

TP LoRaWAN

2e semestre 2021

Contents

1	Objectif	1
2	Partie 1 : Compréhension	1
3	Partie 2 : Accès serveur	2
4	Partie 3 : Premier noeud	3
4.1	Préparation environnement	3
4.2	Mode ABP	5
4.3	Code	5
5	Partie 4 : Second noeud	5
5.1	Préparation environnement	5
5.2	Mode OTAA	6
5.3	Code	6

1 Objectif

Le but de ce TP est de comprendre et utiliser la technologie LoRaWAN dans un scénario plausible. Il s'agit ici de contrôler l'ouverture d'une porte en recensant au serveur les passages. Un premier capteur à l'entrée de la porte comprendra un lecteur RFID. Si l'ID scanné est autorisé ou non, celui-ci enverra un log au serveur comme tentative d'ouverture. De l'autre côté de la porte se situe un capteur de proximité afin de détecter tout passage et le recenser également au serveur. En théorie, le capteur n'est pas en mesure de détecter si le passage est entrant ou sortant, nous enverrons par défaut tout passage. Consultez la documentation jointe si vous souhaitez de plus amples explications.

Déroulé :

- Comprendre la technologie
- Répondre aux questions
- Accéder à la passerelle
- Créer application
- Lier premier noeud (ABP)
- Coder son comportement
- Lier second noeud (OTAA)
- Coder son comportement

2 Partie 1 : Compréhension

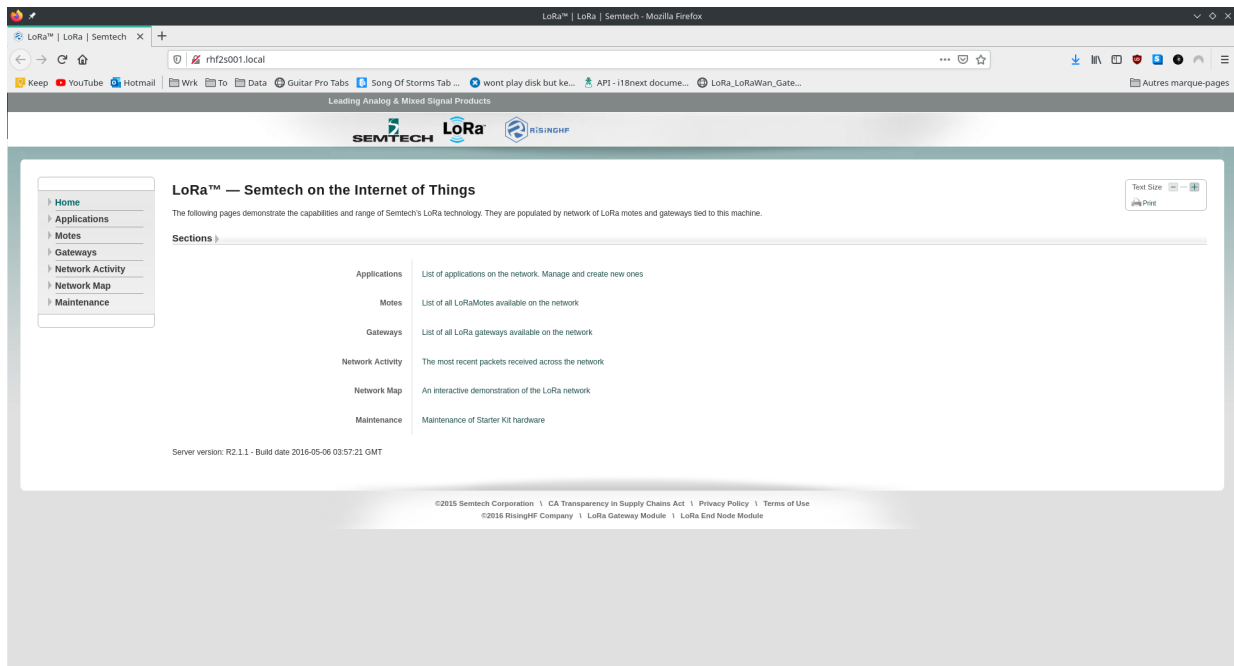
Documentez vous dans un premier temps sur la technologie puis répondez aux questions. Vous pouvez débuter vos recherches [ici](#) et [ici](#).

- Qu'est-ce que LoRaWAN ?
- Quelle est la différence entre LoRa et LoRaWAN ?
- Expliquer : Gateway, Noeud, Application, TTN
- Expliquez la différence entre ABP OTAA
- Quelle est la fréquence autorisée en Europe ?
- Les données sont-elles chiffrées ?

3 Partie 2 : Accès serveur

La gateway et le serveur applicatif ne forment qu'un dans ce TP. Il s'agit d'une Raspberry Pi couplée à un module LoRa. Pour utiliser vos noeuds vous devrez créer une application et associer ses noeuds. Le premier noeud sera recensé en ABP et le second en OTAA.

Accédez au site en renseignant l'url **rhf2s001.local**, notez que vous devez impérativement être sous le même réseau que la gateway.

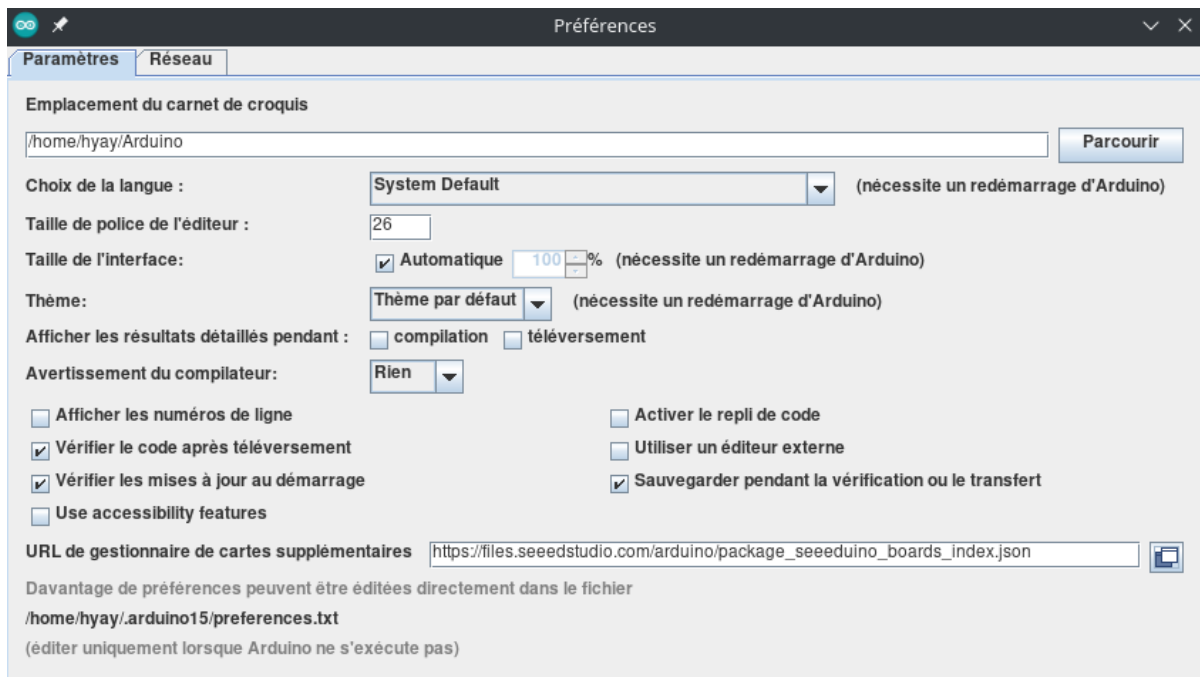


Vous pouvez voir le trafic en temps réel dans l'onglet **Network Activity** en sélectionnant votre application, puis le noeud de votre choix si vous souhaitez un résultat plus restreint. Sachez que les résultats arrivent en hexadécimal ou en plain text selon votre méthode d'envoi.

4 Partie 3 : Premier noeud

4.1 Préparation environnement

Le premier noeud sera la Seeduino LoRaWAN, c'est une Arduino avec une puce LoRa intégrée. Lancez le logiciel Arduino. Nous aurons besoin d'ajouter les drivers de la carte. Pour ce faire naviguez dans Fichiers puis Préférences et ajoutez l'url https://files.seeedstudio.com/arduino/package_seeeduino_boards_index.json dans le champs concerné. Si vous y avez déjà ajouté des lignes, faites simplement un retour à la ligne avant de l'insérer.



Le lien désormais renseigné, naviguez dans le gestionnaire de cartes dans l'onglet Outils et installez les drivers **Seeed SAMD Boards**. S'il n'apparaît pas, assurez vous que l'url que vous avez rentré est correct. Prenez soin de bien renseigner le port (détecté automatiquement) dans l'onglet Outils.

```
#include <SoftwareSerial.h>

void setup() {
  Serial1.begin(9600);
  SerialUSB.begin(115200);
}

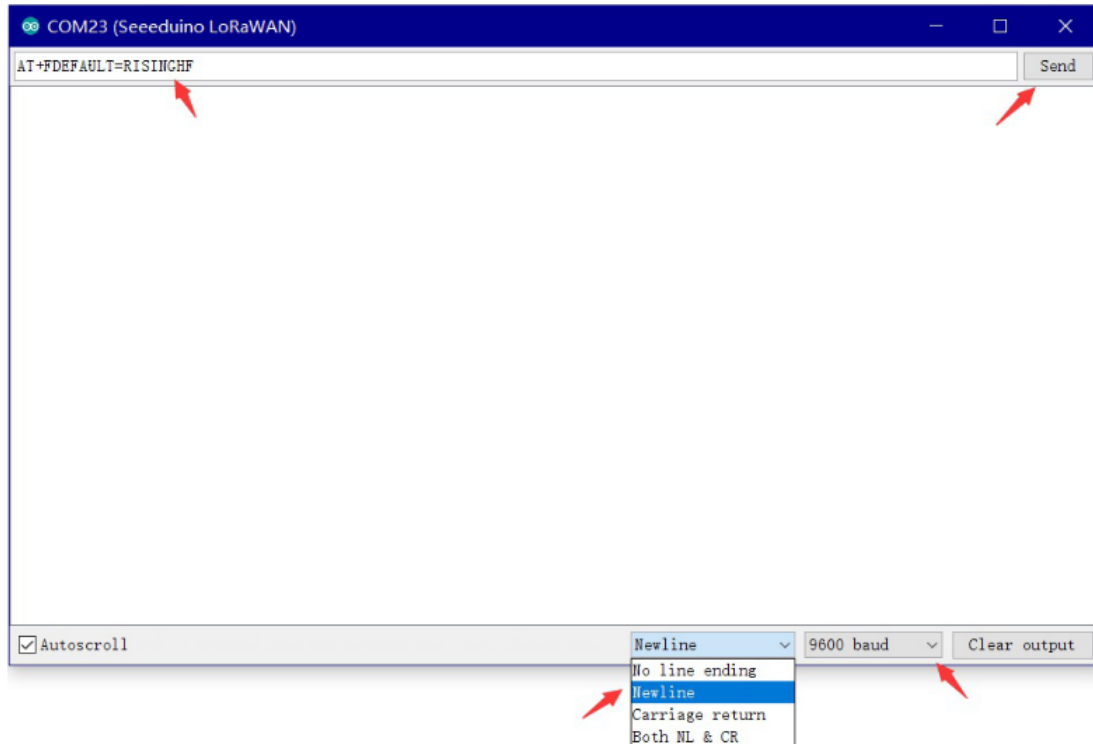
void loop() {
  while(Serial1.available()){
    SerialUSB.write(Serial1.read());
  }

  while(SerialUSB.available()){
    Serial1.write(SerialUSB.read());
  }
}
```

Sélectionnez ensuite la carte dans Outils, Type de carte, Seeeduino LoRAWAN. Compilez puis téléversez le programme. Si vous rencontrez des soucis sur Linux, il s'agit sûrement des permissions, référez vous à [la documentation ici](#).

Enfin ouvrez le moniteur de série (la loupe en haut à droite), cochez **nouvelle ligne** et insérez les lignes suivantes :

```
AT+FDEFAULT=RISINGHF
AT+DR=EU86
```



Vous avez, dans un premier temps, besoin de lier le noeud à sa passerelle. Pour ce faire nous devons renseigner les identifiant du noeud sur cette page. Dans le moniteur de série tapez **AT+ID**, vous obtiendrez les identifiant de votre carte Seeeduino LoRaWAN de manière similaire :

```
+ID: DevAddr, 00:81:12:6D
+ID: DevEui, 47:DB:55:80:00:29:00:1F
+ID: AppEui, 52:69:73:69:6E:67:48:46
```

4.2 Mode ABP

Sur le serveur, cliquez sur l'onglet **Application** et créez en une. Renseignez un nom, un propriétaire et l'AppEUI. Rendez vous dans votre application et dans Configure notes. C'est ici que vous attacherez des appareils à votre application.

Renseignez dans la partie **Personalized notes** :

- DevEui AT+ID command
- DevAddr: AT+ID command
- NWKSKEYDefault value 2B7E151628AED2A6ABF7158809CF4F3C
- APPSKEYDefault value 2B7E151628AED2A6ABF7158809CF4F3C

Votre noeud est désormais lié à votre application.

4.3 Code

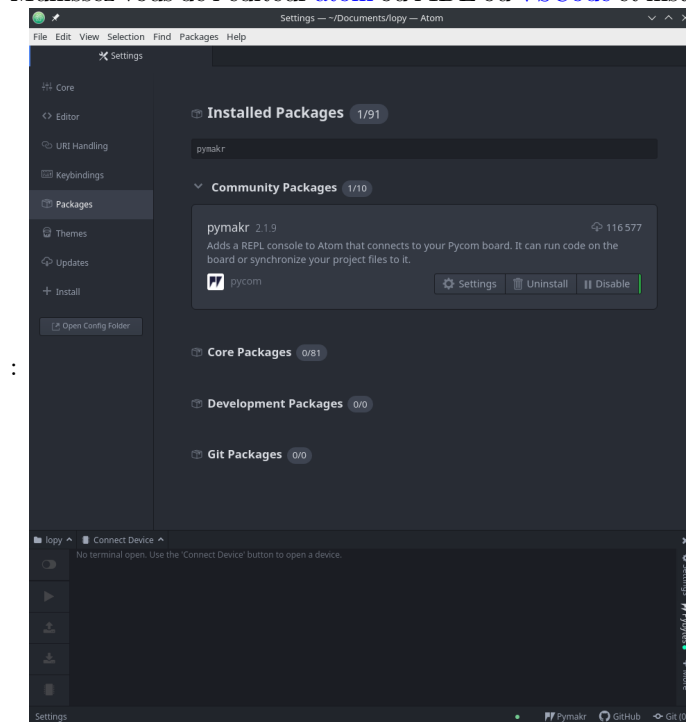
En installant la carte, vous bénéficiez d'exemples fournis pour vous aider à comprendre. Vous devez dans votre cas envoyer un log lorsqu'une carte RFID (votre carte étudiante) est scannée. Et pourquoi pas allumer une led en tant qu'indicateur.

- Seeeuino : [Datasheet](#)
- LED : [Code](#)

5 Partie 4 : Second noeud

5.1 Préparation environnement

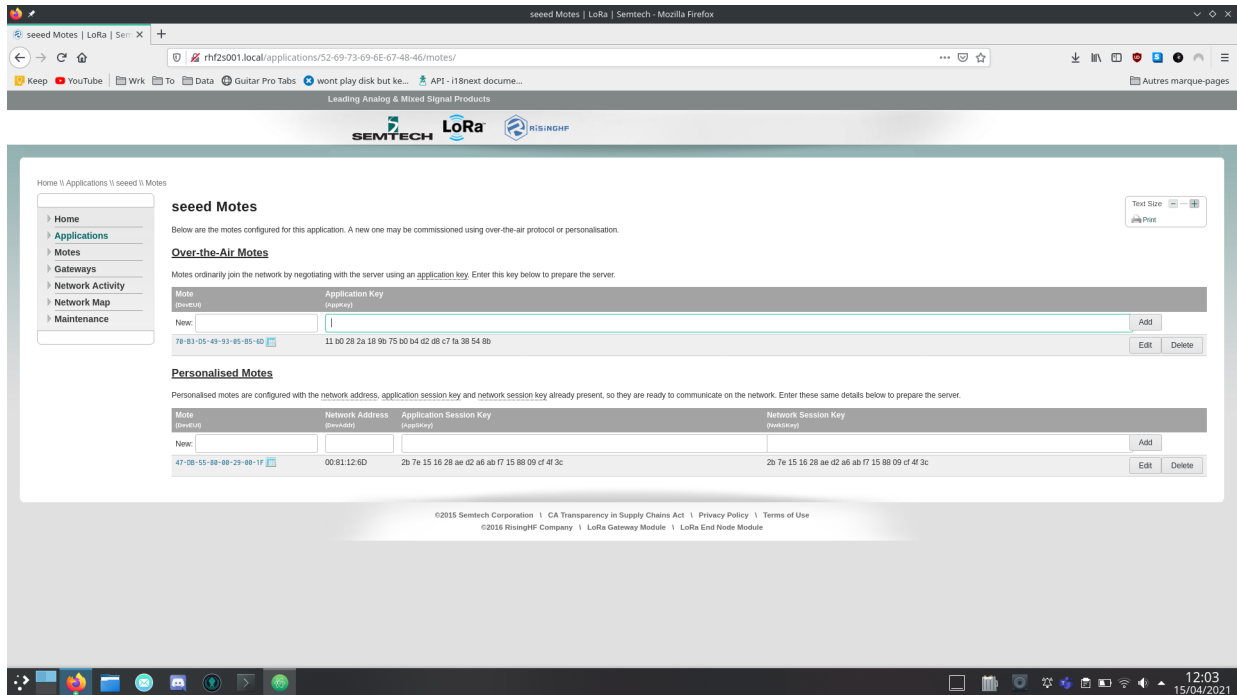
Munissez vous de l'éditeur [atom](#) ou l'IDE ou [VSCode](#) et installez le package **pymakr**. Exemple pour Atom



Vous devriez voir une console apparaître dans la partie inférieure de l'écran. Fixez ensuite le lopy4 sur son socle et branchez le à votre ordinateur. La connexion se fait automatiquement. Vous pouvez exécuter le script que vous visionnez actuellement en cliquant sur le bouton triangulaire ou téléverser le dossier dans lequel vous travaillez avec le bouton upload. Notez que par défaut c'est le script **main.py** qui s'exécute, et doit faire appel à vos autres scripts depuis lui même. Si vous utilisez un linter ou que le programme est capricieux, c'est que vous n'avez pas installé python ou les librairies qu'il utilise.

5.2 Mode OTAA

Comme le mode ABP, vous devez renseigner quelques informations dans configure notes dans votre application. Renseignez dans la partie **Over-the-Air Notes** le DevEUI que vous devez récupérer depuis un script et une clef AppKey à définir. Vous devriez obtenir un résultat similaire après avoir recenser vos deux noeuds :



Ici le minimum requis pour votre code main.py :

```
import network
from network import LoRa
import ubinascii

lora = LoRa(mode=LoRa.LORAWAN, region=LoRa.EU868)
# DevEUI est obtenu via lora.mac()
```

5.3 Code

Votre objectif est de détecter un rapprochement, si c'est le cas, allumer la led et envoyer un log.

- Lopy4 : [Datasheet](#)
- LED : [RGB code](#)
- LoRa : [OTAA](#)
- Useful : [functions](#)

