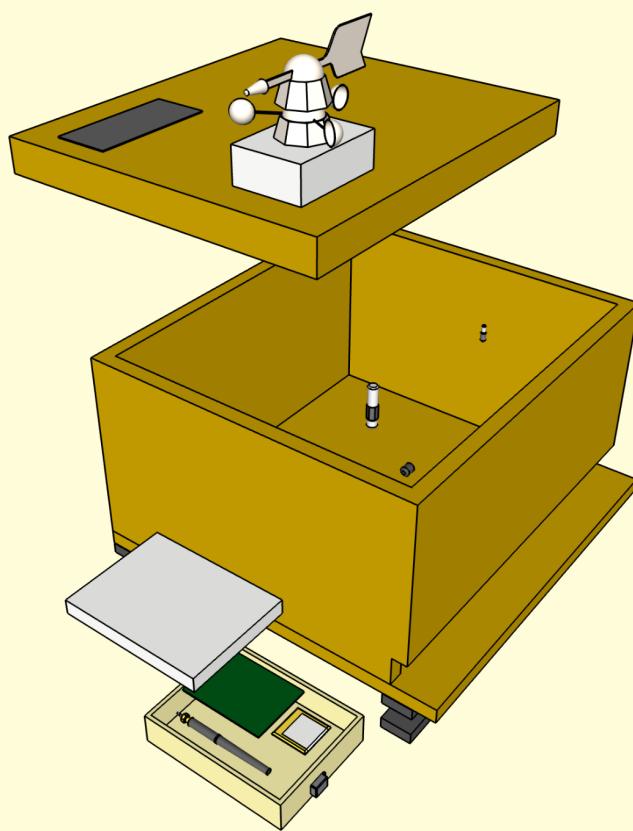
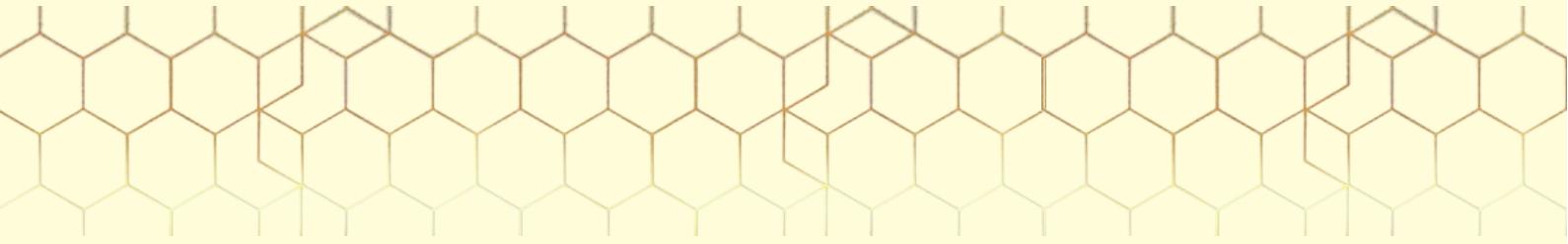

NOTICE D'UTILISATION



SOMMAIRE

Notre but	2
De quoi est composé mon prototype ?	2
Capteurs externes	2
Capteurs internes	3
Comment installer mon prototype ?	4
Mettre le code sur la carte	4
Placement des capteurs	7
Dans le boîtier	8
A l'extérieur de la ruche	14
Dans la ruche	15
Démarrage du prototype	16
Comment accéder à mes données ?	17
Sigfox	17
Première connexion	17
Accéder aux trames de communication	18
Qu'est ce que le Callback ?	20
Ubidots	20
Première connexion	20
Créer un compte	20
Configurer la réception du CallBack	21
Comment personnaliser mon interface ?	28
Accéder à mes données	28
Créer mon tableau de bord	30
Comment créer mes propres alertes ?	37
Puis-je y apporter des modifications ?	40
Modifications matérielle	40
Modifications logicielle	42
Comment réparer mon prototype ?	43
Microcontrôleur	43
Batterie	45
En cas de problème	46
Annexe	47



Nous souhaitons vous remercier de faire confiance à BeeHave pour veiller au bien être de vos abeilles.

Notre but

Les objectifs de ce prototype sont tout d'abord de récolter des données grâce à différents capteurs, puis d'analyser ces données et de les envoyer en temps réel via le réseau Sigfox. Ces données seront ensuite communiquées à une interface homme-machine sur un tableau de bord sur la plateforme Ubidots. L'utilisateur pourra être averti grâce à des alertes sms ou e-mails.

Notre projet s'inscrit dans un triple enjeu. Le premier est économique : nous souhaitons optimiser le rendement du miel pour l'apiculteur, préserver l'essaim et prévenir d'un éventuel essaimage, très coûteux pour l'apiculteur. De plus, notre prototype a pour but d'économiser du temps et des déplacements pour l'apiculteur, afin de faciliter et d'adapter son travail et sa production. Le second enjeu est scientifique : il s'agit d'observer les conditions de vie de la ruche et de prévoir de potentielles anomalies. De plus, nous souhaitons trouver une corrélation entre le son de la ruche et le comportement des abeilles. Enfin, le dernier enjeu est environnemental : en effet, notre prototype est autonome en énergie et basse consommation. Il a pour objectif de réduire la mortalité des abeilles et de permettre une meilleure compréhension de leurs conditions de vie.

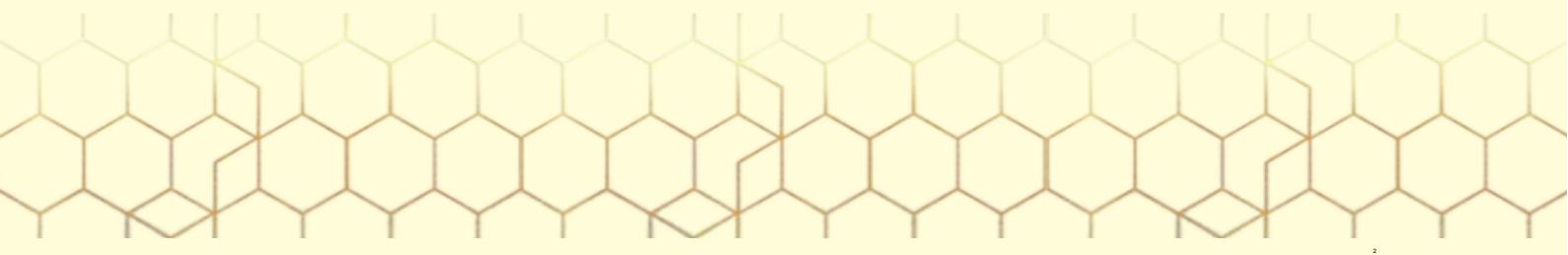
De quoi est composé mon prototype ?

Votre prototype va vous permettre d'acquérir une multitude de données environnementales à l'aide de capteurs placés dans la ruche ou à l'extérieur de celle-ci. Laissez-nous vous présenter les acteurs principaux qui vous permettront de suivre votre ruche. Si vous souhaitez avoir la liste exhaustive des composants accompagnés de leur prix et fournisseurs, veuillez vous référer à l'[annexe](#) se situant en fin de document.

Capteurs externes

Les capteurs externes à la ruche vont permettre d'obtenir les données relatives à la température et à l'humidité extérieures, à la position de la ruche, à l'état de la batterie du système, au poids de la ruche ainsi qu'à la force et la vitesse du vent. On trouvera également le module Wisol accompagné de l'antenne permettant l'envoie de messages sur une plateforme.

La plupart se trouvent dans le petit boîtier, comme l'accéléromètre (MMA8452), la batterie (Accu Li-Ion) et le module Wisol avec l'antenne :

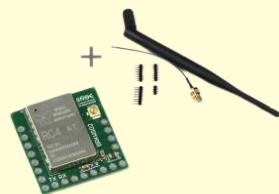




Accéléromètre
position ruche



Batterie
Etat de charge



Antenne
Envoie de messages

Les autres capteurs eux ne se trouvent pas dans la boîte mais à l'extérieur de celle-ci. Leur installation sera détaillée dans la partie suivante.



DHT22
Température &
humidité extérieur



Jauge de poids
Poids de la ruche



Anémomètre et girouette
Force et direction du vent

Capteurs internes

Malgré notre envie d'être le moins invasif possible, certains capteurs doivent se trouver dans la ruche pour récolter des données essentielles au suivi de l'essaim, comme la température et l'humidité ou encore le son de la ruche.

Le placement de ces capteurs dans la ruche sera détaillé dans la partie suivante.



SHT21
Température & humidité
intérieur



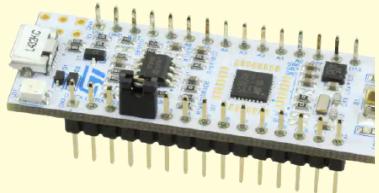
Microphone
Son de la ruche



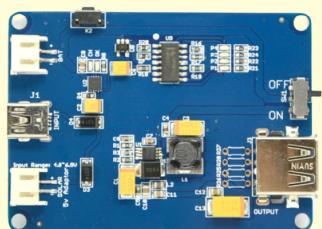
DSB20
Température intérieur



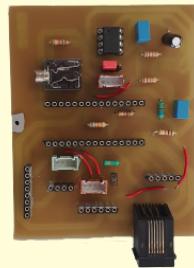
Tous ces capteurs seront reliés à un microcontrôleur et à un Printed Circuit Board (PCB). Ces deux composants vont permettre de faire fonctionner tous les capteurs en même temps et pour acquérir les données de votre ruche. Le Lipo Rider PRO permet quant à lui d'alimenter le système avec la batterie.



Microcontrôleur



Lipo Rider PRO



PCB

Il est important de noter que pour améliorer la durée de vie du prototype, certaines modifications ont été apportées au microcontrôleur. Veuillez vous référer à la partie "[Comment réparer mon prototype?>Microcontrôleur](#)" si votre microcontrôleur est neuf.

Comment installer mon prototype ?

Mettre le code sur la carte

Pour commencer, rendez-vous sur le github créé par nos soins contenant toutes les sources du projet. En effet, les codes sont hébergés sur la plateforme GitHub et sont accessible à partir du lien suivant: https://github.com/Polytech-Sorbonne/EI-SE4_2020-2021_OpenRuche_G8

Les codes doivent être téléchargés en .zip. Pour cela, il suffit de cliquer sur **Code>Download ZIP**. Le code devra être mis dans le microcontrôleur STM32L432KC qui se trouve sur l'image suivante :



Microcontrôleur

Avant d'effectuer toute manipulation, vérifiez bien que vous vous trouvez bien dans la configuration décrite dans la partie [Modifications matérielles>Passer en 5V](#). Une fois la vérification faite,

munissez d'un câble micro-usb vers USB et connectez la carte à votre ordinateur en la branchant à un port USB de votre ordinateur. Une fois branchée, vous devriez voir s'ouvrir une fenêtre avec les fichiers suivant s'ouvrir :

Nom	Modifié le	Type	Taille
DETAILS.TXT	27/05/2004 16:30	Document texte	1 Ko
MBED.HTM	27/05/2006 16:30	Chrome HTML Do...	1 Ko

Si vous vous trouvez bien dans la configuration 5V et qu'un fichier fail.txt est présent en plus de ceux montrés dans la photo, veuillez vous référer à la partie "[Comment réparer mon prototype?>Microcontrôleur](#)".

Pour mettre le code sur la carte, il faut utiliser une interface de développement (IDE). L'équipe de BeeHave vous conseille d'utiliser l'IDE en ligne et gratuite Mbed (<https://os.mbed.com/>). Pour cela, vous devez seulement vous créer un compte et valider votre adresse email. Une fois que ces étapes sont réalisées, rendez-vous sur le lien pour accéder à Mbed et cliquez sur **Compiler** pour accéder au compilateur permettant de mettre le code sur la carte.

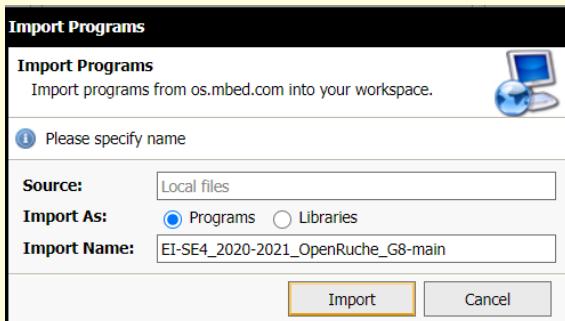
Une fois le compilateur ouvert, vous allez vous retrouver sur la page qui représente l'interface de développement. A droite, dans **Program Workspace**, vous trouverez la liste de tous vos programmes. Elle devrait être vide pour l'instant (on va y importer le programme téléchargé depuis GitHub). Lorsque vous vous retrouverez sur cette page, cliquez sur **Import** comme indiqué sur l'image ci dessous :

The screenshot shows the 'Workspace Management' section of the Mbed IDE. On the left, there's a 'Program Workspace' sidebar with various project icons. In the center, a table lists programs in the workspace, including '08_wakeup', 'ACC', 'batterie', 'Clean', 'consol-led', and 'conso-led'. On the right, there's a 'Workspace Details' panel showing the user 'IgorBGalexQUEFEN' and a 'Recently Modified' list of files. The top navigation bar has several buttons, and a red arrow highlights the 'Import' button.

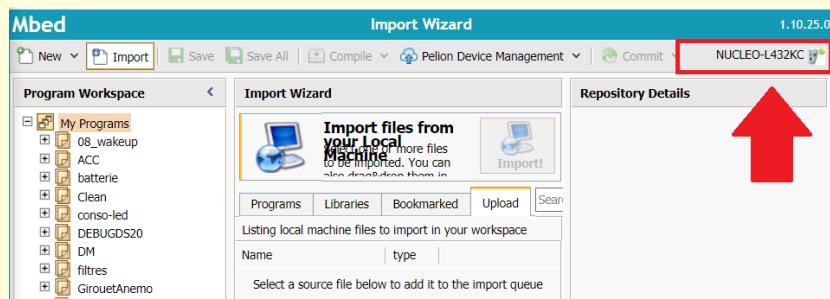
Vous allez ensuite atterrir dans l'Import Wizard. A partir de cette page, cliquez sur **Upload** puis sur **Choisir un fichier** (tout en bas de la page). Après avoir sélectionné le .zip téléchargé plus tôt, cliquez sur **Import** comme indiqué sur la photo.

The screenshot shows the 'Import Wizard' page. It features a central panel titled 'Import files from your Local Machine' with a 'Upload' button highlighted by a red arrow. Below it, there's a file selection area where 'E:\SE4_2020-2021_OpenRuche_G8-main.zip' is listed as a ZIP File. To the right, there's a 'Repository Details' panel. The top navigation bar includes the 'Import' button, which is also highlighted by a red arrow.

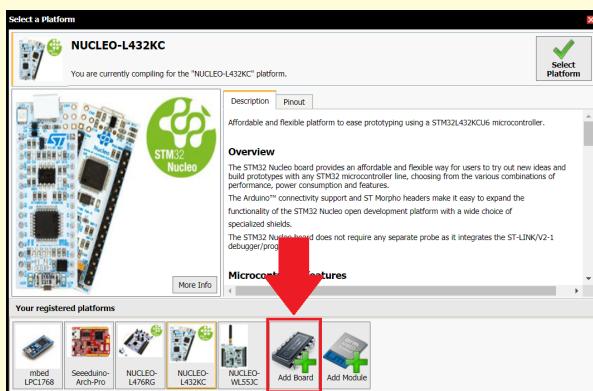
Une fenêtre s'ouvrira, gardez les paramètres par défaut comme indiqué sur l'image suivante puis cliquez sur **Import**.



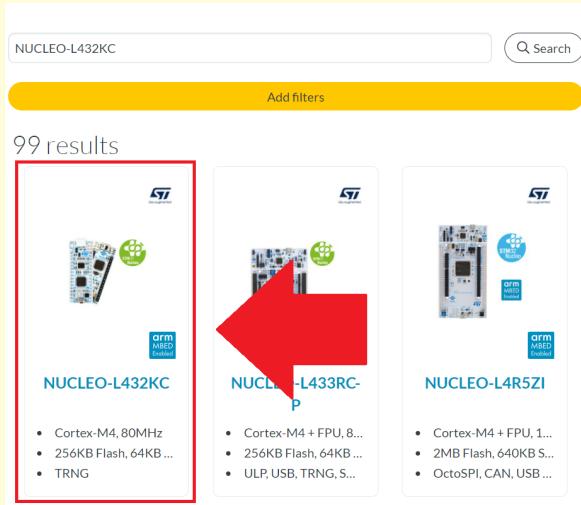
Avant de compiler, vous devrez renseigner la cible pour laquelle vous allez compiler. Pour cela, cliquez le bouton encadré en rouge sur la photo suivante :



La page **Select a platform** va s'ouvrir, vous allez devoir cliquer sur **Add Board** comme indiqué sur l'image ci dessous :



Après avoir cliqué, vous serez redirigé vers la page suivante <https://os.mbed.com/platforms/> ou vous pourrez sélectionner votre microcontrôleur. Pour cela, dans la barre de recherche tapez : "NUCLEO-L432KC" et appuyez sur entrée. Une liste de cartes de développement va se présenter à vous. Sélectionnez la carte possédant le nom recherché dans la barre de recherche comme sur la photo suivante:



Vous serez ensuite redirigé vers la page suivante : <https://os.mbed.com/platforms/ST-Nucleo-L432KC/>. Parcourez la page pour trouver la mention Add to your Mbed Compiler et cliquez dessus comme représenté sur l'image suivante :



Une fois la cible renseignée, retournez sur l'IDE Mbed et cliquez sur le nom de votre projet (se trouvant dans **Programs Workspace**) et appuyez ensuite sur le bouton suivant : 

Si une erreur se produit à la compilation, veuillez vous référer à la partie [En cas de problème](#)

Si la compilation réussit, l'exécutable du programme sera généré et Mbed vous proposera de le télécharger. Après l'avoir téléchargé, il suffira de placer le fichier d'extension .bin dans le microcontrôleur.

Lorsque le microcontrôleur est connecté en USB à l'ordinateur, celui-ci se comporte comme une clé USB. Il suffit donc juste de copier le fichier .bin dans la carte comme on copierait un fichier dans une clé USB.

Placement des capteurs

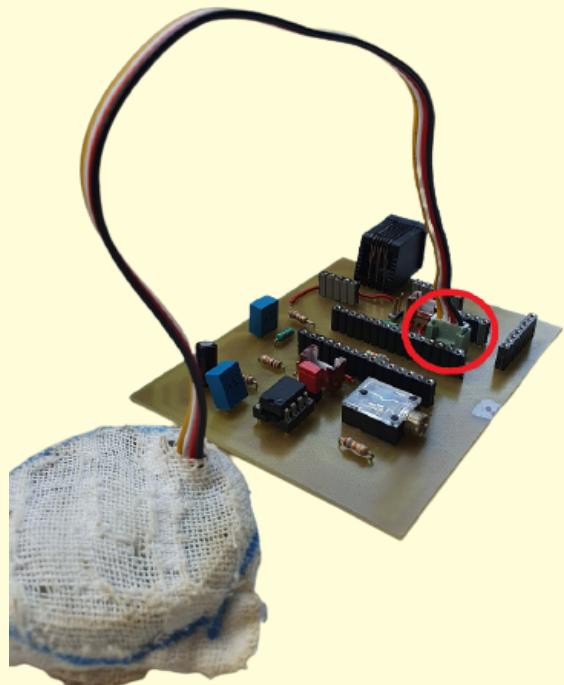
Pour aider à l'implémentation de la solution qu'on vous propose, nous avons réalisé une vidéo vous expliquant comment assembler votre prototype et comment l'installer sur votre ruche. Cette vidéo se nomme *Présentation prototype - Equipe BeeHave* et se trouve sur https://github.com/Polytech-Sorbonne/EI-SE4_2020-2021_OpenRuche_G8.

Dans le boîtier

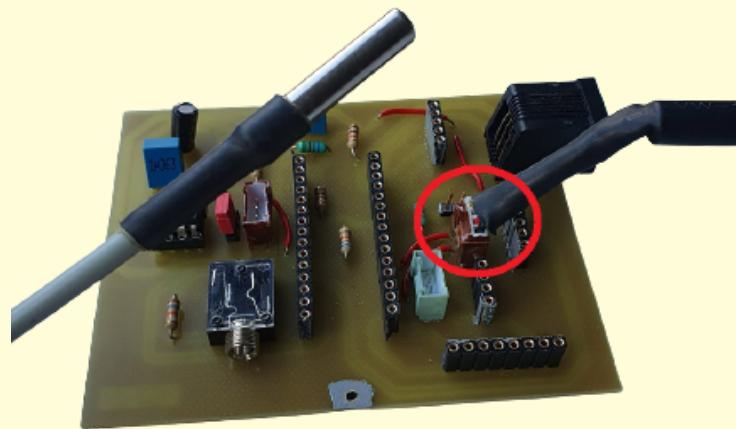
Pour simplifier l'assemblage du prototype et notamment le branchement de chaque capteur sur le PCB, nous avons instauré des astuces impliquant des couleurs que nous allons vous expliciter ici.

Pour commencer, nous avons mis en place un code couleur pour les embouts de type **GROVE**. En effet, nous avons coloré les embouts mâle et femelle de différentes couleurs : un en bleu clair, un en marron et un en rouge.

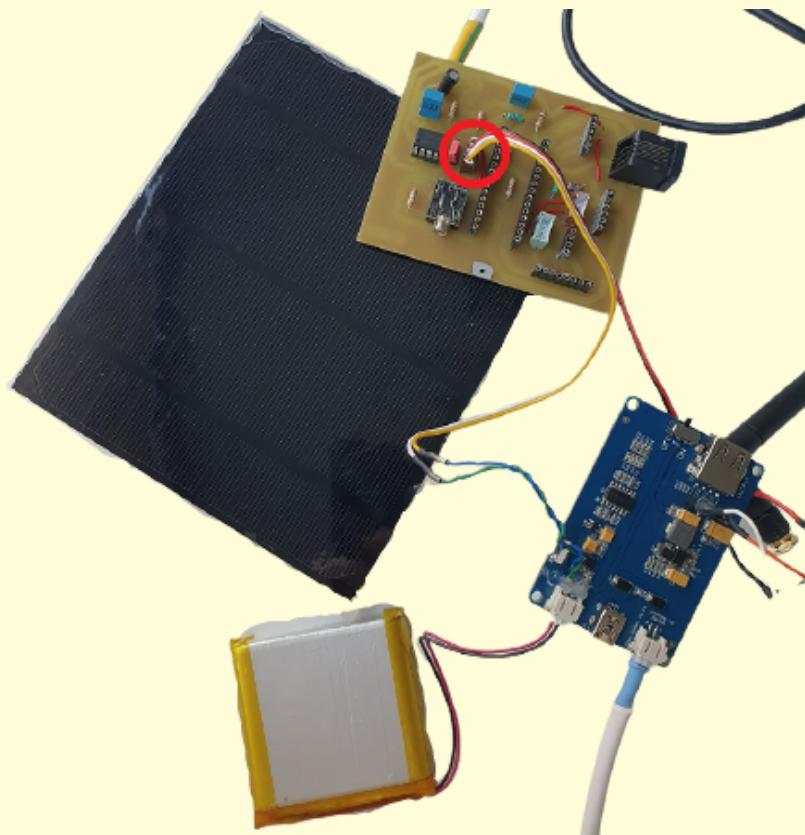
- ❖ Pour le DHT22, la couleur choisie est le bleu clair. Le branchement doit donc être fait de la manière suivante :



- ❖ Pour le DSB20, la couleur choisie est le marron. Le branchement doit donc être fait de la manière suivante :



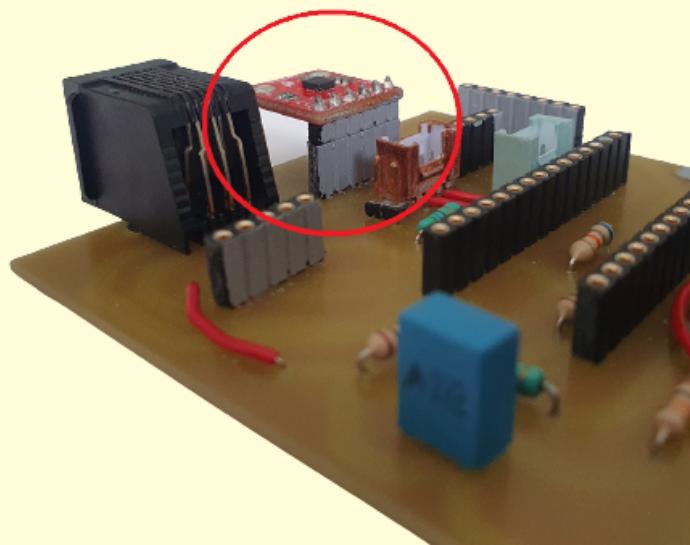
- ❖ Pour l'alimentation, la couleur choisie est le rouge. Le branchement doit donc être fait de la manière suivante :



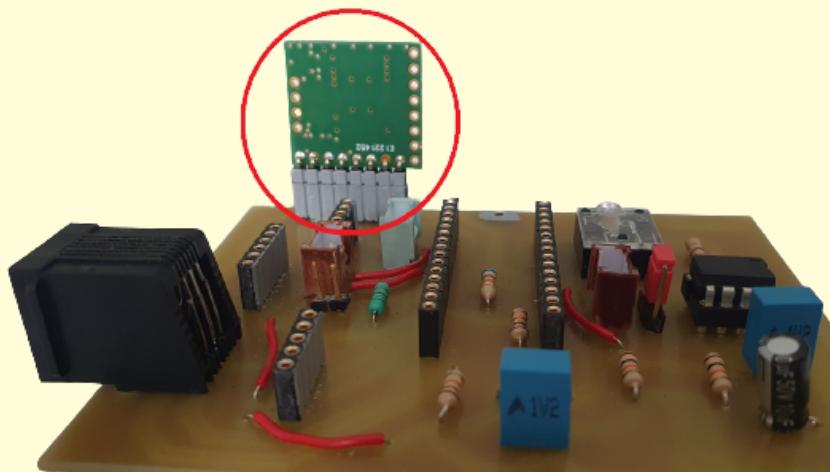
Pour les **headers**, nous avons mis en place un code couleur permettant de savoir dans quel sens doit être brancher le capteur sur le PCB. En effet, nous avons coloré d'une couleur la moitié du header

femelle sur le PCB et la moitié du header mal sur le capteur est coloré de la même couleur. Cela signifie donc que si le capteur est placé correctement, la couleur est continue.

- ❖ Pour l'accéléromètre, le branchement doit être fait de la manière suivante :

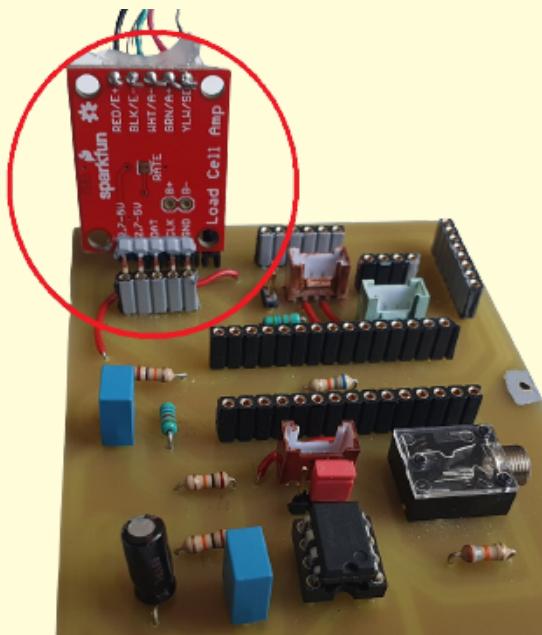


- ❖ Pour le module Wisol et l'antenne, le branchement doit être fait de la manière suivante :



Ne pas oublier de brancher l'antenne au module Wisol

- ❖ Pour le capteur de poids, le branchement doit être fait de la manière suivante :

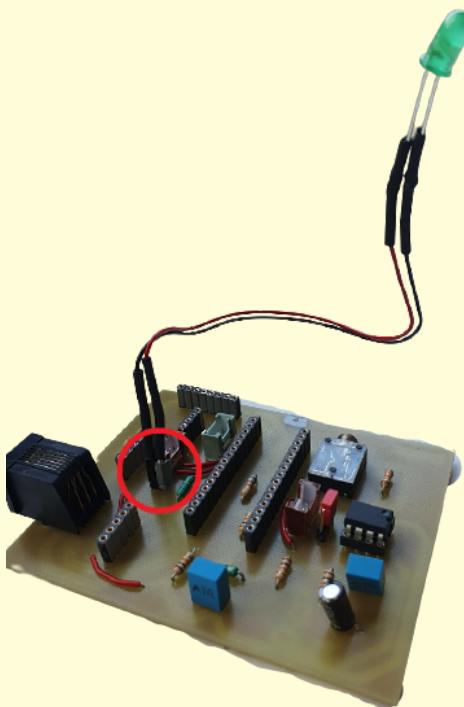


Pour le SHT21 et la LED, nous avons utilisé la même méthode que pour les headers, mais nous avons coloré seulement un header sur le PCB et un header sur le capteur.

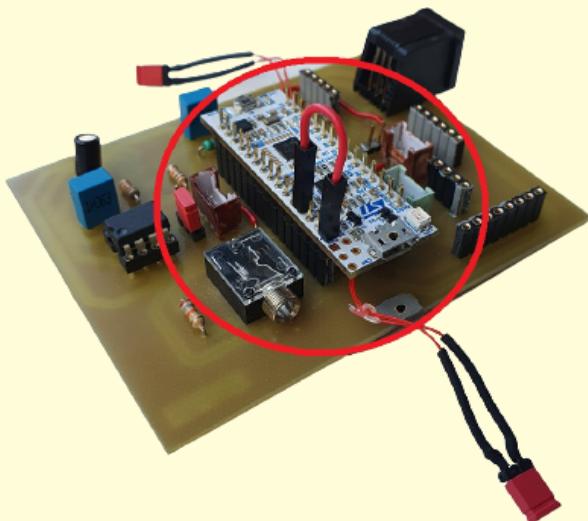
- ❖ Pour le SHT21, le branchement doit être fait de la manière suivante :



❖ Pour la LED, le branchement doit être fait de la manière suivante :



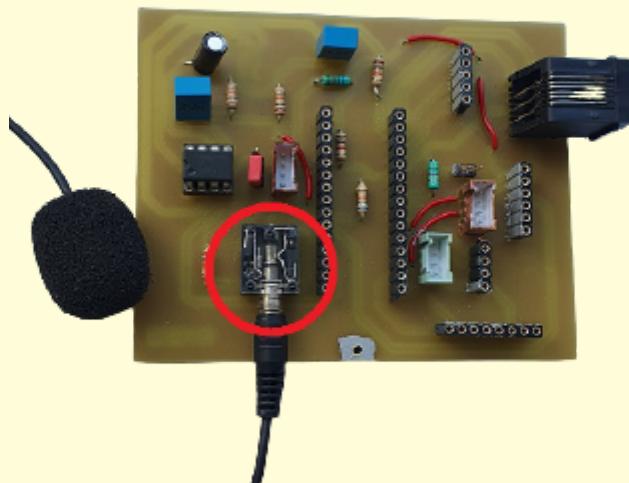
Pour savoir dans quel sens placer le microcontrôleur, nous avons peint un cercle gris avec un point noir sur le microcontrôleur et un autre même cercle avec un point noir sur le PCB. Il faut donc que ces deux marques soient placées l'une au-dessus de l'autre pour que le microcontrôleur soit dans le bon sens.



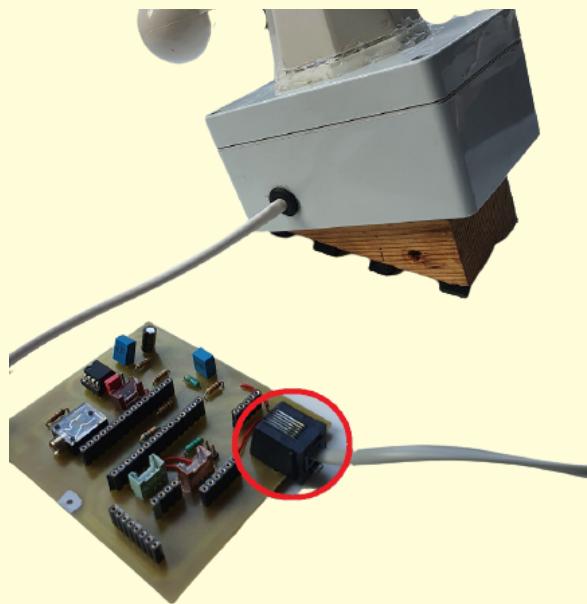
Veuillez à ce que le **câble rouge** entre la pin Vin et la pin 5V du microcontrôleur soit bien branché lors d'une utilisation du prototype en 5V

Enfin pour la girouette/anémomètre et pour le micro, ayant des connecteurs impossible à confondre avec d'autres, nous n'avons pas utilisé d'astuce de couleur.

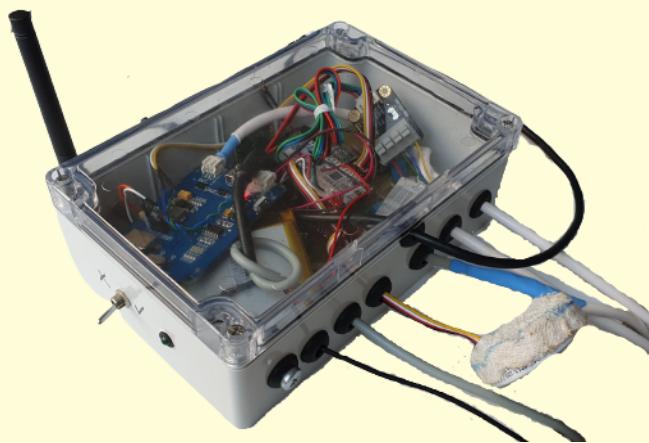
- ❖ Pour le microphone, le branchement doit être fait de la manière suivante :



- ❖ Pour la girouette/anémomètre, le branchement doit être fait de la manière suivante :



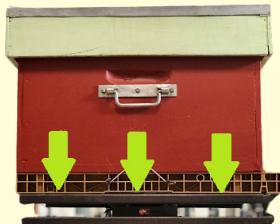
La plupart des capteurs doivent être placés dans le boîtier, notamment l'accéléromètre (MMA8452), la batterie (Accu Li-Ion) ainsi que le module Wi Sol avec son antenne. Après avoir branché tous les capteurs et avoir bien placé les capteurs indiqués précédemment dans la boîte, vous pouvez faire passer les câbles des composants par les trous dans la boîte comme dans la photo suivante, puis fermer la boîte avec le couvercle.



Pour les capteurs n'étant pas présents dans la boîte, leur installation est détaillée dans les parties suivantes.

A l'extérieur de la ruche

D'autres doivent être placés d'une certaine manière à l'extérieur de la ruche :



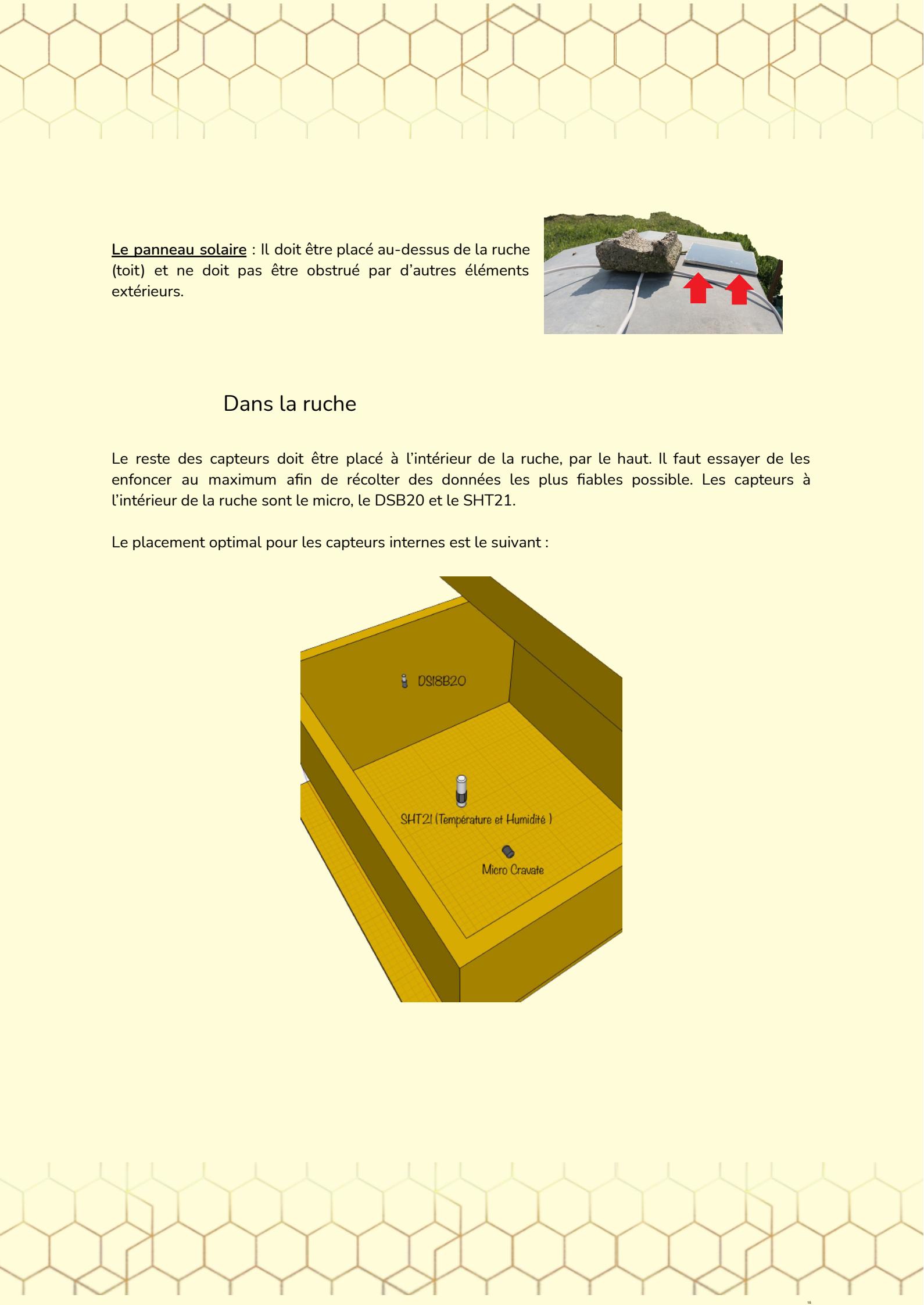
Le capteur de poids : Il doit être placé en dessous de la ruche. Attention au sens : une flèche se trouve sur le capteur pour vous indiquer dans quel sens il doit être placé. La flèche doit pointer vers le sol.



La girouette et l'anémomètre : Elle ne doit pas être à l'abri du vent et doit être placée au-dessus si cela est possible (sur le toit). Il faut impérativement qu'elle soit parallèle au sol et son orientation doit bien correspondre à celle du Nord en fonction de la localisation.



Le DHT22 : Il doit être placé en dessous de la ruche mais en dehors du boîtier.



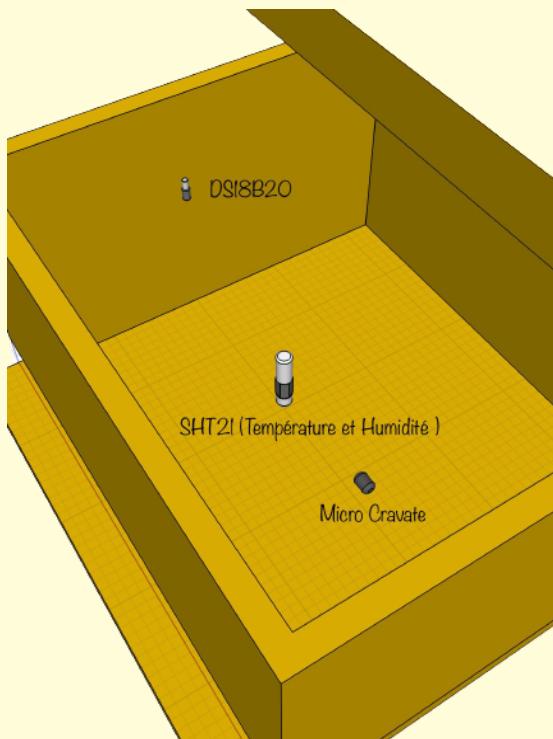
Le panneau solaire : Il doit être placé au-dessus de la ruche (toit) et ne doit pas être obstrué par d'autres éléments extérieurs.



Dans la ruche

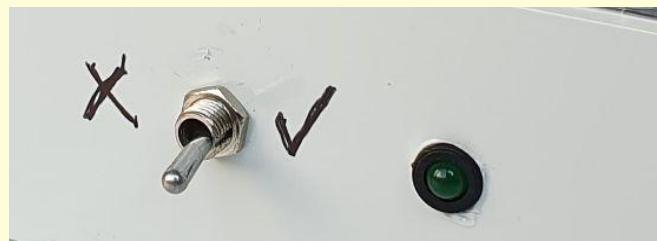
Le reste des capteurs doit être placé à l'intérieur de la ruche, par le haut. Il faut essayer de les enfoncer au maximum afin de récolter des données les plus fiables possible. Les capteurs à l'intérieur de la ruche sont le micro, le DSB20 et le SHT21.

Le placement optimal pour les capteurs internes est le suivant :

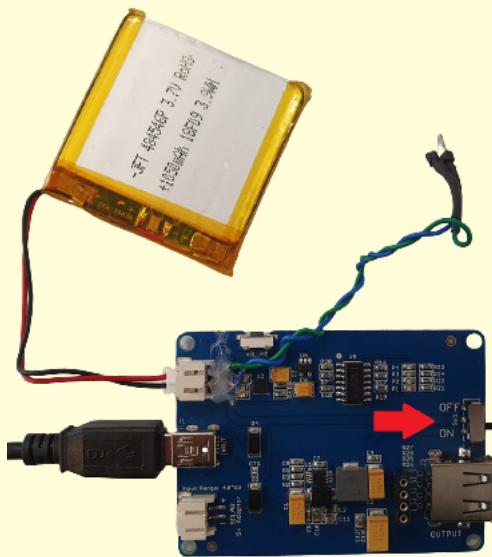


Démarrage du prototype

Pour allumer le prototype, il suffit d'enclencher l'interrupteur qui se trouve sur le boîtier sur la position “ON”, comme on peut le voir sur l'image suivante :



Il faut aussi vérifier que l'interrupteur du Lipo Rider PRO est bien mis sur la position “ON”, comme on peut le voir sur l'image suivante :



Une fois ces étapes réalisées, la LED présente juste à côté de l'interrupteur doit s'allumer pendant quelques secondes. Le prototype est alors fonctionnel.

Pour éteindre le prototype, il suffit de placer les deux interrupteurs cités précédemment sur la position “OFF”.

Comment accéder à mes données ?

Sigfox



Sigfox est un opérateur de télécommunications français de l'IoT. Il permet d'envoyer des données via son réseau à une interface. L'antenne située sur le côté de votre prototype est liée à ce réseau en question.

Première connexion

Pour avoir un accès à vos données et surtout pour activer votre antenne pour lui permettre d'envoyer des messages, vous allez devoir créer un compte sur le Back-end de Sigfox.

- Rendez-vous sur ce lien : <https://buy.sigfox.com/activate>
- Sélectionnez votre pays, ici la France. Cette question est nécessaire pour Sigfox pour savoir si le pays sélectionné est couvert par leur solution ou non. Pour la France, il n'y a aucun problème.

Une fois cette étape passée, vous allez vous retrouver sur une page vous demandant de renseigner les informations suivante :

Provide your DevKit's details for identification

Device ID *

Up to 8 numbers and letters (from A to F)

PAC *

?

Exactly 16 numbers and letters (from A to F)

Ces informations se trouvent sur l'emballage du module WISOL. Après cela, il faudra fournir quelques informations relatives à votre projet.

- Purpose of your project : sélectionnez “Prototype”
- Description : Ajoutez un petit descriptif de votre projet, quelques lignes suffisent.



ex : Connected system placed inside my hive to monitor and track various environmental parameters (temperature, humidity ...) of my bee colony

La prochaine étape va être la création de votre compte. Pour cela, nous vous conseillons d'utiliser une adresse mail Yopmail (pour *Your Own Protection mail*). Yopmail est un système de messagerie en ligne **temporaire** (messages conservés pendant 8 jours) et **gratuit** et qui ne nécessite pas de mot de passe. Pour créer une adresse e-mail Yopmail vous devez :

- Allez sur le lien suivant : <http://www.yopmail.com/>
- Pour l'adresse e-mail, nous vous conseillons de respecter le format suivant :
(ID du module)@yopmail.com

ex : pour un module WISOL ayant un ID de 38CC31 on utilisera l'adresse e-mail
38CC31@yopmail.com

Après avoir créé votre adresse mail Yopmail, vous pouvez créer votre compte. Pour le **login**, renseignez l'adresse e-mail créée précédemment. Le **mot de passe** est libre à vous.

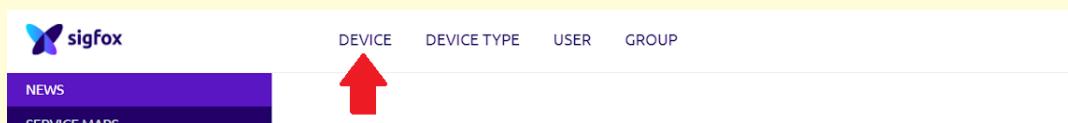
N'oubliez pas de noter le login et le mot de passe choisi.

Une fois tout ceci remplis et valider, retournez sur Yopmail et attendez votre mail de confirmation d'activation et de création du compte. Une fois l'activation terminée, allez sur le site du Back-end de Sigfox (<https://backend.sigfox.com/auth/login>) et connectez vous avec le même identifiant et mot de passe que vous avez défini précédemment.

Et voilà, votre compte est créé. Pour les prochaines connexions, il faudra utiliser l'identifiant et le mot de passe que vous aurez défini ici.

Accéder aux trames de communication

Pour accéder aux données reçues sur le back-end, il vous suffit de vous diriger vers l'onglet "Device" en haut de la fenêtre :



Une nouvelle fenêtre est affichée. Il s'agit de tous les appareils reliés à votre compte Sigfox :

Communication status	Device type	Group	Id	Last seen	Name	Token state
	SNOC_DevKit_1	Honey Stonks	2E6D5B	2021-01-18 12:52:23	SNOC_DevKit_1-device	<input checked="" type="checkbox"/>

Pour cet exemple, le nom de l'appareil en question est SNOC_DevKit_1_device. Il est fortement déconseillé de modifier ses paramètres. Pour accéder aux messages émis par le prototype, il faut cliquer sur l'ID de l'appareil. Ici, **2E6D5B**.

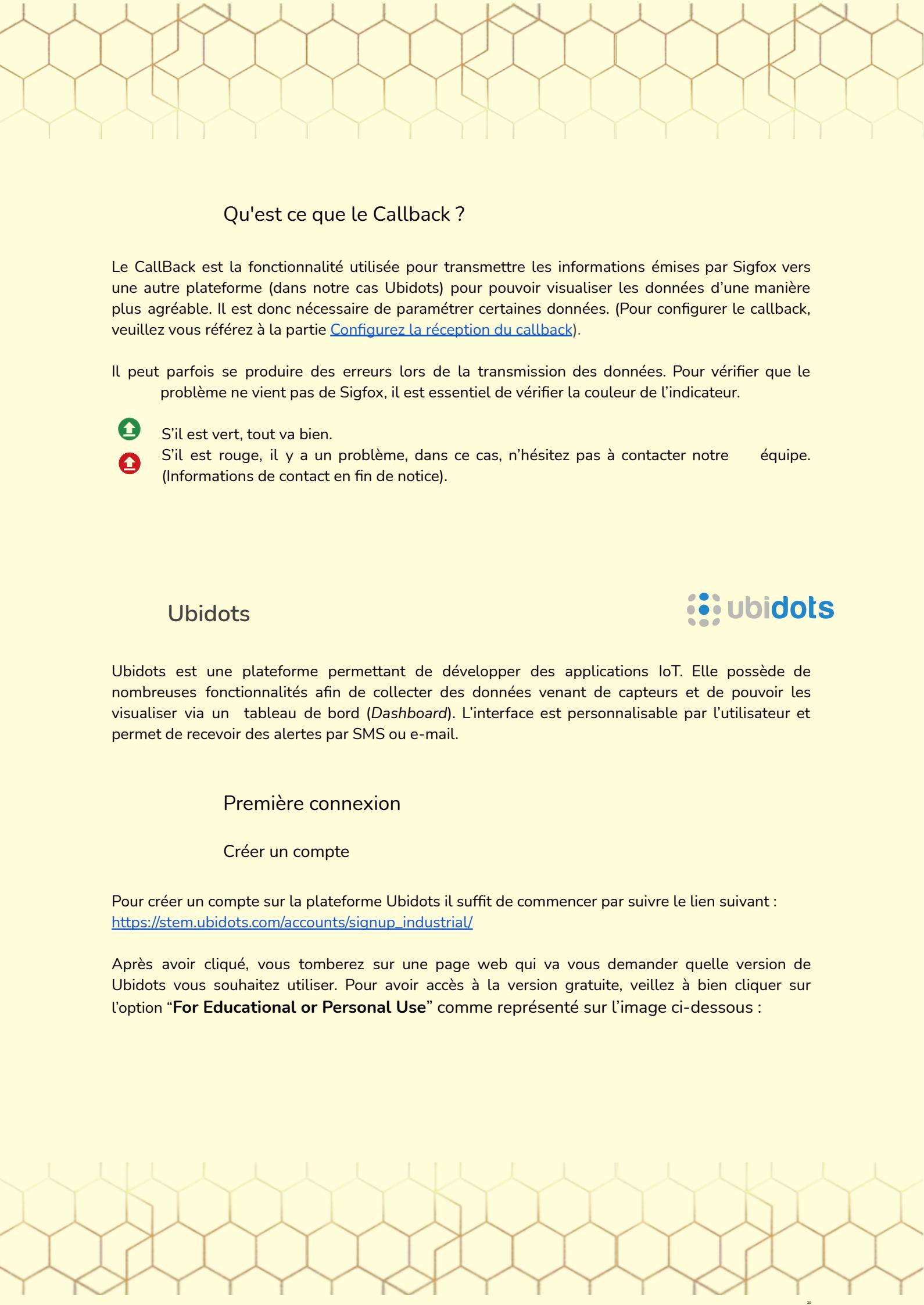
Une fois cette tâche réalisée, votre fenêtre doit ressembler à la photo suivante :

The screenshot shows the Sigfox WEB interface. On the left, there's a sidebar with navigation links: INFORMATION, LOCATION, MESSAGES (which is highlighted in purple), EVENTS, STATISTICS, and EVENT CONFIGURATION. The main content area has tabs at the top: DEVICE, DEVICE TYPE, USER, and GROUP. Below these, it says "Device 2E6D5B - Messages". There are input fields for "From date" and "To date", and buttons for "RESET", "FILTER", and a magnifying glass icon. The main part is a table with the following data:

Time	Seq Num	Data/Decoding	LQI	Callbacks	Location
2021-01-18 12:52:23	1306	850439163843000000000000000000			
2021-01-18 12:47:32	1305	6504399738c30000000000000000000			
2021-01-18 12:30:40	1304	840538d53843000000000000000000			
2021-01-18 12:29:39	1303	840538d53843000000000000000000			
2021-01-18 12:28:39	1302	840538d53883000000000000000000			

A gauche, se trouve une colonne avec les différentes fonctions proposées par l'application WEB. Celle qui nous intéresse est la rubrique "**Messages**".

Un tableau rassemble toutes les données reçues. La première colonne correspond à la date, la deuxième au numéro du message, la troisième à la force du signal, la quatrième à ce qu'on appelle le *CallBack*, et la dernière qui permet de localiser son appareil.



Qu'est ce que le Callback ?

Le CallBack est la fonctionnalité utilisée pour transmettre les informations émises par Sigfox vers une autre plateforme (dans notre cas Ubidots) pour pouvoir visualiser les données d'une manière plus agréable. Il est donc nécessaire de paramétrer certaines données. (Pour configurer le callback, veuillez vous référer à la partie [Configurez la réception du callback](#)).

Il peut parfois se produire des erreurs lors de la transmission des données. Pour vérifier que le problème ne vient pas de Sigfox, il est essentiel de vérifier la couleur de l'indicateur.



S'il est vert, tout va bien.



S'il est rouge, il y a un problème, dans ce cas, n'hésitez pas à contacter notre équipe. (Informations de contact en fin de notice).

Ubidots



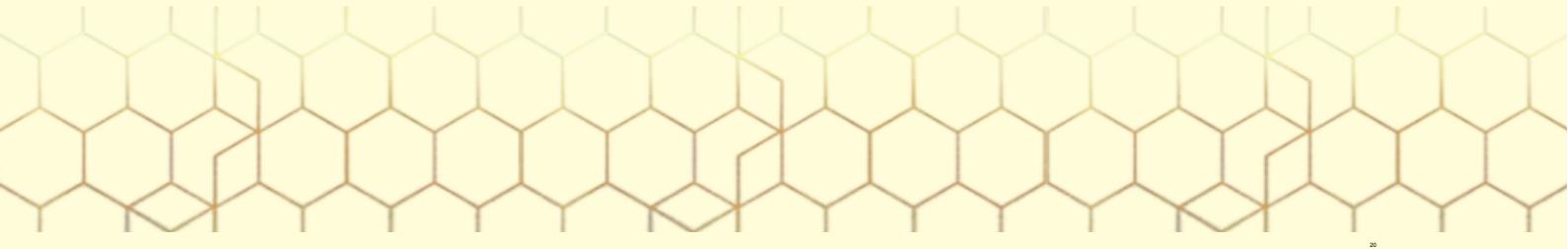
Ubidots est une plateforme permettant de développer des applications IoT. Elle possède de nombreuses fonctionnalités afin de collecter des données venant de capteurs et de pouvoir les visualiser via un tableau de bord (*Dashboard*). L'interface est personnalisable par l'utilisateur et permet de recevoir des alertes par SMS ou e-mail.

Première connexion

Créer un compte

Pour créer un compte sur la plateforme Ubidots il suffit de commencer par suivre le lien suivant :
https://stem.ubidots.com/accounts/signup_industrial/

Après avoir cliqué, vous tomberez sur une page web qui va vous demander quelle version de Ubidots vous souhaitez utiliser. Pour avoir accès à la version gratuite, veillez à bien cliquer sur l'option **“For Educational or Personal Use”** comme représenté sur l'image ci-dessous :



How would you like to use Ubidots?

Whether you're prototyping or going to market,
Ubidots is designed to accelerate your IoT initiatives.

For Educational or Personal Use



Join 60,000+ students, makers and professors
using our FREE Ubidots STEM platform to
prototype, learn, or teach IoT.

[TAKE ME TO UBIODOTS STEM](#)

For Business



Join 1,000+ System Integrators, OEMs and IoT
Entrepreneurs building connected products and
services with Ubidots.

[CONTINUE →](#)

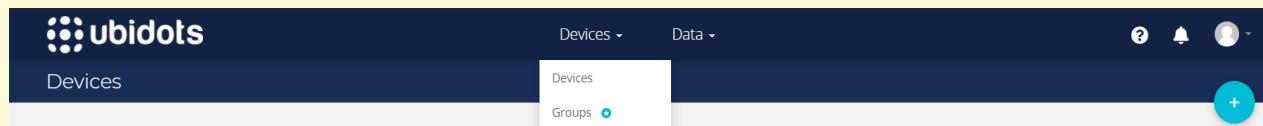
Vous serez ensuite amené sur une page vous demandant de renseigner un nom d'utilisateur, une adresse e-mail et un mot de passe. Par souci de simplicité, nous vous conseillons d'utiliser la même adresse e-mail et le même mot de passe que ceux renseignés pour l'inscription sur le back-end de Sigfox (cf [Sigfox > Créer un compte](#)). Pour le nom d'utilisateur vous êtes libre d'utiliser celui qui vous convient.

N'oubliez pas de noter le login et le mot de passe choisi.

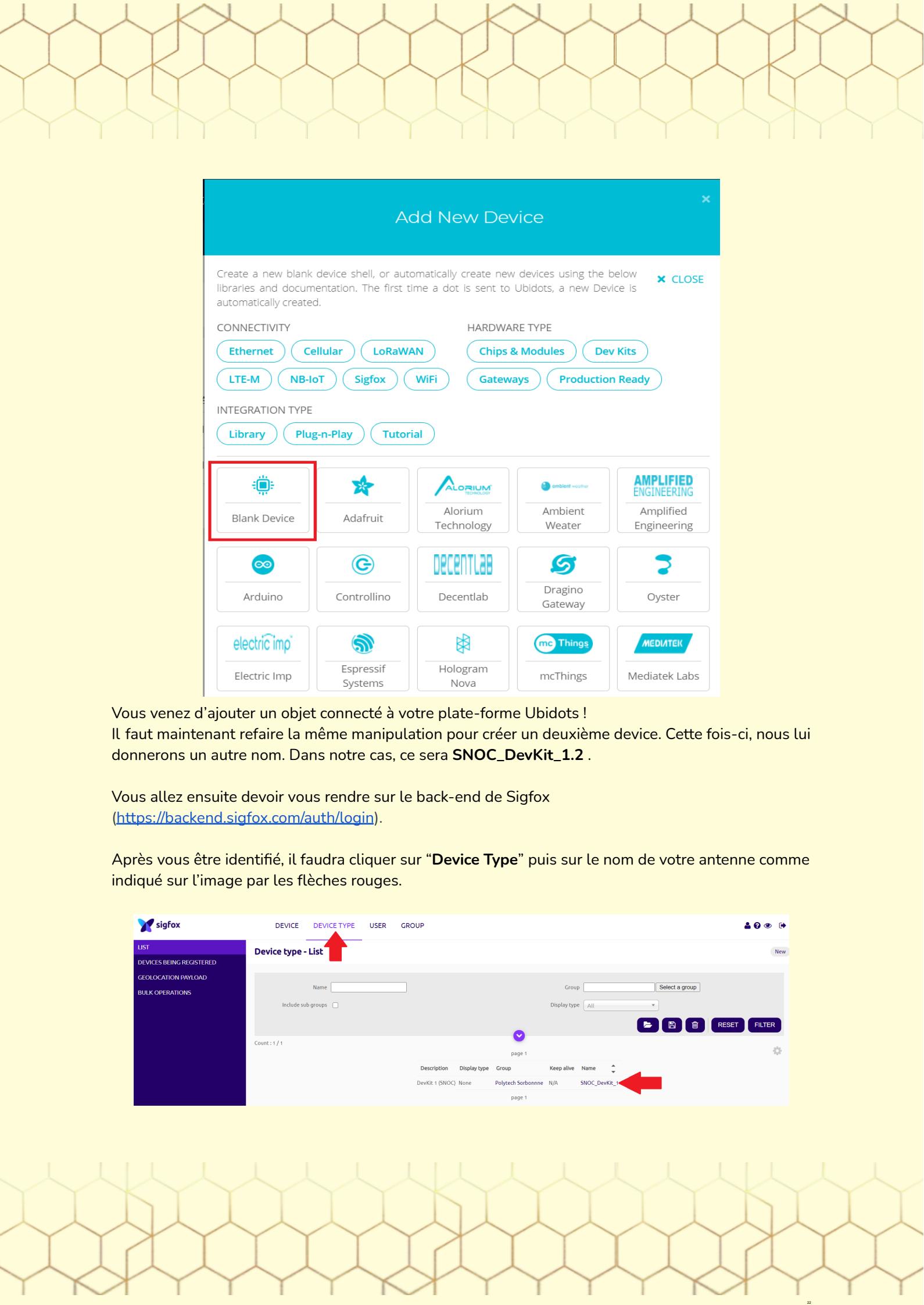
Configurer la réception du CallBack

Pour configurer la réception de votre callback, tout d'abord, vous devez aller sur Ubidot (https://stem.ubidots.com/accounts/signup_industrial/) pour créer un objet avec un identifiant unique représentant votre système.

Pour ce faire, il faut commencer par cliquer sur le menu déroulant “Devices” ou “Dispositifs” et cliquer sur “Devices”



Une fois cette nouvelle page ouverte, il faut cliquer sur le signe “+” en haut à droite de votre écran. L'onglet “Add New Device” s'ouvre. Il faut créer un Device vierge. C'est-à-dire qu'il faut sélectionner le bouton “Blank Device”. Il vous suffira alors d'entrer un nom à votre device. Dans notre cas, il s'agit d'entrer **SNOC_DevKit_1**.



Add New Device

Create a new blank device shell, or automatically create new devices using the below libraries and documentation. The first time a dot is sent to Ubidots, a new Device is automatically created.

X CLOSE

CONNECTIVITY

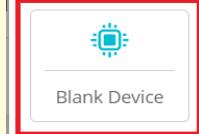
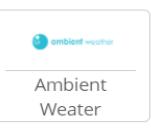
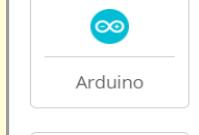
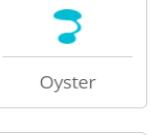
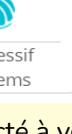
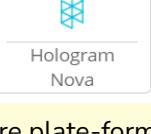
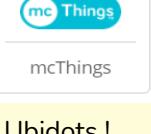
- [Ethernet](#)
- [Cellular](#)
- [LoRaWAN](#)
- [LTE-M](#)
- [NB-IoT](#)
- [Sigfox](#)
- [WiFi](#)

HARDWARE TYPE

- [Chips & Modules](#)
- [Dev Kits](#)
- [Gateways](#)
- [Production Ready](#)

INTEGRATION TYPE

- [Library](#)
- [Plug-n-Play](#)
- [Tutorial](#)

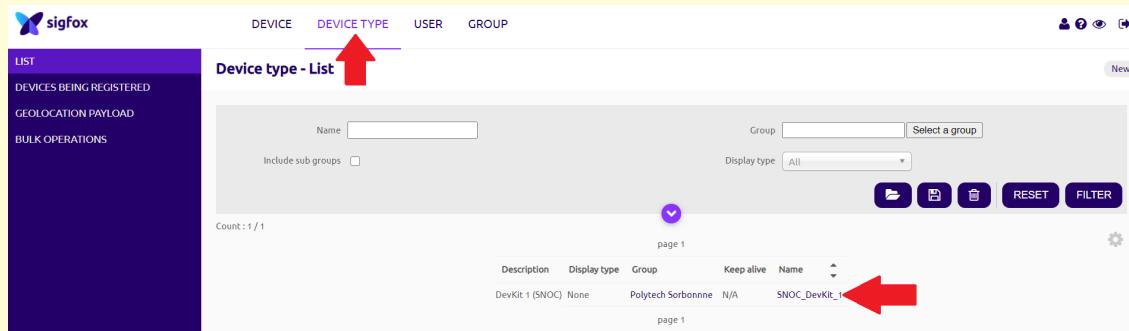
 Blank Device	 Adafruit	 Alorium Technology	 Ambient Weather	 Amplified Engineering
 Arduino	 Controllino	 Decentlab	 Dragino Gateway	 Oyster
 Electric Imp	 Espressif Systems	 Hologram Nova	 mcThings	 Mediatek Labs

Vous venez d'ajouter un objet connecté à votre plate-forme Ubidots !

Il faut maintenant refaire la même manipulation pour créer un deuxième device. Cette fois-ci, nous lui donnerons un autre nom. Dans notre cas, ce sera **SNOC_DevKit_1.2**.

Vous allez ensuite devoir vous rendre sur le back-end de Sigfox
(<https://backend.sigfox.com/auth/login>).

Après vous être identifié, il faudra cliquer sur “Device Type” puis sur le nom de votre antenne comme indiqué sur l'image par les flèches rouges.



The screenshot shows the Sigfox Device Type - List interface. At the top, there are tabs for DEVICE, DEVICE TYPE, USER, and GROUP. The DEVICE TYPE tab is highlighted with a red arrow. Below the tabs, there is a search bar with 'Name' and 'Group' fields, and a 'Display type' dropdown set to 'All'. There are also buttons for 'New', 'Edit', 'Delete', 'RESET', and 'FILTER'. A table lists one device entry: 'DevKit 1 (SNOC)' with 'None' under 'Display type', 'N/A' under 'Group', and 'SNOC_DevKit_1' under 'Name'. A red arrow points to the 'Name' field in this row.



Vous allez ensuite tomber sur une page et sur le bandeau bleu foncé présent sur le côté gauche de votre écran, vous trouverez la rubrique "**CallBack**". Cliquez dessus.



DEVICE DEVICE TYPE USER GROUP

INFORMATION
LOCATION
ASSOCIATED DEVICES
DEVICES BEING REGISTERED
STATISTICS
EVENT CONFIGURATION
CALLBACKS ←
BULK OPERATIONS

Device type 'SNOC_DevKit_1' - Information

Id: 604a2bb3988a371299e7f4b3
Name: SNOC_DevKit_1
Description: DevKit 1 (SNOC)
Keep alive: N/A
Subscription automatic renewal: ⓘ
Group: Polytech Sorbonne
Payload display: None
Downlink mode: CALLBACK
Contracts:
1. polytech_9887_1513f (no token left - geoloc: yes, end date: 2023-03-11)
Alert Email:
Creation date: 2021-03-11 15:39:47
Created by: buy.sigfox.com
Last edition date: 2021-03-11 15:39:46
Last edited by: buy.sigfox.com

La page sur laquelle vous allez vous trouverez contient la liste de tous les Callback que vous avez créé. Normalement, celle-ci est vide. Pour créer un nouveau callback cliquez sur “**New**” comme représenté sur l’image ci dessous :

Device type 'SNOC_DevKit_1' - Callbacks

Sélectionnez “Custom Callback” qui est encadré en rouge sur l’image suivante :

Device type 'SNOC_DevKit_1' - New Callback	
INFORMATION	
LOCATION	
ASSOCIATED DEVICES	
DEVICES BEING REGISTERED	
STATISTICS	
EVENT CONFIGURATION	
CALLBACKS	 Custom callback Creates a new callback from SigFox cloud to your own server. This is the "default" callback type. You can create a full custom request (http method, content type, headers, etc).
BULK OPERATIONS	
AWS IoT	AWS IoT is a managed cloud platform that lets connected devices easily and securely interact with cloud applications and other devices. AWS IoT can

La page sur laquelle vous allez arriver sera vide et vous allez devoir la remplir de la manière suivante:





sigfox

DEVICE DEVICE TYPE USER GROUP

INFORMATION
LOCATION
ASSOCIATED DEVICES
DEVICES BEING REGISTERED

Device type SNO_C_DevKit_1 - Callback edition

Callbacks

Type: DATA / UPLINK
Channel: URL

Custom payload config: batt:0:uint:4 girou:0:uint:3:3 move:0:bool:0 anemo:1:uint:4 humEXT:1:uint:4::3 templ:2::3
 URL syntax: http://host/path?id={device}&time={time}&key1={var1}&key2={var2}...
 Available variables: device, time, data, seqNumber, deviceType
 Custom variables: customData#batt, customData#girou, customData#move, customData#anemo, customData#humEXT, customData#tempINT_SHT, customData#tempEXT, customData#temp_DSB, customData#hum_INT_SHT, customData#poids, customData#amp225285, customData#amp100, customData#amp3000, customData#amp1100
 Url pattern: https://industrial.api.ubidots.com/api/v1.6/devices/SNO_C_DevKit_1/

Use HTTP Method: POST
Send SNI: (Server Name Indication) for SSL/TLS connections
Headers: x-auth-token: BBFF-nXMsLgj5nuLcO67gsi70kda5lmyWfk
header value

Content type: application/json

Body:

```
[{"Batt": {"value": "customData#batt"}, "girou": {"value": "customData#girou"}, "move": {"value": "customData#move"}, "anemo": {"value": "customData#anemo"}, "humEXT": {"value": "customData#humEXT"}, "tempINT_SHT": {"value": "customData#tempINT_SHT"}, "tempEXT": {"value": "customData#tempEXT"}, "temp_DSB": {"value": "customData#temp_DSB"}, "hum_INT_SHT": {"value": "customData#hum_INT_SHT"}, "poids": {"value": "customData#poids"}, "amp225285": {"value": "customData#amp225285"}, "amp100": {"value": "customData#amp100"}, "amp3000": {"value": "customData#amp3000"}, "amp1100": {"value": "customData#amp1100"}]
```

Ok Cancel

Les 2 premiers champs “**Type**” et “**Channel**” sont correctement renseignés dès le début. (Type : DATA; UPLINK)

Pour le champ “**Custom payload config**”, veuillez copié-coller la phrase suivante dans l’emplacement dédié :

```
batt:0:uint:4 girou:0:uint:3:3 move:0:bool:0 anemo:1:uint:4 humEXT:1:uint:4::3  

templINT_SHT:2:uint:10 tempEXT:3:uint:6::5 temp_DSB:4:uint:10 hum_INT_SHT:5:uint:6::5  

poids:6:uint:16 amp225285:8:uint:8 amp100:9:uint:8 amp3000:10:uint:8 amp1100:11:uint:8
```

Pour le champ “**URL-pattern**” vous devez vous seulement renseigner l’URL suivant: <https://industrial.api.ubidots.com/api/v1.6/devices/{device}/> en remplaçant {device} par le nom de votre device précédemment créé.

Dans notre cas, le premier se nomme **SNO_C_DevKit_1**, notre URL est donc le suivant: https://industrial.api.ubidots.com/api/v1.6/devices/SNO_C_DevKit_1/

Pour le champ “**Use HTTP Method**”, pensez à bien sélectionner “**POST**”.

Pour le champ “**Headers**”, dans le premier emplacement renseignez “x-auth-token” comme représenté sur la photo précédente. Le code présent dans le second emplacement est unique à chaque Device. Pour accéder aux Tokens de votre objet, vous devez vous rendre sur Ubidot, dans la rubrique Devices.

Une fois dans la liste, cliquez sur le premier Device, qui se nomme **SNOC_DevKit_1**. Une nouvelle fenêtre s'affiche:

The screenshot shows the Ubidots Device details page for 'snoc_devkit_1'. The top navigation bar has a blue header with the device name. Below it, there's a 'Description' section with a 'Change description' button, an 'API Label' set to 'snoc_devkit_1', and an 'ID' of '604a434c0ff4c34b8d151fe1'. A 'Token' field is present, which is highlighted with a red border. To the right, there are four orange cards showing real-time data: 'anemo' (0.00), 'batt' (10.00), and 'girou' (2.00). Each card includes a 'Last activity' timestamp of '13 minutes ago'.

Dans la case "**Token**", récupérez la chaîne de caractères. Copiez-Collez le Token dans le 2e emplacement du champ "**Headers**" qui se nomme "**Value**".

Pour le champ "**Content type**", renseignez : [application/json](#)

Et enfin, pour le champ "**Body**", vous n'avez qu'à réalisé un copié collé du texte suivant dans l'emplacement dédié :

```
{  
  "batt": {"value": "{customData#batt}"},  
  "girou": {"value": "{customData#girou}"},  
  "move": {"value": "{customData#move}"},  
  "anemo": {"value": "{customData#anemo}"},  
  "humEXT": {"value": "{customData#humEXT}"},  
  "tempINT_SHT": {"value": "{customData#tempINT_SHT}"},  
  "tempEXT": {"value": "{customData#tempEXT}"},  
  "temp_DSB": {"value": "{customData#temp_DSB}"},  
  "hum_INT_SHT": {"value": "{customData#hum_INT_SHT}"}  
}
```

Il s'agit maintenant de créer un deuxième Callback. Il faut reprendre la même procédure. Cette fois-ci 4 champs vont être modifiés:

- Custom payload config, pour lequel il faudra écrire :
[poids:6:uint:16 amp225285:8:uint:8 amp100:9:uint:8 amp3000:10:uint:8 amp1100:11:uint:8](#)
- Url pattern, pour lequel il faudra écrire l'URL associée à notre deuxième device, <https://industrial.api.ubidots.com/api/v1.6/devices/{device2}>. Dans notre cas, l'URL sera : https://industrial.api.ubidots.com/api/v1.6/devices/SNOC_DevKit_1.2/

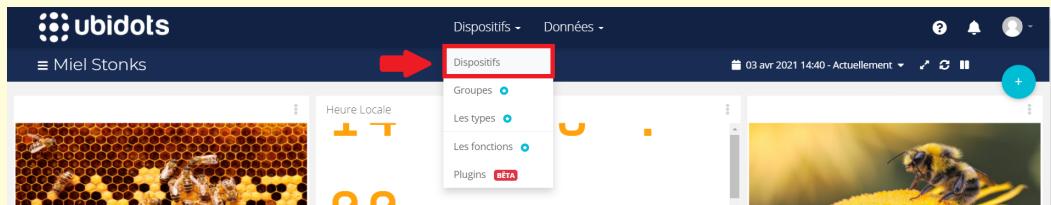
→ Value, le champ associé à "Header", pour lequel il faudra insérer le token du deuxième Device créé sur Ubidot.

→ Le body, du Content Type, pour lequel il faudra écrire:

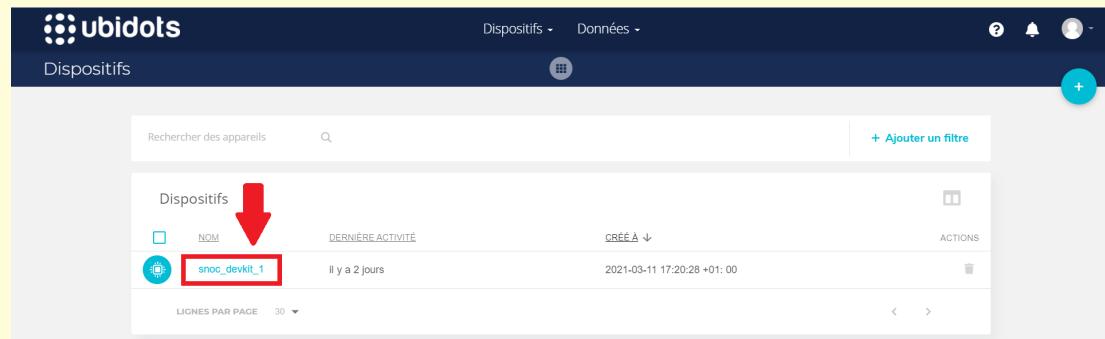
```
{  
  "poids": {"value": "{customData#poids}"},  
  "amp225285": {"value": "{customData#amp225285}"},  
  "amp100": {"value": "{customData#amp100}"},  
  "amp3000": {"value": "{customData#amp3000}"},  
  "amp1100": {"value": "{customData#amp1100}"}  
}
```

Vous avez maintenant réussi à configurer le callback du côté de Sigfox. Nous devons maintenant aller sur Ubidot pour créer certaines variables personnalisées. Par souci de légèreté dans les données envoyées, il a été nécessaire d'effectuer des opérations mathématiques sur vos données avant l'envoi. Il faut donc effectuer les opérations inverses à la réception.

Rendez-vous donc sur Ubidot et cliquez sur *Dispositifs > Dispositifs* comme représenté sur l'image ci-dessous :



Ensuite cliquez sur le nom de votre antenne :



Par la suite, descendez en bas de la page et cliquer sur "Ajouter une variable" puis sur "Synthétique":

The screenshot shows a dashboard interface with a hexagonal background pattern. At the top, there are six data cards arranged in two rows of three. Each card displays a value, a variable name, and a timestamp of the last activity.

- Row 1:**
 - Value: 40.00, Variable: Real_humEXT, Last activity: 30 minutes ago
 - Value: 50.00, Variable: Real_hum_INT_SHT, Last activity: 30 minutes ago
 - Value: 31.40, Variable: Real_temp_INT_DSB, Last activity: 29 minutes ago
- Row 2:**
 - Value: 31.40, Variable: Real_temp_INT_SHT, Last activity: 31 minutes ago
 - Value: 321.00, Variable: temp_dsb, Last activity: 3 minutes ago
 - Value: 13.00, Variable: temptext, Last activity: 3 minutes ago

In the center of the dashboard is a dashed box containing a grey circle with a plus sign and the text "Add Variable". To the right of this box is a white card labeled "Raw" at the top. Below it, there is a red-bordered button labeled "Σ Synthetic". A large red arrow points from the text "Synthetic" in the "Raw" card towards this button.

Vous allez ensuite tomber sur la page suivante où il faudra simplement sélectionner parmi la liste située à droite la variable et appliquer l'opération qu'on va vous décrire dans un tableau ci-dessous.

The screenshot shows a modal dialog titled "snoc_devkit_1 Synthetic Variable". The title bar has a close button (X) in the top right corner.

The main area contains the following elements:

- Input Field:** "Enter your synthetic variable expression. [Learn more about synthetic variables](#)" with a placeholder "batt (snoc_devkit_1) *1d". A green arrow points to this field with the label "Opération". A red arrow points to the "batt" part of the expression with the label "Variable".
- Timezone Selection:** "Timezone: Europe/Berlin" with a dropdown arrow.
- List of Variables:** "Click on a variable to add it to your expression:" followed by a scrollable list of variables:
 - humext
 - hum_int_sht
 - move
- Buttons:** "Cancel" and "Accept" at the bottom.

Vous allez devoir créer en tout 6 variables. Les opérations et les variables à sélectionnées sont les suivantes :

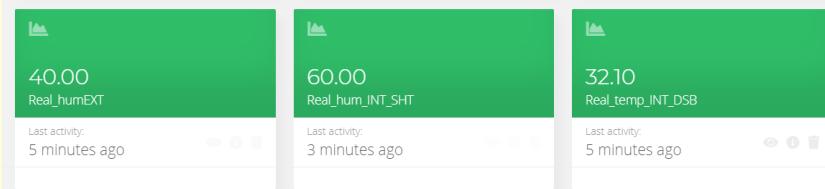
Variable	Opération	Final
anemo	*10	anemo *10
batt	*10	batt *10
humext	*10	humext *10
hum_int_sht	*10	hum_int_sht *10
temp_dsb	/10	temp_dsb /10
tempint_sht	/10	tempint_sht /10

Une fois ces 6 variables créées, vous avez fini de configurer la réception de votre callback.

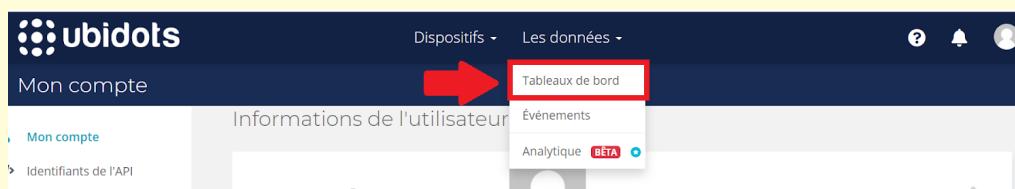
Comment personnaliser mon interface ?

Accéder à mes données

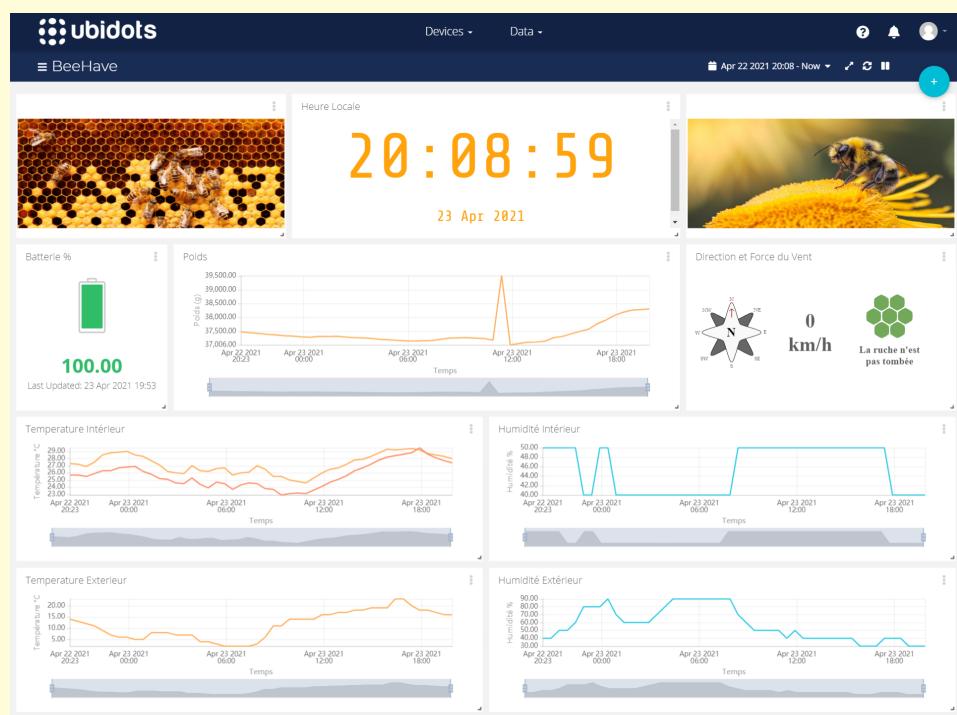
Il existe deux manières d'accéder à vos données. Tout d'abord, comme présenté précédemment (cf. [Configurer la réception du CallBack](#)), vous pouvez y accéder à partir de **Devices>Devices** et vous aurez les données sous forme brute avec la valeur à un instant t.



Il est possible d'avoir accès à ces données de manière plus "friendly" à travers un tableau de bord. Pour avoir vos données sous cette forme, il vous suffit de cliquer sur l'onglet "**Data**" puis sur "**Dashboard**" :



Vous devriez alors voir les informations concernant votre ruche sous la forme suivante :



Si les données(Real_temp_INT_SHT & Real_hum_INT_SHT) renvoyées par le SHT donnent une température de 1°C et une humidité de 0% ou si la batterie renvoie 0% veuillez vous référer à la partie "[Comment réparer mon prototype?](#)"

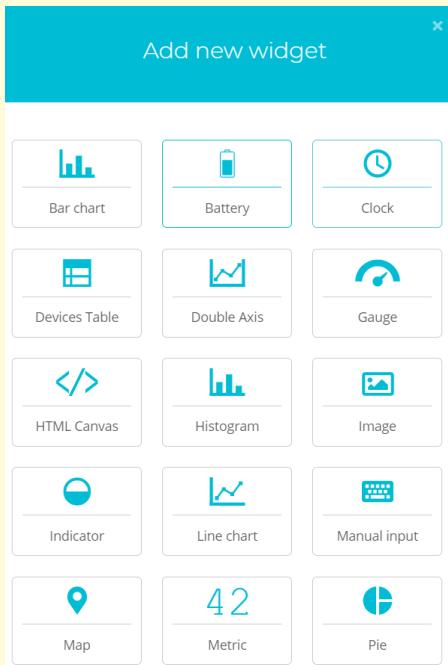
Pour recréer le tableau de bord qu'on vous propose ou créer votre propre interface, veuillez vous référer à la section qui suit (cf. [Créer mon tableau de bord](#)).

Créer mon tableau de bord

Le tableau de bord permet de personnaliser sa page de présentation globale de sa ruche.

Nous allons vous montrer comment procéder pour créer chaque module.

Pour ajouter un module, il faut cliquer sur le signe “plus” en haut à droite de votre écran. Un menu s'affiche contenant différents modules pré-créés pour les utilisateurs:



Il faut choisir un module en fonction de la grandeur que vous voulez afficher sur votre tableau de bord. En sélectionnant un widget, un nouveau menu apparaîtra dans lequel il faudra entrer différents champs, comme par exemple le nom et la variable à associer.

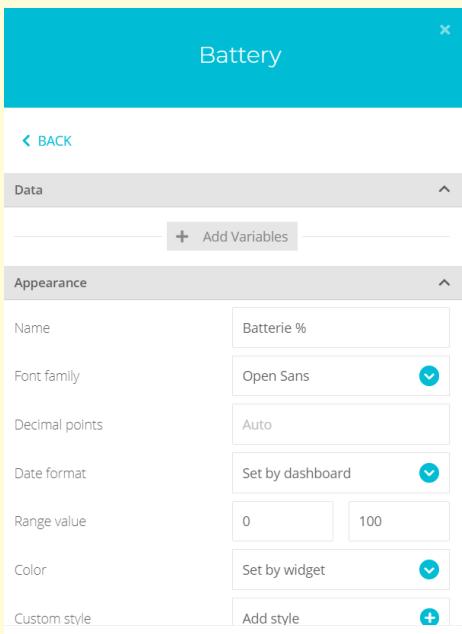
L'interface que nous vous conseillons de faire contient ces différents paramètres:

- Etat de la batterie
- Poids
- Direction du vent, Vitesse du vent
- Stabilité de la ruche
- Température intérieure et extérieure
- Humidité intérieure et extérieure
- L'état de la batterie.

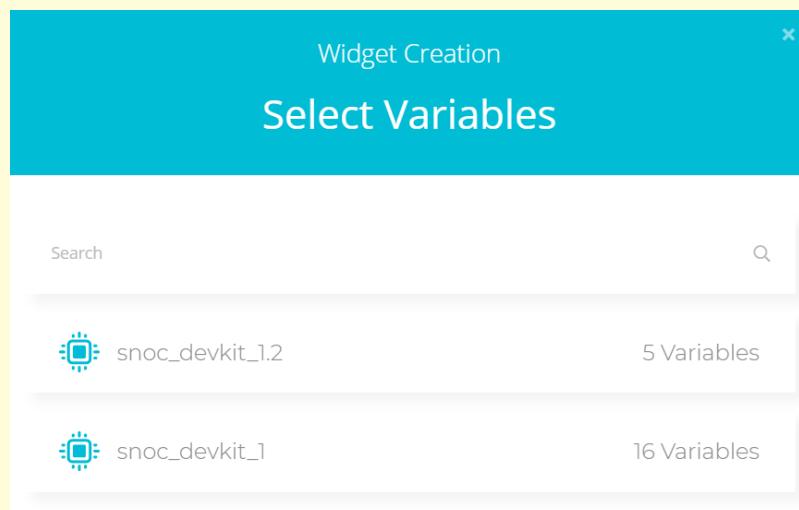
L'exemple que nous faisons est celui décrit ci-dessus.

Cliquez sur le signe “plus” en haut à droite de votre écran pour ajouter un nouveau widget.

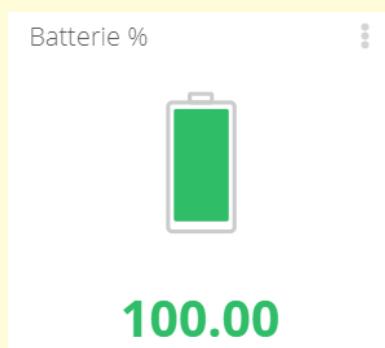
- ❖ Pour la batterie, nous sélectionnons “Battery”:



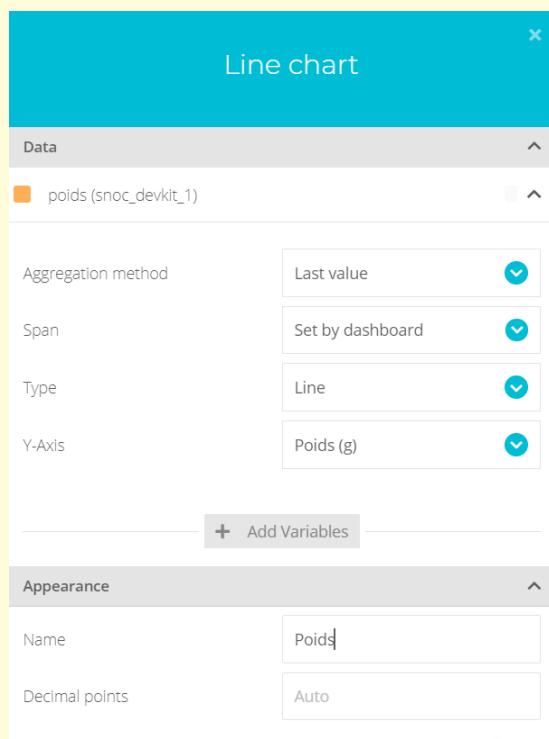
Dans le champ “Name”, choisissez le nom de votre widget. Dans notre cas, ce sera “**Batterie %**”. Il faut maintenant associer la valeur reçue sur la plateforme à ce widget. Pour cela, cliquez sur “**Add Variables**”. Une nouvelle fenêtre s’affiche:



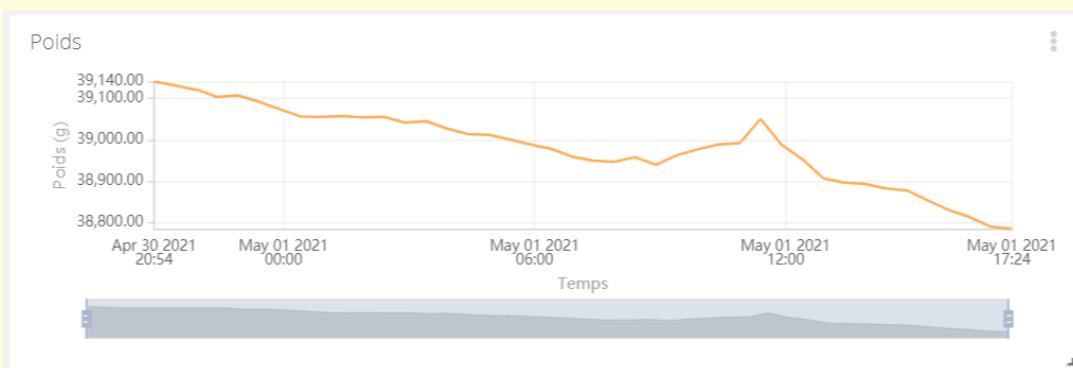
Cliquez sur “**snoc_devkit_1**”, puis sur “batt”. Le widget est maintenant prêt. Il suffira de cliquer sur la flèche verte en bas à droite de l’écran pour valider la création du widget. Voici à quoi devrait ressembler ce widget:



- ❖ Pour le poids, nous sélectionnons “Line chart”.



Pour le nom nous utilisons "Poids". Pour ajouter la variable de poids, cliquez sur "**Add Variables**", puis "**snoc_devkit_1.2**" et finalement "**poids**". Cliquez ensuite sur les flèches vertes pour valider le nouveau widget. Vous devez alors avoir le module qui ressemble à:



- ❖ Pour les températures intérieures, nous sélectionnons "Line Chart".

Nous utiliserons le nom "Températures Intérieures". Il faut ensuite ajouter les deux variables correspondants aux deux capteurs de température. Cliquez sur "Add Variables", puis "**snoc_devkit_1**". Sélectionnez ensuite "**Real_temp_INT_DSB**" et "**Real_temp_INT_SHT**". Cliquez sur les flèches vertes pour valider le nouveau widget qui doit ressembler à :



- ❖ La procédure est la même pour la température extérieure et pour l'humidité intérieure et extérieure, nous résumons ici les différents choix à faire de manière synthétique.

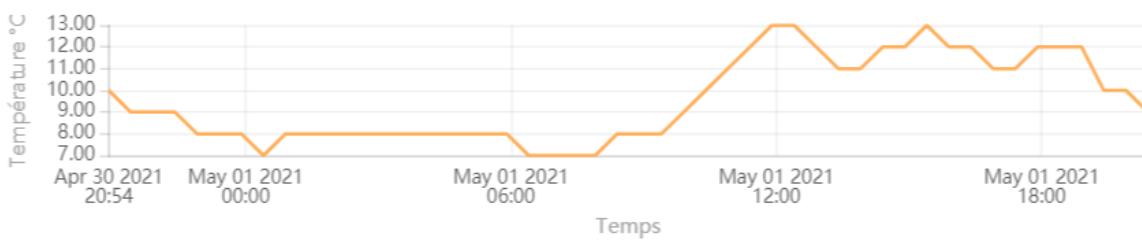
A chaque fois créer un nouveau widget.

-> Pour la température extérieure:

Nom: Température Extérieure

Cliquez sur "Add Variables", puis "**snoc_devkit_1**", et ensuite "**tempext**"

Température Extérieure



-> Pour l'humidité intérieure:

Nom: Humidité Intérieure

Cliquez sur "Add Variables", puis "**snoc_devkit_1**", et ensuite "**Real_hum_INT_SHT**"

Humidité Intérieure

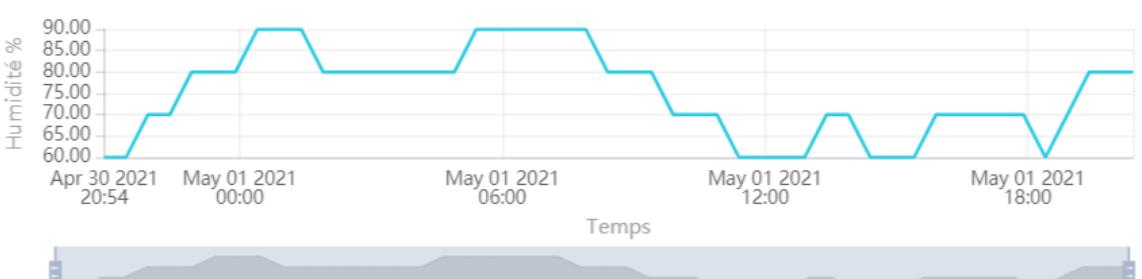


-> Pour l'humidité Extérieure:

Nom: Humidité Extérieure

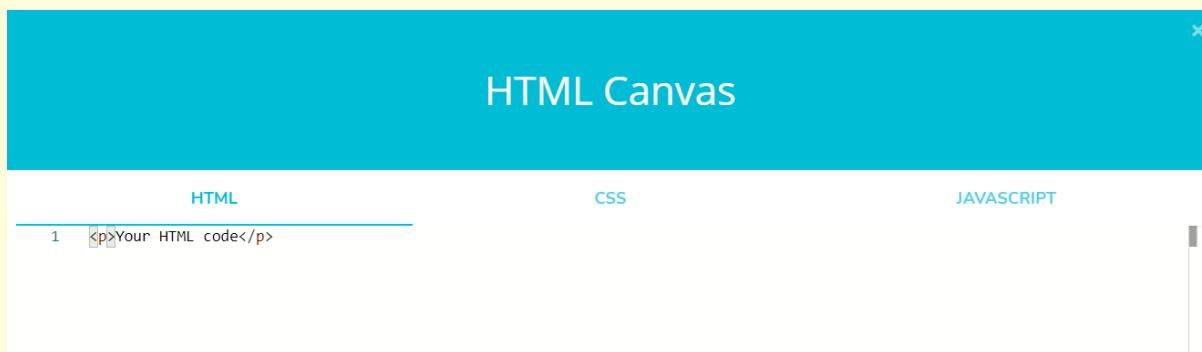
Cliquez sur "Add Variables", puis "**snoc_devkit_1**", et ensuite "**Real_humEXT**"

Humidité Extérieure



- ❖ Pour la vitesse, la direction du vent, et la stabilité de la ruche,, nous sélectionnons “**HTML Canvas**”.

Nous mettons comme nom de module : “**Direction et Force du vent / Stabilité**”
Dans la rubrique “**Body**”, cliquez sur “**Open Editor**”. Une nouvelle fenêtre s'affiche.



Elle contient 3 onglets, “**HTML**”, “**CSS**” et “**JAVASCRIPT**”.

Allez sur le lien suivant et téléchargez le dossier:

https://github.com/Polytech-Sorbonne/El-SE4_2020-2021_OpenRuche_G8/tree/main/HTML_CANVAS

Chaque fichier correspond aux lignes à écrire dans les différents onglets de cette fenêtre.

Pour commencer, nous allons ouvrir le fichier “**html.txt**”. Copiez-Collez le contenu dans le fichier dans l'onglet “**HTML**”. Il faut faire de même avec le fichier “**CSS.txt**”

Le fichier “**JAVASCRIPT.txt**” est un peu plus particulier, il contient des variables qu'il va falloir modifier nous même.

```
var TOKEN = 'ENTRER_TOKEN';
var VARIABLE = 'ID_DIRECTION_DU_VENT'; // direction du vent

var VARIABLE1 = 'ID_ANEMOMETRE'; //force du vent
var VARIABLE2 = 'ID_MOVE'; // Ruche tombée ou pas
```

Pour la variable **TOKEN**, remplacez **ENTRER_TOKEN**, par le token du premier device que nous avions nommé “**Snoc_Devkit_1**”. Pour récupérer le token, il faut aller dans l'onglet “**Devices**” en haut de votre écran, puis sélectionner le dispositif se nommant “**Snoc_Devkit_1**”. Sur l'interface du device, vous verrez une ligne correspondant au Token, comme nous l'avons vu dans la partie Ubidot de la notice.

Pour les trois variables suivantes, il faut remplacer le contenu entre apostrophes par les identifiants des 3 différentes variables, à rappeler, la direction du vent, la force du vent, et la stabilité de la ruche. Pour récupérer l'identifiant d'une variable il faut, toujours dans l'interface du device “**Snoc_Devkit_1**”, sélectionner la variable en question.

The screenshot shows the Snoc DevKit interface. On the left, there's a sidebar with 'snoc_devkit_1' selected. It includes fields for 'Description' (Change description), 'API Label' (snoc_devkit_1), 'ID' (604a434c0ff4c34b8d151fe1), and a 'Token' field which is redacted. Below these are 'Tags' (Add new tag), 'Last activity' (7 minutes ago), and 'Device type' (Set Device Type). On the right, there are six orange cards representing sensor data:

Variable	Value	Last activity
anemo	0.00	9 minutes ago
batt	10.00	9 minutes ago
girou	2.00	9 minutes ago
humext	9.00	9 minutes ago
hum_int_sht	8.00	9 minutes ago
move	0.00	a month ago

Le premier identifiant à entrer correspond à la direction du vent, variable que nous avons nommée “**girou**”. Cliquez sur le carré qui représente cette variable comme sur l’image ci-dessus.

Vous verrez alors une nouvelle fenêtre contenant les différents paramètres de cette variable. Le champ “**ID**”, correspond à la valeur à Copier-Coller dans le fichier “**JavaScript.txt**” qui remplace **ID_DIRECTION_DU_VENT**.

This is a detailed view of the 'girou' variable configuration. It shows the variable name 'girou' at the top. Below it are fields for 'Description' (Change description), 'API Label' (girou), and an 'ID' field which contains the value '604a466a0ff4c3662dbeebdd'. Other sections include 'Allowed range' (From: Min to: Max), 'Unit' (Add unit), and 'Tags' (Add new tag).

La procédure est la même pour la force du vent et la stabilité de la ruche.

Dans le device “**Snoc_Devkit_1**”, la force du vent correspond à la variable “**anemo**” et la stabilité de la ruche correspond à la variable “**move**”. Il faut récupérer les identifiants de ces deux variables et les insérer dans le fichier “**JavaScript.txt**”.

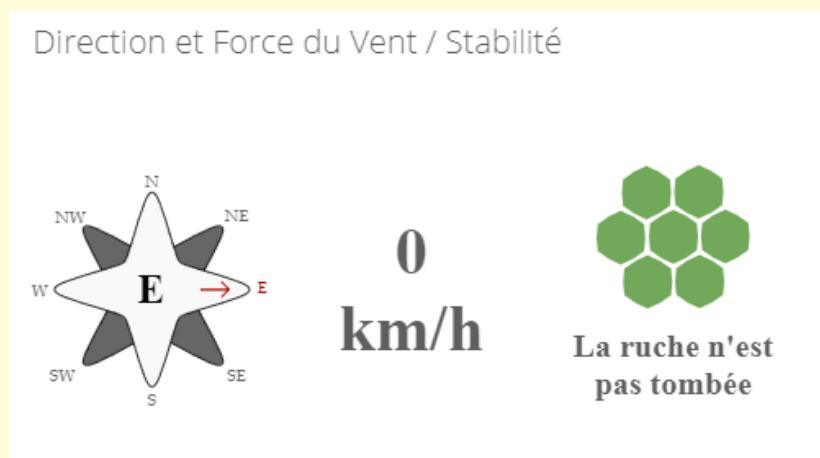
L’identifiant de “**anemo**” est à insérer à la place de **ID_ANEMOMETRE** et l’identifiant de “**move**” à la place de **ID_MOVE**. Il suffit ensuite de copier-coller le contenu de ce fichier dans l’onglet **JAVASCRIPT** de la création du widget.

Une fois les 3 onglets complétés, cliquez sur la flèche verte pour valider le code.

Ensuite, dans la section “**3rd party librairies**”, entrez la ligne suivante:

<https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js>

Vous pouvez maintenant valider votre widget en cliquant à nouveau sur la flèche verte. Votre nouveau widget doit ressembler à:



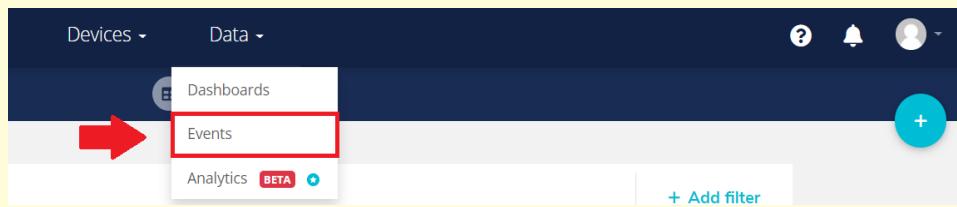
La personnalisation du tableau de bord est donc finie.

Comment créer mes propres alertes ?

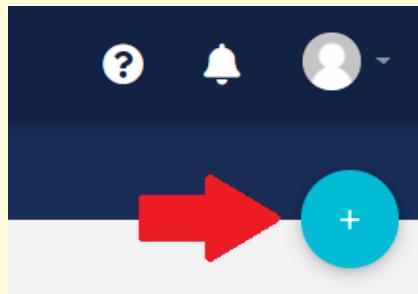
Comme nous savons que chaque colonie d’abeilles est différente et que ses besoins en termes de surveillance ne sont pas les mêmes qu’un autre essaim, nous avons choisi une plateforme qui vous permet de définir vous même vos alertes personnalisées.

Pour créer une alerte, veuillez suivre la procédure suivante :

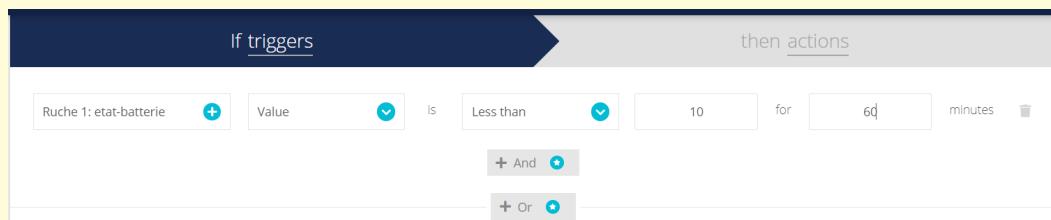
- Connectez-vous avec votre identifiant et votre mot de passe à la plateforme Ubidot.
- Cliquez sur **Data > Events**



→ Cliquez ensuite sur "Create Event"



→ Il suffit ensuite de simplement remplir les champs présentés ici :



- ❖ Le 1er champ concerne la valeur qu'on veut surveiller
- ❖ le 2e champ doit rester sur Value
- ❖ Le 3e champ permet de définir le temps d'action : égal à, plus petit que, supérieur à, etc...
- ❖ Le 4e champ représente la valeur associée à l'action
- ❖ Le dernier champ concerne la durée avant déclenchement

Dans l'exemple qui sert d'illustration, on teste si l'état de la batterie est inférieur à 10 pendant 60 minutes.

Une fois ces paramètres validés on appuie sur la flèche en bas à droite permettant de continuer la création de l'alerte.

→ On arrive ensuite sur une page qui va vous permettre de définir comment souhaitez-vous recevoir l'alerte.

If triggers

then actions

Trigger actions

You may trigger several actions within the same event.

Add new action

- Send Email
- Send SMS
- Send Telegram
- Voice Call
- Set Variable
- Slack
- Trigger WebHook
- UbiFunction

- Si nous sélectionnons une alerte par email par exemple, nous atterrissions sur la page suivante ou nous pouvons définir l'adresse e-mail de la personne recevant l'alerte. On se voit aussi offrir la possibilité de personnaliser le message qui nous sera envoyé.

Email Address

alexandra.deac@etu.sorbonne-universite.fr

Add comma separated emails

ACTIVE TRIGGER

BACK TO NORMAL

Subject

Variable name alert!

Message

Hey there, Variable name was Trigger value at Trigger timestamp.

Repeat action every 60 Minute(s) up to 10 times while trigger is active.

- Une fois ces champs remplis, il suffit de valider à l'aide du bouton suivant :



Et voilà, votre nouvelle alerte personnalisée est créée.

Puis-je y apporter des modifications ?

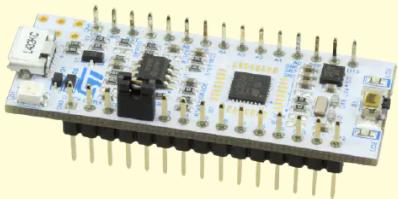
Pour vous offrir une pleine emprise sur le prototype, nous avons mis à votre disposition de possibles modifications que l'on va décrire ici.

Modifications matérielle

Tout d'abord, au niveau matériel, nous vous offrons la possibilité de jouer sur la consommation du système. Le prototype que vous avez entre les mains est actuellement alimenté en 3.3V pour que celui-ci ne consomme pas beaucoup d'énergie et puisse donc durer plus longtemps. Cependant, il est tout à fait possible de faire le choix d'alimenter le système en 5V.

Nous rappelons que le système actuel possède une durée de vie de 1 mois sur une charge de batterie en conditions normales d'utilisation (printemps, été). Si vous décidez de passer en 5V, la durée de vie sur une charge descendra à 1 semaine.

Pour réaliser ces modifications, il va falloir dans un premier temps éteindre le système. On rappelle que la procédure pour éteindre le système est détaillée dans la partie "[Démarrage du prototype](#)". Ensuite, il faut ouvrir la boîte et sortir le système car il va falloir toucher au PCB (*Printed Circuit Board*) et au microcontrôleur.



Microcontrôleur

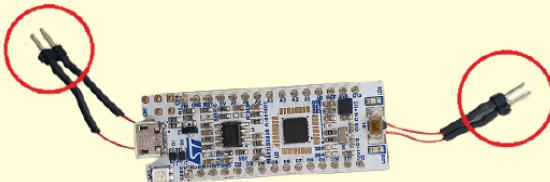


PCB

VEILLEZ À NE PAS DÉBRANCHER LES AUTRES CÂBLES
en cas de problème référez vous à la partie "[Placement des capteurs>Dans le boîtier](#)"

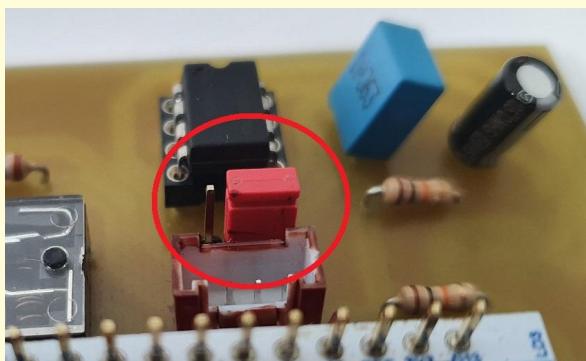
- Passer en 3.3V :

Pour alimenter le système en 3.3V, il va falloir dans un premier retirer les deux jumpers du microcontrôleur comme illustré sur la photo ci-dessous :



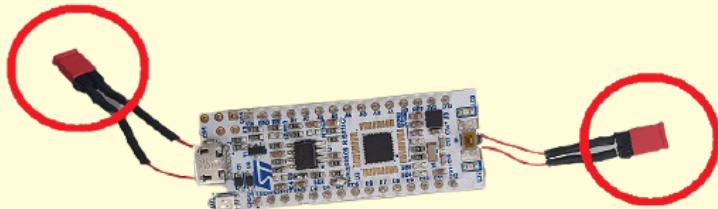
Il faut aussi noter que retirer ces jumpers coupe l'alimentation entre le debugger et le microcontrôleur, ce qui entraîne l'impossibilité de reprogrammer la carte avec un nouveau programme. Si vous avez besoin de reprogrammer la carte, vous devez vous mettre dans la configuration 5V décrite ci-dessous.

De plus, sur le PCB, il va falloir placer le jumper de la manière suivante :

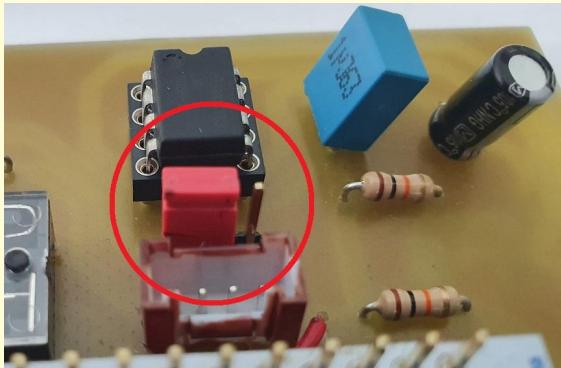


- Passer en 5V :

Pour alimenter le système en 5V, il va falloir, contrairement au 3.3V, placer les deux jumpers du microcontrôleur comme illustré sur la photo ci-dessous :



De plus, cette fois-ci, il va falloir placer le jumper de la manière suivante sur le PCB :



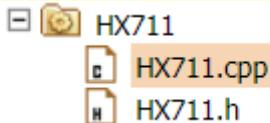
Si vous avez besoin de reprogrammer la carte avec un nouveau programme, vous devez vous mettre dans cette configuration.

Une fois ces étapes réalisées, vous pouvez alimenter votre système comme décrit dans la partie "[Démarrage du prototype](#)", fermer la boîte et réinstaller votre prototype comme décrit dans la partie "[Comment installer mon prototype](#)".

Modifications logicielle

Il est possible de modifier la tare de la jauge de poids pour l'adapter au poids de votre ruche à vide. Par défaut, le programme considère que la ruche pèse 13,2 kilogrammes. Si ce n'est pas le cas pour votre ruche, veuillez suivre les étapes suivantes :

- Ouvrir le code source et se diriger vers le dossier HX711 et ouvrir le fichier HX711.cpp.



- Ensuite, dirigez vous vers la ligne 47 du fichier.

```
47 int poids_ruche_initial = 13200; //En grammes
48 this->setOffset(poids_ruche_initial*609.9979);
```

- Cette ligne correspond à l'initialisation du poids de départ de la ruche, c'est-à-dire le poids de la ruche vide. Pour fixer votre propre valeur, veuillez indiquer après le signe '=' le poids de la ruche en grammes.
- Il faudra ensuite recompiler le code et le glisser dans la carte électronique et veiller à bien remettre les Jumpers.

Comment réparer mon prototype ?

Microcontrôleur

Pour améliorer l'autonomie et les performances du système, certaines modifications ont été apportées au microcontrôleur.



Microcontrôleur

- Tout d'abord, nous avons **retiré les résistances R18 et R19** se trouvant à l'arrière du microcontrôleur. Ces résistances permettent d'alimenter une LED se trouvant sur le microcontrôleur qui consomme de la puissance inutilement.

Pour pouvoir les retirer, il suffit de prendre une pince et de les enlever du microcontrôleur. Il faut néanmoins faire bien attention de retirer que ces deux résistances et pas d'autres composants.

- De plus, nous avons **retiré les solder bridge SB9 et SB14** présents à l'arrière du microcontrôleur. Ces deux Solder Bridge contrôlent le debugger qui permet à un utilisateur de mettre un programme sur le microcontrôleur. Le problème de celui-ci est qu'il consomme énormément et en permanence. Enlever ces solder bridge du microcontrôleur permet donc de supprimer le debugger et d'augmenter donc l'autonomie de notre système.

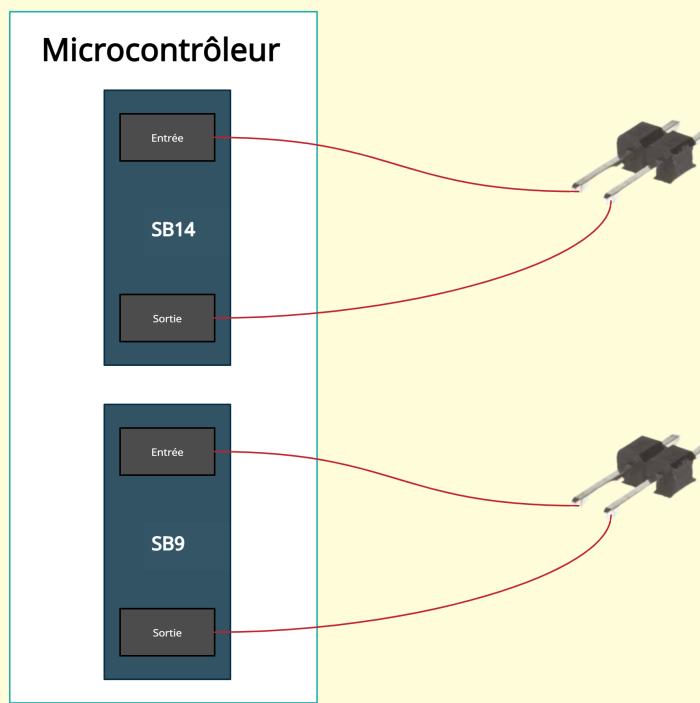
Néanmoins, il peut être utile de pouvoir reprogrammer la carte en cas de problème. Pour cela, nous avons soudé l'entrée et la sortie de chaque solder bridge avec un fil fin et un header male au bout. Cela permet de reconnecter les solder bridge en cas de besoin avec un jumper (fourni dans la boîte dans un sachet en plastique).



Header male



Jumper



Les jumpers devront être mis de l'autre côté des headers mâles dans le cas où on souhaite l'utiliser en 5V. Pour pouvoir utiliser le système en 3.3V et donc augmenter considérablement la durée de vie, il va donc falloir retirer les solder bridge souder les câbles et headers comme indiqué plus haut. Mettez de la colle chaude par-dessus vos soudures pour les fortifier car elles seront fragiles.

On vous a répertorié ici deux problèmes auxquels vous pourrez faire face :

- Si un **problème** intervient avec le **capteur de température et d'humidité SHT21** et qu'il renvoie des données incohérentes, alors il va falloir retirer le Solder Bridge SB18 se trouvant à l'arrière du microcontrôleur. Ce solder bridge relie les pins A4 et D4 du microcontrôleur.

Cependant, comme le microcontrôleur possède peu de pins et que nous avons besoin de toutes, il est nécessaire de casser cette liaison.

Pour cela, comme pour les résistances R18 et R19, munissez vous d'une pince et enlever le solder bridge SB18

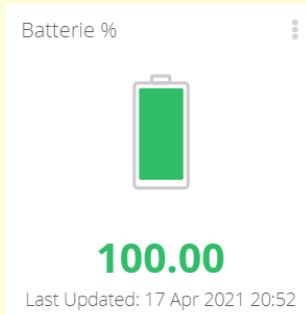
- Si un **problème** intervient lorsque vous trouvez en configuration 5V et que vous branchez la carte à votre ordinateur (fichier “**fail.txt**”), cela signifie qu'il y a sûrement un des câbles reliés à SB9 ou SB14 qui s'est déssoudé. Il faut alors enlever la colle mise pour fortifier et reprendre les soudures comme décrite précédemment.

Batterie

Si lorsque vous allez voir vos données vous observer que l'état de la **batterie indique 0%**, cela signifie que la batterie du prototype est complètement déchargée.

Deux possibilités s'offrent alors à vous. Soit vous changez de batterie en reprenant le même modèle. Soit vous rechargez votre batterie à l'aide du LiPo Rider PRO.

Pour recharger la batterie, il faut vous munir d'un câble mini-usb, de le brancher au Lipo-Rider et brancher l'autre côté en USB à votre ordinateur portable par exemple.



Si d'autres problèmes que ceux cités précédemment interviennent, vous pouvez vous référer au schéma du PCB présent sur le GitHub (https://github.com/Polytech-Sorbonne/EI-SE4_2020-2021_OpenRuche_G8) du projet qui contient tout le schéma électrique du prototype. Sinon, il est aussi possible d'aller voir l'article que nous avons écrit sur notre prototype sur la plateforme Hackster.io (<https://www.hackster.io/380297/connected-beehive-liberate-beekeepers-80869c>).

Enfin, si vous trouvez de réponse nulle part, référez vous à la partie “[En cas de problème](#)” ou nos coordonnées personnelles sont renseignées pour répondre à toutes vos questions.

En cas de problème

Si vous rencontrez une quelconque difficulté dans la mise en place ou l'utilisation de votre prototype, n'hésitez pas à contacter un membre de l'équipe d'HoneyStonks. Nous essaierons au mieux de vous venir en aide dans les plus brefs délais. Voici nos coordonnées:

Contact					
Adresse mail	alexandradeac 44@gmail.com	bennisismail9 @gmail.com	ameliekirady@ gmail.com	shehadehfadi @orange.fr	wyt109376335 4@gmail.com
N° Téléphone	+3365156569 2	+3365081231 0	+3361917715 5	+3376793645 4	+33668109365

Il est aussi bien sur possible de nous contacter sur l'**adresse mail du projet:**

beehave.projet@gmail.com

Annexe

Composant	Quantité	Prix à l'unité	Datasheet	Fournisseur
Microcontrôleur STM32L432KC	1	10,98€	https://www.st.com/resource/en/datasheet/stm32l432kc.pdf	https://fr.farnell.com/stmicroelectronics/nucleo-l432kc/carte-de-developpement-nucleo-32/dp/2580786
Antenne Sigfox BRKWS01	1	23,88€	https://yadom.fr/media/product-attachment//home/magento///File-1571651613.pdf	https://yadom.fr/carte-breakout-sfm10r1.html
Batterie Accu Li-Ion 3,7V 5000 mAh	1	5,08€	https://fr.aliexpress.com/item/33024844443.html?src=google&albch=shopping&acnt=248-630-5778&isdl=y&slnk=&plac=&mtp=&albtt=Google_7_shopping&aff_platfrom=google&aff_short_key=UneMJZVf&albagn=888888&isSmbAutoCall=false&needSmbHouyi=false&albcn=11446264977&albaq=108768739141&trqt=898305777525&crea=fr3302484443&netw=u&device=c&albpq=898305777525&albpd=fr3302484443&aclid=CjwKCjwKCAjwnPOEBhA0EiwA609ReTRdBIOM2pzbqoFkD4pLuW7iMmHT94PfP3G_U-T-8j5QVXYxzVkJ3URoCC3oQAvD_BwE&qcls=aw.ds	https://fr.aliexpress.com/item/33024844443.html?src=google&albch=shopping&acnt=248-630-5778&isdl=y&slnk=&plac=&mtp=&albtt=Google_7_shopping&aff_platfrom=google&aff_short_key=UneMJZVf&albagn=888888&isSmbAutoCall=false&needSmbHouyi=false&albcn=11446264977&albaq=108768739141&trqt=898305777525&crea=fr3302484443&netw=u&device=c&albpq=898305777525&albpd=fr3302484443&aclid=CjwKCjwKCAjwnPOEBhA0EiwA609ReTRdBIOM2pzbqoFkD4pLuW7iMmHT94PfP3G_U-T-8j5QVXYxzVkJ3URoCC3oQAvD_BwE&qcls=aw.ds
Carte LiPo Rider Pro	1	15,60€	https://wiki.seeedstudio.com/Lipo_Rider_Pro/	https://www.gotronic.fr/art-carte-lipo-rider-pro-106990008-19050.htm

Cellule solaire SOL3W	1	19,00€	https://www.gotronic.fr/art-cellule-solaire-sol3w-18996.htm	https://www.gotronic.fr/art-cellule-solaire-sol3w-18996.htm
Capteur de température et d'humidité DHT22	1	10,30€	https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf	https://www.gotronic.fr/art-capteur-d-humidite-et-de-t-grove-101020019-18964.htm
Capteur de température et d'humidité SHT21	1	24,20€	https://github.com/DFRobot/DFRobot_SHT20/raw/master/Sensor-Humidity-SHT20-Datasheet.pdf	https://www.gotronic.fr/art-sonde-d-humidite-et-de-temperature-sen0227-27842.htm
Capteur de température DS18B20	1	7,90€	https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf	https://www.gotronic.fr/art-capteur-de-temperature-grove-101990019-23842.htm
Microphone	1	1,34€	https://fr.aliexpress.com/item/4000986752635.html?spm=a2g0o.placeorder.0.0.4a486ef8ZouRiH&mp=1	https://fr.aliexpress.com/item/4000986752635.html?spm=a2g0o.placeorder.0.0.4a486ef8ZouRiH&mp=1
Jauge de poids HX711	1	10,77€	https://cdn.sparkfun.com/datasheets/Sensors/ForceFlex/hx711_english.pdf	https://urlz.fr/eVty
Accéléromètre MMA8452Q	1	11,95€	http://cdn.sparkfun.com/datasheets/Sensors/Accelerometers/MMA8452Q.pdf	https://www.gotronic.fr/art-accelerometre-3-axes-bob-13926-25824.htm
Anémomètre et Girouette LEXCA007	1	28,00€	https://www.lextronic.fr/ensemble-capteurs-girouette-anemometre-57683.html	https://www.lextronic.fr/ensemble-capteurs-girouette-anemometre-57683.html
Interrupteur	1	6,40€	https://docs.rs-online.com/a689/0900766b8141f8a3.pdf	https://fr.rs-online.com/web/p/interrupteurs-a-levier/1900642/?cm_mmc=FR-PLA-DS3A-_-google-_-PLA_FR_FR_InterrupteursWhoop-_(FR:Whoop)+Interrupteurs+%C3%A0+levier-_1900642&matchtype=&pla

				323865561782&gclid=Cj0KCQjwyN-DBhCDARIsAFOELTmwVJGblqBESBAp2LqiMwnkGzMPzzLChhXjATodYU527QrqtDSttQaAmprEALw_wcB&gclsrc=aw.ds
Régulateur de tension LDO - TS5205CX533	1	0,62€	https://www.mouser.fr/ProductDetail/Taiwan-Semiconductor/TS5205CX533-RFG?qs=QEzMIVqgwc15FQoplXgHbw==	https://www.mouser.fr/ProductDetail/Taiwan-Semiconductor/TS5205CX533-RFG?qs=QEzMIVqgwc15FQoplXgHbw==
LM386	1	1,10€	https://www.ti.com/lit/ds/symlink/lm386.pdf?ts=1613363268166&ref_url=https%253A%252F%252Fwww.google.com%252F	https://www.digikey.fr/fr/product-highlight/t/texas-instruments/lm386-low-voltage-audio-power-amplifiers
LED	1	0,24€	https://docs.rs-online.com/eb75/0900766b8138289d.pdf	https://fr.rs-online.com/web/p/led/2284985/?cm_mmc=FR-PLADS3A--google--CSS_FR_FR_Afficheurs_et_opto%C3%A9lectronique_Whoop_-(FR:Whoop!)+LED--2284985&matchtype=&aud-827186183886:pla-300807720454&gclid=Cj0KCQjwyN-DBhCDARIsAFOELTmtXV0had-Y13gHxK98K5kYnsP5AKm4rCfcovH4cyveQ9taMGOtsxUaAoxdEALw_wcB&gclsrc=aw.ds
Condensateurs 10 uf	1	0,40€	https://www.mouser.fr/datasheet/2/293/euvz-1219460.pdf	https://www.mouser.fr/ProductDetail/Nichicon/UVZ1H100MDD1TD?qs=ZMpy%252BEepF5Nor746e8tj0Q%3D%3D&mgh=1&vip=1&gclid=Cj0KCQjwyN-DBhCDARIsAFOELTmwI3F4i4KjlgT8x7

				<u>oJ20J1JJt5m51iVmzQ-1TlIsSBLiw2nwSPfoUaAhEaEALw_wcB</u>
Condensateurs 1uf	2	0,50€	<u>https://docs.rs-online.com/238b/0900766b800274c2.pdf</u>	<u>https://fr.rs-online.com/web/p/condensateurs-electrolytiques/2287142/?cm_mmc=FR-PLA-DS3A--google-PLA_FR_FR_Passifs_Whoop--(FR:Whoop)+Condensateurs+%C3%A9lectrolytiques--2287142&matchtype=&pla-338854523223&gclid=Cj0KCQjwyN-DBhCDARlsAFOELTkoRD2WodXwLtNi4lEP865Be6jYxfmm6632pkYNJvZ_R04nkWZKqUaAgiBEALw_wcB&gclsrc=aw.ds</u>
Condensateurs 2.2uf	1	0,51€	<u>https://www.mouser.fr/datasheet/2/315/ABAA0000C1035-1316563.pdf</u>	<u>https://www.mouser.fr/ProductDetail/Panasonic/EEU-HD1H2R2B?qs=RSvpvxwo2RNq7NZk11U90w%3D%D3D&mgh=1&vip=1</u>
Condensateurs 470p	1	1,17€	<u>http://www.farnell.com/datasheets/2722825.pdf</u>	<u>https://fr.farnell.com/murata/rde5c1h471j0m1h03a/condensateur-470pf-50v-mlcc-radial/dp/2990775?gclid=Cj0KCQjwyN-DBhCDARlsAFOELThTO57p84JkYYu05yx5UfdxH6M66cCu6wziGLQP69KKQbqZS8AzNSwaaPhJEALw_wcB&mc_kv=s0kb9bpex_dclpcrid459810025845 plid kword match slid product 2990775 partid 107317281773 partid 127173559982:pla-300800139254 &CMP=KNC-GFR-SHOPPING-SMEC-01-M</u>

				<u>ar-21_Desk-Hi&gross</u> <u>_price=true</u>
Résistance 18k	1	0,45€	http://www.farnell.com/datasheets/2860633.pdf	https://fr.farnell.com/multicomp/mf25-18k/resistance-18k-0-25w-1/dp/9341447?gclid=Cj0KCQjwyN-DBhCDARIsAFOELTnQXKe3hX7HO5r8ccDdtLGwUl0c3UP0H-1OW8siST4YRFzp9L164QaAjKfEALw_wcB&mckv=somrT8qtH_dclpcrid459810025839lpid kword match slid product 9341447 product 107317281573 product 127173559982:pla-340957824222 &CMP=KNC-GFR-SHOPPING-SMEC-01-Mar-21_Desk-Hi&gross_price=true
Résistance 68k	1	0,10€	https://www.mouser.fr/datasheet/2/447/Yaqeo_LRCFR_1-1670055.pdf	https://www.mouser.fr/ProductDetail/Yaqeo/CFR-12JT-52-68K?qs=LdthLRB3J3Qy%2FjPi4rnrNq%3D%3D&mgh=1&vip=1&qclid=Cj0KCQjwyN-DBhCDARIsAFOELTnHcQ8FWREoTAU7q1Up7qkm0mCd7klKV9l9KsU68Abn2rWzt-XBzxMaAophEALw_wcB
Résistance 10k	2	0,42€	http://www.farnell.com/datasheets/2860633.pdf	https://fr.farnell.com/multicomp/mf25-10k/resistance-10k-0-25w-1/dp/9341110?gclid=Cj0KCQjwyN-DBhCDARIsAFOELTnXbaCRTW8968s0RshTyVEqGDKpWF9cPhA5yw4bb4GIDH0K982Dhp8aALA-EALw_wcB&mckv=se24jYnvP_dclpcrid459805093413 plid kword match

				<u>lslid product 9341110 pgrid 108011006136 ptaid aud-127173564062:pla-341596380665 &CMP=KNC-GFR-SHOPPING-SMEC-01-Mar-21_Desk-Hi&gross_price=true</u>
TOTAL	25	180,91€		

Annexe 1 : Liste exhaustive de tous les composant du prototype renseignant le nombre d'unité, le prix, la fiche technique et le fournisseur de chaque composant