beehive through Ubidots

With this website, it will send you alerts via mail and text message if your battery level is too low or if your beehive has been stolen.

Thanks to this interface, you can monitor the condition of your beehive and if everything is functioning properly.

# VI – Installing the system on the beehive

The system on the beehive is set up like this :

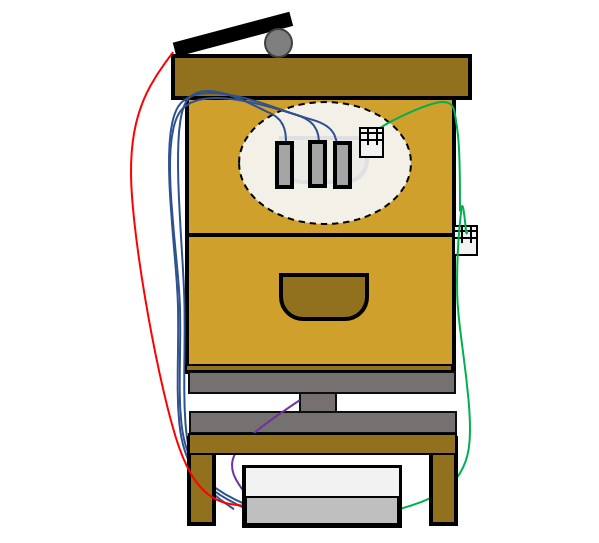


Figure 7 - Schematic of the beehive with the sensors

There are three temperature sensors and one DHT22 in the inside of the beehive and one DHT22 on the oustide. On the top of the beehive you have the solar cell. And it is on the weight gauge :

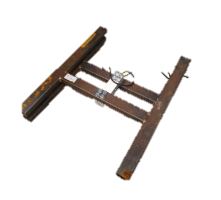


Figure 8 - Weight gauge

More precisely, the sensors are placed like this :

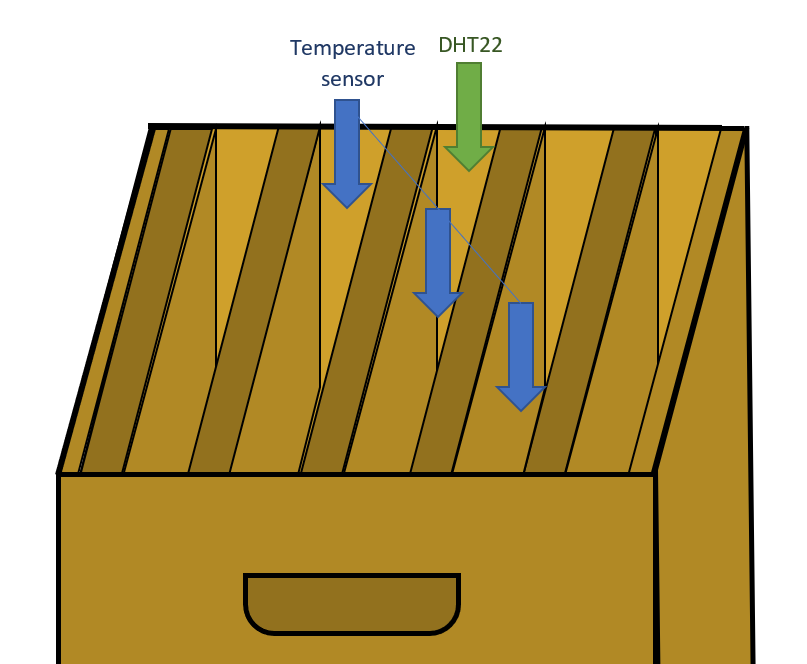


Figure 9 - Detailed position of the sensors

# Conclusion

With this project, you should be able to monitor the status of your beehive and take action if needed for the well-being of the bees !

Thank you for reading this user guide and hope that everything is clear !