

Météo Concept



Enseignants : M.Quere et M.Auclair

Sommaire:

<b>Partie personnelle :</b>	<b>3</b>
Présentation :	3
Introduction :	3
Diagramme de déploiement :	4
Diagramme de séquence :	6
Cahier des charges :	6
Gantt personnel :	7
Mes recherches:	8
C'est quoi le Mac version?	9
C'est quoi le paramètre de région?	9
C'est quoi l'ADR algorithm?	10
À quoi sert l'AppKey , AppEui et DevEui?	10
Qu'es que c'est Antenna Gain?	11
Qu'es que c'est que Channel-plan?	11
Installation de Chirpstack (physique) :	11
Installation de Chirpstack (ISO) :	12
Configuration du Serveur Chirpstack :	12
Configuration de la passerelle depuis ISO Chirpstack :	14
Partie Downlink:	16
Création d'un logiciel embarqué pour le lora:	17
Compte rendu sur l'avancement du projet :	18
Conclusion:	19

Partie personnelle :

Présentation :

Je suis Kylliann, née en 2005, je suis actuellement en BTS Ciel (Cybersécurité, Informatique et Électronique) dans la spécialisation de informatique et réseau. j'ai passée mon bac au lycée bréquigny par la voie de la STI2D. Tout au long de ma scolarité et pu apprendre beaucoup de choses que ça soit dans l'informatique ou dans le réseau, même si je voudrais faire que du développement j'ai appris que c'est mieux d'apprendre quelque leçon sur le réseau car cela pouvait le servir dans la vie de tous les jours.

Introduction :

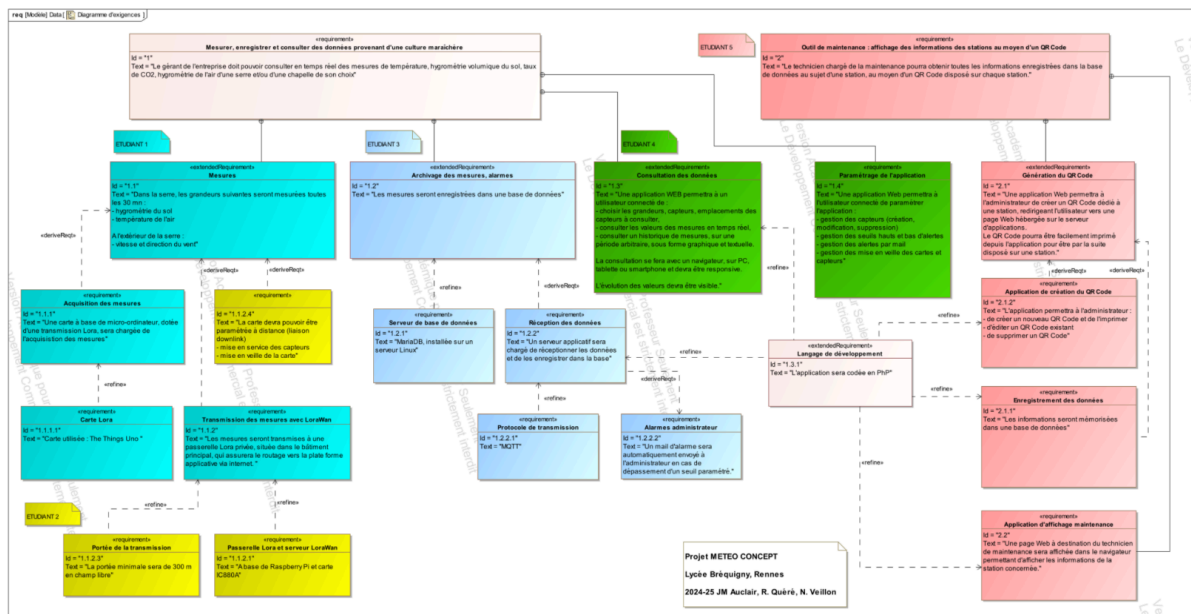
Dans le cadre de mon projet, je travaille sur la mise en place d'une liaison entre les capteurs et la base de données. Pour cela, j'ai conçu une passerelle permettant de centraliser les données collectées par les capteurs et de les transmettre efficacement vers la base de données pour traitement et analyse.

lien du github: <https://github.com/mathias-lmsl/Projet-E6-Meteo-Concept>

Diagrammes :

Diagramme d'exigence :

Dans ce diagramme d'exigence, il explique quel tâche peut faire chaque personne



je suis représenté en tant que l'étudiant jaune et mes tâches sont la carte lora devra se mettre en veille ou non, la portée utilisé doit être de minimum 300 m, et les matériaux choisis sont une raspberry PI et une carte IC880A.

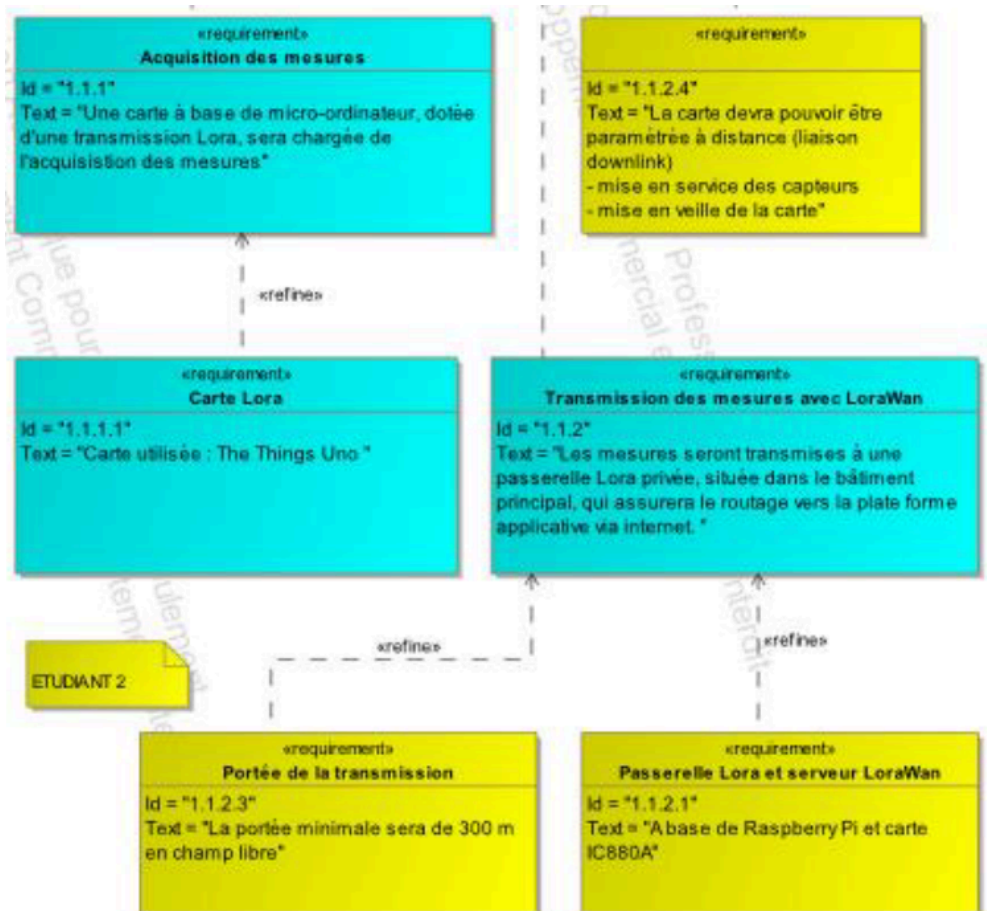


Diagramme de déploiement :

dans ce diagramme, mon bloc est le  $\mu$ -ordinateur auquel dans ce bloc on retrouve le Chirpstack Gateway OS Full qui à les drivers IC880A et le serveur Chirpstack qui à aussi trois sous bloc le bloc de Chirpstack Gateway os full .On retrouve le Router,je mqtt-broker et

handler.

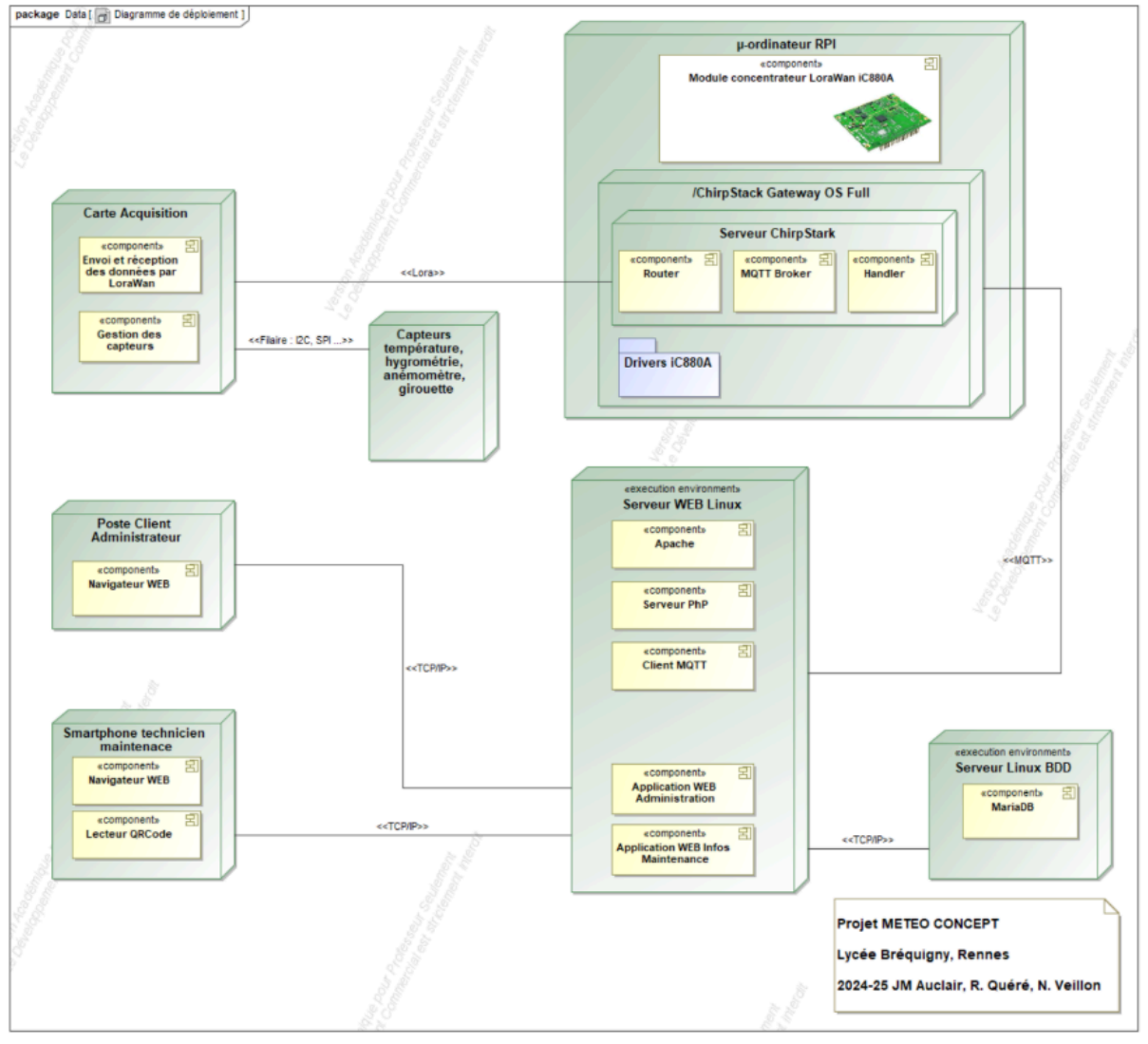
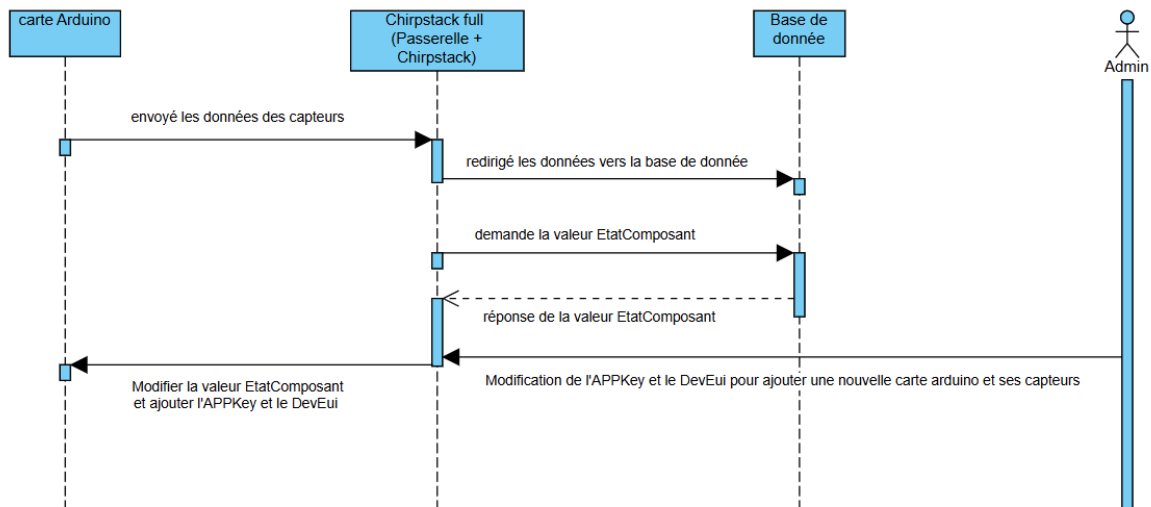


Diagramme de séquence :



pour le diagramme de séquence, on a trois objets (carte Arduino , Chirpstack gateway os full et Base de donnée), il a un acteur qui est l'admin et pour mettre

Cahier des charges :

Voici mon cahier des charges que je dois respecter:

- Mettre en œuvre la passerelle IMST800A + RPI Chirpstack full afin d'assurer la réception et la récupération des données des cartes jusqu'à la passerelle installée dans le bâtiment principal,
- Permettre une liaison descendante (downlink) des cartes vers le serveur, permettant à l'étudiant 4 de

satisfaire l'exigence 1.1.2.4 : commander à distance la mise en veille de cartes et/ou capteurs,

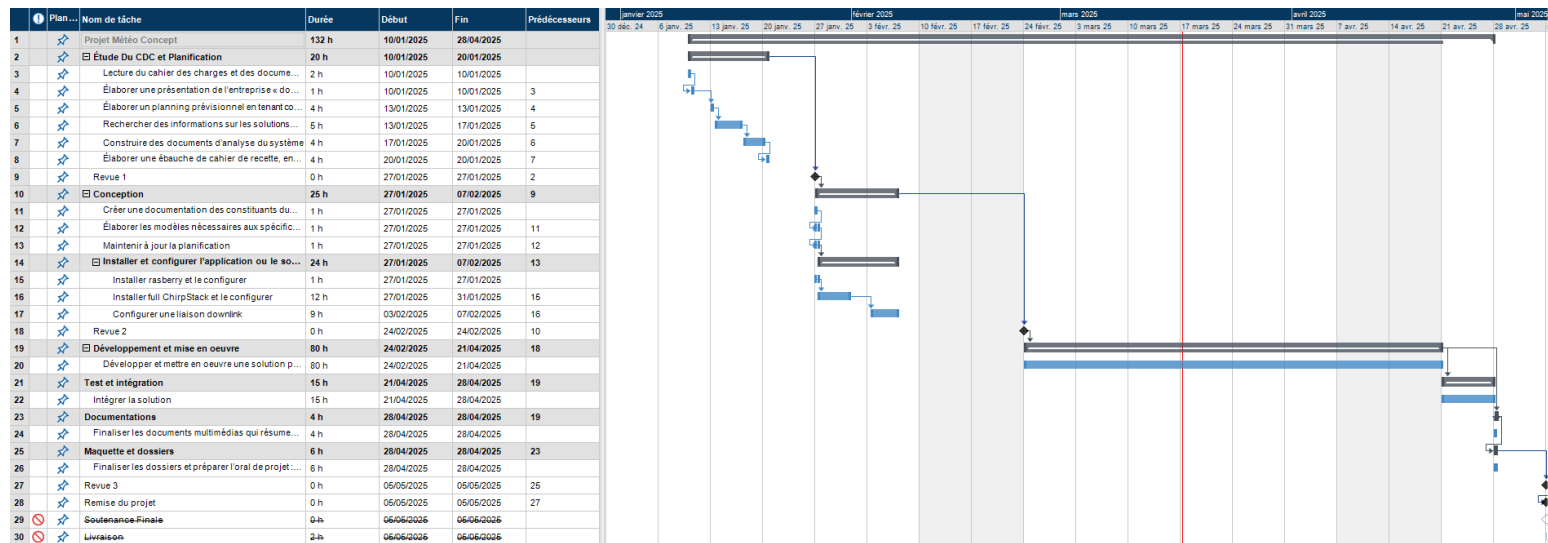
- Mettre en œuvre le logiciel embarqué dans la carte permettant à celle-ci de déclarer un capteur actif

ou inactif (en veille), et de se mettre en veille pour une durée déterminée,

- Valider une portée de transmission d'au moins 300 m.

Gantt personnel :

Voici mon gantt personnel auquel ce que j'avais prévu :



Ceci est le détaille du gantt

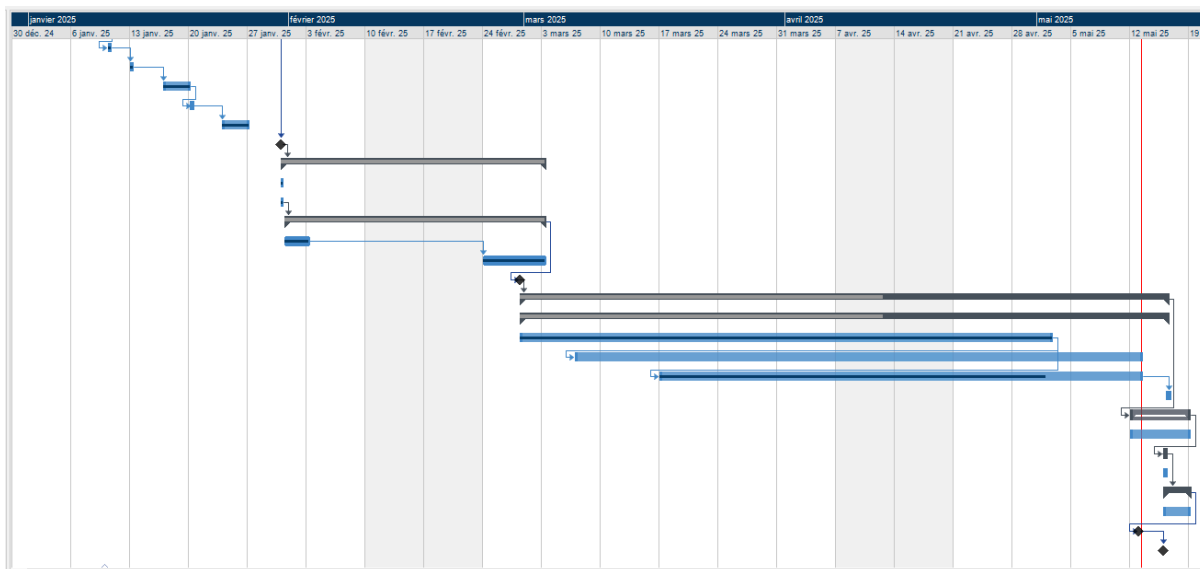
Plan...	Nom de tâche	Durée	Début	Fin	Prédécesseurs
1	Projet Météo Concept	132 h	10/01/2025	28/04/2025	
2	Étude Du CDC et Planification	20 h	10/01/2025	20/01/2025	
3	Lecture du cahier des charges et des docume...	2 h	10/01/2025	10/01/2025	
4	Élaborer une présentation de l'entreprise « do...	1 h	10/01/2025	10/01/2025	3
5	Élaborer un planning prévisionnel en tenant co...	4 h	13/01/2025	13/01/2025	4
6	Rechercher des informations sur les solutions...	5 h	13/01/2025	17/01/2025	5
7	Construire des documents d'analyse du système	4 h	17/01/2025	20/01/2025	6
8	Élaborer une ébauche de cahier de recette, en...	4 h	20/01/2025	20/01/2025	7
9	Revue 1	0 h	27/01/2025	27/01/2025	2
10	Conception	25 h	27/01/2025	07/02/2025	9
11	Créer une documentation des constituants du...	1 h	27/01/2025	27/01/2025	
12	Élaborer les modèles nécessaires aux spécific...	1 h	27/01/2025	27/01/2025	11
13	Maintenir à jour la planification	1 h	27/01/2025	27/01/2025	12
14	Installer et configurer l'application ou le so...	24 h	27/01/2025	07/02/2025	13
15	Installer raspberry et le configurer	1 h	27/01/2025	27/01/2025	
16	Installer full ChirpStack et le configurer	12 h	27/01/2025	31/01/2025	15
17	Configurer une liaison downlink	9 h	03/02/2025	07/02/2025	16
18	Revue 2	0 h	24/02/2025	24/02/2025	10
19	Développement et mise en oeuvre	80 h	24/02/2025	21/04/2025	18
20	Développer et mettre en oeuvre une solution p...	80 h	24/02/2025	21/04/2025	
21	Test et intégration	15 h	21/04/2025	28/04/2025	19
22	Intégrer la solution	15 h	21/04/2025	28/04/2025	
23	Documentations	4 h	28/04/2025	28/04/2025	19
24	Finaliser les documents multimédias qui résume...	4 h	28/04/2025	28/04/2025	
25	Maquette et dossiers	6 h	28/04/2025	28/04/2025	23
26	Finaliser les dossiers et préparer l'oral de projet:...	6 h	28/04/2025	28/04/2025	
27	Revue 3	0 h	05/05/2025	05/05/2025	25
28	Remise du projet	0 h	05/05/2025	05/05/2025	27
29	Soutenance-Finale	0 h	06/06/2025	06/06/2025	
30	Livraison	2 h	06/06/2025	06/06/2025	

Dans le gantt original, je pensais que je faisais que du chirpstack mais après avoir relue, j'ai vu qu'il me manquait deux partie qui sont Permettre une liaison descendante (downlink) des cartes vers le serveur, permettant à l'étudiant 4 de satisfaire l'exigence 1.1.2.4 : commander à distance la mise en veille de cartes et/ou capteurs,

- Mettre en œuvre le logiciel embarqué dans la carte permettant à celle-ci de déclarer un capteur actif

ou inactif (en veille), et de se mettre en veille pour une durée déterminée,

du coup j'ai fait des modifications sur le gantt d'origine pour arriver à ce résultat:



	Plan...	Nom de tâche	Durée	Début	Fin	Prédécesseurs	Progression
1		Projet Météo Concept	161,66 h	10/01/2025	19/05/2025		56%
2	✓	Étude Du CDC et Planification	24 h	10/01/2025	27/01/2025		100%
3	✓	Lecture du cahier des charges et des docume...	2 h	10/01/2025	10/01/2025		100%
4	✓	Élaborer une présentation de l'entreprise « do...	1 h	10/01/2025	10/01/2025	3	100%
5	✓	Élaborer un planning prévisionnel en tenant co...	4 h	13/01/2025	13/01/2025	4	100%
6	✓	Rechercher des informations sur les solutions...	5 h	17/01/2025	20/01/2025	5	100%
7	✓	Construire des documents d'analyse du système	4 h	20/01/2025	20/01/2025	6	100%
8	✓	Élaborer une ébauche de cahier de recette, en...	4 h	24/01/2025	27/01/2025	7	100%
9	✓	Revue 1	0 h	31/01/2025	31/01/2025	2	100%
10	✓	Conception	28 h	31/01/2025	03/03/2025	9	100%
11	✓	Créer une documentation des constituants du...	1 h	31/01/2025	31/01/2025		100%
12	✓	Maintenir à jour la planification	1 h	31/01/2025	31/01/2025		100%
13	✓	Installer et configurer l'application ou le so...	27 h	31/01/2025	03/03/2025	12	100%
14	✓	Installer full ChirpStack et le configurer	5 h	31/01/2025	03/02/2025		100%
15	✓	Création d'un code de test arduino	15 h	24/02/2025	03/03/2025	14	100%
16	✓	Revue 2	0 h	28/02/2025	28/02/2025	13	100%
17	✓	Développement et mise en oeuvre	107,25 h	28/02/2025	16/05/2025	16	56%
18	✓	Développer et mettre en oeuvre une solutio...	107,25 h	28/02/2025	16/05/2025		56%
19	✓	Mise en oeuvre de la passerelle Chirpstack	82,08 h	28/02/2025	02/05/2025		100%
20	✓	Configurer une liaison downlink	93,66 h	07/03/2025	13/05/2025	19	0%
21	✓	Mise en oeuvre d'un logiciel embarqué pour...	75,5 h	17/03/2025	13/05/2025	19	80%
22	✓	Tester la connection (environ 300m)	1 h	16/05/2025	16/05/2025	21	0%
23	✓	Test et intégration	15 h	12/05/2025	19/05/2025	17	0%
24	✓	Intégrer la solution	15 h	12/05/2025	19/05/2025		0%
25	✓	Documentations	4 h	16/05/2025	16/05/2025	23	0%
26	✓	Finaliser les documents multimédias qui résu...	4 h	16/05/2025	16/05/2025		0%
27	✓	Maquette et dossiers	6 h	16/05/2025	19/05/2025	25	0%
28	✓	Finaliser les dossiers et préparer l'oral de proj...	6 h	16/05/2025	19/05/2025		0%
29	✓	Revue 3	0 h	13/05/2025	13/05/2025	27	0%
30	✓	Remise du projet	0 h	16/05/2025	16/05/2025	29	0%
31	✗	Soutenance Finale	0 h	10/01/2026	10/01/2026		0%
32	✗	Livraison	2 h	23/06/2026	23/06/2026		0%

j'ai du me rajouter du temps car ayant eu un problème avec l'os chirpstack gateway full qui arrivais plus à reconnaître ma passerelle du coup ça m'a prit plus de temps à comprendre où était le problème que de pouvoir avancer sur les autres tâches.

Mes recherches:

La plupart de mes travaux de recherche ont eu lieu sur le site officiel de Chirpstack, qui offre toutes les informations nécessaires pour l'implémentation et la configuration des systèmes LoRa. Ce site a prouvé être une source d'information fiable et exhaustive pour répondre à la majorité de mes questions.



Cependant, si je fais face à un manque d'informations spécifiques ou à des difficultés techniques, je me tourne vers le forum communautaire Chirpstack. Ce forum m'a permis de consulter des discussions similaires et d'établir si d'autres utilisateurs avaient rencontré les mêmes soucis. Cela m'a souvent aidé à trouver des solutions concrètes ou des lignes de réflexion pour avancer dans mon projet.

C'est quoi le Mac version?

La version MAC est importante pour organiser la communication entre les appareils et le serveur. Elle définit les règles à suivre pour envoyer les données, comme le choix du canal, la vitesse d'envoi, la puissance du signal et d'autres paramètres. Grâce à ce protocole, les échanges sont plus efficaces et sécurisés, tout en utilisant au mieux la bande passante disponible.

Il existe différentes versions du protocole MAC, chacune apportant des améliorations ou des modifications par rapport à la précédente :

- LoRaWAN 1.0.x : La première version, qui a posé les bases du réseau LoRaWAN.
- LoRaWAN 1.1.x : Introduit des fonctionnalités supplémentaires de sécurité, une meilleure gestion de la batterie et une optimisation des communications.
- LoRaWAN 1.2.x : Améliorations sur la gestion des canaux et des débits de données, ainsi qu'une meilleure sécurité.
- LoRaWAN 1.3.x : La version la plus récente, avec des avancées pour supporter des réseaux plus complexes et des fonctionnalités de sécurité renforcées.

Alors pourquoi la version MAC est-elle importante dans ChirpStack ?

Elle garantit que les échanges de données entre les dispositifs et le serveur se déroulent sans problème, tout en respectant les normes de sécurité et d'efficacité de la communication.

C'est quoi le paramètre de région?

Les paramètres régionaux représentent des valeurs spécifiques à chaque région qui ne sont pas incluses dans les spécifications directes de LoRaWAN. Les éléments considérés incluent les canaux de transmission, la taille maximale des données envoyées, ainsi que la corrélation entre débit de données et paramètres tels que le facteur d'étalement ou la bande passante, sans oublier les réglages de puissance d'émission. Ces paramètres peuvent subir des modifications au fil du temps indépendamment de toute nouvelle version du protocole LoRaWAN.

C'est quoi l'ADR algorithm?

L'ADR algorithm permet d'ajuster automatiquement le débit de données (data rate) et la puissance d'émission des dispositifs LoRaWAN en fonction des conditions du réseau, afin d'améliorer l'efficacité des communications et de prolonger la durée de vie des batteries des capteurs IoT.

L'algorithme ADR présente plusieurs avantages importants pour un réseau LoRaWAN :

1. La portée du réseau s'optimise : ADR ajuste puissance et débit selon la qualité du signal pour étendre la couverture sans ressources gaspillées.
2. L'ADR autorise les appareils à diminuer leur consommation d'énergie en ajustant automatiquement la puissance d'émission et le débit de données ce qui demeure crucial pour les dispositifs alimentés par batterie.
3. L'efficacité du réseau connaît des améliorations : Les paramètres de communication s'ajustent dynamiquement grâce à l'ADR qui permet une gestion réseau optimisée tout en renforçant sa fiabilité.

À quoi sert l'AppKey , AppEui et DevEui?

Dans les réseaux LoRaWAN, il est essentiel que chaque appareil connecté puisse établir son identité et assurer une communication sécurisée avec le réseau. Le fonctionnement repose sur trois composants distincts : le DevEUI, l'AppEUI et l'AppKey.

- Chaque dispositif reçoit un identifiant unique appelé Le DevEUI (Device EUI). Le capteur possède une sorte de carte d'identité. Le réseau obtient la capacité d'identifier avec précision l'appareil qui cherche à établir une connexion. Chaque DevEUI possède une unicité mondiale qui garantit qu'aucun appareil ne sera confondu avec un autre.
- L'AppEUI (Application EUI) représente l'identifiant spécifique qui lie un appareil à son application ou serveur associé. Il établit la connexion entre un dispositif et l'application destinée à recevoir ses informations. L'AppEUI garantit que les données transmises par l'appareil atteignent leur destination correcte.
- Le dispositif utilise l'AppKey (Application Key) comme une clé de sécurité privée pour s'authentifier lors de sa connexion au réseau. Elle permet également de sécuriser les échanges de données par cryptage pour maintenir la confidentialité des communications et restreindre l'accès au réseau à l'appareil autorisé uniquement.

Les trois éléments mentionnés jouent un rôle crucial pour l'identification des dispositifs, leur lien avec des applications spécifiques et la sécurisation des communications au sein d'un réseau LoRaWAN.

Qu'est-ce que c'est Antenna Gain?

Le gain d'antenne représente une méthode pour évaluer combien une antenne peut focaliser un signal vers une direction spécifique. Une antenne dotée d'un gain élevé possède la capacité d'émettre ou de recevoir des signaux renforcés dans une direction déterminée ce qui lui confère une performance supérieure par rapport à une antenne isotrope qui diffuse uniformément dans toutes les directions.

Le gain se mesure en décibels isotropes (dBi). Il faut noter qu'une antenne à gain élevé ne produit pas davantage de puissance mais seulement une meilleure orientation de celle-ci. Cela permet d'améliorer la portée ou la qualité du signal dans les zones nécessaires tout en évitant le gaspillage d'énergie dans d'autres endroits.

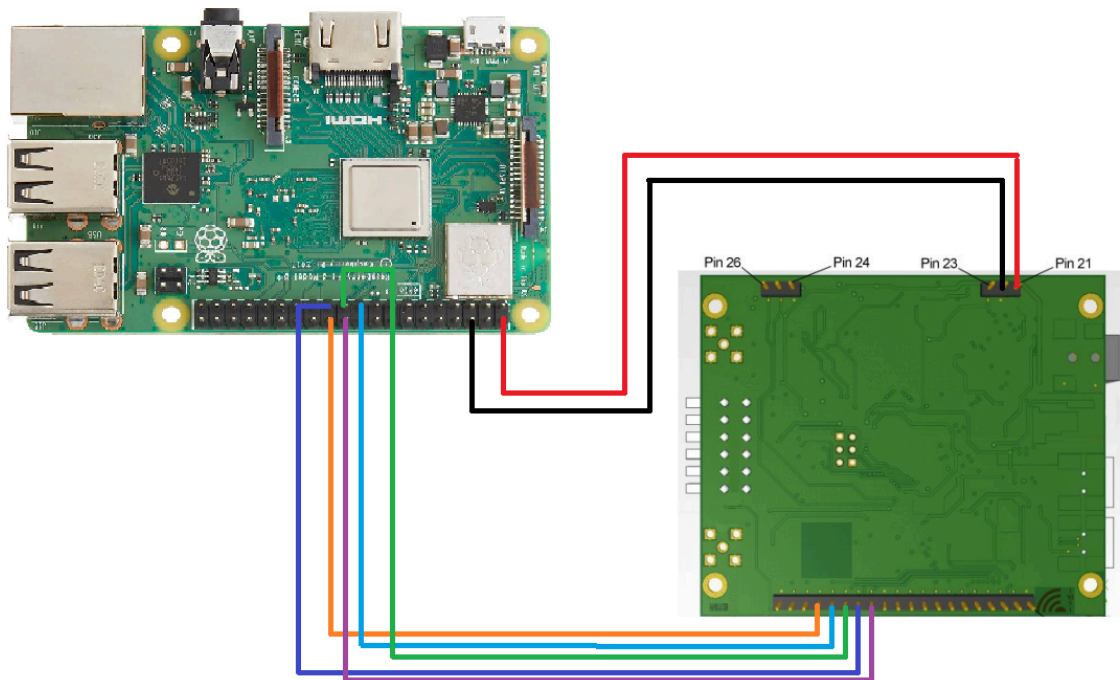
Qu'est-ce que c'est Channel-plan?

Un Channel Plan, ou plan de canaux, constitue un outil pour organiser l'utilisation des diverses fréquences au sein d'un réseau sans fil. Il s'agit d'une sorte de carte qui détermine quelles plages de fréquences doivent être utilisées et à quels emplacements pour empêcher plusieurs appareils d'utiliser simultanément le même canal ce qui provoquerait des interférences.

Ce plan garantit des communications sans perturbations tout en maintenant des performances réseau optimisées. Les réseaux Wi-Fi ainsi que les systèmes radio et tous les autres systèmes sans fil l'utilisent.

Installation de Chirpstack (physique) :

Pour l'installation de la passerelle sur la carte arduino, j'ai dû me renseigner sur le site lora pour avoir cette installation



Le câble orange est le NSS qui permet de sélectionner le concentrateur LoRa via SPI pour permettre la communication avec le microcontrôleur ou la passerelle.

Le câble bleu clair est le MOSI qui permet d'envoyer les données de la raspberry pi 3 B+ vers la passerelle ICM880A.

Le câble vert est MISO qui permet d'envoyer les données de la passerelle ICM880A vers la raspberry pi 3 B+.

Le câble bleu foncé est le SPI Clock qui permet de rythmer la communication entre les deux cartes.

Installation de Chirpstack (ISO) :

Une fois que vous avez correctement connecté votre ICM880A à la Raspberry Pi 3B+ l'étape suivante consiste à installer ChirpStack (version Full) pour gérer et configurer votre réseau LoRaWAN.

Il faut installer l'iso chirpstack de type full qui nous permettra de faire la configuration de la passerelle et du serveur chirpstack.

on va aller sur un navigateur et mettre <http://ip-du-serveur-chirpstack>, ce qui va nous emmener sur une page où on peut voir chirpstack et la configuration de la passerelle.

Ce qui va nous intéresser pour l'instant, c'est la configuration de chirpstack.

Configuration du Serveur Chirpstack :

Sur la partie Chirpstack, on va devoir créer un projet pour pouvoir commencer à faire la configuration de la passerelle

Tenants / ChirpStack / Device profiles / Device BTS 2025

**Device BTS 2025** device profile id: 9b639a4b-6959-4580-beb7-fa0d22398109 [Delete device profile](#)

[General](#) [Join \(OTAA / ABP\)](#) [Class-B](#) [Class-C](#) [Codec](#) [Relay](#) [Tags](#) [Measurements](#) [Select device-profile template](#)

\* Name  
Device BTS 2025

Description

\* Region  
EU868

Region configuration ⓘ

\* MAC version ⓘ  
LoRaWAN 1.0.3

\* Regional parameters revision ⓘ  
A

\* ADR algorithm ⓘ  
Default ADR algorithm (LoRa only)

Flush queue on activate ⓘ ☒

Allow roaming ⓘ ☐

\* Expected uplink interval (secs) ⓘ 3600

Device-status request frequency (req/day) ⓘ 1

RX1 Delay (0 = use system default) ⓘ 0

On devra ajouter un nom , la région ,le MAC version, le paramètre de région,l'ADR algorithm et l'intervalle pour le uplink.

## Partie Gateway

En cliquant sur gateway, on pourra créer la ID de la passerelle.

Tenants / ChirpStack / Gateways

**Gateways** [Add gateway](#) [Selected gateways](#)

<input type="checkbox"/>	Last seen	Gateway ID	Name	Region ID	Region common-name
<input checked="" type="checkbox"/> Online	2025-03-07 15:54:32	c2fe30d26efc1b41	Gateway BTS 2025	eu868	EU868

En cliquant sur le bouton “add gateway”, page de configuration apparaît, on va devoir mettre un nom et mettre / générer une ID pour la passerelle.

**Gateway BTS 2025** gateway id: c2fe30d26efc1b41 [Delete gateway](#)

[Dashboard](#) [Configuration](#) [TLS certificate](#) [LoRaWAN frames](#)

[General](#) [Tags](#) [Metadata](#)

\* Name  
Gateway BTS 2025

Description

\* Gateway ID (EUI64)  
c2fe30d26efc1b41

\* Stats interval (secs) ⓘ 30

## Partie device:

Suite à la création de l'ID de la passerelle, on ira faire la communication entre la carte lora et le serveur chirpstack full

Tenants / ChirpStack / Applications / Application BTS 2025 / Devices / carte de test

carte de test device eui: 0004a30b00216c4c Delete device

Dashboard Configuration OTAA keys Activation Queue Events LoRaWAN frames

Device Tags Variables

\* Name

carte de test

Description

carte de test pour pouvoir faire la liaison.

\* Device EUI (EUI64)

0004a30b00216c4c

Join EUI (EUI64)

e7bc29efe7c14e1e

MSB ↕ ↻ 🗑

\* Device profile

Device BTS 2025

Device is disabled

☐

Disable frame-counter validation

☐

Submit

on va devoir mettre un nom , mettre le device EUI de la carte lora , le Join EUI (qui est plus référencer en APP Eui) qui va être le même que le DevEui mais on peut aussi en générer un cela marche aussi et mettre le device profile qu'on a créer.

En allant, sur OTAA Keys à côté de configuration,on va pouvoir générer une Application Key.

Tenants / ChirpStack / Applications / Application BTS 2025 / Devices / carte de test

carte de test device eui: 0004a30b00216c4c Delete device

Dashboard Configuration OTAA keys Activation Queue Events LoRaWAN frames

Flush OTAA device nonces

\* Application key

b7f767f6bdd1ed9b4d9e68822be9b4d2d

MSB ↕ ↻ 🗑

Submit

## Configuration de la passerelle depuis ISO Chirpstack :

On va juste activer le Chirpstack Concentrator, puis aller dans SX1301 pour mettre l'ID de la passerelle.

ChirpStack Concentrator

ChirpStack Concentrator provides a unified API interface to LoRa(R) concentrator hardware. Please refer to the [ChirpStack Concentrator Hardware](#) page for supported hardware and configuration options.

Global configuration

SX1301

SX1302 / SX1303

2.4GHz

SX1301

Configure the fields below if you have selected the SX1301 chipset.

Antenna gain (dBi)

2

Shield model

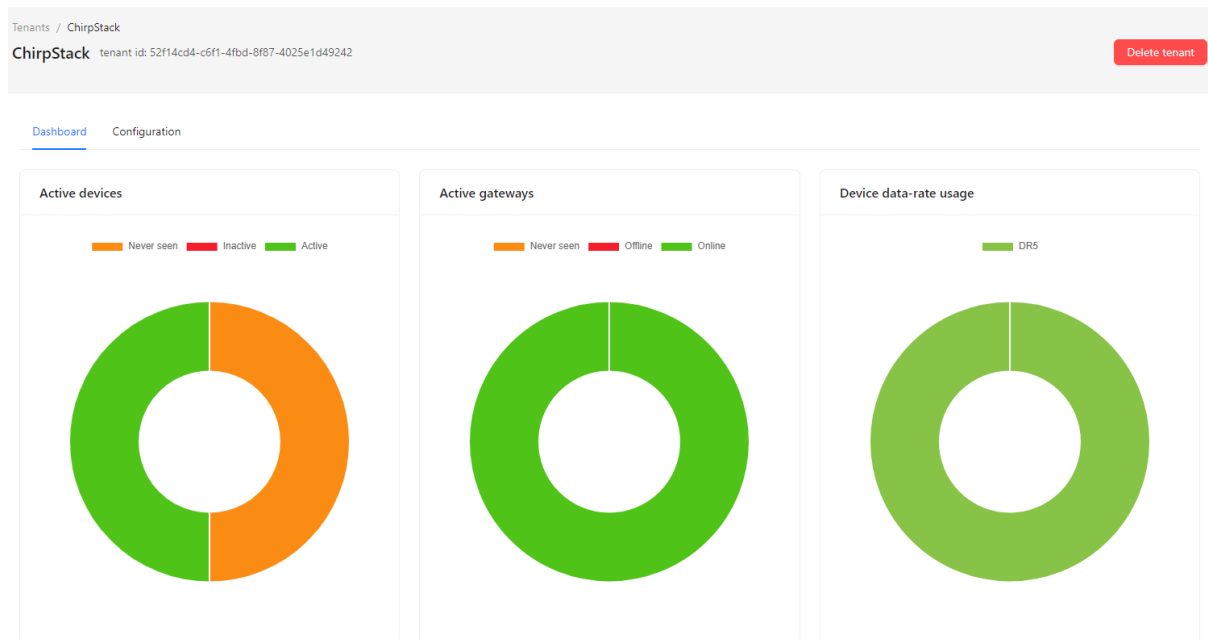
IMST - iC880A

Channel-plan

EU868 - Standard channels + 867.1, 867.3, ... 867.9

Select the channel-plan to use. This must be supported by the selected shield.

Pour finir on peut voir si tout est bien configuré en allant dans chirpstack.

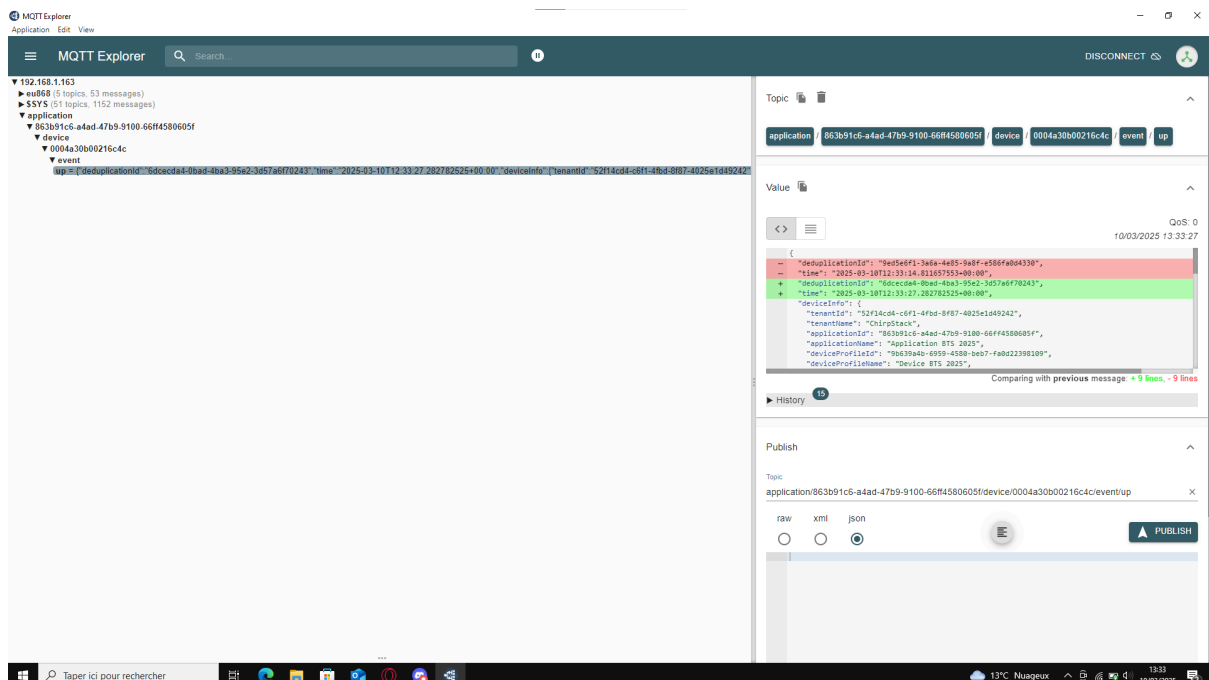


Si vous avez du rouge dans l'active gateways, c'est que la passerelle est mal configurée ou elle ne marche pas.

Si vous avez du rouge dans l'active devices, cela veut dire peut-être de la configuration de la carte Lora, du DevEUI que vous avez mal mis ou sinon la carte n'envoie pas de donnée.

Pour voir la transmission de la donnée dans chirpstack full, on peut utiliser plusieurs méthodes soit aller dans l'application MQTT Explorer, soit aller dans les événements des appareils.

Utilisation de MQTT Explorer:



utilisation de event dans chirpstack

Tenants / ChirpStack / Applications / Application BTS 2025 / Devices / carte de test

carte de test device eui: 0004a30b00216c4c Delete device

Dashboard Configuration OTAA keys Activation Queue Events LoRaWAN frames

Download

2025-04-28 09:48:19	<span>up</span>	DR: 5 Data: 003405450067013e FCnt: 0 FPort: 1
2025-04-28 09:48:14	<span>join</span>	DevAddr: 005535e1
2025-04-28 09:45:00	<span>up</span>	DR: 5 Data: 003405450067013e FCnt: 0 FPort: 1
2025-04-28 09:44:55	<span>join</span>	DevAddr: 01f59ede
2025-04-04 14:55:23	<span>join</span>	DevAddr: 0098ab41
2025-04-04 09:40:54	<span>join</span>	DevAddr: 01d39e40
2025-04-04 09:22:26	<span>up</span>	DR: 5 Data: 003405450067013e FCnt: 0 FPort: 1
2025-04-04 09:22:21	<span>join</span>	DevAddr: 00d24c2c
2025-04-04 09:20:08	<span>join</span>	DevAddr: 00dc5591
2025-04-04 09:19:50	<span>join</span>	DevAddr: 016db11c

Dedans on peut voir si c'est du downlink et du uplink comme indication up pour uplink et txack pour le downlink.

2025-05-12 16:46:27	<span>up</span>	DR: 5 Data: 01 FCnt: 3 FPort: 1
2025-05-12 16:46:26	<span>txack</span>	
2025-05-12 16:46:26	<span>up</span>	DR: 5 Data: 00 FCnt: 2 FPort: 1
2025-05-12 16:46:15	<span>txack</span>	
2025-05-12 16:46:15	<span>up</span>	DR: 5 Data: 01 FCnt: 1 FPort: 1
2025-05-12 16:46:15	<span>status</span>	Battery level: 0% Battery level unavailable: true External power source: false Margin: 7

## Partie Downlink:

Je recherche des solutions pour pouvoir envoyer les valeurs 00 et 01 pour contrôler l'état.

Le programme ci-dessus me permet de voir si une donnée est bien reçue sur la carte lora.

```
#include <TheThingsNetwork.h>
#include <Arduino.h>

// Set your AppEUI and AppKey
const char *appEui = "0004A30B00216C4C"; // Remplacez par votre AppEUI
const char *appKey = "bf767f6bddled0b4d0e68822be9b4d2d"; // Remplacez par votre AppKey

#define loraSerial Serial1 // Utilisation de Serial1 pour la communication LoRa
#define debugSerial Serial // Utilisation de Serial pour le débogage
```



```
debugSerial.println("-- STATUS");
ttn.showStatus(); // Affiche l'état du module LoRa

debugSerial.println("-- JOIN");
ttn.join(appEui, appKey); // Connexion au réseau LoRaWAN avec l'AppEUI et l'AppKey
}

void loop() {
  // Exemple pour envoyer un message uplink
  byte payload[] = { 0x01, 0x02, 0x03, 0x04 }; // Un tableau d'octets à envoyer en payload
  debugSerial.println("Envoi du message uplink");
  ttn.sendBytes(payload, sizeof(payload)); // Envoi du message

  // Vous n'avez pas besoin de gérer explicitement les downlinks ici. TheThingsNetwork
  // gère cela en interne via des callbacks. Les messages downlink arrivent automatiquement
  // selon les périodes ou les événements du réseau LoRaWAN (programmé via Chirpstack par exemple).
  debugSerial.println("Message envoyé. Attente de Downlink...");

  // Vous pouvez ici définir une pause avant le prochain envoi, par exemple :
  delay(10000); // Attendez 10 secondes avant d'envoyer le prochain message
}
```

### Création d'un logiciel embarqué pour le lora:

Je dois créer un code pour pouvoir mettre en veille ou mettre en état active.

Pour l'instant, je fais des recherches pour pouvoir voir si la donnée du downlink passe bien dans la carte Lora.

```
void messageCallback(const uint8_t *payload, size_t size, port_t port) {
  if (size > 0) {
    debugSerial.print("Réception du downlink: ");
    for (size_t i = 0; i < size; i++) {
      debugSerial.print(payload[i], HEX);
      debugSerial.print(" ");
    }
    debugSerial.println();

    if (payload[0] == 0x00) {
      // Arrêter l'envoi de données
      sendData = false;
      debugSerial.println("Envoi des uplinks désactivé.");
    } else if (payload[0] == 0x01) {
      // Réactiver l'envoi de données
      sendData = true;
      debugSerial.println("Envoi des uplinks réactivé.");
    }
  }
}
```

Dans ce bout de code, il permet de comparer la valeur du downlink pour soit arrêter l'envoi des données en uplink ou renvoyer/ continuer d'envoyer les données.

```
// Fonction pour envoyer les données à TTN
void sendDataToTTN() {
    if (sendData) {
        byte payload[] = { 0x01 }; // Message à envoyer
        ttn.sendBytes(payload, sizeof(payload)); // Envoi sans affichage des messages
    }
    else {
        debugSerial.println("Envoi bloqué, aucune donnée ne sera envoyée.");
    }
}
```

Cette partie de code me permet d'envoyer des données sur le uplink et dire s'il est arrêté.

```
void loop() {
    // Vérifier les messages et envoyer des données si l'envoi est autorisé
    ttn.poll(); // Vérifie les messages downlink
    sendDataToTTN(); // Envoie des données si l'envoi est activé

    // Message de débogage pour suivre le processus
    debugSerial.println("Cycle terminé. Attente du prochain envoi...");
    delay(10000); // Attendre avant de recommencer
}
```

Dans cette partie de code, elle regarde qu'elle est la donnée du downlink et envoie les données s'il est activé.

Le résultat que j'ai :

```
16:43:48.977 -> Successful transmission. Received 01
16:43:48.977 -> Réception du downlink: 1
16:43:48.977 -> Envoi des uplinks réactivé.
16:43:48.977 -> Envoi du message uplink...
16:43:48.977 -> Sending: mac tx uncnf 1 01
16:43:51.289 -> Successful transmission
16:43:51.289 -> Cycle terminé. Attente du prochain envoi...
16:44:01.281 -> Sending: mac tx uncnf 1 00
16:44:02.439 -> Successful transmission. Received 00
16:44:02.439 -> Réception du downlink: 0
16:44:02.439 -> Envoi des uplinks désactivé.
16:44:02.439 -> Envoi bloqué, aucune donnée ne sera envoyée.
16:44:02.439 -> Cycle terminé. Attente du prochain envoi...
```

Ce n'est pas complètement terminé car quand arrête d'envoyer des données , il m'envoie encore une donnée qui n'est plus 01 mais 00.

Compte rendu sur l'avancement du projet :

J'ai réussi à installer une passerelle qui connecte les capteurs à la base de données qui avait un problème car la passerelle a commencé à arrêter de fonctionner.

Maintenant tout remarche bien sans problème.

Cependant, le projet n'est pas encore terminé. Le downlink doit encore être configuré pour permettre la communication vers les capteurs et je finalise mon programme embarqué pour pouvoir modifier l'état du capteur. Je prévois aussi d'ajouter un système avec des potentiomètres pour faire des tests plus détaillés et de laisser la passerelle tourner pendant 1 semaine.

Du coup sur l'état du projet, je suis en retard sur mes tâches mais l'avancée du projet est plus rapide depuis que l'incident sur Chirpstack a été trouvé.

A	B	C	D
N°	Action	Attendu	
1	installer chirpstack full	le systeme chirpstack est bien installé	ok
2	configuration de chirpstack	la passerelle prend bien le gateway id afficher en vert dans l'interface web de chirpstack et on a les données de AppEui (JoinEui) et AppKey	ok
3	la carte lora se connecte bien sur la passerelle	le device se connecte bien à la passerelle ,afficher en vert dans l'interface web de chirpstack	ok
4	connection de downlink sur la carte arduino	les données envoyer sur le chirpstack permet de arrêter ou renvoyer/continuer d'envoyer les données en uplink	pas ok
5	faire la liaison entre le serveur chirpstack avec la page web	quand il demande d'arrêter à la carte lora il doit passer sur chirpstack pour après envoyer vers la carte lora et inversement pour reactiver ou continuer l'envoi des données	pas ok

Conclusion:

Chirpstack est une plateforme simple et efficace. Même un débutant peut facilement comprendre comment la configurer et la faire fonctionner. Cependant, certaines parties du site utilisent des informations d'anciennes versions ou manquent de détails, ce qui peut parfois créer des confusions. Malgré cela, les ressources disponibles et l'aide de la communauté permettent généralement de trouver des solutions et d'avancer.