



# INITIATION AU RESEAU LORAWAN

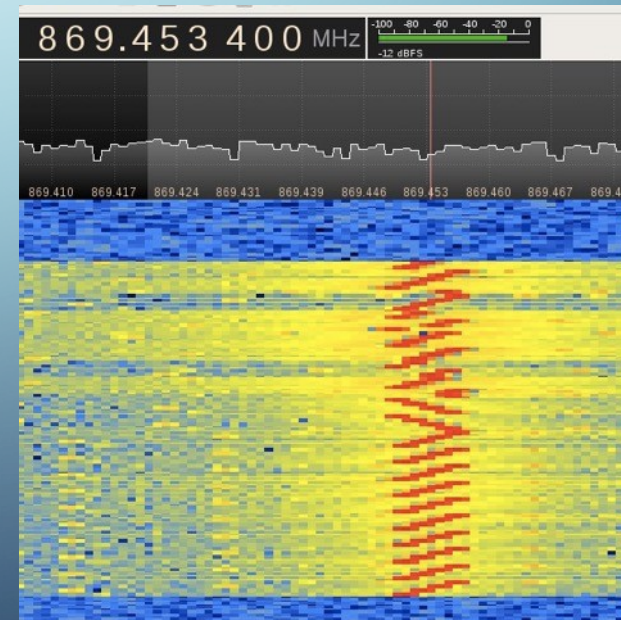
AVEC UN KIT ARDUINO LORA RADIO NODE V1.0

# PRÉSENTATION LoRaWAN™

- LoRaWan est un réseau de communication **longue distance** et **bas débit** à très basse consommation d'énergie.
- Né à GRENOBLE en 2009
- Acheté 5M\$ par Semtech en 2012
  - 1 fondeur de chip + 1 sous licence
- LoRaWan 1.0 released en 2015
- Déployé par les opérateurs télécoms locaux
  - 5 pays déployés connus
- Déployable a titre privé
  - 1 réseau mondial ouvert (TTN)

# PRÉSENTATION LoRaWAN™

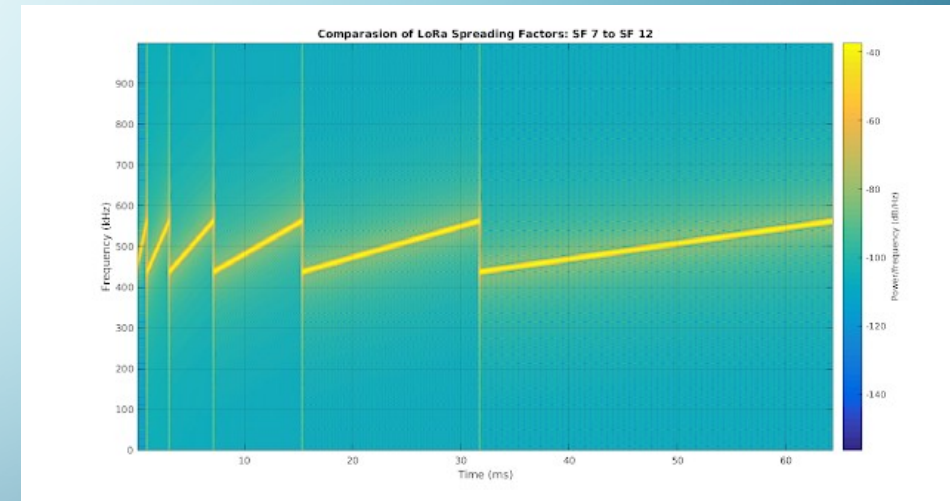
- Il repose sur quelques principes:
  - Un mode partiellement déconnecté
  - Une communication par étalement de spectre
  - L'utilisation de fréquences libres (868MHz - EU)
  - Le respect d'un « Duty Cycle » (1%)
- Communications possibles sur 10km – 15km environ



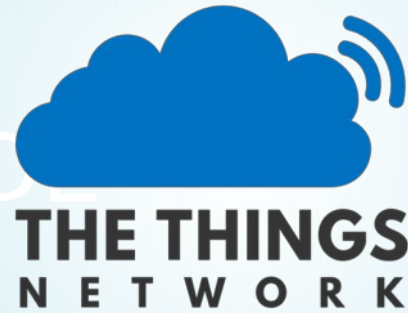


# PRÉSENTATION LoRaWAN™

- Quelques paramètres:
  - Des canaux de communication
    - 3 standards définis par la norme (permettant de se connecter)
    - 5 autres pouvant être ajoutés
  - Une vitesse de communication (SF) de 250b/s à 5400b/s
  - Des identifiants / clefs :
    - DEV-EUI – Un identifiant de l'objet (adresse MAC qui doit être unique sur le réseau)
    - APP-EUI – Un identifiant du type de l'objet (application) pour diriger le trafic.
    - APP-KEY – Une clé utilisée dans le chiffrement et la signature des données transmises.

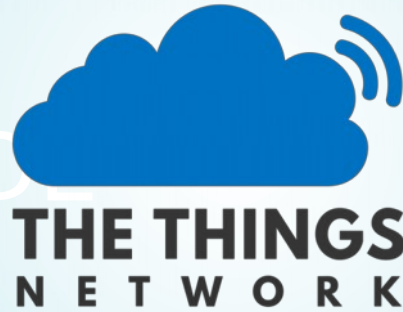


# PRÉSENTATION D



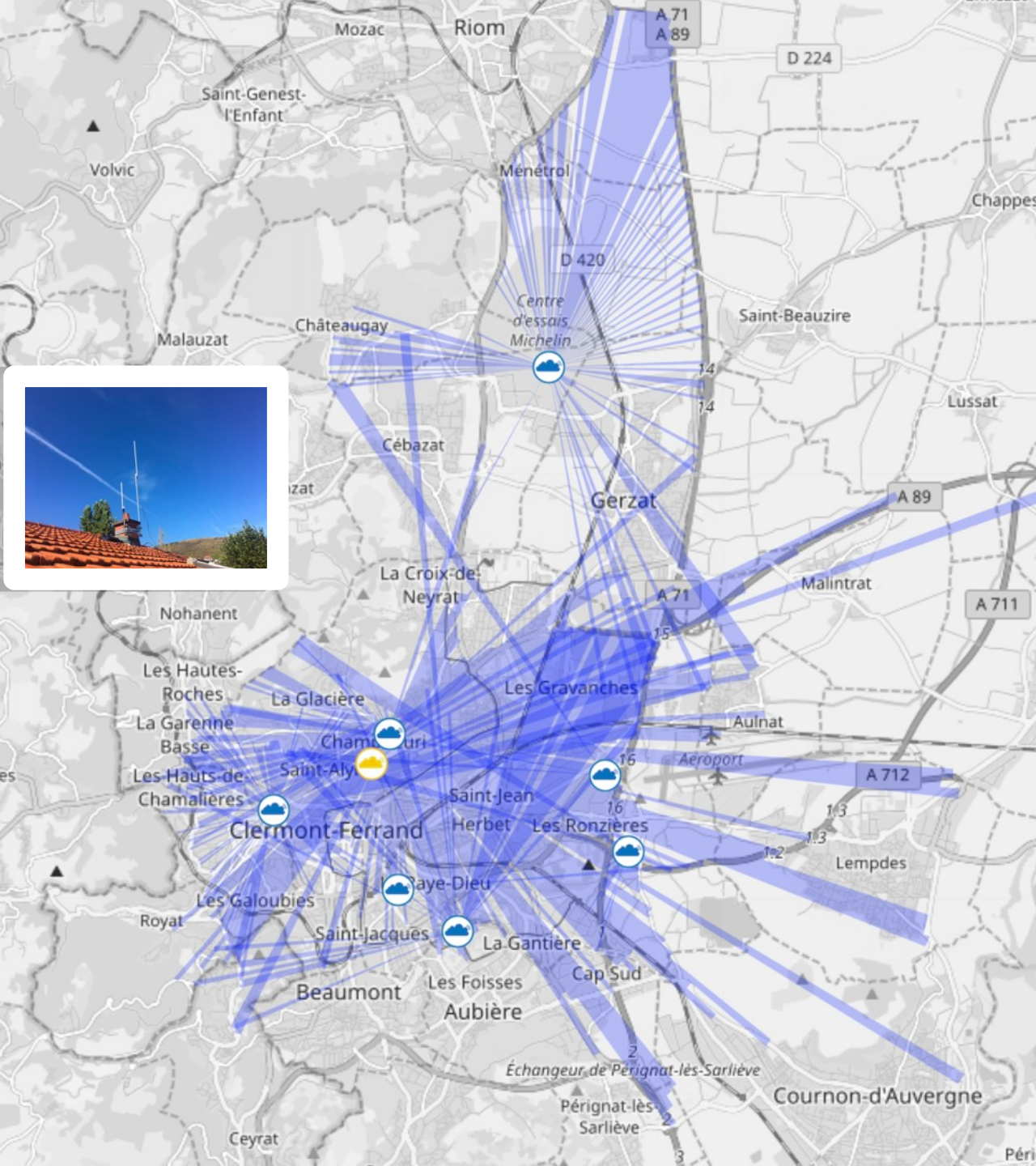
- LoRaWan fonctionne avec un point de concentration des données que l'on appelle un cœur de réseau. TheThingsNetwork est un cœur de réseau Open Source dont les antennes sont Crowd Sourced.

# PRÉSENTATION D



- Il y a donc d'un côté une « organisation » TTN qui fournit l'infrastructure de serveurs centrale et les interfaces d'administration
- Et de l'autre côté des **initiateurs** qui déploient des antennes pour étendre le réseau mondial.





# CLERMONT-FERRAND

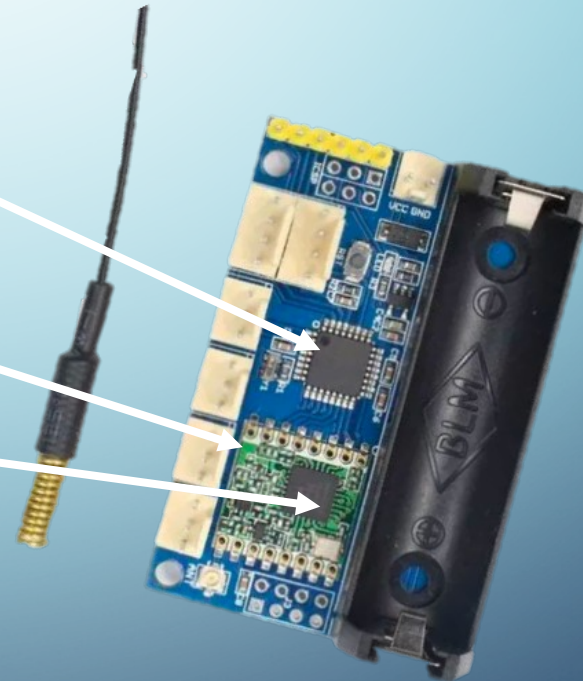
- Plusieurs antennes déployées sur des toits pour offrir un couverture large. Déploiement toujours en cours.
- Un réseau facile à étendre
  - Installation toit : environ 500€
  - Installation intérieur: environ 70€



# PRÉSENTATION D



- LoRa Radio Node v1.0 est un **Arduino**
- Associé à un **module RFM95**
- Qui lui-même intègre un **SX1276** de Semtech

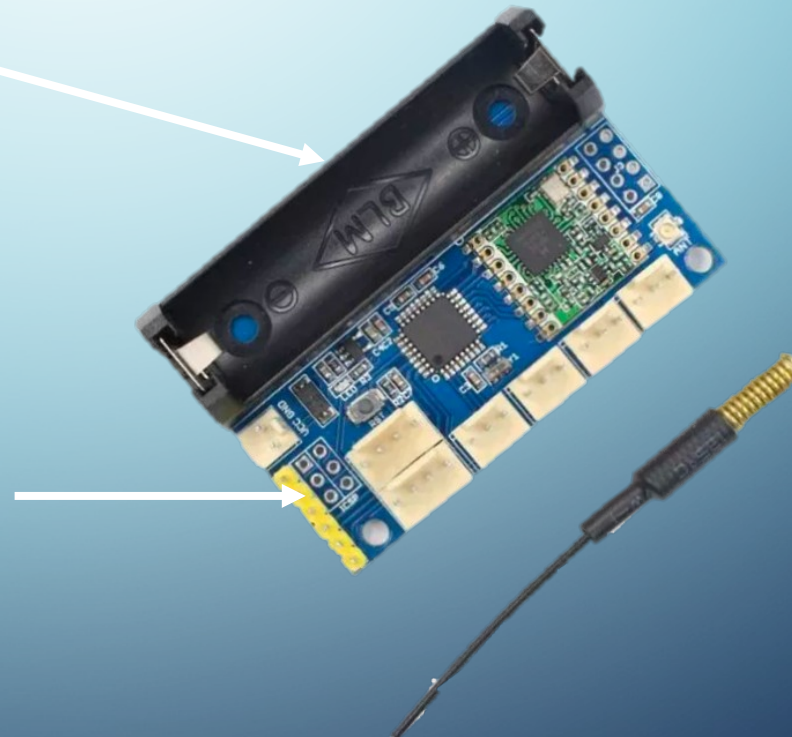




# PRÉSENTATION D



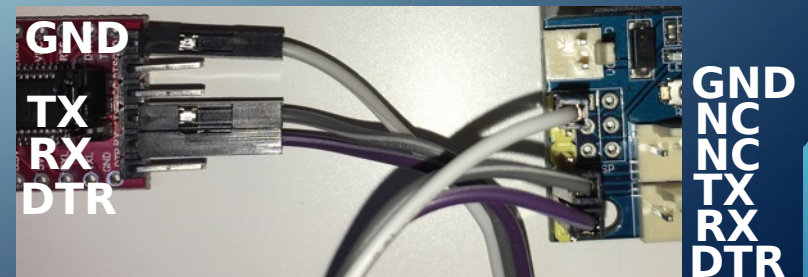
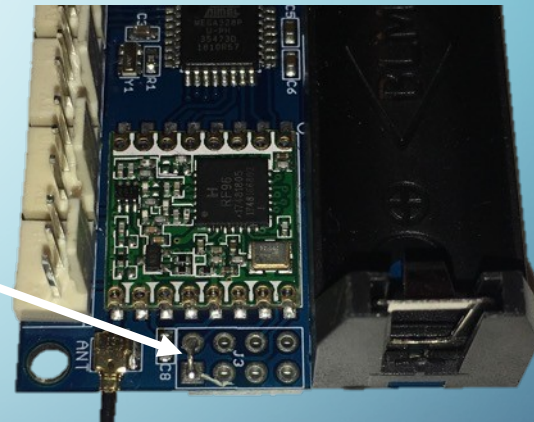
- Il est alimenté par une pile fournissant entre 3V et 3,7V capable de délivrer  $> 150\text{mA}$  (Lithium, LiFe4, Li-Ion)
- Il se programme avec un câble série 3.3V depuis l'IDE Arduino. Il faut choisir « Arduino Pro ou Arduino Mini » et le sous type 3.3V / 8MHz



# PRÉSENTATION D



- Pour l'utiliser en LoRaWan il faut connecter le signal DIO1 (interruption utilisée par le SX1276) sur la broche D5
- Pour la programmation nous allons connecter le FTDI (interface USB-UART) sur le connecteur jaune comme suit. (Rx et Tx sont croisés). Ajouter DTR qui sur la photo n'est pas visible.







# INSTALLATION DE

- Installer l'IDE gratuite si ce n'est pas fait depuis :

<https://www.arduino.cc/en/Main/Software>

Le tutoriel utilise la version 1.8.9 (Mai 2019)

- Une fois lancée, nous devons configurer notre carte dans le menu « **Outils** » comme indiqué ici:

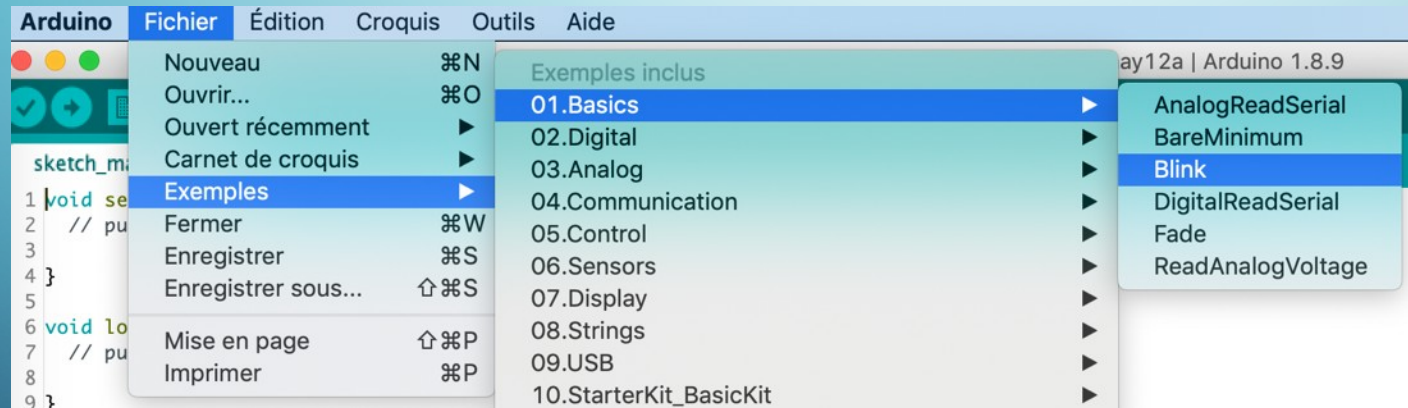


Type de carte: "Arduino Pro or Pro Mini"  
Processeur: "ATmega328P (3.3V, 8 MHz)"  
Port



# INSTALLATION DE

- Nous allons compiler un premier exemple pour vérifier que la carte fonctionne : Blink



- Ensuite il faut le téléverser (vérifiez éventuellement que le port choisi soit le bon)

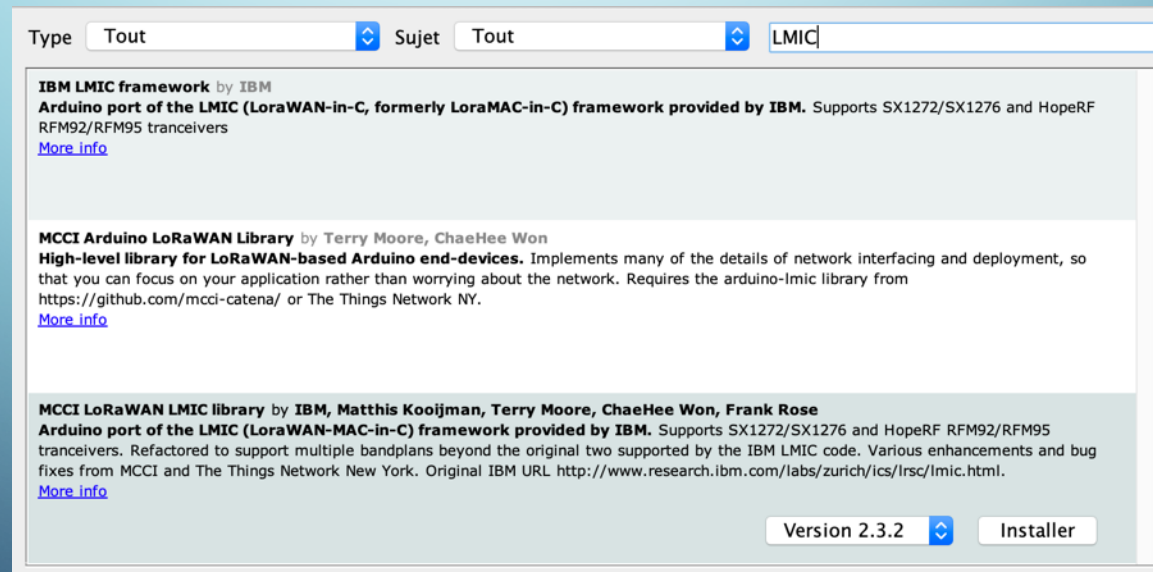




# INSTALLATION DE LA LIB

- Nous ajoutons cette librairie avec  
« Outil » « Gérer les bibliothèques »

1- Chercher LMIC



2- Choisir MCCI...

Version 2.3.2

# INSTALLATION DE LA LIB

- Maintenant nous devons faire une petite configuration de cette bibliothèque pour choisir la Zone dans laquelle nous allons émettre (Europe - 868MHz).
- Editer le fichier dans

**Arduino/library/MCCI\_LoRaWAN\_LMIC\_library/project\_config/lmic\_project\_config.h**

```
// project-specific definitions
#define CFG_eu868 1
//#define CFG_us915 1
//#define CFG_au921 1
//#define CFG_as923 1
//#define LMIC_COUNTRY_CODE LMIC_COUNTRY_CODE_JP
//#define CFG_in866 1
#define CFG_sx1276_radio 1
//#define LMIC_USE_INTERRUPTS
```



# INSTALLATION DE LA LIB LMIC

- Nous allons maintenant partir d'un projet permettant de simplifier l'utilisation de cette bibliothèque:

1- Télécharger

<https://github.com/disk91/arduino-lora-radio-node-demo/archive/master.zip>


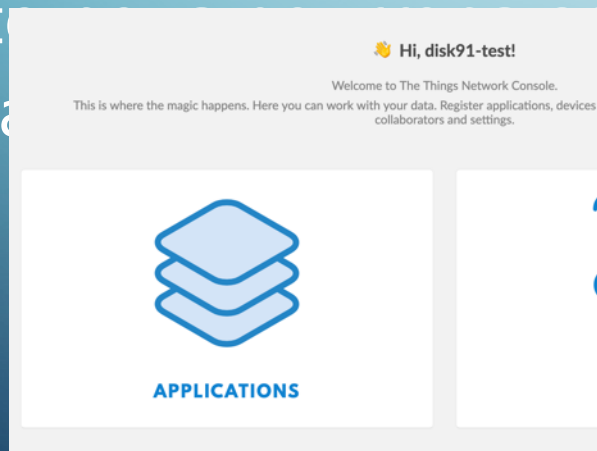
2 - décompresser le fichier

3 - Ouvrir Lora-Node.ino depuis l'IDE Arduino

# CREATION DES IDENTIFIANTS



- Nous devons créer un compte sur <https://console.thethingsnetwork.org>
- Ensuite, nous allons créer une application objet:



### CREATE AN ACCOUNT

Create an account for The Things Network and start exploring the world of Internet of Things with us.

**USERNAME**  
This will be your username — pick a good one because you will not be able to change it.

**EMAIL ADDRESS**  
You will receive a confirmation email, as well as occasional account related emails. If this email address is managed by a third party (such as for corporate email addresses), this third party might block emails coming from The Things Network. This email address is not public.

**PASSWORD**  
Use at least 6 characters.

Create account



# CREATION DES IDENTIFIANTS



- Nous créons en premier un application

**ADD APPLICATION**

**Application ID**  
The unique identifier of your application on the network

app-accolab-1 ✓

**Description**  
A human readable description of your new app

Workshop accolab ✓

**Application EUI**  
An application EUI will be issued for The Things Network block for convenience, you can add your own in the application settings page.

EUI issued by The Things Network

**Handler registration**  
Select the handler you want to register this application to

ttn-handler-eu ✓

Cancel Add application

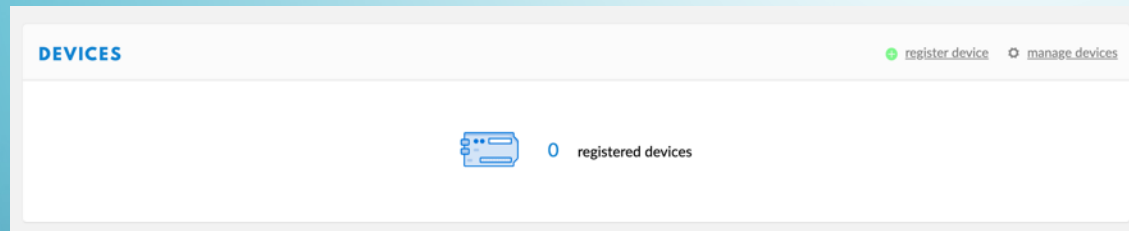
APP-ID : app-accolab-1

APP-EUI : sera généré

# CREATION DES IDENTIFIANTS



- Nous créons ensuite un objet



**REGISTER DEVICE** [bulk import devices](#)

**Device ID**  
This is the unique identifier for the device in this app. The device ID will be immutable.

device-1 ✓

**Device EUI**  
The device EUI is the unique identifier for this device on the network. You can change the EUI later.

✎ [Cliquer ici](#) this field will be generated

**App Key**  
The App Key will be used to secure the communication between you device and the network.

✎ this field will be generated

**App EUI**

70 B3 D5 7E D0 01 BF FF



# CREATION DES IDENTIFIANTS



- Nous avons maintenant les ID nécessaire à connecter notre objet

Device EUI	<>	↔	00 8B 40 AC [REDACTED]	📋
Application EUI	<>	↔	70 B3 D5 7E [REDACTED]	📋
App Key	<>	↔	👁 B5 5C 7F 52 26 54 B5 73 11 12 DF 32 [REDACTED]	📋

- Il faut mettre ces données dans le fichier loraWan.cpp de notre projet Arduino. Attention à l'ordre des octets APP-EUI et DEV-EUI sont inscrits à l'envers par rapport à ce que vous voyez ici.



# COMMUNICATION SUR

- Maintenant nous pouvons téléverser notre sketch sur l'Arduino.
- Ouvrir le Console (9600 Bps) pour vérifier le déroulé.

Overview **Data** Settings

### APPLICATION DATA

|| pause clear

Filters: uplink downlink activation ack error

	time	counter	port	
▲	17:43:36	36	1	payload: 00 01 02 03 04 05 06 07 08 09 0A 0B bit: [1,0,0,0,0,0,0,0] value1: 0 value2: 2
▲	17:43:14	35	1	payload: 00 01 02 03 04 05 06 07 08 09 0A 0B bit: [1,0,0,0,0,0,0,0] value1: 0 value2: 2
▲	17:42:51	34	1	payload: 00 01 02 03 04 05 06 07 08 09 0A 0B bit: [1,0,0,0,0,0,0,0] value1: 0 value2: 2