

## Projet de ruche connectée B-Smart

 **“B-Smart, la ruche qui veut plus!”**

 **“B-Smart, the hive that is seeking for more!”**

---

Année : 2020-2021  
Ecole : Polytech Sorbonne  
Spécialité : Electronique et Informatique,  
parcours systèmes embarqués  
(année 4)


Supervisé par :  
Yann Douze  
Tarik Larja  
Sylvain Viateur


Equipe composée de :  
Maxime Lebreton  
Nikolaï Birolini  
Hadrien Gourdet  
Alexandra Hulot

# Sommaire / Summary


|   |           |
|---|-----------|
| <b>Contexte / Context</b>   | <b>2</b>  |
| <b>Solution</b>   | <b>2</b>  |
| <b>Capteurs &amp; Grandeurs physiques / Sensors &amp; Physical Quantities</b>           | <b>3</b>  |
| <b>Choix des capteurs / Sensors choice</b>  | <b>4</b>  |
| <b>Schéma/ Schematic (disponible en annexe / available in the appendix)</b>             | <b>4</b>  |
| <b>PCB</b>  | <b>5</b>  |
| <b>Transmission des données / Data transmission</b>                                     | <b>8</b>  |
| La 0G / The 0G  | 8         |
| Module réseau / Network module  | 8         |
| Description du module / Module description  | 8         |
| Activation du module / Module activation  | 10        |
| Envoi des données / Data transmission   | 11        |
| Présentation générale / General presentation  | 11        |
| Fonctionnement  | 12        |
| Réception des données / Data Reception  | 13        |
| <b>Stockage des données sur le cloud/ Data storage</b>                                  | <b>14</b> |
| <b>Visualisation des données / Data visualization</b>                                   | <b>15</b> |
| <b>Autonomie / Self sufficiency</b>   | <b>15</b> |
| <b>Conclusion</b>   | <b>16</b> |
| <b>Annexes</b>  | <b>18</b> |
| Schémas PCB / PCB Schematics  | 18        |
| Liste des composants / Components list  | 20        |
| Coûts / Costs   | 22        |
| Configuration du callback sur Sigfox Backend / Callback configuration on Sigfox Backend | 24        |

## Contexte / Context

 Les abeilles sont des insectes pollinisateurs, elles assurent la reproduction de plus de 80 % des espèces végétales du globe. Comme le disait Albert Einstein ; « Si l'abeille disparaissait de la surface du globe, l'homme n'aurait plus que quatre années à vivre ». De ce fait, il est essentiel de préserver cette espèce afin de protéger à la fois notre écosystème et d'assurer la longévité de notre espèce. surface du globe, l'homme n'aurait plus que quatre années à vivre ». De ce fait, il est essentiel de préserver cette espèce afin de protéger à la fois notre écosystème et d'assurer la longévité de notre espèce.


 Bees are pollinating insects that reproduce more than 80% the world's plant species. As Albert Einstein said "If the bee disappeared from the face of the earth, man would have only four years to live." Therefore, it's essential to preserve this species in order to protect our ecosystem and ensure the longevity of our species

## Solution

 Il est impossible à notre échelle de changer considérablement le climat ou de mettre fin aux pesticides.

Cependant, il est possible de surveiller le bien-être et la santé des colonies afin de déterminer les environnements propices à leurs proliférations.

Notre ruche connectée B-Smart permettra cette surveillance à distance et garantira aux apiculteurs une télésurveillance de leurs abeilles tout en améliorant leurs productivités.

 It's impossible on our scale to stop the climate change or put an ending to the use of pesticide.

However, it's possible to monitor their health and their behaviour to avoid the end of the colony.

Our connected hive B-Smart will ensure this surveillance and will optimise the productivity of beekeepers.

De plus, cette ruche sera open-source ([https://github.com/Polytech-Sorbonne/EI-SE4\\_2020-2021\\_OpenRuche\\_B-Smart\\_G4](https://github.com/Polytech-Sorbonne/EI-SE4_2020-2021_OpenRuche_B-Smart_G4)), c'est-à-dire qu'elle appartiendra entièrement aux apiculteurs et pourra être modulée librement.

Mais qu'est ce qui détermine le bien être des abeilles ? Pour protéger les colonies apicoles, il nous faut définir les différentes grandeurs physiques à surveiller.

Ainsi, nous avons décidé de surveiller :

- La température intérieure de chaque cadre de la ruche
- La température extérieure
- L'humidité interne de la ruche
- La luminosité reçue
- La force du vent et sa direction
- Son poids
- Le son

Moreover, this hive will be open-source ([https://github.com/Polytech-Sorbonne/EI-SE4\\_2020-2021\\_OpenRuche\\_B-Smart\\_G4](https://github.com/Polytech-Sorbonne/EI-SE4_2020-2021_OpenRuche_B-Smart_G4)), so all rights will be given to beekeepers. They can change anything they want and have guaranteed free access to the system.

What determines the well being of the bees? To answer this, we firstly need to find physical parameters to modelise an healthy environment for our bees.

So, we decided to use :

- Internal temperatures per frame
- External temperatures
- Internal humidity
- External luminosity
- Wind strength
- Wind direction
- Hive's weight
- Internal sound

## Capteurs & Grandeurs physiques / Sensors & Physical Quantities



Ces paramètres ont été déterminés grâce à une longue recherche bibliographique.

La mesure de la température sur chaque cadre permet de déterminer approximativement le nombre d'abeille dans la colonie.

La mesure de la température externe permet de vérifier si l'isolation de la ruche est fonctionnelle.

La mesure de l'humidité dans la ruche permet de prévenir l'apparition de certaines maladies (notamment la Nosemose qui scelle le destin de la ruche car elle est incurable) si jamais la ruche est trop humide.



With an intensive bibliographical research, we choose these parameters.

The internal temperature measurement per frame allows the system to approximately calculate the population of bees.

The external temperature measurement allows the system to ascertain the thermal isolation of the hive.

The measure of the internal humidity prevents the apparition of certain diseases (such as Nosema, an incurable disease).

Le luminosité, la force du vent et sa direction aident l'apiculteur pour le placement de sa ruche. Par exemple, si un vent est trop fort la quantité de pollen récoltée sera plus faible.

La mesure du poids est cruciale. Elle permet de visualiser les miellées ou de notifier l'apiculteur des baisses de poids anormales

Enfin, la mesure du son est une donnée permettant de déterminer le comportement de la colonie. Notre ruche connectée sera capable de déterminer 4 comportements à l'aide d'une analyse du spectre sonore à l'intérieur de la ruche :

- Colonie fonctionnelle
- Colonie sans reine
- Colonie stressée
- Colonie en plein essaimage

Then, these values will inform him about the external environment of his hives.

For example, if the wind is too strong, the harvest of the pollen will be poorer.


The weight of the hive is the heart of the B-Smart system. It's used to quantify the honey flow and notify beekeepers about anormal weights.

Finally, the sound will define the behaviour of the colony.

B-Smart detects 4 behaviours :

- A functional colony
- A colony without a queen
- A stressed colony
- A swarming colony

## Choix des capteurs / Sensors choice

 Le choix des capteurs a été la recherche d'un équilibre parfait entre leurs consommations, leurs précisions et leurs robustesses.

En effet, les capteurs devaient consommer peu d'énergie afin d'optimiser l'autonomie de notre système, ils devaient être suffisamment précis (notamment la température interne) et ils devaient être robuste face à l'environnement interne ou externe de la ruche.




The choice of our sensors was a research of a perfect balance between their precision, their energy consumption and their solidity.

Indeed, to have a great autonomy, his consumption must be low, to have valid values, the system must be precise.

Finally, our device must be autonomous, so it can survive in the internal and external environment of the hive.

## Schéma/ Schematic

*(disponible en annexe / available in the appendix)*

 Le schéma proposé est à jour et prend en compte les dernières améliorations réalisées sur le PCB avant démarrage de la campagne de tests.



The released schema is up to date and it uses our last PCB improvements before launching the test campaign.


8 connecteurs Grove sont reliés à 8 E/S numériques afin d'y interfacer les capteurs tout en permettant une évolutivité de notre système embarqué.

Un LM358 est utilisé pour amplifier le signal audio en provenance du microphone. Un potentiomètre sert à ajuster le gain et à éviter la saturation.

Nous utilisons deux LDO TS5205CX533 en commutation. L'un pour contrôler l'alimentation des capteurs, l'autre pour l'alimentation de notre module SigFox. Les deux régulateurs sont activés ou non via une même pin du MCU grâce à un inverseur logique CD4090 CMOS. De ce fait la broche ENABLE d'un des deux LDO est à l'état bas quand l'autre est à l'état haut. Pour économiser des E/S numériques de notre MCU nous utilisons une broche convertisseur analogique numérique (CAN) du MCU afin de générer le signal d'activation des LDO.

Une DEL en série avec un cavalier permet de vérifier si le MCU transmet bien des données au module SigFox. Cette DEL étant par défaut alimentée, l'envoi d'une trame se traduira par un rapide scintillement de la DEL.

## PCB

 Le PCB correspond à l'implémentation de notre schéma électronique. Il est réalisé en fonction des contraintes de notre Atelier. Les librairies d'empreintes et de symboles utilisés sont jointes au projet.

Les dimensions de notre PCB sont inférieures à 12 cm par 10 cm.

8 Grove connectors are plugged into 8 digitals I/O in order to interface varied sensors with the embedded system. Thus, the system can still be easily scalable.

An LM358 sensor is used in order to amplify the audio signal from the microphone. A potentiometer is used to avoid saturation by fixing the gain.

We are using two switching TS5205CX533 LDO. One as a sensor power supply and the other as a SigFox module supply. Linear regulators are enabled by the same MCU pin, thanks to a CD4090 CMOS hex inverter. Thus, one LDO is enabled when the other is shut down. In order to save a digitals I/O we are using a MCU digital to analog (DAC) pin to generate LDO's enable signal.

In serial with a jumper, a LED is blinking each time a SigFox module is transmitted.



The PCB is matching with the schematic described above. It was built according to our workshop's specificities. Used libraries are included in the project files.

Our PCB size is 12 cm by 10 cm. We have fixed I/O attribution in order to simplify PCB routing.

Le choix des E/S sur le MCU a été optimisé pour la réalisation du PCB.

Tous nos connecteurs sont des connecteurs Grove hormis un RJ11 pour l'anémomètre et un connecteur jack pour le son.

La distance minimum entre nos pistes est de 0.4 mm.

Les composants CMS et le plan de masse sont sur la face bottom.

Nos pistes de signaux ont une largeur qui varie entre 1 mm et 0.8 mm. Nos pistes d'alimentations font 1,2mm de largeur afin de pouvoir distinguer plus facilement.

Afin de simplifier le PCB nous avons utilisé 3 ponts, "BRIDGE1", "BRIDGE2" et "BRIDGE3". Il y a un bridge qui passe sous le MCU mais cela ne posera pas de soucis car le MCU est sur un support.

Les trous font un diamètre de 0.8mm et l'empreinte associée fait 1.5mm de diamètre.

L'espacement entre chaque broche du connecteur Grove est de 2 mm. Les autres broches (MCU, module SigFox) sont espacées de 2.54 mm.

Le connecteur entre notre PCB et le module Sigfox est de type femelle 9 pin connector.

Des modifications mineures prises en compte sur le schéma et réalisées à la main ne sont pas prises en compte dans le routage joint. Il s'agit des éléments suivants:

- Modification de la valeur de certaines résistances et condensateurs
- UART TX passe de PA\_3 à PA\_9
- UART RX passe de PA\_2 à PA\_10
- Pontages entre plusieurs parties du plan de masse qui étaient disjointes à cause d'un espace trop étroit entre pistes.
- Augmenter la largeur des 4 trous latéraux permettant de fixer le PCB sur entretoises.

We have only Grove connectors, an RJ11 connector (anemometer) and a jack connector (sound acquisition)

The smallest distance between two traces is 0.4 mm.

CMS components and ground plane are on the PCB bottom face. Supply traces have a width between 0.8mm and 1 mm. Supply traces have a width of 1.2 mm in order to distinguish them easily.

In order to simplify the PCB we are using 3 bridges "BRIDGE1", "BRIDGE2" and "BRIDGE3". One bridge is under the MCU but it's normal because the MCU is socketed.

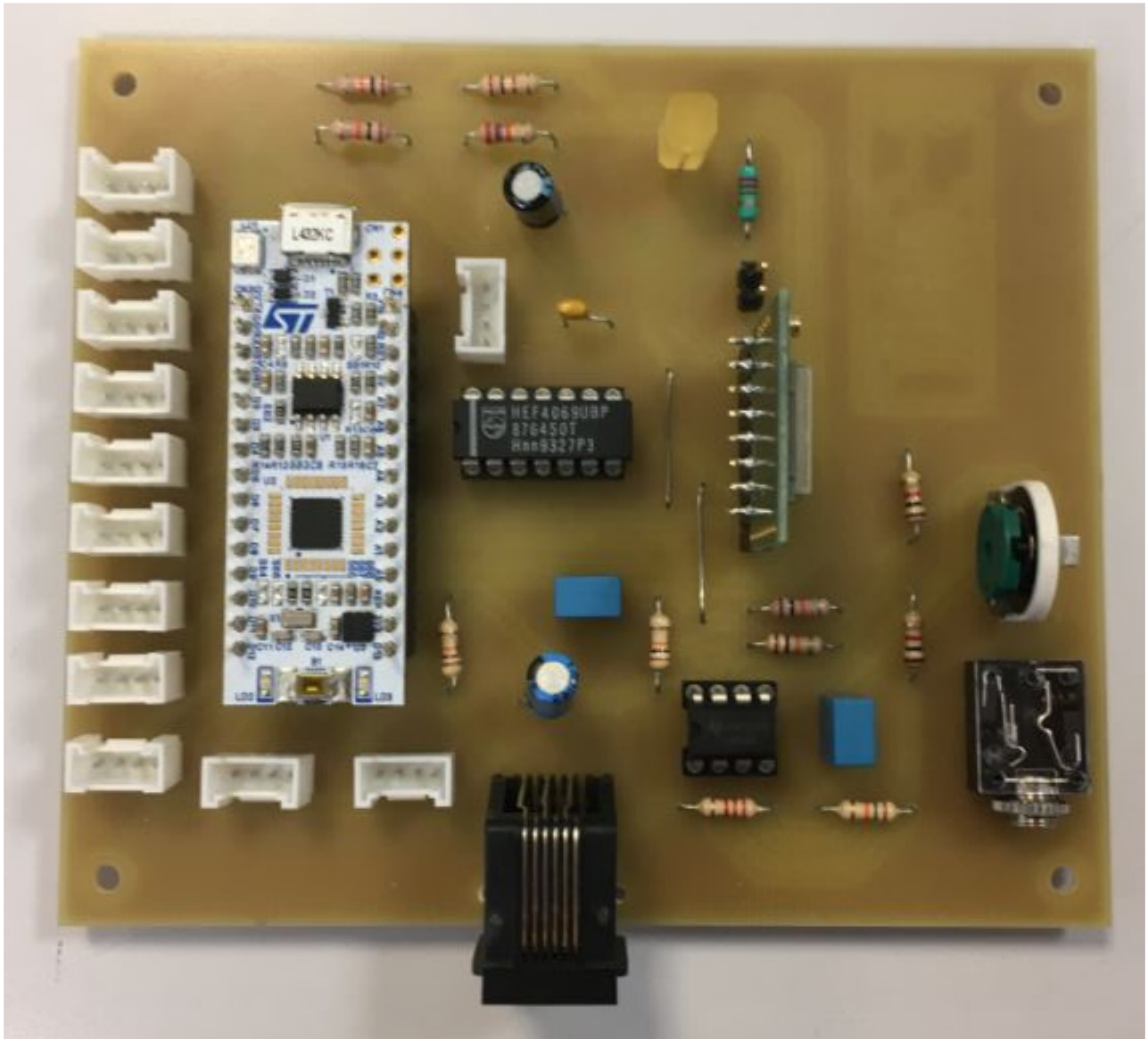
All holes have 0.8 mm diameter and a 1.5 mm footprint.

We fixed 2 mm between each Grove pin and 2,54 mm for other pins (MCU socket, SigFox module socket).

The SigFox module is plugged to the PCB thanks to a female 9 pin connector.

We had some improvements to our PCB after the printing (the schema is up to date with this improvement):

- Resistors and capacitor values
- UART TX is set to PA\_3 instead of PA\_9
- UART RX is set to PA\_2 instead of PA\_10
- Bridge between parts of the ground plane due to bad connections ( small space between two traces).
- Increase the diameter of PCB fixing holes (a hole at each PCB corner)




PCB B-Smart



# Transmission des données / Data transmission

## 1. La 0G / The 0G


 Qui dit transmission de données, dit réseau. De ce côté nous avons un large choix : 5G, 3G,...voire 0G ! Cependant, il faut savoir qu'un traitement de flux massifs de données est d'autant plus énergivore qu'il y a de données. Lorsque l'autonomie est un critère primordial, la 5G & co est donc à bannir !

De plus, notre système n'envoie que quelques octets de données, ce qui nous permet d'utiliser le top du top niveau consommation : la 0G.

Pour des messages de très faible taille, ses avantages sont les suivants :

- envoi sur de très longues distances,
- très faible coûts énergétiques,
- simplicité : pas de carte sim

( Ce qui en fait le réseau de prédilection des IoT !)

 On the network side, multiple offers are given : 5G,...,3G or even the 0G.

However, you may know that - the more data are exchanged, the more energy-intensive the system is. Hence the 5G & co are banned when the main criterion is autonomy.

Besides, the system sends only a few bytes of data, which allows us to use the best of the best in terms of energy consumption : the 0G !


For small messages, the advantages are as follows :

- data exchanges over long distances,
- very low energy costs,
- simplicity : no card sim


(That's why the 0G is a method of choice for IoT !

## 2. Module réseau / Network module

### Description du module / Module description

 Les opérateurs de télécommunications offrant leurs services 0G sont principalement Sigfox et LoRa. La différence entre les deux n'est pas très grande : la première fonctionne avec son propre kit de modules, tandis que le second fonctionne avec un abonnement annuel (comme n'importe quel opérateur téléphonique).

Après plusieurs comparaisons, nous nous sommes arrêtés sur un kit avec un module de télécommunication Sigfox et une antenne

 Lora and Sigfox are the main 0G telecommunication providers on the market.

The first one runs with his own environment kit and the other one works with a monthly subscription like a phone operator.

After several researches, we chose the french solution : The Sigfox communication kit and his antenna (cf.components list)

The kit includes a one year subscription, which allows to :

intégrée. De la même manière que pour un opérateur classique, il nous faut souscrire à un abonnement.

Le kit offre 1 an d'abonnement avec lequel vous pouvez, par jour :

- envoyer 140 messages,
- recevoir 4 messages.

Un message fait 12 octets maximum.

- send 140 messages maximum per day
- receive 4 messages maximum per day

A message is composed of 12 bytes.



**Kit Carte Breakout Sigfox BRKWS02-RC4 + Antenne 920Mhz<sup>1</sup>**

<sup>1</sup> <https://yadom.fr/kit-carte-breakout-sigfox-brkws01-rc4-antenne.html> (05/01/2021)

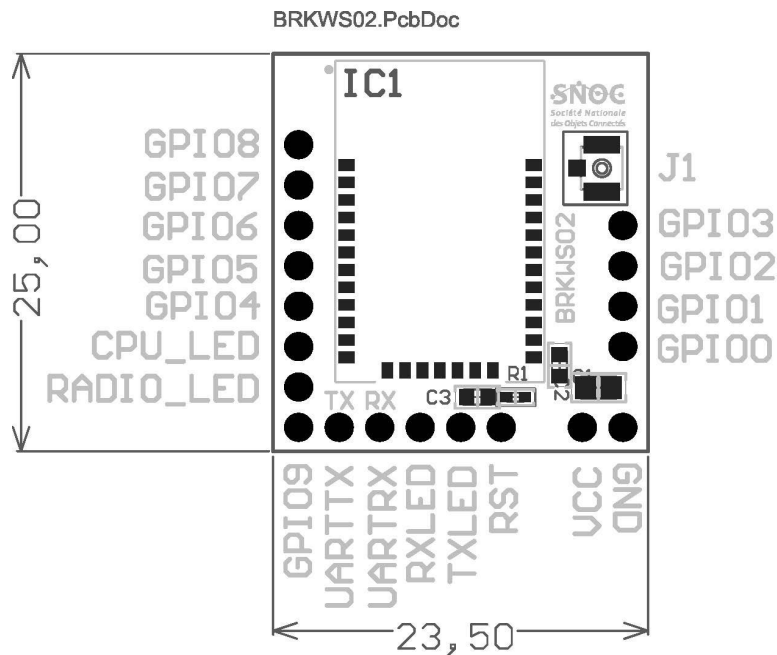



Schéma de la carte

## Activation du module / Module activation

 Pour pouvoir envoyer des messages, il nous faut impérativement activer le module Wisol. Pour cela, il faut se rendre sur la page d'activation de Sigfox :

<https://buy.sigfox.com/activate>


Il vous sera demandé, entre autre, de renseigner successivement les informations suivantes :

- votre pays,
- l'identifiant unique (ID) et le code PAC précisés sur l'emballage du module,
- une adresse mail et un mot de passe.

S'il n'est pas inclus, vous devrez souscrire à un abonnement annuel pour l'envoi d'au minimum 100 variables par jour (11 euros HTC).

Une fois l'activation terminée, vos messages envoyés seront visualisables sur le backend de sigfox :

<https://backend.sigfox.com/auth/login> .

 The activation of the Wisole module (sigfox module) is necessary for sending messages. To achieve this, subscribe on : <https://buy.sigfox.com/activate>

They will ask you to enter the following informations :

- Your country
- Your ID and your PAC (wrote on your sigfox module)
- Your mail and a password


If not included, an annual subscription is required. It must include a minimum of 100 variables per day (roughly £ 9,81 /year).

Once the subscription is done, please check your messages on the sigfox backend :

<https://backend.sigfox.com/auth/login>

### 3. Envoi des données / Data transmission

#### Présentation générale / General presentation


 Le module est contrôlé avec des commandes série AT (9600 bauds) envoyées sur les broches TX / RX. (voir la *datasheet* pour plus de détails).

La commande pour envoyer un message SIGFOX (sans message de retour) est :  
`AT$SF=XXXXXXXXXXXX` (Valeur Hexadécimal).

Le message envoyé est la trame des données mesurées. Cependant, tel quel nous ne pouvons pas envoyer beaucoup de données : une mesure flottante réserverait à elle seule 4 bits sur les 12 disponibles. Il faut donc optimiser la manière de stocker nos variables pour au maximum envoyer une trame de 12 octets, quitte à perdre en précision...

Dans ce projet, nous irons un peu plus loin : en effet, en évaluant l'ensemble des critères de précision de nos mesures, nous verrons qu'il est possible de tout stocker dans un message d'environ 7 octets, sachant que chaque octet gagné réduira la consommation énergétique (la phase d'envoi des données étant une phase très énergivore !).

*N.B. L'ensemble du code pour la partie réseau est inclus dans le fichier `main.cpp`. Ce fichier devra être compilé et téléversé sur la cible L432KC.*

 The sigfox module is monitored by AT serial controls (9600 bauds), sent to the TX/RX pins (please refer to the datasheet for more details).

The control to send a SIGFOX message (without reply) is : `AT$SF=XXXXXXXXXXXX` (Hexadecimal value).

The message sent is the frame of the measured data. However, as such, we cannot send a lot of data: a floating measure alone would reserve 4 bits of the 12 available. It is therefore necessary to optimize the way variables are stored in order to send a frame of 12 bytes maximum, at the risk of losing accuracy...

In this project, we will go a little further: in fact, by evaluating all the accuracy criteria of our measurements, you will see that it is possible to store everything in a message of about 7 bytes, knowing that each byte gained will reduce energy consumption (the transmission phase is a very energy-consuming phase!).

*N.B. The entire code for the network part is included in the file `main.cpp`. This file must be compiled and uploaded to the L432KC target.*

## Fonctionnement



Notre trame est représentée par une structure que nous appelons *Frame*.

Elle a pour champs un tableau d'octets *field* de taille *size* et l'index *offset* du dernier bit écrit.

Pour créer une *Frame*, nous faisons appel à la fonction *new\_frame()*. Cette dernière fixe l'offset à 0 et la taille à 7 octets ( macro *SIZE*).

La plupart de nos données ont des critères spécifiques, propres à leurs mesures ( précision , min/max , etc.) . Le stockage de chaque variable est donc géré par une fonction qui leur est propre (par exemple *stock\_temp()* pour la température).

Par soucis d'économie d'octets, nous avons les spécifications suivantes :

- nos valeurs commencent tous à 0. Ainsi une valeur minimale de 18 ou - 5 par exemple, est ramenée à 0.
- les flottants sont stockés en int. Par exemple, pour une précision de 1/2 (précision de la température) nous aurons 2 fois plus de valeurs que pour une précision de 1.

### Exemple pour la température :

```
int bits = (int)((temp + 18) * 2 + 0.5f);
```

Ces entiers, sont par la suite décomposés bit à bit dans la *frame* sur n bits suivant le nombre maximal de valeurs.

### Exemple pour la température :

```
stock_int(bits, 7, frame);
```

### Ligne de code pour récupérer le bit numéro i d'une valeur :

```
value & (1 << (taille_en_bits - i - 1))
```

Le stockage bit à bit est codé dans la fonction *stock\_bit()*. Elle détermine notamment le numéro de l'octet courant et le numéro du bit donné en paramètre de fonction.



Our frame is represented by a structure, called *Frame*.

It has an array of *size* field bytes and an index of the last bit written.

To create a *Frame*, we use the *new\_frame()* function. This function sets the offset to 0 and the size to 7 bytes ( macro *SIZE*).

Most of our data have specific criteria, depending on their measurements (accuracy, min/max, etc.). The storage of each variable is therefore managed by a specific function for each (for example *stock\_temp()* for the temperature).

In order to save some data bytes, the code has the following specifications :

- our values all start at 0. So a minimum value of 18 or - 5 for example, is reduced to 0.
- floats are stored in int. For instance, an accuracy of 1/2 (temperature accuracy) will have twice as many values as for an accuracy of 1.

### Example for the temperature :

```
int bits = (int)((temp + 18) * 2 + 0.5f);
```

These integers are then broken down bit by bit in the *frame* to n bits according to the maximum number of values.

### Example for the temperature :

```
stock_int(bits, 7, frame);
```

### Line of code to retrieve the bit number i of a value:

```
value & (1 << (taille_en_bits - i - 1))
```

Bit-to-bit storage is encoded in the *stock\_bit()* function. Especially, it determines the number of the current byte and the number of the given bit as a function parameter.

Enfin, notre frame est envoyée depuis la fonction `send_frame()` - via le commande `AT$SF` vu précédemment.

N.B. Une fonction de `DEBUG` est également disponible et permet d'afficher sur un terminal les données envoyées au module sigfox.

Finally, our frame is sent by the `send_frame()` function - via the `AT$SF` control seen earlier.

N.B. A `DEBUG` function is also available to display the data sent to the sigfox module on a terminal.

## Réception des données / Data Reception



Un callback permet de réceptionner les données envoyées par le module sur le sigfox backend.

Le module sigfox devrait apparaître dans la liste des *Devices liés à votre compte*.

Pour configurer un callback, cliquer sur *Device Type > callbacks > new*.

*N.B. La configuration complète du callback est détaillée en annexe.*

Une fois terminé, vous pourrez voir vos messages/ trames reçues depuis *Device > Id > Messages*.

*Vous pourrez y voir :*

- un signal vert vous indiquant si le callback s'est bien effectué ( en cliquant dessus, vous aurez toutes les valeurs des variables configurées dans le callback).
- la date et l'heure de réception,
- la trame reçue en hexadécimale.



A callback allows you to receive the data sent by the module on the sigfox backend.

The sigfox module should appear in the Devices list on your SigFox account.

To configure a callback, click *Device Type > callbacks > new*.

*N.B. The complete configuration of the callback is detailed in the appendix.*

Once completed, you will be able to view your received messages/frames from *Device > Id > Messages*.

You will be able to see:

- a green signal telling you whether the callback was correctly performed ( by clicking on it, you will have all the values of the variables configured in the callback).
- the date and time of receipt,
- the received frame in hexadecimal.

| Time                | Seq Num | Data / Decoding | LQI | Callbacks | Location |
|---------------------|---------|-----------------|-----|-----------|----------|
| 2020-12-08 18:26:44 | 1369    | 7136dce812b3d0  |     |           |          |

*Exemple de message / Example of message*

## Stockage des données sur le cloud/ Data storage



Lorsque nous avons commencé notre projet, nous hébergeâmes et visualisâmes nos données sur une seule et même plateforme : Ubidots.

Cette dernière avait l'avantage d'être très intuitive et simplifiée.

Cependant, lorsque le projet nécessite de travailler sur beaucoup de données, il vaut mieux passer son chemin. En effet, chez B-Smart nous avons beaucoup de capteurs et qui dit beaucoup de capteurs dit beaucoup de données à envoyer.

Dans un premier temps, la version gratuite d'Ubidots ne permet que de traiter au maximum 10 variables, ce qui est peu !

Dans un second temps, le réseau 0G nous limite à l'envoi de messages de 12 octets au maximum, ce qui nous oblige à optimiser notre frame (comme décrit précédemment). Seulement, les opérations bit à bit sur Ubidots sont payantes encore une fois. En comparant les coûts que cela engendre, nous avons préféré changer de plateforme.

Puisque Sigfox ne permet pas de stocker nos données et de les exploiter, il faudra donc les sauvegarder sur un cloud dans un premier temps . Vous pouvez choisir n'importe lequel mais nous recommandons OVH metrics. En effet, ce service d'OVH offre énormément de possibilités pour seulement 1,18 euros.

Vous pourrez stocker entre autres 100 variables ( séries) par mois ! ( bien plus que sur Ubidots par exemple, qui à défaut de payer plus de 50 euros, n'offre que 10 variables/mois !)

Lorsque vous stockez vos données, l'hébergeur vous donne accès à un token de lecture et un d'écriture.



When we started our project, we hosted and visualized our data on the same platform : Ubidots.

The latter had the advantage of being very intuitive and simplified.

However, when the project needs to work on a lot of data, it is better to seek for something else. Indeed, at B-Smart we have a lot of sensors - and a lot of sensors means a lot of data to send...

At first, the free version of Ubidots only allows you to process a maximum of 10 variables, which is not much!


Second, the 0G network limits us 12 bytes per message, which forces us to optimize our frame (as described above). Alas, bit-to-bit operations on Ubidots aren't free... Comparing the costs of each solution, we decided to change for another platform.

Since Sigfox does not allow us to store our data and use it, we will have to back it up to a cloud at first. You can choose any, but we recommend OVH metrics. Indeed, this service offers a lot of possibilities , for only £1.05/month.


Among other things, you can store up to 100 variables (series) per month! ( much more than Ubidots for example, which costs roughly £44.48 to operate on more than 10 variables/month!)

When you store your data, the hosting provider gives you access to a read and a write token.

## Visualisation des données / Data visualization


 Pour visualiser nos données, nous avons choisi les tableaux de bord Grafana. Cette plateforme en ligne est open source et propose donc gratuitement ses services. Elle est également reconnue pour ses tableaux de bords jugés plus professionnels et qualitatifs, et offre notamment la possibilité de générer des alertes (sms, mail, etc.).

Les données stockées sur le cloud sont communiquées à Grafana via le token d'écriture et un protocole d'échange de données. Pour ce dernier, nous avons choisi OpenTSDB, orienté séries temporelles distribuées et hautes performances.

 To view our data, we chose the Grafana dashboards. This online platform is open source and therefore offers its services free of charge. It is also recognized for its dashboards - considered more professional and qualitative, and offers especially the possibility of generating alerts (sms, mail, etc.).


The data stored on the cloud is communicated to Grafana via the write token and a data exchange protocol. For the latter, we chose OpenTSDB, oriented - distributed time series and high performance.

## Autonomie / Self sufficiency

 La question de l'autonomie est essentielle dans le cadre de ce projet et en système embarqué en général.

Afin que le système soit autonome, nous avons décidé de coupler une batterie et un panneau photovoltaïque grâce notamment à l'utilisation du module lipo rider pro. Cela permet d'assurer un fonctionnement autonome de notre système.

Il est important de notifier que des coupures de plusieurs heures peuvent arriver notamment en hiver où lorsqu'il y a une longue période d'intempérie avec donc un très faible taux d'ensoleillement, la batterie n'est pas suffisamment rechargée.


 The issue of autonomy is essential in this project and in the embedded system in general.


In order for the system to be autonomous, we decided to couple a battery and a photovoltaic panel thanks to the use of the lipo rider pro module. This allows our system to function autonomously.

It is important to notify that cuts of several hours can happen especially in winter when there is a long period of bad weather with a very low rate of sunshine, the battery is not sufficiently charged.



## Conclusion

 En six mois, nous avons pu produire un système embarqué permettant l'analyse du comportement d'une ruche. En partant d'un petit prototype sur platine d'essais, nous avons pu implémenter un nombre toujours croissant de capteurs. En parallèle, nous avons pu effectuer suffisamment de veille technique et d'analyses bibliographiques afin de pouvoir monter en compétences et mieux utiliser les capteurs que nous avons implémentés. Une première série d'essais en extérieur d'une journée a pu nous permettre d'identifier des points clés en matière d'ergonomie, nécessaires à prendre en compte (position des capteurs, emplacement de la ruche, démontage, etc.). Une deuxième série d'une semaine sur un prototype intermédiaire a pu nous permettre de mettre en évidence certaines carences en matière d'autonomie du système embarqué B-Smart. Ces carences se sont confirmées lors de la troisième campagne d'essais (sur ruche) mais la faible consommation de notre système (quelques  $\mu\text{A}$  lorsque le MCU est éteint et environ  $50\text{ mA}$  avec le MCU allumé) ont pu nous permettre un rechargement suffisant de la batterie afin de recevoir des trames de manières périodiques. Nous avons pu observer que certaines valeurs à transmettre, les trames associées n'étaient pas transmises. Le système embarqué produit par la team B-Smart est fonctionnel mais néanmoins très perfectible. En effet, un travail prioritaire sur la gestion des trames ainsi que sur l'autonomie (ajout d'un panneau solaire plus performant et d'une batterie de capacité supérieure) peuvent permettre à B-Smart de gagner davantage en fiabilité.

 In six months, we were able to produce an on-board system to analyze the behavior of a hive. Starting with a small prototype on a test stage, we were able to implement an ever increasing number of sensors. At the same time, we were able to carry out sufficient technical monitoring and bibliographic analyses in order to be able to improve our skills and make better use of the sensors we implemented. A first series of one-day outdoor tests allowed us to identify key ergonomic points that needed to be taken into account (position of the sensors, location of the hive, disassembly, etc.) .A second series of a week on an intermediate prototype allowed us to highlight some shortcomings in terms of self-sufficiency of the B-Smart on-board system. These deficiencies were confirmed during the third test campaign (on hive) but the low power consumption of our system (a few  $\mu\text{A}$  when the MCU is turned off and about  $50\text{ mA}$  with the MCU turned on) have been able to allow us to sufficiently recharge the battery to receive frames in periodic ways. We observed that some values entailed some problems , like frames that were not transmitted. The onboard system produced by the B-Smart team is functional but nevertheless very perfectible. Indeed, a priority work on frame management as well as on autonomy (addition of a more efficient solar panel and a higher capacity battery) can allow B-Smart to gain more reliability.

#### Les points forts de B-Smart sont les suivants:

- La première série d'essais réalisée dans un rucher était satisfaisante car les nombreuses trames reçues ont pu nous permettre d'étudier attentivement le comportement de la ruche.
- L'utilisation de plusieurs capteurs de températures permet d'obtenir des informations intéressantes comme la taille de la colonie.
- Le prototype, facilement transportable, s'installe de manière intuitive au sein d'une ruche.
- L'optimisation des trames réseau permet évolutivité et le panel Grafana peut s'adapter facilement à l'usage des usagers.

#### Les points faibles de B-Smart sont les suivants:

- Pour certaines valeurs de données, les trames ne sont pas transmises.
- Carences en autonomie. Le panneau solaire n'est pas suffisamment efficace et ne permet pas une recharge rapide du système embarqué. La Lipo Rider Pro doit être remplacée par un circuit d'alimentation spécifique, implémenté sur le même PCB que la partie logique.
- La FFT n'est pas suffisamment sensible.

#### The strengths of B-Smart are the followings:

- The first series of tests carried out in an apiary were concluent because the numerous frames received, allowed us to study carefully the behaviour of the apiary.
- The use of several temperature sensors provides interesting information such as colony size.
- The prototype, easily transportable, is installed intuitively within a hive.
- Network frame optimization allows scalability and the Grafana dashboard is a friendly-use tool.

#### The weaknesses of B-Smart are the followings:

- For some data values, frames are not transmitted.
- Lack of autonomy. The solar panel is not efficient enough and does not allow a fast charging of the onboard system. The Lipo Rider Pro must be replaced by a specific power circuit, implemented on the same PCB as the logic part.
- FFT is not sensitive enough and must be revisited.

# Annexes

## Schémas PCB / PCB Schematics

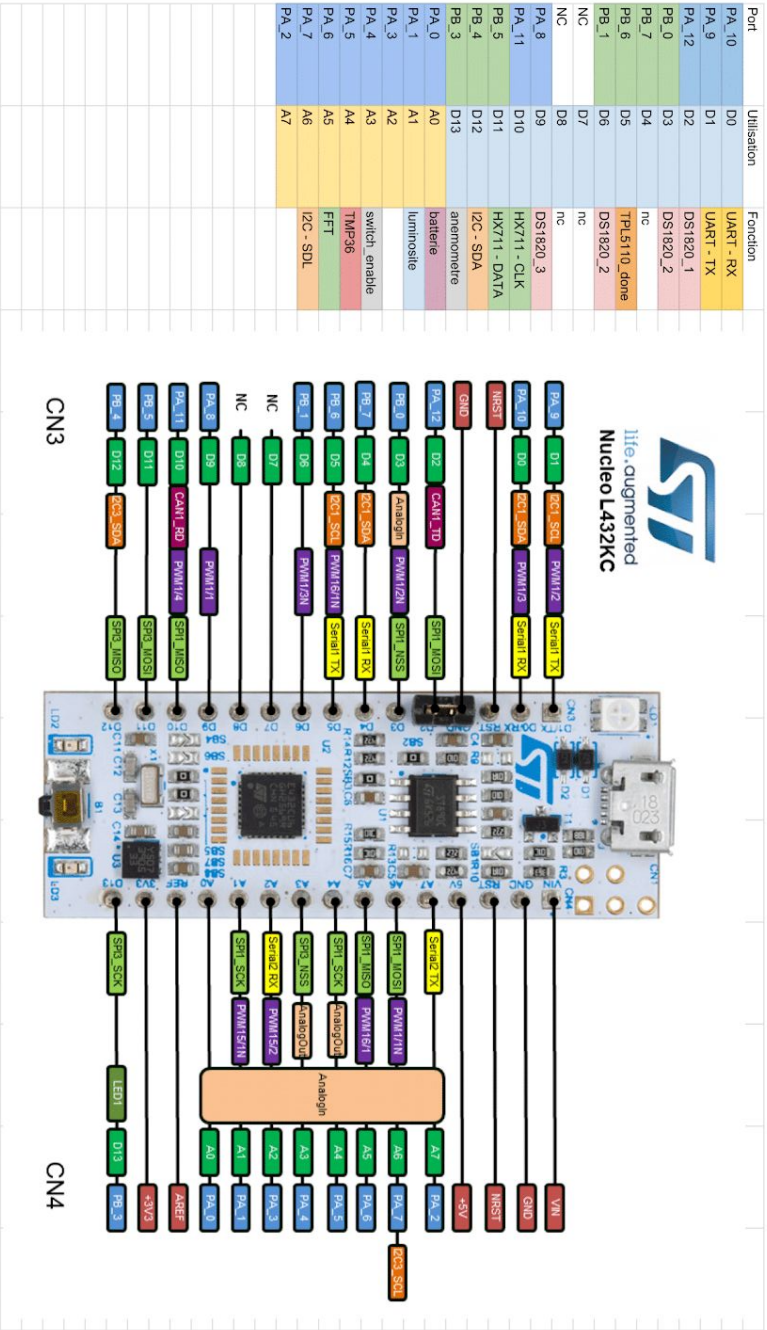


Schéma d'utilisation du microcontrôleur  
Usage scheme of the microcontroller

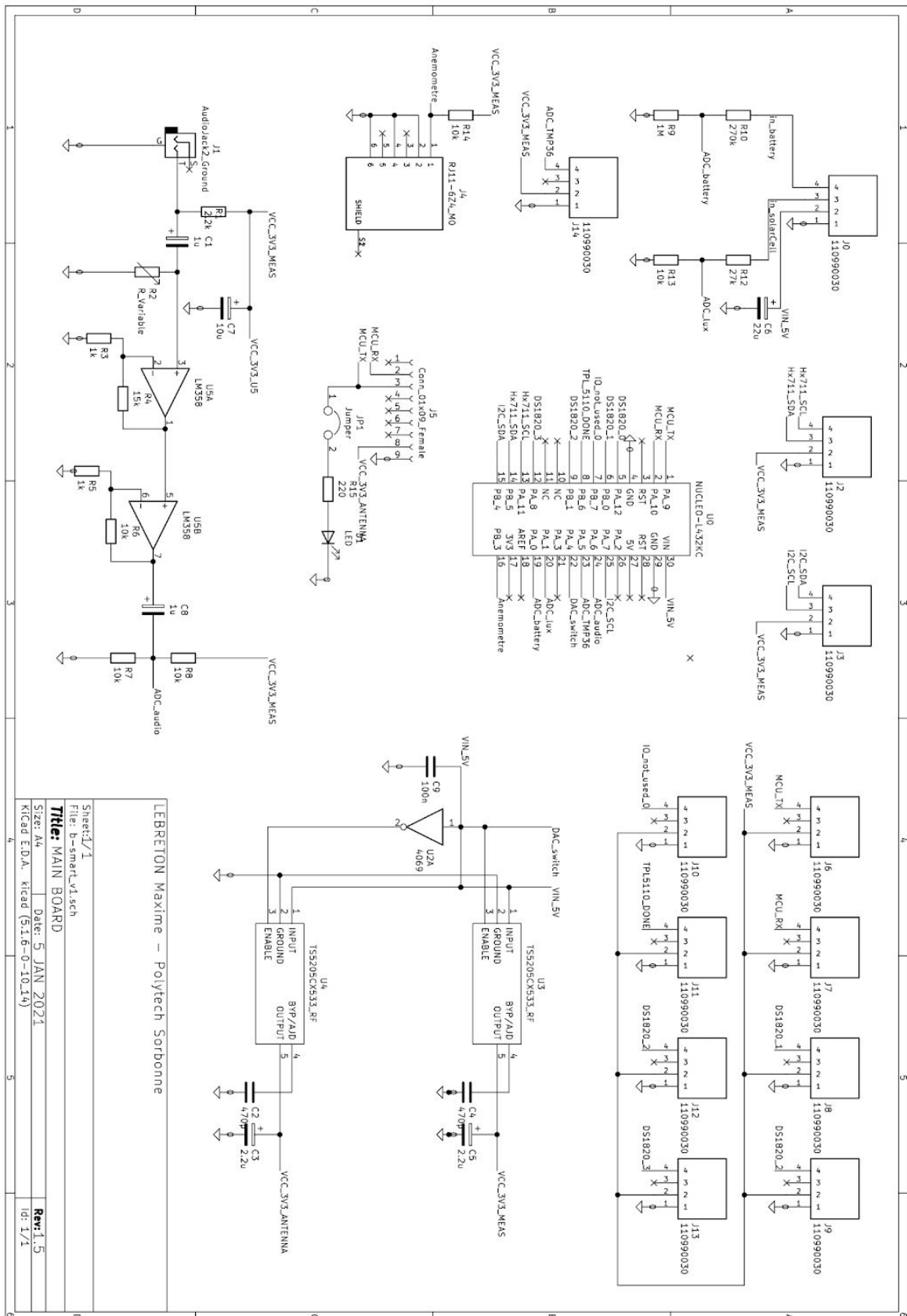


Schéma PCB  
PCB Schematic

Components list V1.0

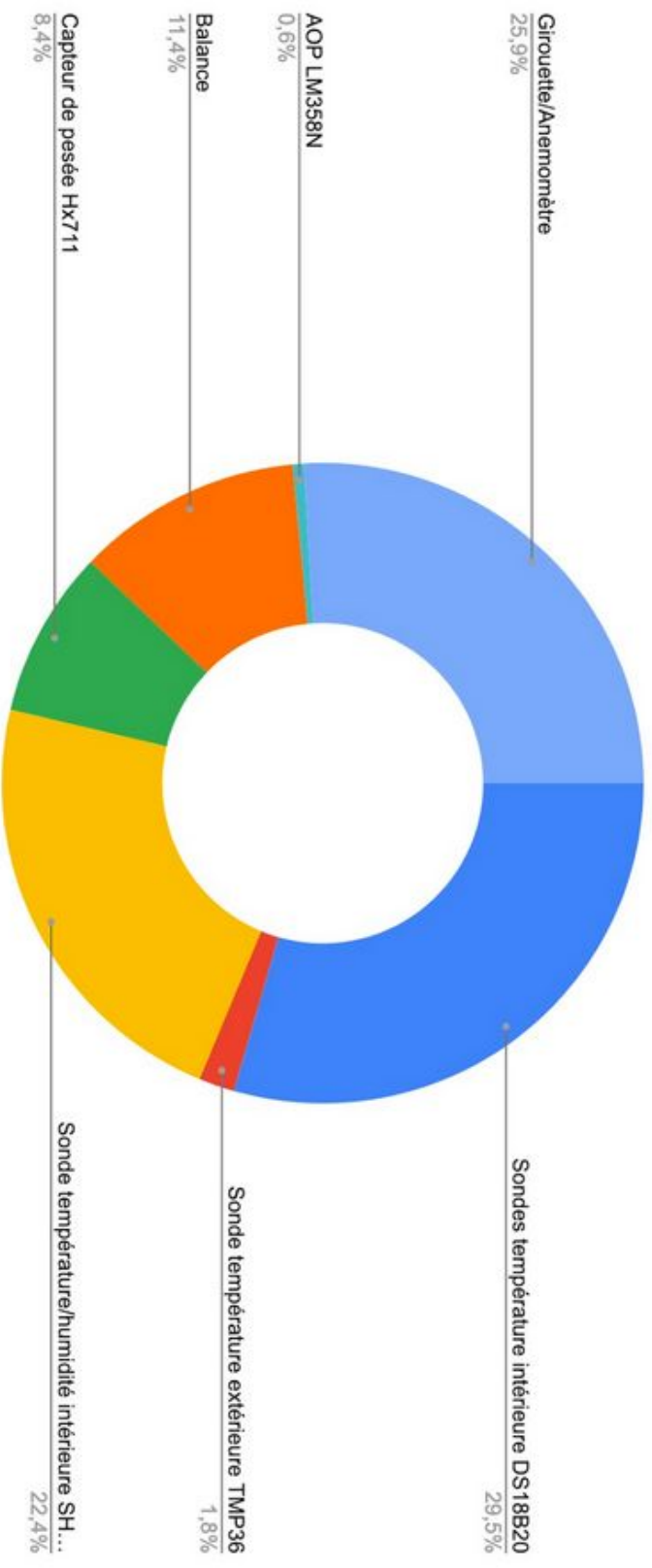
| MAIN BOARD |        |   |  |   |
|------------|--------|---|--|---|
| Name       | Number | Type  | Description  | Value   |
| J6         | 1      | Grove connector   | Connector: NS_Tech_Grove_1x04_P2mm_Vertical                    |   |
| J7         | 2      | Grove connector   | Connector: NS_Tech_Grove_1x04_P2mm_Vertical                    |   |
| J8         | 3      | Grove connector   | Connector: NS_Tech_Grove_1x04_P2mm_Vertical                    |   |
| J9         | 4      | Grove connector   | Connector: NS_Tech_Grove_1x04_P2mm_Vertical                    |   |
| J10        | 5      | Grove connector   | Connector: NS_Tech_Grove_1x04_P2mm_Vertical                    |   |
| J11        | 6      | Grove connector   | Connector: NS_Tech_Grove_1x04_P2mm_Vertical                    |   |
| J12        | 7      | Grove connector   | Connector: NS_Tech_Grove_1x04_P2mm_Vertical                    |   |
| J13        | 8      | Grove connector   | Connector: NS_Tech_Grove_1x04_P2mm_Vertical                    |   |
| U3         | 9      | Taiwan Semiconductor TSS205CX533 RF, LDO Voltage Regulator, 150mA, 3.3 V, +/-1.5% 5-Pin, SOT-25 | TSS205CX533_RF:SOT95P270X125-5N                                | Taiwan Semiconductor TSS205CX533 RF, LDO Voltage Regulator, 150mA, 3.3 V, +/-1.5% 5-Pin, SOT-25 |
| C5         | 10     | Polarized capacitor   | Capacitor_SMD:C_0805_2012Metric                                | 2.2u  |
| C4         | 11     | Unpolarized capacitor   | Capacitor_SMD:C_0805_2012Metric                                | 470p  |
| U4         | 12     | Taiwan Semiconductor TSS205CX533 RF, LDO Voltage Regulator, 150mA, 3.3 V, +/-1.5% 5-Pin, SOT-25 | TSS205CX533_RF:SOT95P270X125-5N                                | Taiwan Semiconductor TSS205CX533 RF, LDO Voltage Regulator, 150mA, 3.3 V, +/-1.5% 5-Pin, SOT-25 |
| C3         | 13     | Polarized capacitor   | Capacitor_SMD:C_0805_2012Metric                                | 2.2u  |
| C2         | 14     | Unpolarized capacitor   | Capacitor_SMD:C_0805_2012Metric                                | 470p  |
| J1         | 15     | Audio Jack, 2 Poles (Mono / TS), Grounded Sleeve  | Connector_Audio:Jack_3.5mm_Ledino_K835PRS_Horizontal           |   |
| R1         | 16     | Resistor  | Resistor_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P10.16mm_Horizontal | 2.2k  |
| C1         | 17     | Polarized capacitor   | Capacitor_THT:CP_Radial_D10.0mm_P5.00mm                        | 1u  |
| R2         | 18     | Variable resistor   | Potentiometer_THT:Potentiometer_ACP_CA14-H5_Horizontal         | R_Variable 2.2k   |
| U5         | 19     | Low-Power, Dual Operational Amplifiers, DIP-8/SOIC-8/TQ-99-8                                    | LM358N_NOPB:DIP794W:SP254L959H508Q8N                           | LM358   |
| R4         | 20     | Resistor  | Resistor_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P10.16mm_Horizontal | 15k   |
| R3         | 21     | Resistor  | Resistor_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P10.16mm_Horizontal | 1k  |
| R6         | 22     | Resistor  | Resistor_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P10.16mm_Horizontal | 10k   |
| R5         | 23     | Resistor  | Resistor_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P10.16mm_Horizontal | 1k  |
| R8         | 24     | Resistor  | Resistor_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P10.16mm_Horizontal | 10k   |
| R7         | 25     | Resistor  | Resistor_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P10.16mm_Horizontal | 10k   |
| R10        | 26     | Resistor  | Resistor_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P10.16mm_Horizontal | 270k  |
| R9         | 27     | Resistor  | Resistor_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P10.16mm_Horizontal | 1M  |
| R13        | 28     | Resistor  | Resistor_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P10.16mm_Horizontal | 10k   |
| R12        | 29     | Resistor  | Resistor_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P10.16mm_Horizontal | 27k   |
| J0         | 30     | Grove connector   | Connector: NS_Tech_Grove_1x04_P2mm_Vertical                    | 110900030   |
| U2         | 31     | Hex inverter  | HCF4069UBEV:DIP850W:50P254L2000H510Q14N                        | 4069  |
| J3         | 32     | Grove connector   | Connector: NS_Tech_Grove_1x04_P2mm_Vertical                    | 110900030   |
| J2         | 33     | Grove connector   | Connector: NS_Tech_Grove_1x04_P2mm_Vertical                    | 110900030   |
| J4         | 34     | Grove connector   | RJ11-624_MO:TE_RJ11-624_MO                                     | RJ11-624_MO   |
| R14        | 35     | Resistor  | Resistor_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P10.16mm_Horizontal | 10k   |



|             |    |  |   |  |                   |
|-------------|----|--|---|--|-------------------|
|             |    | Generic connector, single row, 01x09, script generated (kicad-library-utis/schib/autogen/connector/) |   | Generic connector, single row, 01x09, script generated (kicad-library-utis/schib/autogen/connector/) |                   |
| J5          | 36 |  | Connector_PinHeader_2.54mm:PinHeader_1x09_P2.54mm_Vertical                                |  | Conn_01x09_Female |
| Jp1         | 37 | Jumper, normally closed  | Connector_PinHeader_2.54mm:PinHeader_1x02_P2.54mm_Vertical                                | Jumper, normally closed  | Jumper            |
| R15         | 38 | Resistor   | Resistor_THT_R_Axial_DIN0207_L6.3mm_D2.5mm_P10.16mm_Horizontal                            | Resistor   | 220               |
| D1          | 39 | Light emitting diode   | LED_THT_LED_D5.0mm  | Light emitting diode   | LED               |
| C6          | 40 | Polarized capacitor  | Capacitor_THT_CP_Radial_D10.0mm_P5.00mm   | Polarized capacitor  | 22u               |
| J14         | 41 | Grove connector  | Connector_NS-Tech_Grove_1x04_P2mm_Vertical  |  | 110990030         |
| U0          | 42 | NUCLEO-L432KC  | NUCLEO-L432KC[2]-MODULE_NUCLEO-L432KC   |  | NUCLEO-L432KC     |
| C7          | 43 | Polarized capacitor  | Capacitor_THT_CP_Radial_D10.0mm_P5.00mm   | Polarized capacitor  | 10u               |
| C8          | 44 | Polarized capacitor  | Capacitor_THT_CP_Radial_D10.0mm_P5.00mm   | Polarized capacitor  | 1u                |
| C9          | 45 | Unpolarized capacitor  | Capacitor_THT_CP_Radial_D10.0mm_P5.00mm   | Unpolarized capacitor  | 100n              |
| XX          | 46 | Sigfox Module  | Kit_Carte_Breakout_Sigfox_BRKWS01   |  | 868M              |
| XX          | 47 | Sigfox Antenna   | 868 MHz antenna   |  |                   |
| POWER BOARD |    |  |   |  |                   |
| U0          | 1  | Adafruit TPL5110 Low Power Timer Breakout  |   |  |                   |
| R0          | 2  | Resistor   |   |  | 100K              |
| SENSORS     |    |  |   |  |                   |
| J1          | 1  | Jack Microphone  |   |  |                   |
| J2          | 2  | HX711 Balance Module   |   |  |                   |
| J2          | 3  | 200 Kg Weight sensor   | 200Kg-Electronic-Platform-Scale-Load-Cell-Pressure-Balanced-Cantilever-Load-Weight-Sensor |  |                   |
| J3          | 4  | SHT20 Humidity Sensor  |   |  |                   |
| J14         | 5  | TMMP36 Temperature Sensor  |   |  |                   |
| J4          | 6  | Anemometer   |   |  |                   |
| J8          | 7  | DS1820 Temperature Sensor  |   |  |                   |
| J9          | 8  | DS1820 Temperature Sensor  |   |  |                   |
| J12         | 9  | DS1820 Temperature Sensor  |   |  |                   |
| J13         | 10 | DS1820 Temperature Sensor  |   |  |                   |
| OTHERS      |    |  |   |  |                   |
| XX          | 1  | Solar Panel  | 2W Solar Panel  |  | 2W                |
| XX          | 2  | 5V Power Supply from Battery + Solar Panel   | LIPO RIDER PRO BOARD  |  |                   |

# Estimation du coût de notre prototype

## Capteurs



# Estimation du coût de notre prototype

## Alimentation



| Nom                 | Prix Unitaire TTC (EUR) |
|---------------------|-------------------------|
| Batterie 1050 mAh   | 10,9                    |
| Panneau solaire 2 W | 12,9                    |
| Lipo Rider Pro      | 15,2                    |



# Configuration du callback sur Sigfox Backend / Callback configuration on Sigfox Backend



Vous devrez remplir les champs suivants:

- Type **DATA UPLINK**

(pour envoyer des données sans en recevoir)

- Channel **URL**
- Custom payload config : **<à définir !>**

Dans le custom payload vous allez définir la valeur de vos variables suivant leur position dans la trame.

Exemple : temp\_exte:0:uint:7  
luminosity:1:uint:4::1

où la variable temp\_exte est un unsigned int codé sur les 7 premier bits de l'octet numéro 0,

la luminosité est un unsigned int codé sur 4 bits en partant de l'avant dernier bit de l'octet numéro 1.

Sans précision, les index sont suivants le little endian.

- Url pattern : **https://DESC:<token de lecture>@opentsdb.gra1.metrics.ovh.net/api/put**

où le token de lecture est le token de lecture donné par OVH

opentsdb est le protocole de gestion de base de données orientée séries temporelles distribué, hautes performances,

gra1 est la région<sup>2</sup> où se situe votre module,

<sup>2</sup>

<https://build.sigfox.com/sigfox-radio-configurations-rc>



Fill in the following fields :

- Type **DATA UPLINK**

(to send data without receiving any)

- Channel **URL**
- Custom payload config : **<à définir !>**

In the custom payload, define the value of your variables according to their index in the frame.

Example : temp\_exte:0:uint:7  
luminosity:1:uint:4::1

where the variable temp\_exte is an unsigned int encoded on the first 7 bits of the byte number 0.

the luminosity is an unsigned int encoded on 4 bits, starting from the second last bit of the byte number 1.

Without any information, the little endian rule is applied.

- Url pattern : **https://DESC:<token de lecture>@opentsdb.gra1.metrics.ovh.net/api/put**

where the reading token is the one given by OVH

opentsdb is a distributed, scalable time series database

gra1 is the area<sup>3</sup> where your module is located

<sup>3</sup>

<https://build.sigfox.com/sigfox-radio-configurations-rc>

- Use HTTP Method **POST**
- Send SNI : **checked**
- Headers **000 000**
- Content type **application json**

Vous devrez remplir dans le body toutes les valeurs que vous souhaitez réceptionner. Le code qui change entre chaque bloc ( 1 bloc = 1 donnée) est le nom de la variable écrite dans “metric” et “value” .

- Use HTTP Method **POST**
- Send SNI : **checked**
- Headers **000 000**
- Content type **application json**

At the end, you must fill in all the values you want to receive - inside the body section. The code between each block (1 block = 1 data) that is the most likely to change is the variable's name in the “metric” and “value” fields

The screenshot displays the Sigfox web interface. The top navigation bar includes the Sigfox logo and tabs for 'DEVICE', 'DEVICE TYPE', 'USER', and 'GROUP'. The left sidebar shows 'DEVICES' and 'DELETED DEVICES'. The main content area is titled 'Device - List' and features a search bar with fields for 'Id', 'State', 'Last seen from date', and 'Last seen to date'. Below the search bar is a table with columns: 'Communication status', 'Device type', 'Group', 'Id', 'Last seen', 'Name', and 'Token state'. The table contains one entry for 'SNOC\_DevKit\_1' with a status of 'polytechtp' and a last seen time of '2020-12-08 18:26:44'. Below the table is a 'page 1' indicator. The bottom section shows the 'Content type' set to 'application/json' and a 'Body' section with a JSON array containing two objects, each with 'metric', 'value', and 'tags' fields.

Content type **application/json**

Body

```
[
  {
    "metric": "temp_exte",
    "value": {customData#temp_exte},
    "tags": {
      "host": "web01",
      "dc": "lga"
    }
  },
  {
    "metric": "temp_int",
    "value": {customData#temp_int},
    "tags": {
      "host": "web01",
      "dc": "lga"
    }
  }
],
```

