

LoRaWAN

- Introduction à l'loT -

Support de Travaux Pratiques



Contacts

Dimitri Boudier – Responsable du module – TP
dimitri.boudier@ensicaen.fr

Ressources

Toutes les ressources (supports CM, TP et outils) sont sur la page Moodle du cours :
<https://foad.ensicaen.fr/course/view.php?id=213>

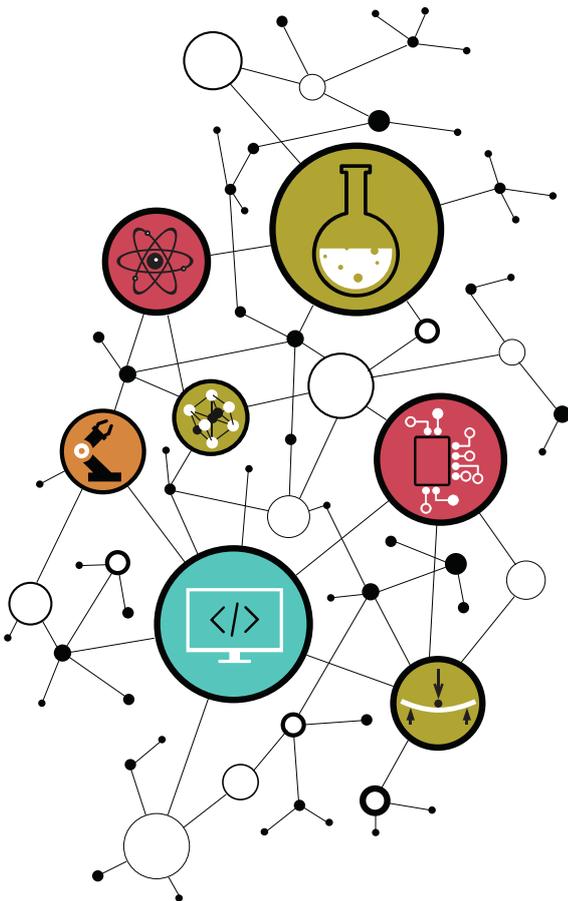


Except where otherwise noted, this work is licensed under
<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Contacts.....	2
Ressources.....	2
Partie 1 Prélude.....	5
I. Mise en contexte.....	6
I.1. Capteurs et systèmes connectés.....	6
I.2. Organisation du travail.....	7
II. Besoins en matériels et logiciels.....	8
II.1. En bref.....	8
II.2. Présentation du matériel.....	9
III. Prise en main de l'environnement.....	12
III.1. Création du projet.....	12
III.2. Étude du projet généré.....	13
III.3. Manipuler les GPIOs avec la HAL.....	14
III.4. Utiliser les interruptions.....	16
Partie 2 Liaison Série Asynchrone / UART.....	21
I. Présentation.....	22
II. Spécifications.....	23
III. Implémentation sur STM32.....	24
III.1. Création du projet.....	24
III.2. Étude du projet généré.....	25
III.3. UART en transmission.....	26
III.4. UART en réception.....	29
III.5. Bridge UART (pour les plus avancés).....	32
IV. Synthèse.....	35
Partie 3 Bus SPI.....	37
I. Spécifications.....	38
II. Constitution d'une trame.....	39
III. Application : Light click.....	40
III.1. Étude du capteur.....	40
III.2. Configuration du périphérique SPI.....	41
III.3. Driver light_click.....	42
IV. Application : Accel 2 click.....	47
IV.1. Étude du capteur.....	47
IV.2. Configuration du périphérique SPI.....	48
IV.3. Driver accel_2_click.....	49
V. Synthèse.....	57
Partie 4 Bus I²C.....	59
I. Description.....	60
II. Application : Weather click.....	61
II.1. Étude du capteur.....	61
II.2. Configuration du périphérique I ² C.....	62
II.3. Driver weather_click.....	62
II.4. Synthèse.....	73
Partie 5 Glossaire.....	75

PARTIE 1

DE L'IOT À LORAWAN



I. The Internet of Things

L'**Internet des Objets**, que l'on nommera régulièrement **IIoT** pour **Internet of Things**, désigne une catégorie de systèmes embarqués ayant eu une évolution flagrante ces dernières années¹.

Ces produits se distinguent certes par une connectivité aux différents réseaux existants (Sigfox, Wi-Fi, 4G, 5G, ...) mais aussi par leur faible consommation et leur faible coût (tout cela allant de pair avec une faible taille et de faibles performances de calcul). La connectivité de ces appareils peut se faire sur différents protocoles, chacun apportant un compromis différent dans le triptyque portée-débit-consommation, comme le montre la figure suivante².

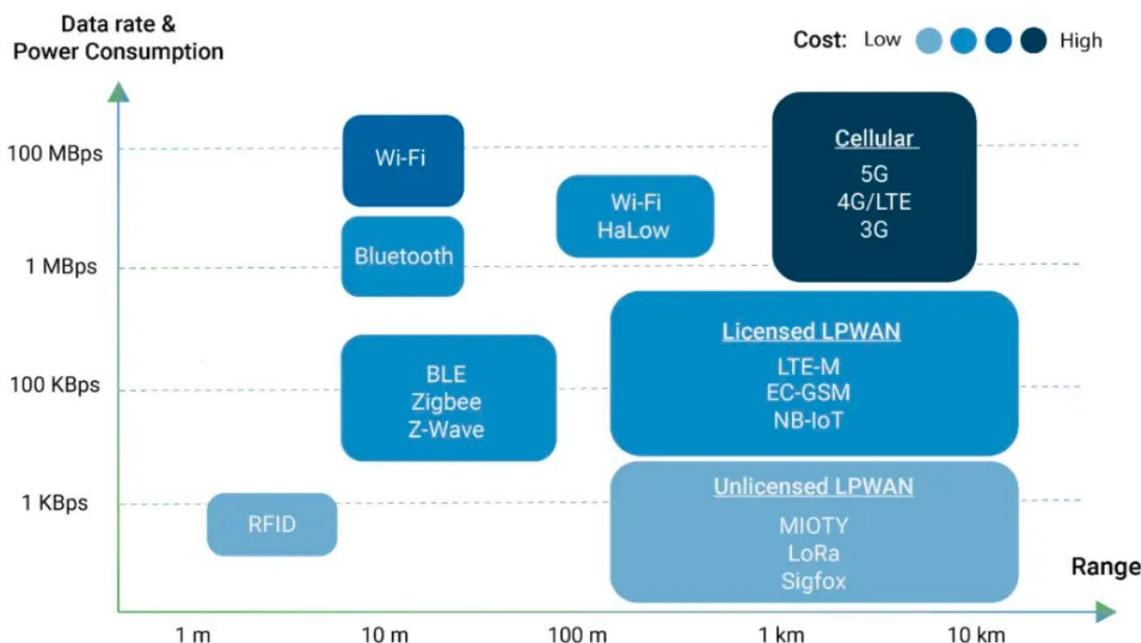


Figure 1: Comparaison de différents protocoles radio

Cette figure met en avant que les protocoles radio ayant la plus grande portée et la plus faible consommation sont appelés **LPWAN**, pour **Low-Power Wide Area Network**. Parmi ces protocoles on cite Sigfox (société française mondialement connue) et LoRa (sujet de cet TP).

II. Modulation LoRa

LoRa (Long Range) est donc un **protocole radio** ayant pour principales caractéristiques un faible débit mais une faible consommation et une très grande portée. En Europe, ce protocole peut utiliser les bandes libres à 433 MHz et 868 MHz.

Les détails de la modulation ne sont pas utiles à la compréhension ou à la réalisation du reste du TP. Ainsi l'aspect « couche physique » de notre système n'est pas abordé ici. Si vous souhaitez plus d'informations, nous vous conseillons le premier des deux livres rédigés par l'Université Savoie Mont Blanc et approuvés par la LoRa Alliance : « LoRa – LoRaWAN et l'Internet des Objets »³.

1 [https://iiot-analytics.com/number-connected-iiot-devices/](https://iot-analytics.com/number-connected-iiot-devices/)

2 <https://www.mokolora.com/lora-and-wireless-technologies/>

3 <https://www.univ-smb.fr/lorawan/livre-gratuit/>

III. Architecture réseau LoRaWAN

La figure ci-dessous montre l'architecture d'un réseau LoRaWAN. Pour rappel, l'architecture d'un réseau désigne l'ensemble des équipements matériels et protocoles d'échanges utilisés.

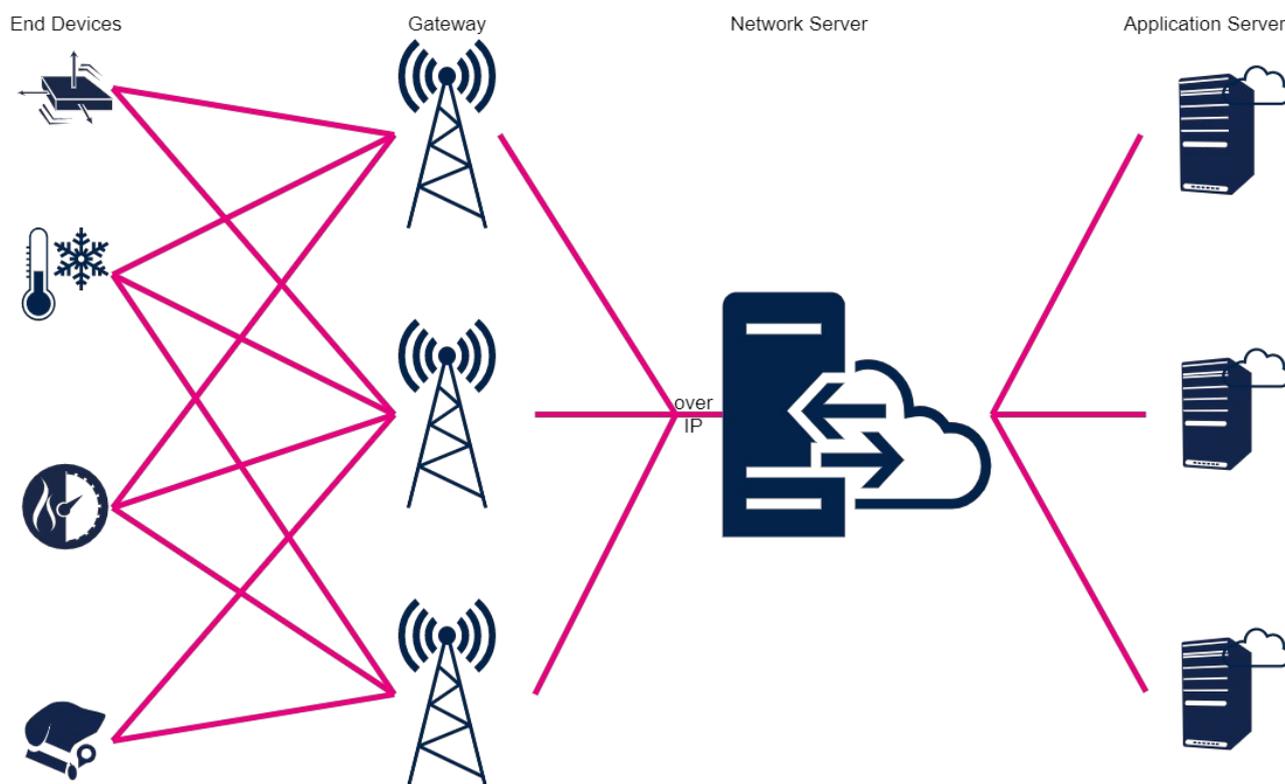


Figure 2: Architecture du réseau LoRaWAN

III.1. End device

Tout à gauche du schéma se trouve les **End Devices** ou **Nodes**. Ce sont les appareils IoT qui échangent des informations en modulation LoRa. Ces composants n'ont pas de cible particulière : tout appareil dans la zone de couverture recevra le message.

III.2. Gateway

Les **passerelles** ou **gateways** sont en principe les destinataires des *end devices*. Elles possèdent donc une interface en LoRa permettant d'échanger des données avec les appareils IoT.

De l'autre côté, les passerelles sont connectés à un *Network Server*. Elles disposent donc d'une seconde interface connectée à Internet (4G, 5G, Ethernet, Wi-Fi, ...).

Leur rôle est donc de faire une « conversion » du protocole LoRa (utilisé par les *end devices*) vers un protocole IP (utilisé par les serveurs).

Toutes les gateways possèdent un identifiant unique (64-bit EUI) permettant de les associer à un *Network Server*.

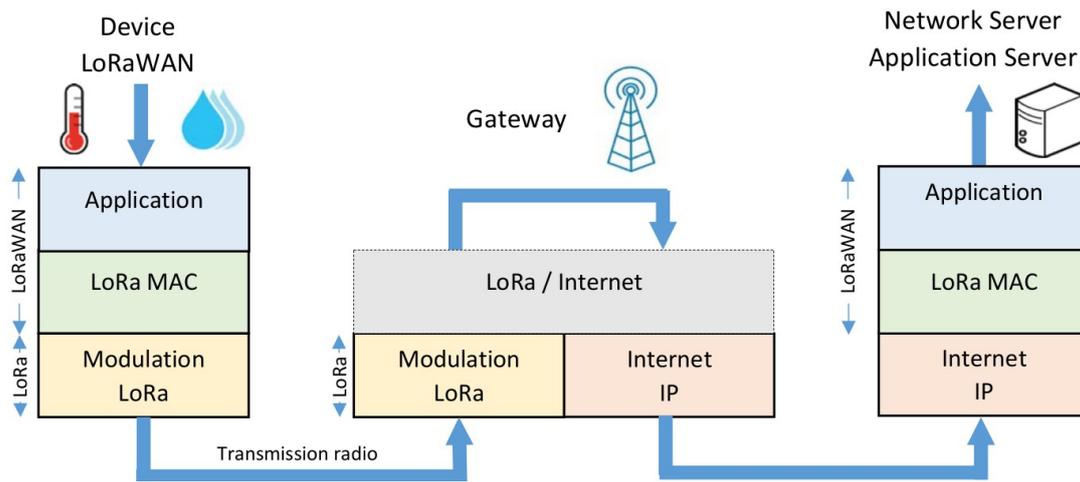


Figure 3: Gateway LoRaWAN

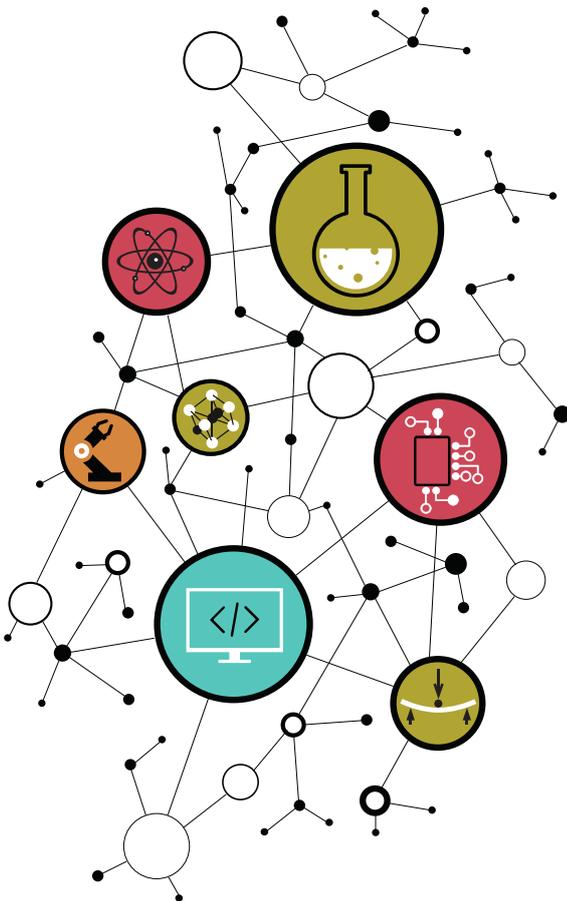
Gateway

End Device → Gateway = uplink l'autre = downlink

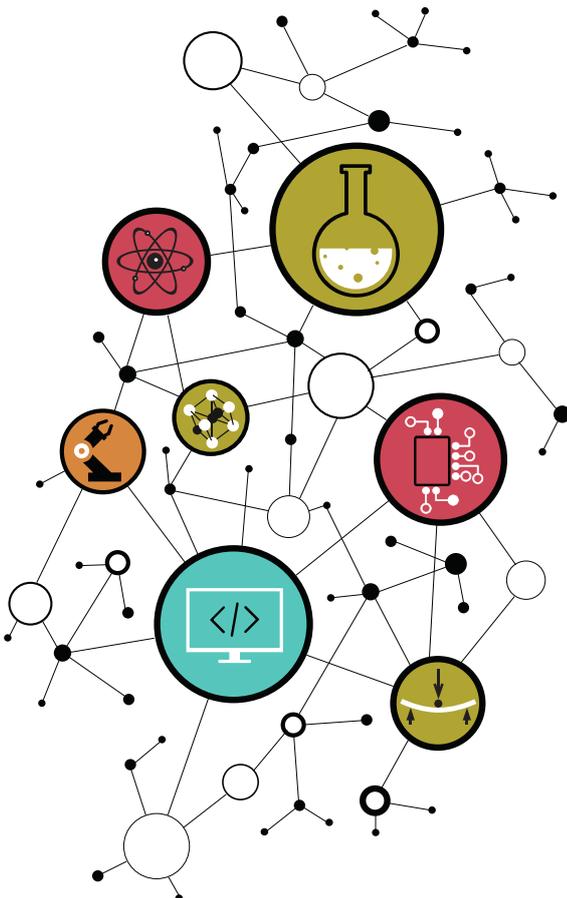
LoRaWAN en particulier désigne la surcouche protocolaire mise en place pour lier un

PARTIE 2

LoRA END DEVICE



PARTIE 3
LoRAWAN



I. Gateway

Une passerelle LoRaWAN dispose de deux interfaces : une interface LoRa avec les appareils IoT et une interface Internet avec le *Network Server*. Concernant l'interface LoRa, il n'y a rien à configurer puisqu'une passerelle écoutera tous les échanges par construction. En revanche il faut s'assurer que l'interface Internet soit fonctionnelle, ce qui revient à vérifier que la passerelle est bien connectée à un réseau filaire ou Wi-Fi.

I.1. Passerelle « The Things Indoor Gateway »

Vous avez à disposition plusieurs passerelles de modèle « The Things Indoor Gateway » (que l'on appellera TTIG désormais). C'est une passerelle **LoRa-Wi-Fi** conçue par The Things Network.

Nous le verrons plus tard, mais The Things Network est un réseau communautaire et open-source qui maintient des serveurs LoRaWAN (*Network Server* et *Application Server*). Toutes les spécifications techniques de la passerelle sont fournies par le revendeur Adafruit⁴.



La documentation complète de la passerelle est sur le site The Things Stack : <https://www.thethingsindustries.com/docs/gateways/models/thethingsindoorgateway/>

Commençons par relever toutes les informations de la passerelle. Ceci correspond aux points 2 et 3 de la section « Prerequisites ».

- Gateway EUI : _____ (16 caractères)
- Wi-Fi MAC : _____
- Wi-Fi password : _____

Plutôt que de traiter toute cette documentation (dont l'objectif est d'associer la passerelle au *Network Server*), nous décidons de nous concentrer sur la partie « New Gateways » qui indique comment relier une passerelle TTIG à un réseau Wi-Fi. Suivez les instructions et connectez la passerelle au réseau Wi-Fi dont les paramètres vous sont fournis par l'enseignant.

Si sa configuration est correcte, la LED de la passerelle est soit au vert clignotant (0.25 s, connecté au Wi-Fi) soit au vert fixe (connecté au Wi-Fi et au *LoRaWAN Network Server*). Vous pouvez consulter la section « LED States » pour identifier d'éventuels problèmes.

Note : les connexions suivantes doivent être autorisées par le pare-feu de l'infra réseau :

- IPv4 TCP <cluster>.cloud.thethings.network 443 CUPS
- IPv4 TCP <cluster>.cloud.thethings.network 8887 LNS
- IPv4 TCP rjs.sm.tc 9191 Root CUPS
- IPv4 UDP your DNS Server(s) 53 DNS

⁴ <https://www.adafruit.com/product/4345>

I.2. Passerelle « The Things Gateway »

Nous avons également une autre passerelle conçue par The Things Network, la passerelle « The Things Gateway »⁵. Celle-ci est une passerelle **LoRa-Ethernet** et **LoRa-Wi-Fi** (par opposition à la TTIG qui est une passerelle LoRa-Wi-Fi uniquement). Elle n'a normalement pas vocation à être manipulée, mais la démarche d'initialisation est consignée ici pour information.

Au démarrage, cette passerelle effectue une série de tests. À ce titre elle doit avoir le droit de manipuler les protocoles et ports suivants :

- HTTPS (443) sortant
- ping sur **8.8.4.4** (serveur DNS de google)
- NTP (123) sur **pool.ntp.org**
- MQTTS (8881) sur **eu1.cloud.thethings.network**

Pour information, son adresse MAC est **50:10:ec:41:28:55**.



⁵ <https://www.thethingsnetwork.org/docs/gateways/gateway/>

II. The Things Whatever

Cette section n'a pas d'intérêt technique, mais permet de comprendre l'écosystème dans lequel évoluent les serveurs vus dans les pages suivantes.



**THE THINGS
NETWORK**

Comme déjà dit dans la partie précédente, **The Things Network (TTN)** est une communauté néerlandaise ayant pour objectif de fournir une infrastructure LoRaWAN open-source. La naissance de ce projet fait écho aux protocoles LoRa et LoRaWAN qui sont eux-mêmes open-source. Dans cette infrastructure, TTN propose un *Network Server* et un *Application Server*.



**THE THINGS
INDUSTRIES**

En grandissant, The Things Network a créé la société **The Things Industries (TTI)**. C'est un modèle économique courant chez les communautés libres: le cœur de métier reste le développement d'une solution open-source (donc gratuite) en partie financée par une société de services (ici TTI) offrant du support, de la maintenance, un accès aux fonctionnalités avancées ...



**THE THINGS
STACK**

En repartant de zéro, TTI a développé un nouveau *LoRaWAN Network Server* appelé **The Things Stacks (TTS)**⁶. C'est un logiciel open-source⁷, donc utilisable gratuitement, mais qui doit être déployé sur des serveurs physiques. TTI assure le développement et le maintien de TTS et propose différents niveaux de services payants. En parallèle TTN a conservé son objectif de fournir une infrastructure LoRaWAN gratuite et open-source. C'est **The Things Stack Sandbox**⁸ (anciennement *The Things Stack Community Edition*, renommé le 21/12/2023).

Pour résumer :

- The Things Network est une communauté open-source dédiée aux réseaux LoRaWAN ;
- The Things Industries est une société à but lucratif, créée suite au projet TTN ;
- The Things Stack est un LoRaWAN Server open-source, développé par TTI ;
- TTN assure aujourd'hui le maintien de TTS Sandbox, une infrastructure LoRaWAN gratuite.

⁶ <https://www.thethingsindustries.com/docs/getting-started/the-things-stack-basics/>

⁷ <https://github.com/TheThingsNetwork/lorawan-stack>

⁸ <https://www.thethingsindustries.com/docs/reference/ttn/>

III. LoRaWAN Network Server (LNS)

Ayant connecté une passerelle LoRaWAN à Internet (via Wi-Fi ou Ethernet), il faut maintenant associer cette passerelle à un *LoRaWAN Network Server* ou LNS. Ceci se fait grâce à The Things Network.

Allez sur le site <https://www.thethingsnetwork.org/>, connectez-vous en utilisant les identifiants ci-dessous :

- login : enseignements-sate@ensicaen.fr
- password : *sateENSICAEN2024*

Une fois connecté, cliquez sur le compte en haut à droite, puis sur **Console**.



Sélectionnez ensuite le cluster **Europe 1 (eu1 - Dublin, Ireland)**.

Vous arrivez ensuite sur un tableau de bord, dans lequel les deux options sont **Go to applications** et **Go to gateways**. Le premier concerne le *LoRaWAN Application Server* (vu dans la section suivante) et le deuxième concerne le *LoRaWAN Network Server*. Cliquez sur celui-ci.

Reprenons la documentation de la passerelle The Things Indoor Gateway :
<https://www.thethingsindustries.com/docs/gateways/models/thethingsindoorgateway/>

Suivez les instructions de la section « Claiming The Things Indoor Gateway ». Après avoir cliqué sur « Claim gateway », la LED de la passerelle devrait s'agiter pendant quelques secondes à quelques minutes, ce qui correspond à la configuration de la passerelle et la connexion au LNS. La LED passera ensuite au vert fixe, indiquant que la connexion au LNS est établie.

Depuis le tableau de bord des gateways, il est possible de cliquer sur celle tout juste ajoutée afin d'en modifier quelques paramètres (comme le nom par exemple) et aussi de voir en temps-réel les données échangées.

IV. LoRaWAN Application Server (LAS)

Créons enfin le LoRaWAN Application Server.

IV.1. Création de l'Application Server

Revenez à l'écran d'accueil de votre compte TTN : <https://eu1.cloud.thethingsnetwork/>.

Cliquez sur « Go to applications », puis « + Create application ».

Remplissez les différents champs nécessaires à la création de l'application (**Application ID**, **Application name**, **Description**) et cliquez sur « Create application ».

À ce stade le LoRaWAN Application Server est prêt, il ne reste qu'à y ajouter les End devices.

IV.2. Ajout de End Devices au LAS

Les opérations qui suivent sont à répéter autant de fois qu'il y a de End Device à ajouter à l'Application Server.

Depuis l'interface de votre Application Server, cliquez sur **End devices** dans l'onglet de gauche, puis sur **+ Register end device** à droite de la nouvelle fenêtre.

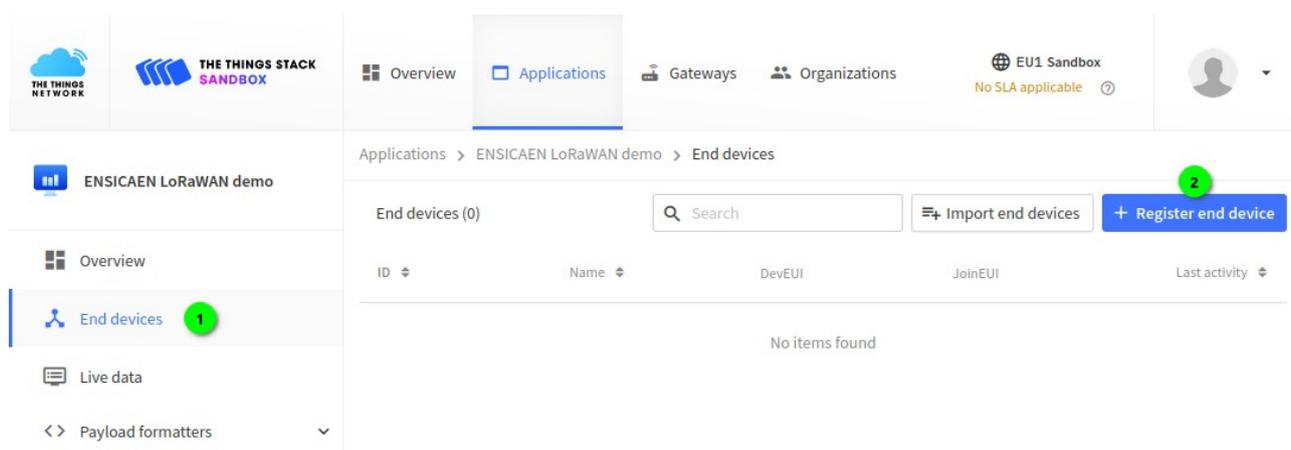


Figure 4: Ajouter un End device depuis l'interface de configuration de l'Application Server.

Sur cette nouvelle fenêtre, toutes les informations concernant le *End Device* sont à ajouter une par une (les menus déroulants apparaissent au fur et à mesure). Ci-dessous se trouvent tous les champs à renseigner.

Renseignez d'abord les informations concernant le **End device type** :

- **End device brand** : STMicroelectronics
- **Model** : NUCLEO-WL55JC1
- **Hardware version** : 1.0
- **Firmware version** : Vous devez récupérer le n° de version du firmware utilisé sur STM32. Ouvrir le fichier *.ioc, aller dans l'onglet **Project Manager** ①, puis dans l'onglet **Project** ②, puis recopier le **Firmware package version** ③ (voir Figure 5).
- **Profile (region)** : EU_863_870
- **Frequency plan** : Europe 863-870 MHz (SF9 for RX2 - recommended)

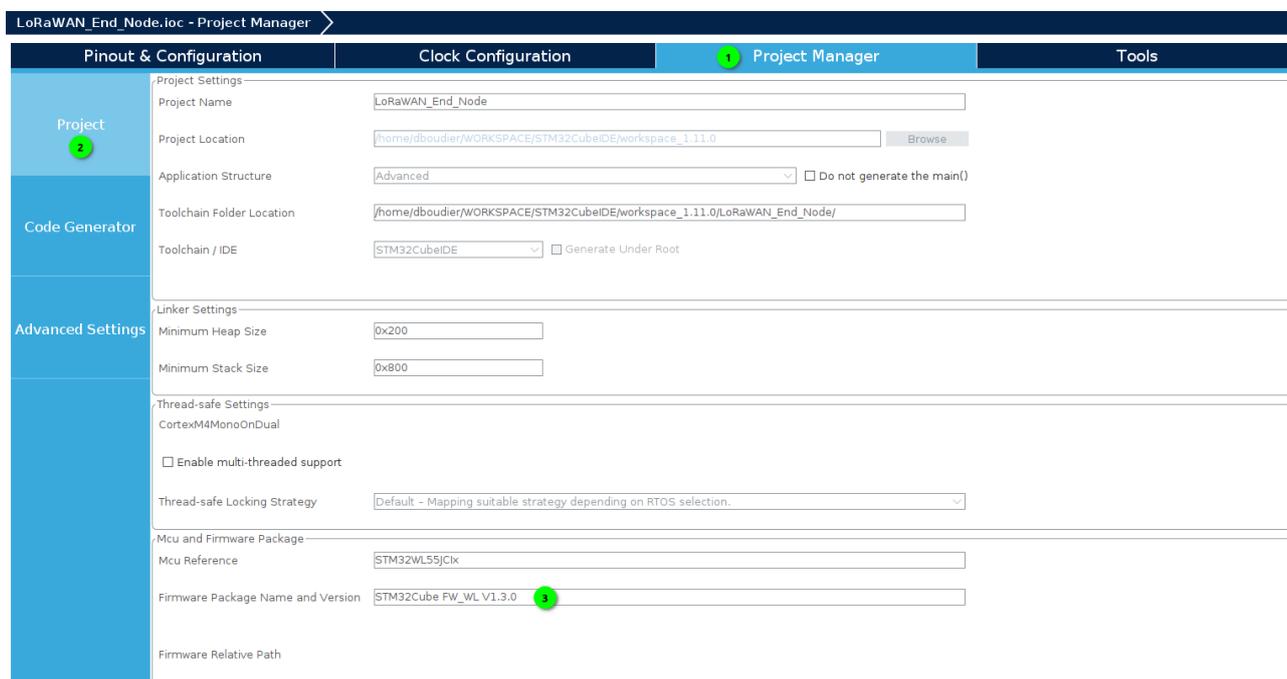


Figure 5: Récupérer le Firmware version depuis STM32CubeMX (fichier ioc)

Renseignez ensuite les informations de connexion, **Provisioning information** :

- **JoinEUI** : un identifiant 8 octets à choisir = _____⁹
- **DevEUI** : l'identifiant 8 octets indiqué sur le MCU STM32WL55JC
- **AppKey** : une clé 16 octets à choisir = _____¹⁰
- **End device ID** : un identifiant du *End Device*, pour le LAS (laisser par défaut).

Au final vous obtiendrez un résultat proche de celui présenté sur la Figure 6 page 17.

⁹ Conservez cet identifiant, il sera à inscrire dans le firmware du *End Device*.

¹⁰ Conservez cet identifiant, il sera à inscrire dans le firmware du *End Device*.

Register end device

Does your end device have a LoRaWAN® Device Identification QR Code? Scan it to speed up onboarding.

End device type

Input method

- Select the end device in the LoRaWAN Device Repository
- Enter end device specifics manually

End device brand *
 Model *
 Hardware Ver. *
 Firmware Ver. *
 Profile (Region) *

NUCLEO-WL55JC1
 LoRaWAN Specification 1.0.4, RP002 Regional Parameters 1.0.1, Over the air activation (OTAA), Class A



The STMicroelectronics NUCLEO-WL55JC1 is a development board based on the STM32WL55JC microcontroller that comes with a built-in temperature sensor and a light sensor. The NUCLEO-WL55JC1 board supports a wide choice of Integrated Development Environments (IDE) and an expansion connector for Arduino UNO V3. Suitable for rapid prototyping of LoRaWAN® end nodes by adding sensors and actuators.

[Product website](#) | [Data sheet](#)

Frequency plan *

Provisioning information

JoinEUI *

This end device can be registered on the network

DevEUI * 0/50 used

AppKey *

End device ID *

This value is automatically prefilled using the DevEUI

After registration

- View registered end device
- Register another end device of this type

Figure 6: Ajout d'un End Device au LoRaWAN Application Server.

PARTIE 4

RETOUR AU END DEVICE

I. Connecter le End Device au LAS

Une fois l'infrastructure du réseau LoRaWAN opérationnelle (Gateway, Network Server, Application Server), nous pouvons configurer les End Devices de sorte à ce qu'ils communiquent avec le LoRaWAN Application Server.

Retournez sur le projet STM32CubeIDE.

Ouvrez le fichier de configuration *.ioc et modifier les paramètres de connexion au serveur.

- **Static Device EUI¹¹** : laisser décoché pour que le firmware récupère le EUI en mémoire du micro-contrôleur (et qu'il ne soit pas figé)
- **App/JoinEUI** : correspond au **JoinEUI** de la note 9 page 16 ;
- **Application key** : correspond au **AppKey** de la note 10 page 16 ;
- **Network key** : correspond au **AppKey** de la note 10 page 16 ;
- **Network session key** : correspond au **AppKey** de la note 10 page 16 ;
- **Application session key** : correspond au **AppKey** de la note 10 page 16.

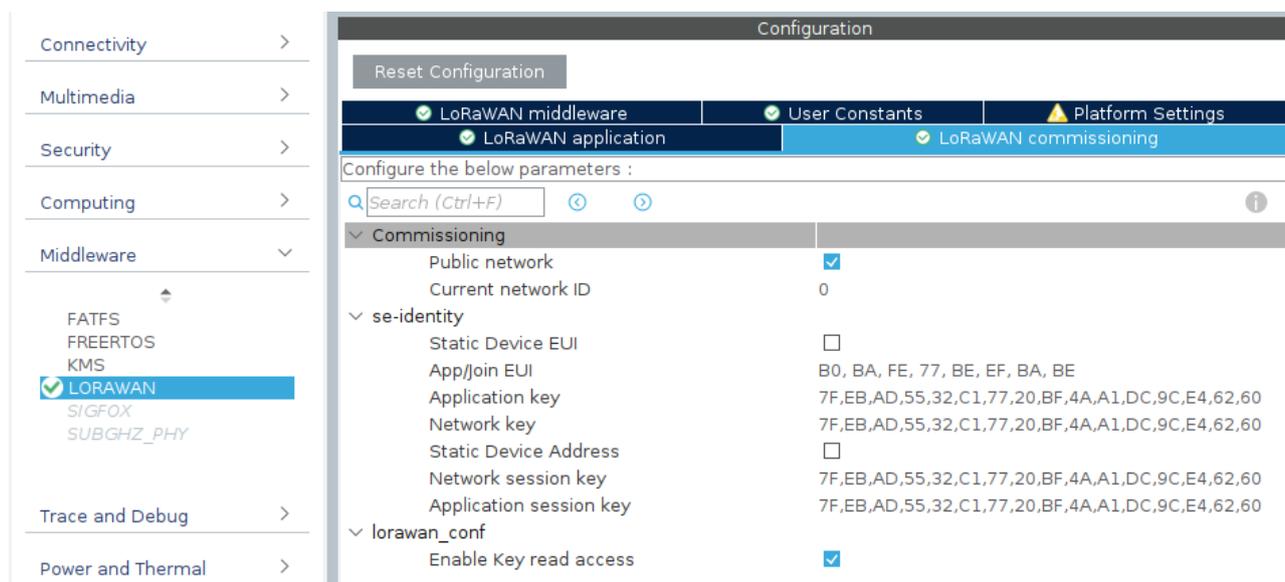


Figure 7: Paramètres d'authentification et de chiffrement auprès du LoRaWAN Application Server.

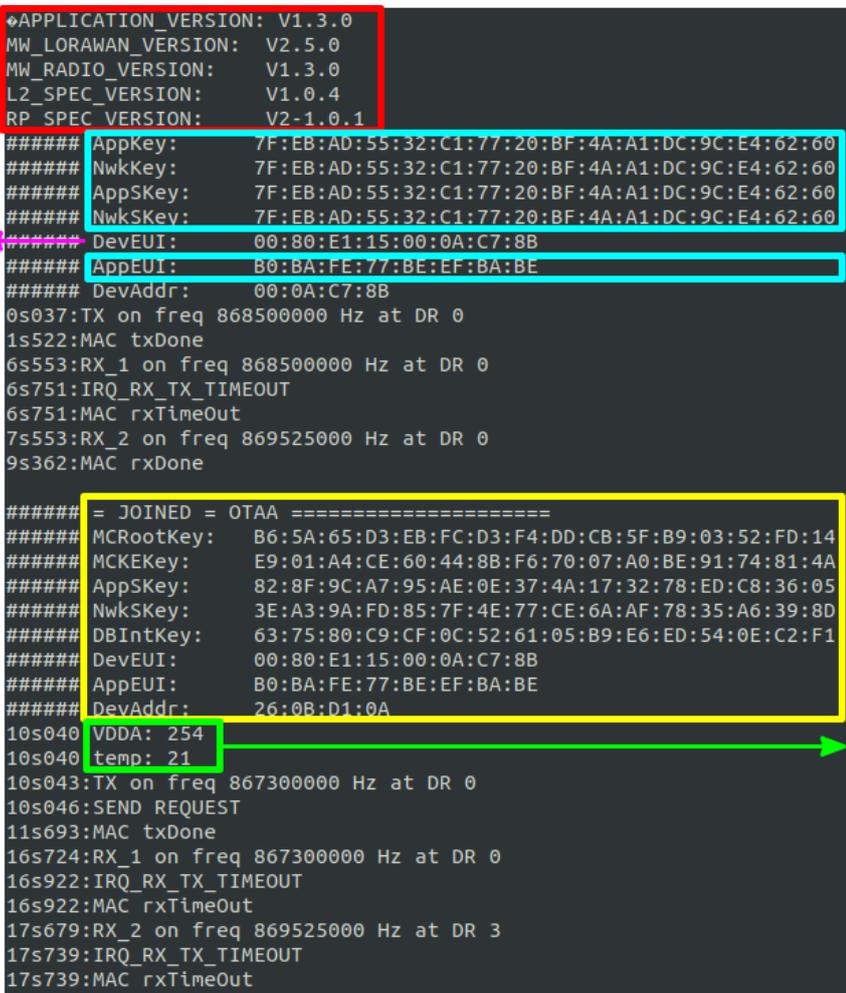
Une fois ces changements effectués, sauvegardez le fichier *.ioc, générez à nouveau le code. Attendez avant de téléverser.

Ouvrez un terminal série pour observer les logs (consignation des informations) de l'application. En même temps gardez un œil sur la fenêtre de configuration du End Device (TheThingsNetwork → Applications → cet Application Server → End Devices → ce End Device).

Téléversez le programme dans la cible et observez les affichages sur les deux interfaces de surveillance (terminal série et panneau du End Device sur le LAS).

¹¹ Cette option n'existe plus sur la version 1.3 du firmware.

Sur le terminal série, vous devriez observer ceci :



```

φAPPLICATION_VERSION: V1.3.0
MW_LORAWAN_VERSION: V2.5.0
MW_RADIO_VERSION: V1.3.0
L2_SPEC_VERSION: V1.0.4
RP_SPEC_VERSION: V2-1.0.1

##### AppKey: 7F:EB:AD:55:32:C1:77:20:BF:4A:A1:DC:9C:E4:62:60
##### NwkKey: 7F:EB:AD:55:32:C1:77:20:BF:4A:A1:DC:9C:E4:62:60
##### AppSKey: 7F:EB:AD:55:32:C1:77:20:BF:4A:A1:DC:9C:E4:62:60
##### NwksKey: 7F:EB:AD:55:32:C1:77:20:BF:4A:A1:DC:9C:E4:62:60
##### DevEUI: 00:80:E1:15:00:0A:C7:8B
##### AppEUI: B0:BA:FE:77:BE:EF:BA:BE
##### DevAddr: 00:0A:C7:8B

0s037:TX on freq 868500000 Hz at DR 0
1s522:MAC txDone
6s553:RX_1 on freq 868500000 Hz at DR 0
6s751:IRQ_RX_TX_TIMEOUT
6s751:MAC rxTimeOut
7s553:RX_2 on freq 869525000 Hz at DR 0
9s362:MAC rxDone

##### = JOINED = OTAA =====
##### MCRotKey: B6:5A:65:D3:EB:FC:D3:F4:DD:CB:5F:B9:03:52:FD:14
##### MCKEKey: E9:01:A4:CE:60:44:8B:F6:70:07:A0:BE:91:74:81:4A
##### AppSKey: 82:8F:9C:A7:95:AE:0E:37:4A:17:32:78:ED:C8:36:05
##### NwksKey: 3E:A3:9A:FD:85:7F:4E:77:CE:6A:AF:78:35:A6:39:8D
##### DBIntKey: 63:75:80:C9:CF:0C:52:61:05:B9:E6:ED:54:0E:C2:F1
##### DevEUI: 00:80:E1:15:00:0A:C7:8B
##### AppEUI: B0:BA:FE:77:BE:EF:BA:BE
##### DevAddr: 26:0B:D1:0A

10s040 VDDA: 254
10s040 temp: 21
10s043:TX on freq 867300000 Hz at DR 0
10s046:SEND REQUEST
11s693:MAC txDone
16s724:RX_1 on freq 867300000 Hz at DR 0
16s922:IRQ_RX_TX_TIMEOUT
16s922:MAC rxTimeOut
17s679:RX_2 on freq 869525000 Hz at DR 3
17s739:IRQ_RX_TX_TIMEOUT
17s739:MAC rxTimeOut
  
```

Application starting (red text pointing to the first five lines)

Printed on the MCU (pink text pointing to the DevEUI line)

Application Server parameters (blue text pointing to the AppKey, NwkKey, AppSKey, NwksKey, AppEUI, and DevAddr lines)

The End Device joined the LAS (blue text pointing to the = JOINED = OTAA section)

Data sent to the LAS (green text pointing to the VDDA and temp lines)

Sur le panneau de surveillance du End Device (sur TTN) :

weather-click-demo-sensor
ID: weather-click-demo-sensor

↑ 8 ↓ 1 • Last activity 7 seconds ago

Overview Live data Messaging Location Payload formatters General settings

General information

End device ID: weather-click-demo-sensor

Frequency plan: Europe 863-870 MHz (SF9 for RX2 - recommended)

LoRaWAN version: LoRaWAN Specification 1.0.4

Regional Parameters version: RP002 Regional Parameters 1.0.1

Created at: Jan 9, 2024 21:47:14

Live data See all activity →

- ↑ 22:04:34 Forward uplink data message DevAddr: 26 0B D1 0A Payload: { b
- ↑ 22:04:34 Successfully processed data message DevAddr: 26 0B D1 0A
- ↑ 22:04:24 Forward uplink data message DevAddr: 26 0B D1 0A Payload: { b
- ↑ 22:04:24 Successfully processed data message DevAddr: 26 0B D1 0A
- ↑ 22:04:14 Forward uplink data message DevAddr: 26 0B D1 0A Payload: { b
- ↑ 22:04:14 Successfully processed data message DevAddr: 26 0B D1 0A

weather-click-demo-sensor
ID: weather-click-demo-sensor

↑ 9 ↓ 1 • Last activity 7 seconds ago

Overview **Live data** Messaging Location Payload formatters General settings

Time Type Data preview Verbose stream Export as JSON Pause Clear

Time	Type	Data preview
↑ 22:03:54	Forward uplink data message	DevAddr: 26 0B D1 0A Payload: { battery_voltage: [-], humidity: [-], light: [-], pressure: [-], temperature: [-] } 00 27 10 15 01 F4 FE 3E 09 00 05 03 AB 00 00 ...
↑ 22:03:54	Successfully processed data message	DevAddr: 26 0B D1 0A
↓ 22:03:44	Schedule data downlink for transmissi...	DevAddr: 26 0B D1 0A Rx1 Delay: 5
↑ 22:03:44	Forward uplink data message	DevAddr: 26 0B D1 0A Payload: { battery_voltage: [-], humidity: [-], light: [-], pressure: [-], temperature: [-] } 00 27 10 15 01 F4 FE 3E 09 00 05 03 AB 00 00 ...
↑ 22:03:44	Successfully processed data message	DevAddr: 26 0B D1 0A
↓ 22:03:35	Schedule data downlink for transmissi...	DevAddr: 26 0B D1 0A Rx1 Delay: 5
↑ 22:03:35	Forward uplink data message	DevAddr: 26 0B D1 0A Payload: { battery_voltage: [-], humidity: [-], light: [-], pressure: [-], temperature: [-] } 00 27 10 16 01 F4 FE 3E 09 00 05 03 AB 00 00 ...
↑ 22:03:35	Successfully processed data message	DevAddr: 26 0B D1 0A
↓ 22:03:25	Schedule data downlink for transmissi...	Schedule on path eu1-58a0cbfffe001afa@ttn: Schedule
↓ 22:03:25	Schedule data downlink for transmissi...	DevAddr: 26 0B D1 0A Rx1 Delay: 5
✎ 22:03:25	Update end device	["activated_at"]
↑ 22:03:25	Forward uplink data message	DevAddr: 26 0B D1 0A Payload: { battery_voltage: [-], humidity: [-], light: [-], pressure: [-], temperature: [-] } 00 27 10 15 01 F4 FE 3E 09 00 05 03 AB 00 00 ...
↑ 22:03:25	Successfully processed data message	DevAddr: 26 0B D1 0A
↑ 22:03:17	Forward join-accept message	DevAddr: 26 0B D1 0A
↑ 22:03:15	Successfully processed join-request	
☞ 22:03:15	Accept join-request	DevAddr: 26 0B D1 0A
⊕ 21:47:14	Create end device	

Data sent to the LAS by the End Device

The End Device joined the LAS

II. Nanana

PARTIE 5

APPLICATION CUSTOM

I. Modifier le firmware du End Device

II. ThingSpeak et formattage du payload

III. Mise en contexte

III.1.a. Notation

-  Question attendant une réponse écrite (résultant d'une analyse des ressources)
-  Répondre en programmant / travaillant sur machine
-  Répondre en relevant une capture d'écran du terminal série ou d'un chronogramme (oscilloscope, analyseur logique, ...)

IV. Passerelle

IV.1. Préparer la passerelle

Soyons honnête

Notez ici les différents champs

- Gateway EUI
 - De la forme 58A0CBxxxxxx
 - Insérer FFFE

PARTIE 6
GLOSSAIRE

Termes classés par thématique

LSb / MSb	<i>Least/Most Significant bit</i>
LSB / MSB	<i>Least/Most Significant Byte</i>
MCU	<i>Microcontroller Unit</i>
CPU	<i>Central Processing Unit</i>
Périphérique	Composant matériel interne au MCU, conçu pour une fonction précise
TTI	
TTN	
TTS	
TTSS	
LNS	LoRaWAN Network Server
LAS	LoRaWAN Application Server
Join Server	
DevAddr	Device Address
NwkSKey	Network Session Key, clé d'authentification
AppSKey	Application Session Key, clé de chiffrement
AppKey	
EUI	Extended Unique Identifier
AppEUI/ JoinEUI	Join server EUI, Identification du Join Server utilisé (8 octets) permet de régénérer des NwkSKey et AppSKey à chaque nouvelle session
AppEUI	Application server EUI Obsolète depuis LoRaWAN v1.0.4, remplacé par JoinEUI
DevEUI	Device EUI
DevNonce	