Ecole Publique d'Ingénieurs en 3 ans

Projet I am ENSICAEN

DOCUMENTATION DE LA GATEWAY THE THINGS NETWORK

le 26 septembre 2022,
version 1.1

Lucille ANTOINE,
3A SATE
lucille.antoine@ecole.ensicaen.fr

Tuteur école : **Hugo Descoubes / Philippe Lefebvre / Dimitri Boudier**

ENSI CAEN
ÉCOLE PUBLIQUE D'INGÉNIEURS
CENTRE DE RECHERCHE

www.ensicaen.fr

TABLE DES MATIERES

TABLE DES FIGURES

MISE EN ROUTE DE LA GATEWAY

1. Contexte

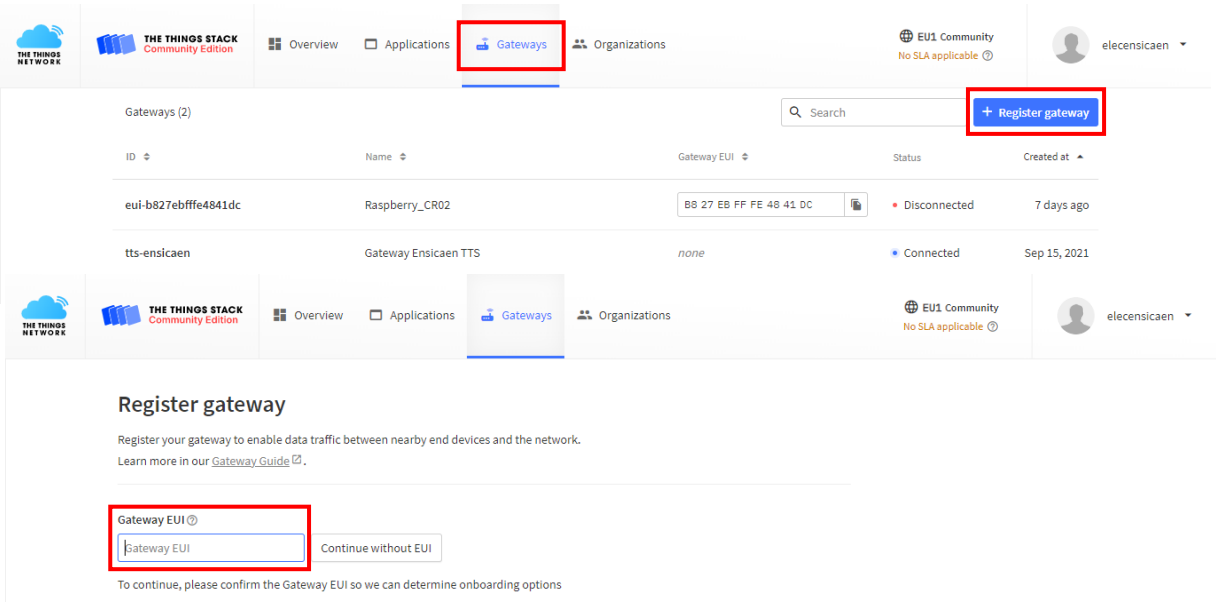
Lors du projet immersif de cette année 2022, qui vise à équiper les bâtiments de l'Ensi de capteurs pour faire un diagnostic de ceux-ci, nous avons décidé d'utiliser la gateway The Things Network (TTN) https://www.thethingsnetwork.org/docs/gateways/gateway/, en parallèle de développer une gateway maison, pour faire le lien entre les capteurs et l'application. Le but d'utiliser cette gateway, plutôt que celle maison, est qu'elle est déjà stable car elle est disponible sur le marché. Nous pouvons alors communiquer via LoraWAN, ce qui n'est pas encore possible à ce jour sur la gateway développée lors de ce projet par d'autres étudiants. La fréquence du LoraWAN en Europe est de 868MHz. Nous travaillerons donc autour de celle-ci. Cette gateway a déjà été utilisée par Philippe Lefebvre sur d'autres projets par le passé, le but étant de l'utiliser dans notre projet.

2. Configuration de la gateway

Pour se connecter à TTN https://eu1.cloud.thethings.network/console/, voici les identifiants :

- Log in : philippe.lefebvre@ensicaen.fr
- Mot de passe : Jastkemt0

Pour configurer la gateway, il faut d'abord l'ajouter sur TTN, en cliquant sur « Register gateway », puis en saisissant l'identifiant de la gateway dans « Gateway EUI » ou non. A voir avec Philippe Lefebvre pour la configuration, il l'a réalisé avant que nous reprenions le projet.



Il faut que cette gateway soit connectée au réseau avec des requêtes spécifiques pour pouvoir y accéder. Voici celle qu'on a utilisé pour demander à la DSI les autorisations:

- Machine d'adresse MAC: 54:10:ec:41:28:55
- Prise 1E11 bureau A108 (la box devrait être placée dans le bureau A202 à terme)
- La machine doit pouvoir faire :
    - HTTPS (443) sortant
    - ping sur 8.8.4.4
    - NTP (123) sur pool.ntp.org
    - MQTTS sur eu1.cloud.thethings.network sur le port 8881

Nous avons communiqué ces informations à la DSI pour qu'ils puissent nous ouvrir les ports et obtenir les autorisations nécessaires. Théoriquement l'adresse mac de cette gateway a été fixée dans le VLAN 214 (réseau étudiant) de l'école pour pouvoir faire les configurations nécessaires. Nous avons principalement interagi avec Laurent Cousin à la DSI.

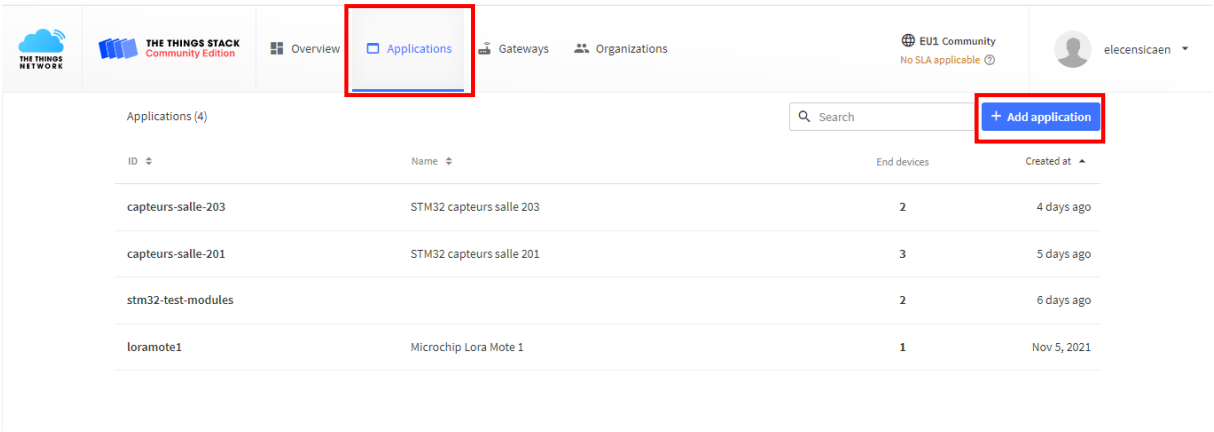
3. Connexion de la gateway avec le module de démo

Il existe un module de démonstration. Une application sur TTN est déjà créée pour l'utiliser sous le nom loramote1. Elle a été développée par Philippe Lefebvre. Il a une trace de la configuration et de la trame à faire pour connecter ce module à la gateway. Contacter Philippe Lefebvre au besoin pour essayer de le rejouer.

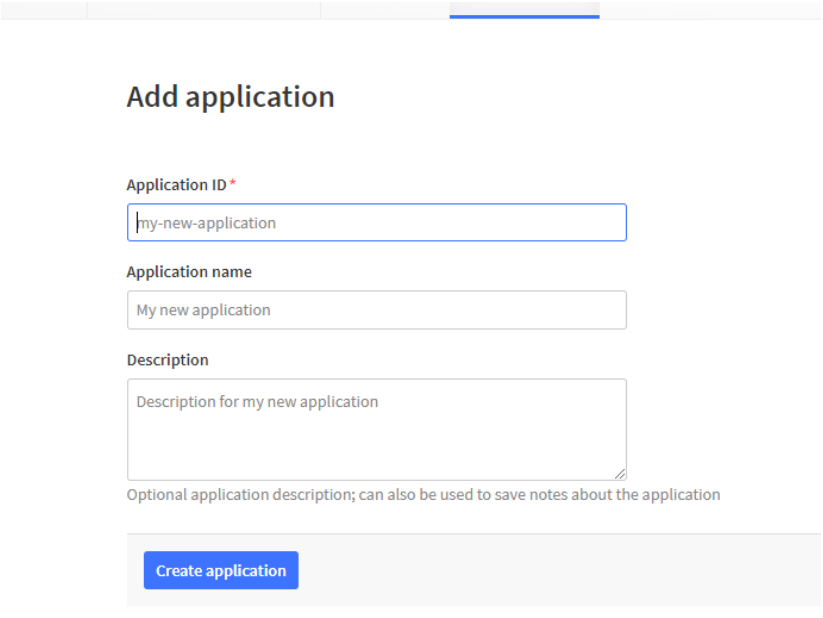
UTILISATION DE LA GATEWAY

4. Construction des applications sur The Things Network

Pour construire une nouvelle application il faut aller dans l'onglet application de TTN et ajouter une nouvelle application en cliquant sur le bouton « *Add application* ». Une application est un espace dans lequel on peut coupler différent devices et leur définir un comportement similaire.

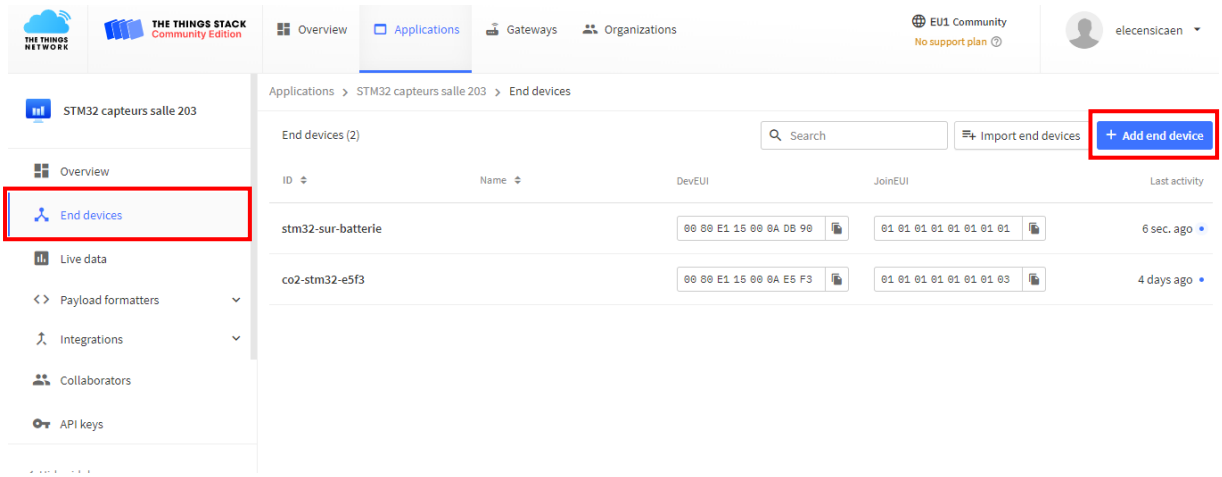


Il faut ensuite donner un ID à l'application dans le champ « *Application ID* ». Il ne doit contenir que des minuscules, chiffres, ou le caractère « - ». C'est facultatif mais conseillé de donner un nom à l'application.

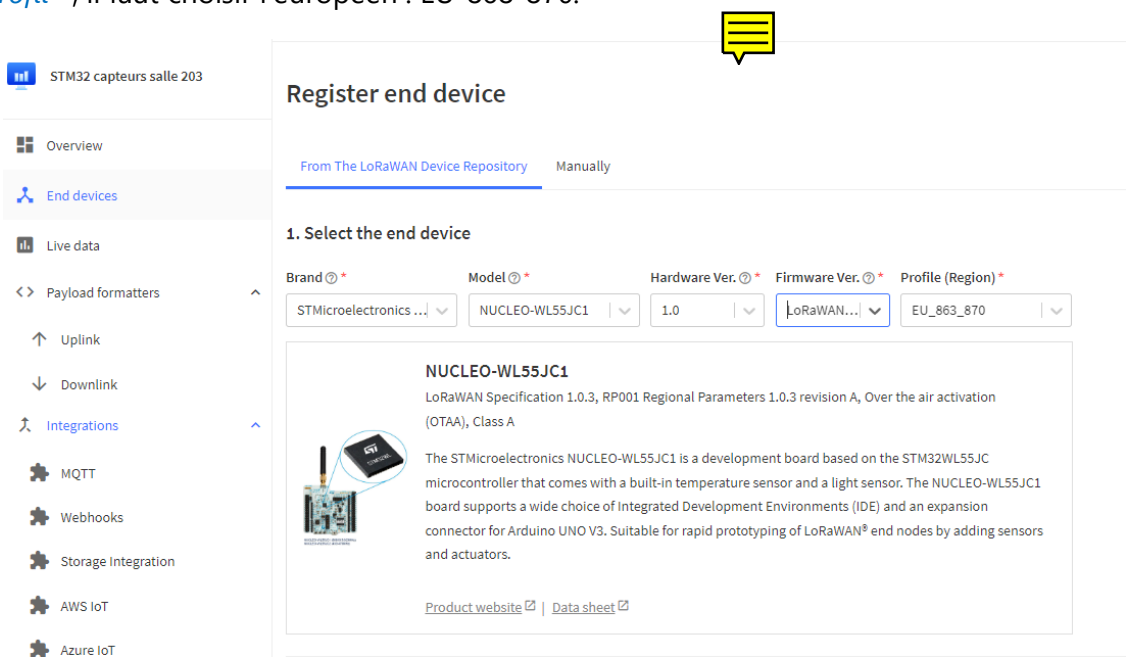


Nous avons choisi d'utiliser une nomenclature particulière pour les IDs : « *capteurs-salle-numéro* » avec le numéro de la salle à définir en fonction de la salle pour laquelle on veut traiter les donner des capteurs via cette application.

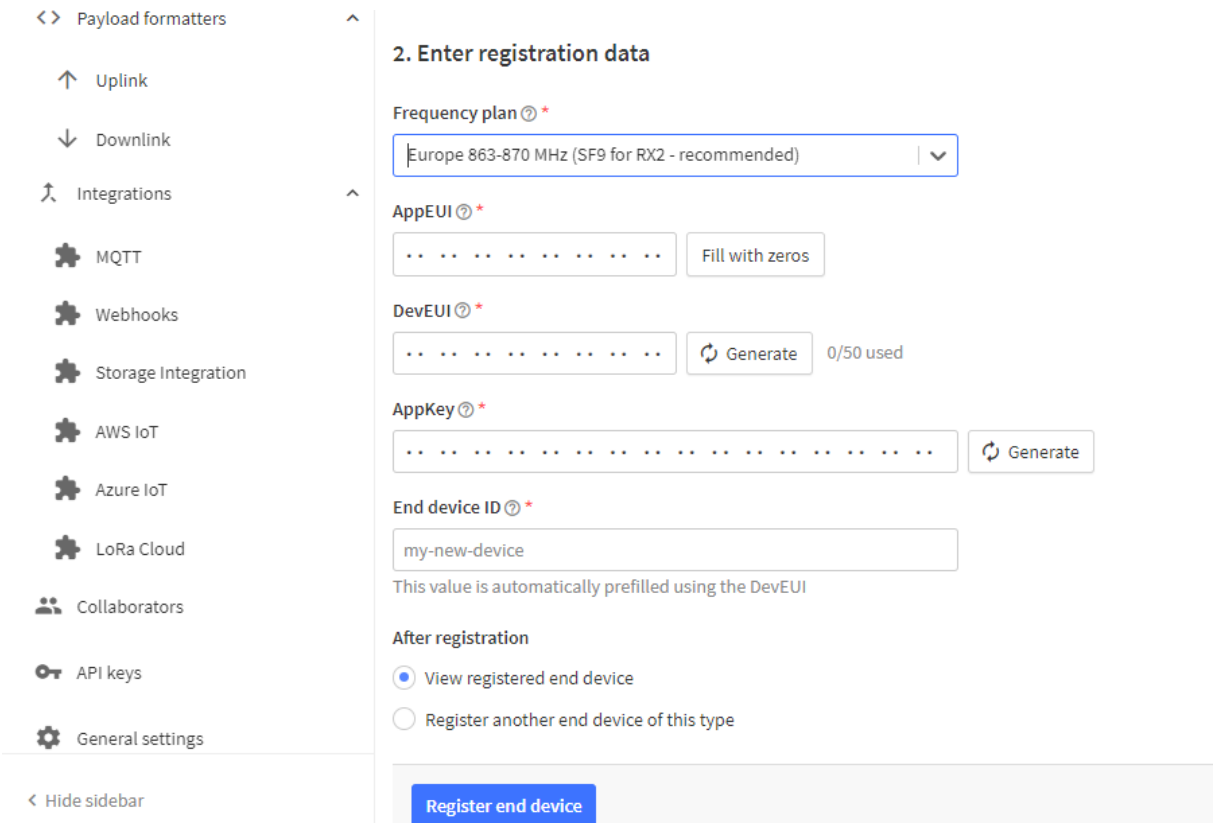
Une fois l'application créée, il faut y ajouter des devices. Un device correspond dans notre cas à une carte STM32 avec son module. Pour ajouter le device il faut aller dans la partie « *End devices* » puis cliquer sur « *Add end device* ».



Il faut ensuite chercher la technologie utilisée : pour nous le NUCLEO-WL55JC1 de chez STMicroelectronics. Nous n'avons pas encore établi avec certitude quel « *Firmware Version* » nous devions utiliser. Mais nous avons testé à la fois la 1.1.0 et la 1.2.0. Pour ce qui concerne le « *Profil* », il faut choisir l'européen : EU-868-870.



Pour le « *Frequency plan* », prendre celui recommandé fonctionne très bien et il correspond à la norme européenne de 868MHz. Il faut ensuite renseigner les EUI et keys correspondant à la carte que nous souhaitons coupler. L'« *AppEUI* » (ou « *JoinEUI* » en fonction de la version du firmware) doit être le même que celui configuré sur STM32 Cube IDE, à voir sur la Figure 1. Le « *DevEUI* » est indiqué sur le STM32 comme sur la zone encadrée sur la Figure 2.



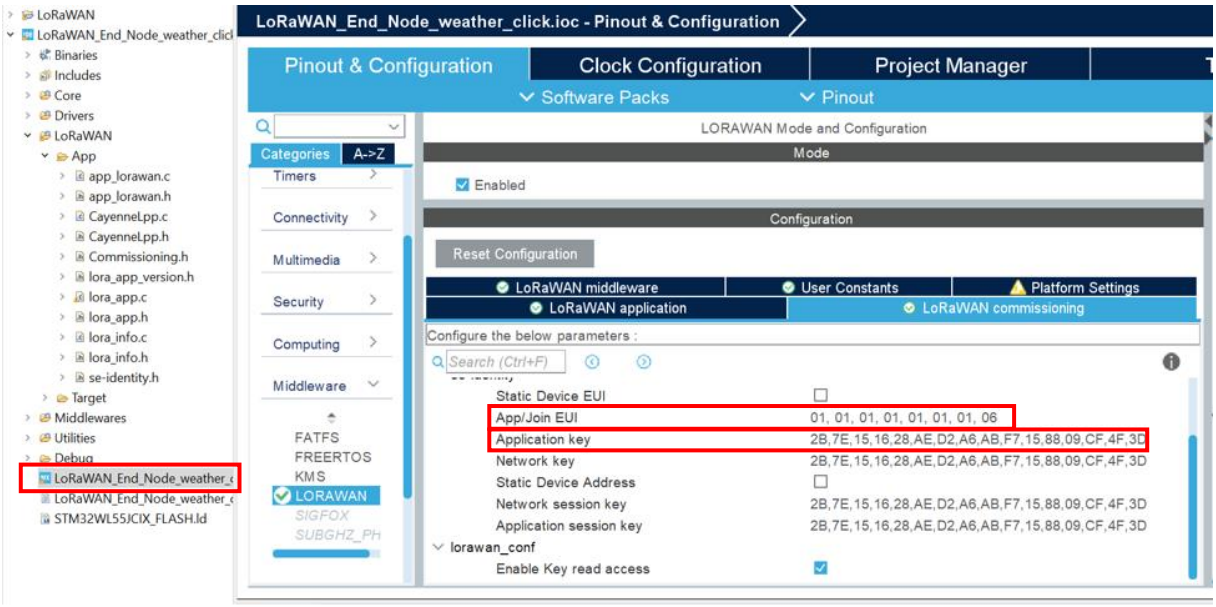


Figure 1 : Localisation de l'App/JoinEUI et de l'AppKey

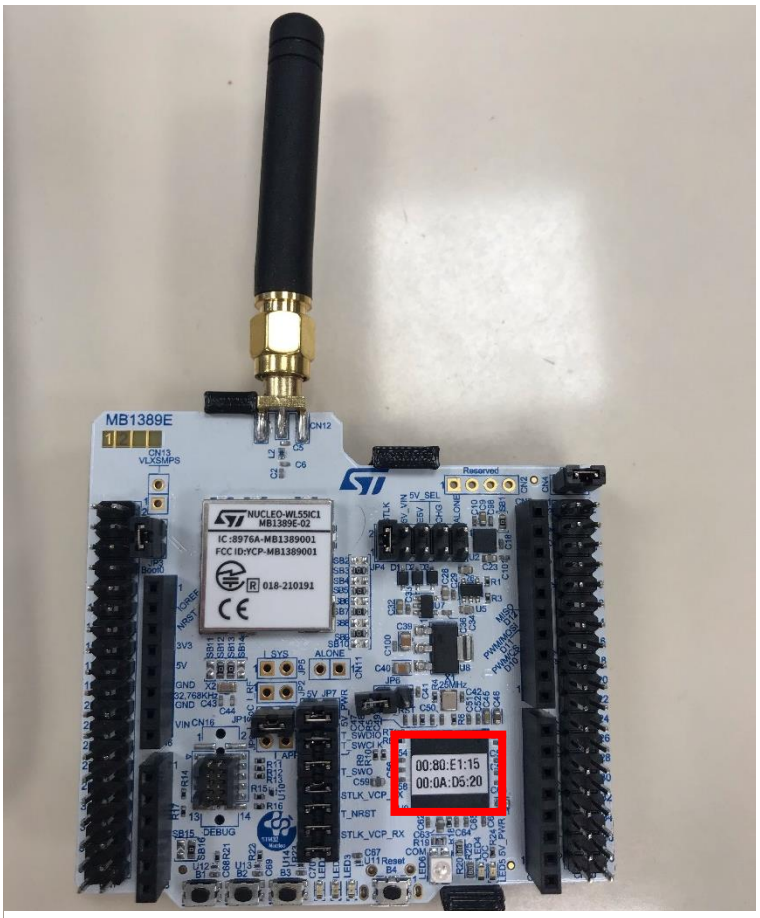


Figure 2 : STM32-localisation du DevEUI

Nous avons fait face à quelques problèmes et interrogations. Tout d'abord le temps de reconnexion du device, après un reset ou déconnexion de celui-ci, était de plus en plus long à chaque fois. Il y avait des messages d'erreur tels que « *Device Nounce too small* » ou « *Device Nounce has already been used* », dans la partie « *Live Data* » qui permet de voir les flux de données sur l'application. Il fallait un certain délai, entre 1 et 2 minutes pour que le capteur se reconnecte. Ceci est visible sur la Figure 3.

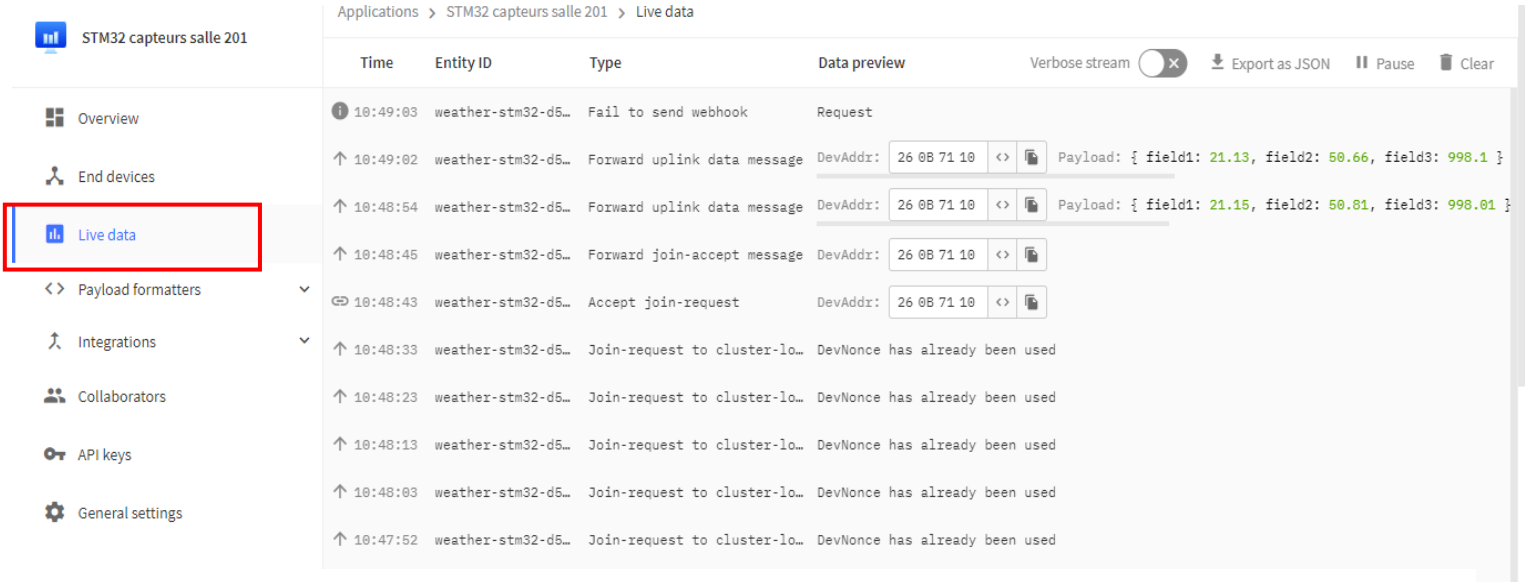


Figure 3 : Mise en évidence du délai de reconnexion d'un device

Nous avons d'abord essayé en ayant le même AppEUI et AppKey pour tous nos devices, mais pour essayer d'améliorer les temps de connexion nous avons décidé d'en créer un différent pour chaque. Toutefois le problème ne semble pas résolu. Le temps est un peu plus court mais il y a toujours plusieurs échecs avant la connexion et la raison nous est encore inconnue. Nous ne savons donc pas s'il y a un intérêt à changer les AppEUI et AppKey, cela reste à investiguer.

Finalement la solution trouvée qui semble être efficace est de changer la macro LORAWAN_FORCE_REJOIN_AT_BOOT de false à true dans le fichier lora_app.h présenté en Figure 4. Le join otaa se fait donc très rapidement, en moins de 30sec après le reset de la carte.

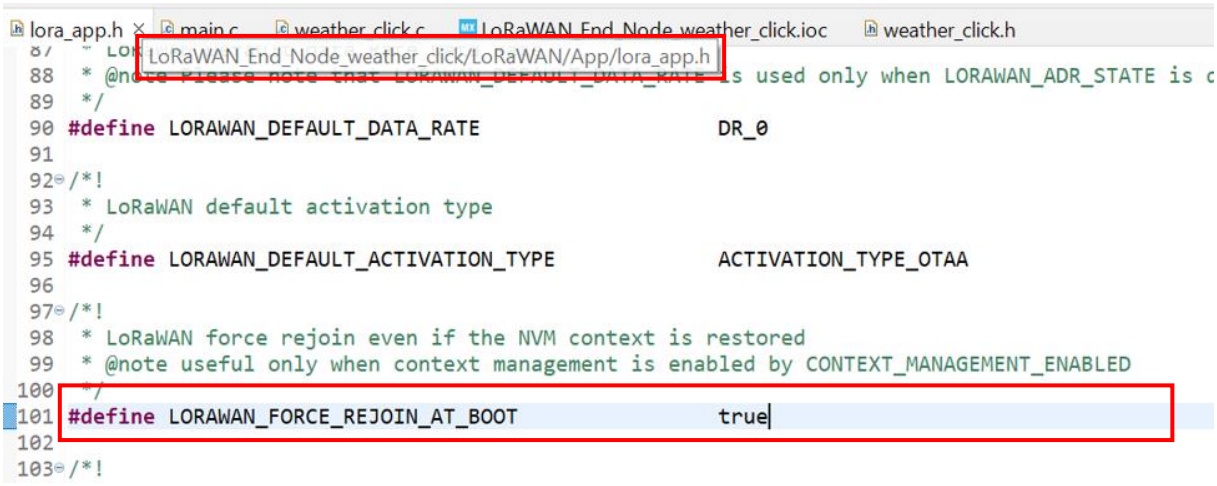


Figure 4 : Mise en évidence de la macro à modifier

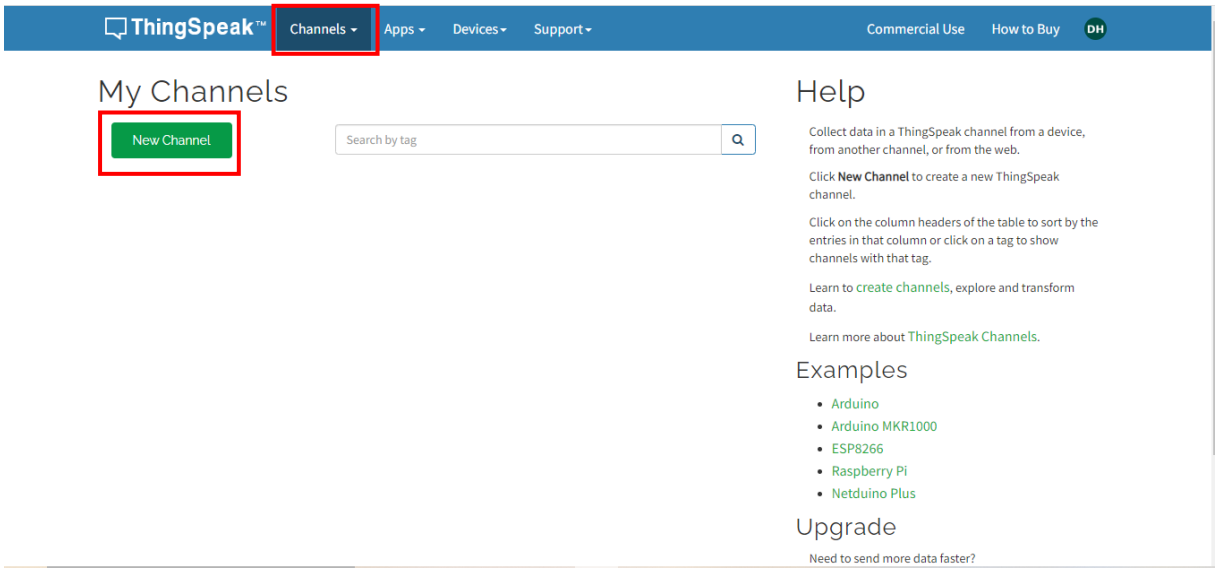
5. Connexion aux canaux de ThingSpeak

Une fois nos devices connectés, nous avons choisi de les relier à ThingSpeak : https://thingspeak.com/channels. Grâce à ce webhook, nous pouvons visualiser rapidement nos données depuis un ordinateur, et également les récupérer via des requêtes http pour collecter les données et ensuite les afficher dans l'application développée par d'autres étudiants. Cet outil fonctionne avec des canaux, qui possèdent chacun une clé d'écriture et une clé de lecture. On peut créer jusqu'à 8 champs dans chaque canal. Pour nous un champ correspond à une variable venant d'un capteur : par exemple la température. De plus cette solution est proposée par MathWorks. Nous avons donc des comptes disponibles avec la licence de l'école. Toutefois un compte étudiant ne permet de créer que 10 canaux maximum. Néanmoins un compte académique permet de créer 250 canaux, d'où le fait que nous avons utilisé le compte de Hugo Descoubes pour faire nos canaux.

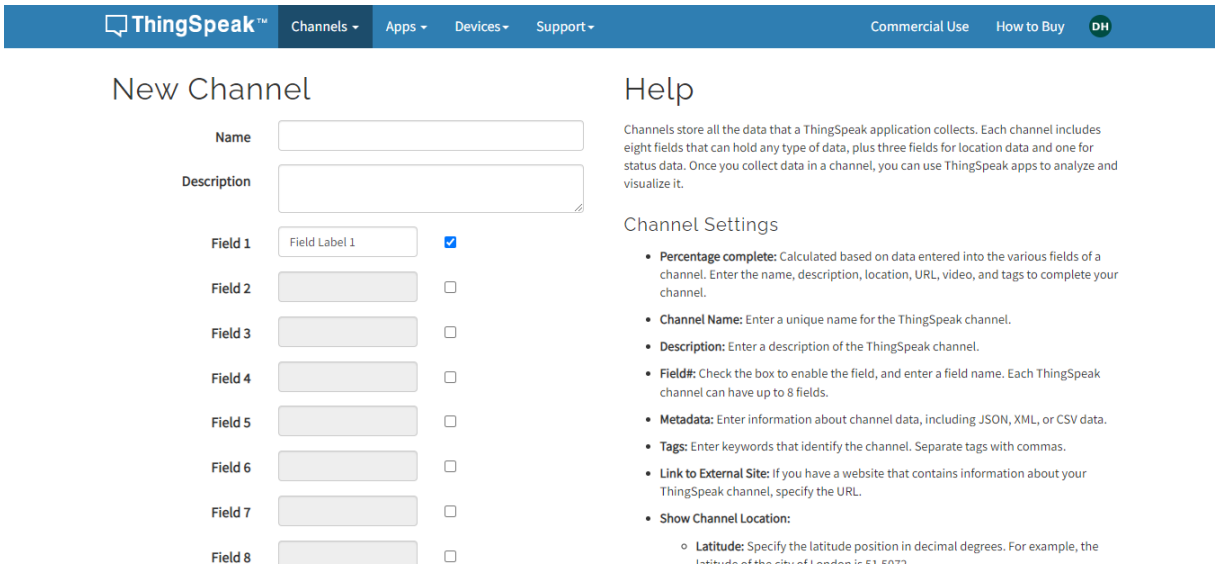
- Identifiants de connexion : hugo.descoubes@ensicaen.fr
- Mot de passe : projetIMMERSIF14

Nous devons également aller voir la DSI afin de savoir s'il est possible d'avoir un compte académique dédié au projet immersif.

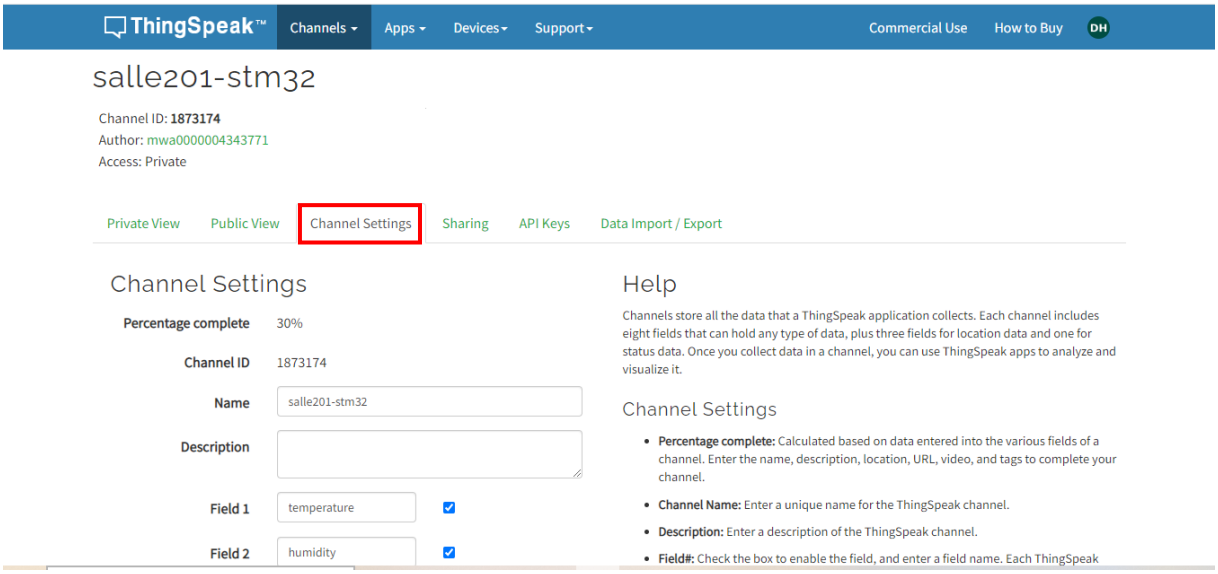
Une fois connecté à ThingSpeak, il faut créer un nouveau canal et le paramétrer. Pour cela il faut cliquer que « *New Channel* » dans l'onglet « *Channel → My Channels* ».



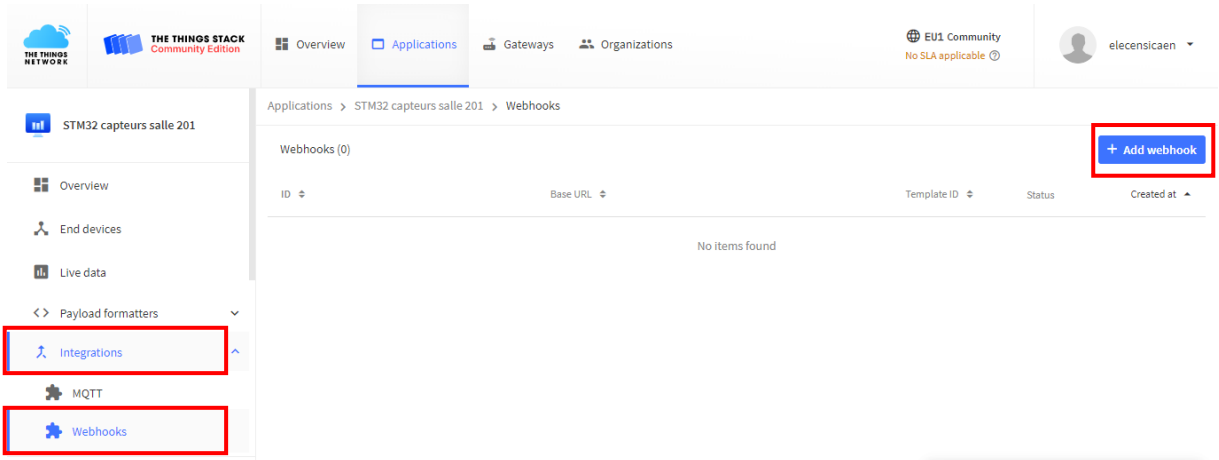
Il faut ensuite donner un nom à ce canal et débloqué autant de fields que souhaité en les cochant. Il faut renommer chaque field avec le nom de la data qu'on veut visualiser dans ce field.



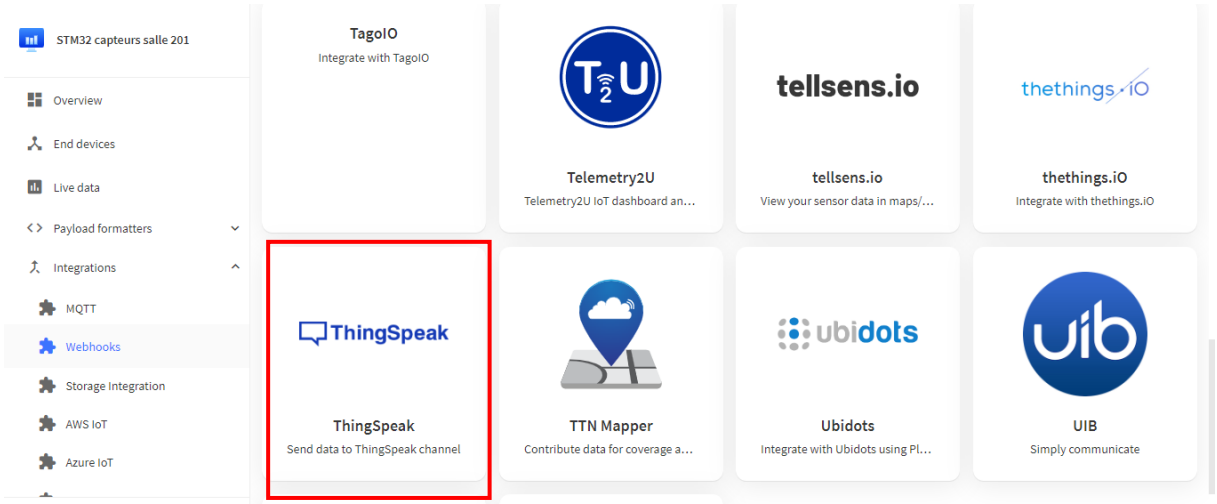
On peut aussi modifier et rajouter les field dans une limite de 8 fields. Il faut donc construire une application par salle, tant qu'il y a moins de 8 capteurs par salle. S'il y a plus de 8 capteurs, il faudra alors créer plusieurs applications pour la même salle. Pour modifier les fields il suffit d'aller dans les settings du canal.



Une fois le canal créé et paramétré à nos souhaits, il faut le relier à TTN. Pour ce faire il faut aller dans « *Integrations* », puis « *Webhook* » et ajouter le webhook via le bouton « *Add webhook* ».



Il faut ensuite faire défiler jusqu'à avoir le logo ThingSpeak.



Pour connceter le canal, il faut lui donner un identifiant interne à TTN, dans « *Webhook ID* ». Pour l'identifiant du cannal, il faut aller le relever dans ThingSpeak. C'est egalement le cas pour l'« *API Keys* », il faut y rentrer la « *Write API Key* » de ThingSpeak. Cf Figure 5.

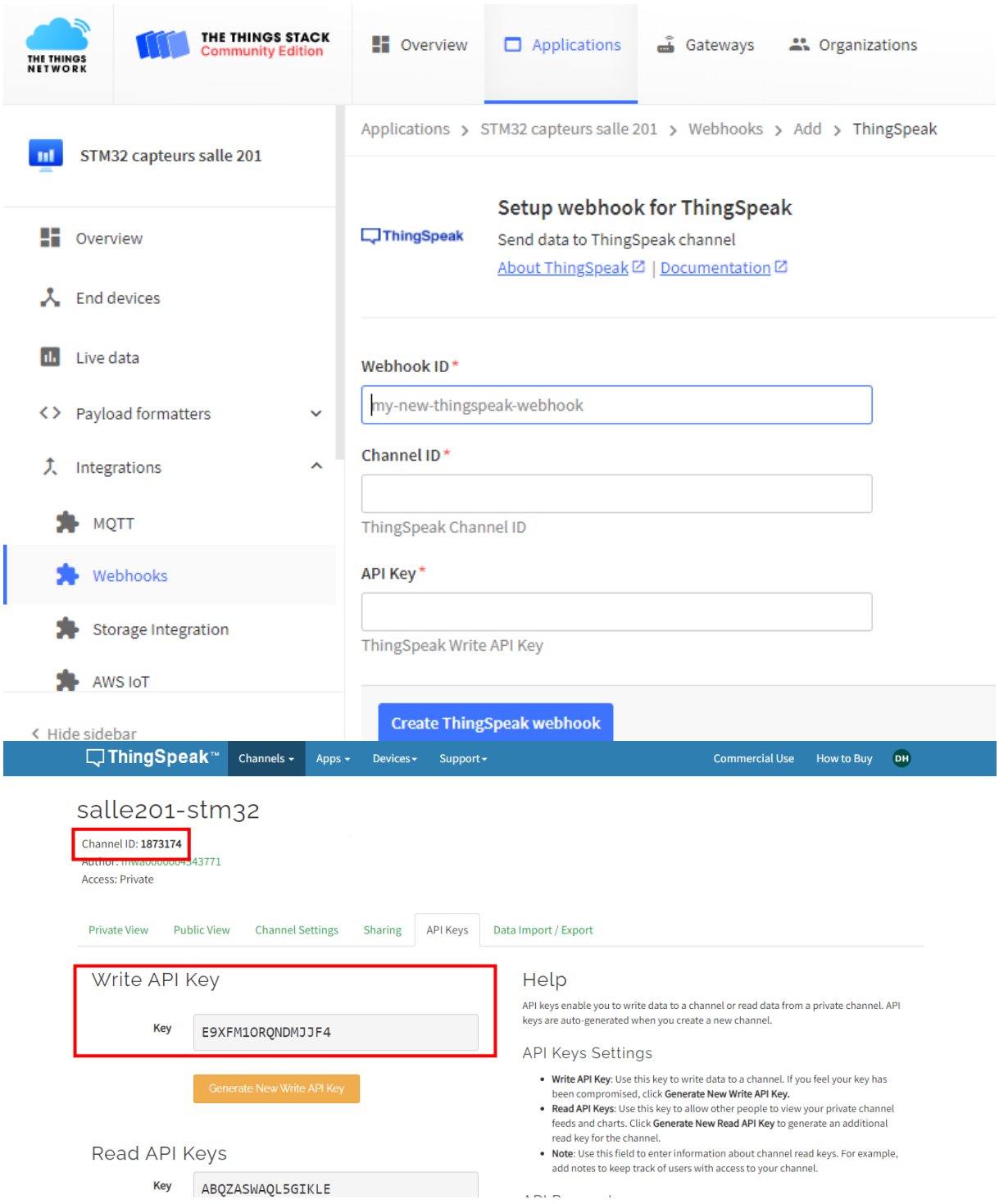
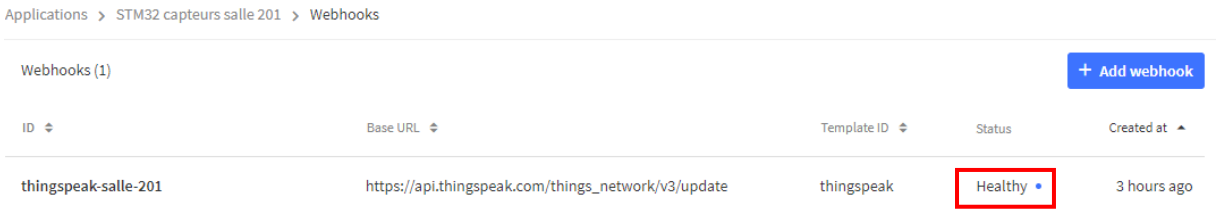
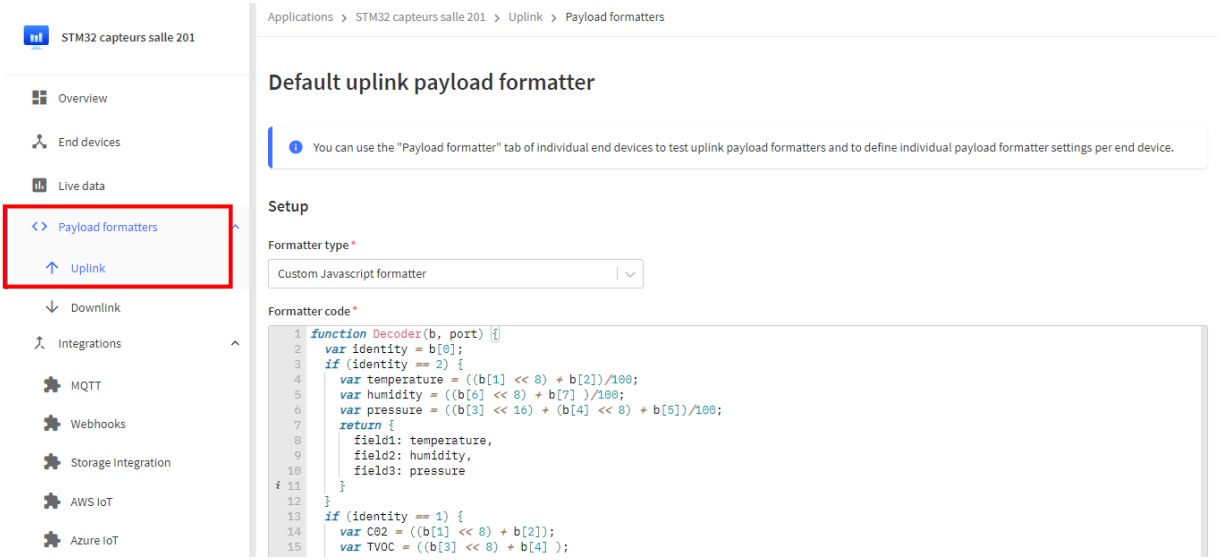


Figure 5 : Channel ID et API Key de ThingSpeak

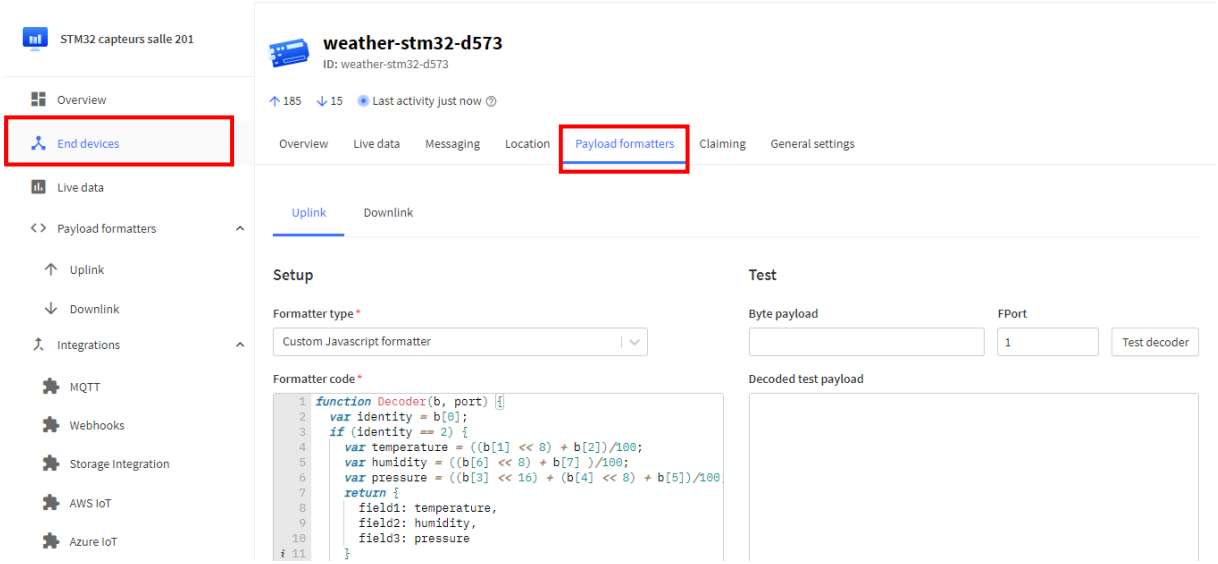
La connection est bien établie lorsque le canal a un statut Healthy sur TTN.



Maintenant que le canal est connecté, il faut y envoyer les bonnes données. Pour appliquer son propre formatage de données, il faut aller dans « *Payload Formatters → Uplink* » et sélectionner le type « *Custom Javascript formatter* ». Il est alors possible d'écrire son formatage.



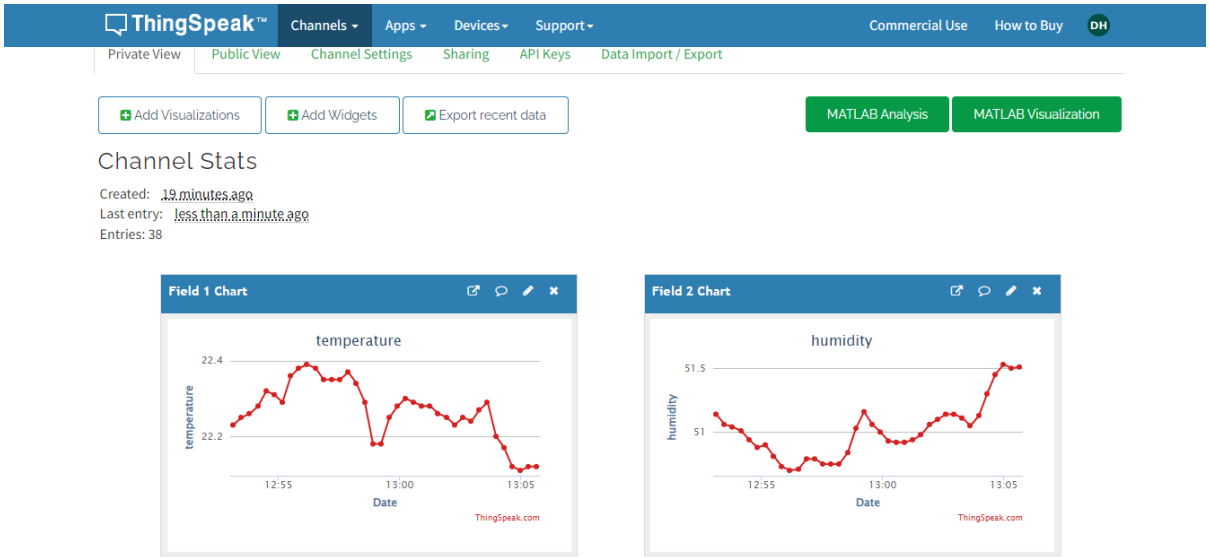
De même dans le « *End devices* », sur chaque device dans « *Payload formatters* », il faut mettre le script de formatage.



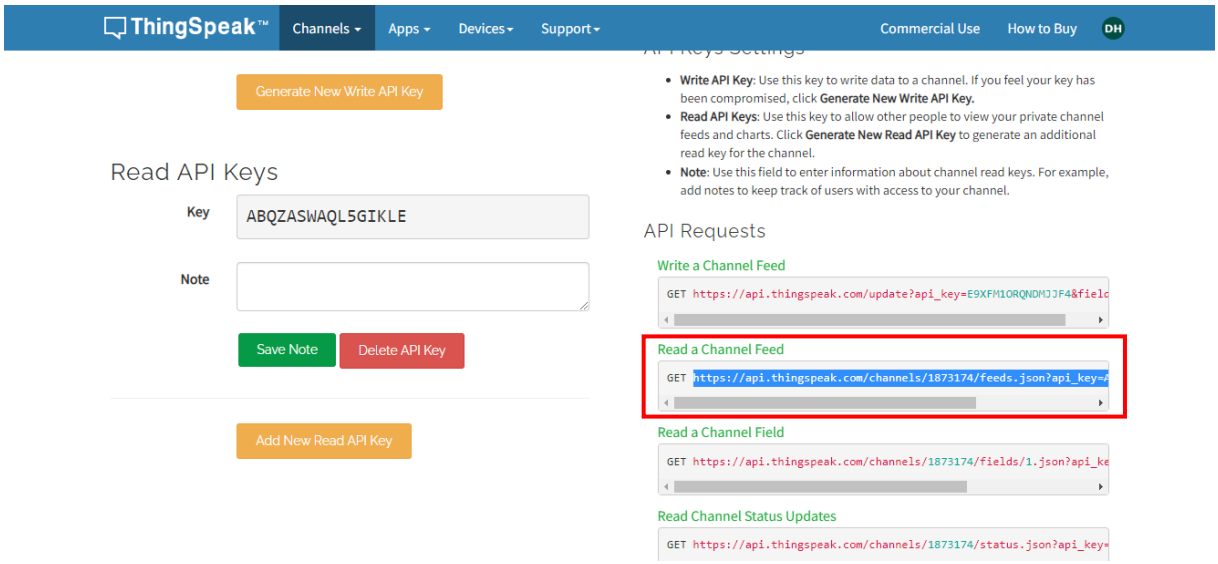
On a identifié chaque capteur sur le premier octet. 0x01 correspond au capteur de CO2/TVOC, celui en 0x02 au capteur de température/humidité/pression et celui en 0x03 au capteur de luminosité. Et ensuite les octets restants sont traités différemment en fonction du capteur pour obtenir les bonnes valeurs pour chaque variable.

```
function Decoder(b, port) {
  var identity = b[0];
  if (identity == 2) {
    var temperature = ((b[1] << 8) + b[2])/100;
    var humidity = ((b[6] << 8) + b[7] )/100;
    var pressure = ((b[3] << 16) + (b[4] << 8) + b[5])/100;
    return {
      field1: temperature,
      field2: humidity,
      field3: pressure
    }
  }
  if (identity == 1) {
    var C02 = ((b[1] << 8) + b[2]);
    var TVOC = ((b[3] << 8) + b[4] );
    return {
      field4: C02,
      field5: TVOC,
    }
  }
    if (identity == 3) {
    var light = b[1];
    return {
      field6: light,
    }
  }
}
```

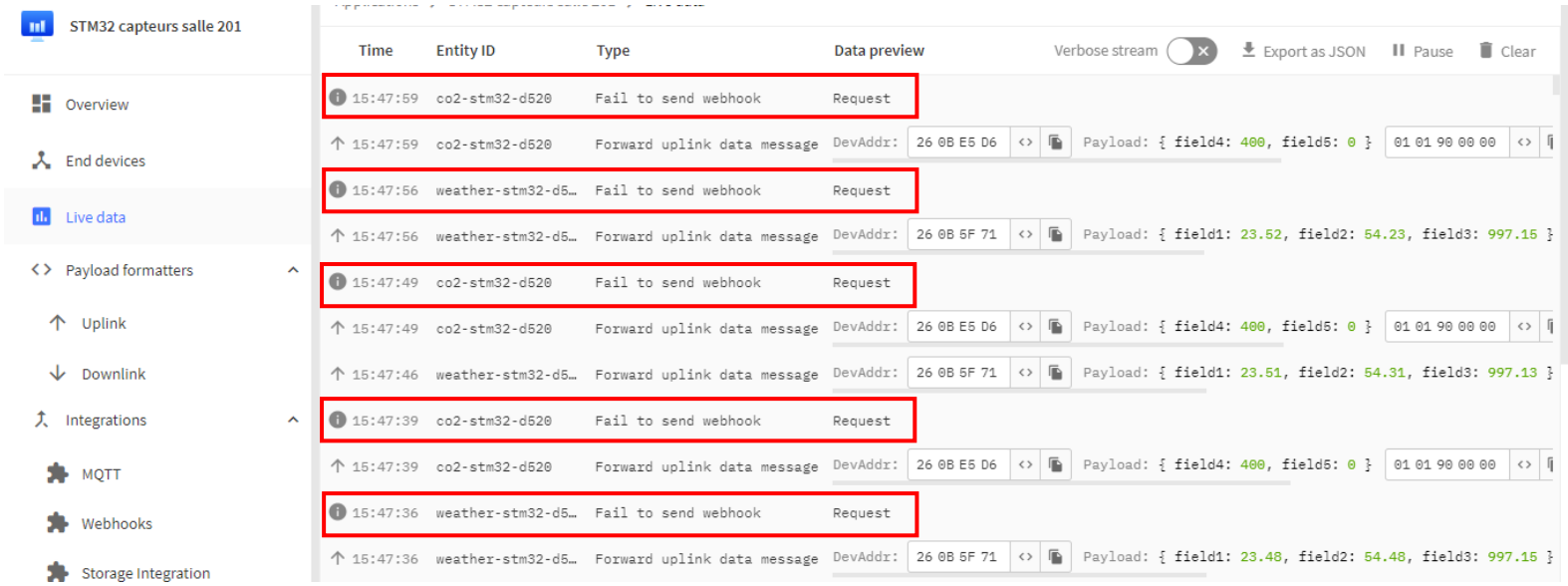
Voici un exemple de valeurs que nous avons pu récupérer et afficher dans l'interface de ThingSpeak.



Pour pouvoir afficher les données dans l'application, il suffit de donner le lien pour lire dans la partie des « *API Keys* », avec aussi la clé de lecture.



On peut repérer quelques axes à améliorer dans cette partie. L'envoi des données échoue assez régulièrement et nous ne savons pas si cela est dû à une trop grande fréquence d'envoi d'information et que le délai de traitement est plus long. Ou bien simplement notre outil n'est pas assez fiable pour établir le lien avec le canal à chaque fois. Il faut continuer d'observer ce comportement et essayer de comprendre pourquoi il y a ces « *Fail to send webhook* » qui apparaissent.



6. Visualisation des données avec Cayenne

Avant de réussir à faire fonctionner ThingSpeak pour les capteurs, nous avons réussi à visualiser les données des capteurs des weather click avec Cayenne comme webhook. Toutefois cette solution n'était pas très modulable car il semblait impossible de créer nos propres widgets et de récupérer les données pour les envoyer vers l'application développée par les étudiants. Nous sommes donc repassés sur ThingSpeak, mais cette étape nous a permis de valider nos capteurs simplement avant de créer notre propre formatage.

L'application qui est couplée avec Cayenne est celle intitulée « stm32-test-modules ».

ENSI CAEN

ÉCOLE PUBLIQUE D'INGÉNIEURS
CENTRE DE RECHERCHE

Ecole Publique d'Ingénieurs en 3 ans

6 boulevard Maréchal Juin, CS 45053
14050 CAEN cedex 04

ministère éducation nationale enseignement supérieur recherche

Normandie Université

Cti Commission des titres d'ingénieur

cdefi Conférence des Directeurs des Écoles Françaises d'Ingénieurs

European Accreditation of Engineering Programmes EUR-ACE®

CONFÉRENCE DES GRANDES ÉCOLES