Créer une application avec Lora32u4 pour The Things Network

Programmez les cartes Lora32u4 pour créer une application IoT enregistrée sur The Things Network

Difficulté **Moyen** | Durée **1 heure(s)** | Catégories **Électronique** | Coût **14 EUR (€)**

Sommaire

Introduction

The Things Network est une initiative communautaire dans le domaine de l'internet des objets (ou IoT) permettant de raccorder des passerelles ou d'utiliser des passerelles mises en place par ses membres pour transmettre par radio des informations issues de capteurs.
La spécificité de la liaison radio est d'utiliser la modulation LoRa, des transmissions longues distances (plusieurs kilomètres) avec une faible consommation d'énergie, et qui peut donc fonctionner longtemps sur batteries dans des zones non couvertes par wifi ou bluetooth.
Les messages envoyées par radio sont au format LoRaWan, qui assure l'intégrité du message ainsi que son cryptage.
La carte LoRa32u4 réuni sur un seul support :

- un émetteur-récepteur LoRa,
- un microcontroleur Atmel 32u4 ,
- un connecteur pour batterie LiPo 3,7V
- un connecteur USB
- un circuit de gestion de charge de la pile par l'alimentation USB
- des broches d'entrée-sortie permettant de raccorder toute sorte de capteur
- la compatibilité avec l'environnement de programmation Arduino

Bref, c'est un moyen très économique de créer son propre capteur IoT à raccorder sur The Thing Network

> La procédure décrite est pour une programmation sous windows

> Pour que les opérations décrites ci-dessous donnent un résultat, il faut bien entendu que vous soyez à portée d'une passerelle The Thing Network

Matériaux

- une carte LoRa32u4 II (choisir la fréquence de 868MHz)

Optionnel :

- une batterie LiPo 3,7V avec connecteur JST PH 2mm
- une antenne 868 MHz avec connecteur u.fl (sinon un morceau de fil électrique de 8cm suffit)

Outils

- un câble USB vers micro USB
- un fer à souder suivant les versions de la carte

Étape 1 - Installer les pilotes et bibliothèques

1 - Télécharger les fichiers Driver windows et Arduino Hardware folder sur la page BSFrance
2 - Pour les drivers, il suffit de dézipper et de cliquer sur adafruit_drivers.exe. Parmi la liste des drivers proposés, il faut choisir Feather32u4
3 - Bon, là, normalement, il est possible de brancher la carte sur le port USB de l'ordinateur.

> Il est possible que vous ayez le message que l'installation du pilote n'a pas été possible. Pas de panique, on y reviendra

3 - Pour les fichiers Arduino, il faut le dézipper dans le répertoire Mes Documents/Arduino/hardware (ce qui est le répertoire par défaut de l'installation de l'environnement Arduino, mais peut-être différent suivant votre installation. si le sous-répertoire hardware n'existe pas, créez le. Cette bibliothèque sert à gérer le microcontrôleur AT Mega32u4 de la carte.
4 - Démarrez l'IDE Arduino. Vous devriez pouvoir trouver la carte dans le menu Outils > Type de carte > LoRa32u4II 868
5 - Dans l'environnement Arduino, à ce stade on sélectionne le port par le menu Outils > Port, mais s'il y a eu l'erreur d'installation de pilote précédemment mentionnée, le port n'apparait pas. Il faut appuyer sur le bouton reset de la carte et sélectionner à nouveau, dans le laps de temps du reset, le menu Outils > Port. Là normalement le port devrait apparaitre quelques instants et on peut le sélectionner.
6 - Il reste encore à installer un bibliothèque : la bibliothèque LMIC qui contient les fichiers pour le protocole LoraWan. Pour cela il y a 2 méthodes :
Méthode 1 :

- La première est d'aller dans le menu Croquis > Inclure une bibliothèque > Gérer les bibliothèques.
- Dans la barre de recherche, du gestionnaire de bibliothèque, tapez "lmic"

> Si vous ne voyez rien apparaitre, vérifiez que les listes déroulantes Type et Sujet soient bien sur "Tout"

- Choisissez d'installer le bibliothèque IBM LMIC Framework

Méthode 2 :

- Téléchargez l'archive du projet GitHub https://github.com/matthijskooijman/arduino-lmic dans le répertoire Mes Documents/Arduino/Libraries. Vous devriez avoir un répertoire arduino-lmic-master

7 - Lorsque cette bibliothèque est bien installée, vous pouvez choisir dans le menu Fichier > Exemples > LMIC-Arduino le sketch ttn-otaa

> Pour faire court, la différence entre les sketches ttn-abp et ttn-otaa vient des deux différentes façon de s'enregistrer sur le réseau The Thing Network (par enregistrement, je parle de l'échange qui a lieu entre notre carte et la passerelle TTN lors de la mise sous tension de la carte)
> La première est l'Activation By Personalization (ou abp) pour laquelle il faut avoir une adresse réseau de la carte appelée DevAddr)
> La seconde est l'Over-The-Air-Activation (ou otaa). Dans ce mode DevAddr est transmis automatiquement pendant la phase d'enregistrement.

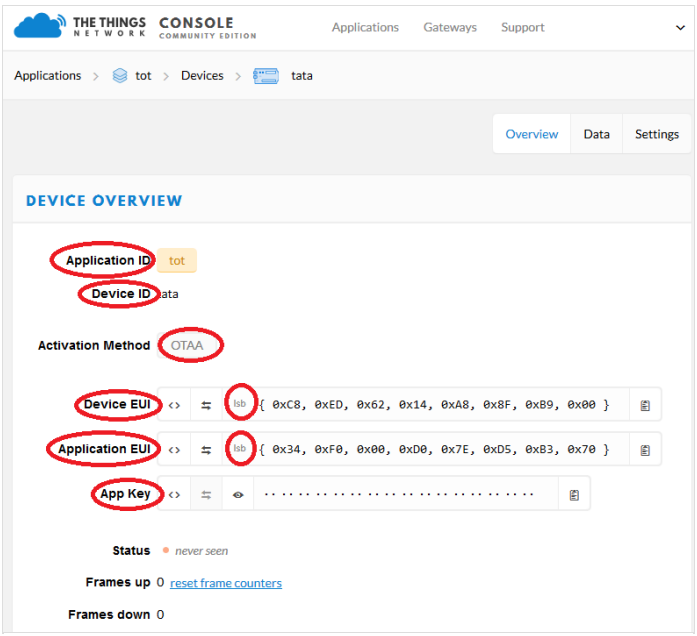
Étape 2 - Préparation de la carte

Lorsque vous avez téléchargé les drivers et les bibliothèques Arduino sur la page de BSFrance, vous avez du voir qu'il y avait un schéma de la carte (Pinout diagram)
Sur ce schéma, vous pouvez voir que qu'il y a une entrée appelée DIO1 en bas à gauche. Or il y a des fortes chances que la carte que vous ayez en main soit une révision 1.2 (c'est marqué derrière) et sur cette révision, DIO1 n'est pas là, mais à l'arrière de la carte (on voit le numéro de version sur la photo, ainsi que les pins IO1, IO2 et IO3 en face des pins 1, 2 et 3 respectivement.
Les broches DIO0 à DIO3 sont des broches de la puce LoRa, elles servent au pilotage de la puce par le microcontroleur.
Le sketch utilise DIO0 et DIO1 qui est en réalité un port de la puce LoRa de la carte en le reliant à une entrée du microcontrôleur (voir la deuxième image ci-contre)
Le tableau en bas à droite du schéma nous éclaire un peu sur ces liaisons. On voit que DIO0 est câblé en interne sur la carte sur la broche 7.

> Ce n'est pas la cas de la broche DIO1 qui doit être câblé par nos soins. Sans cette liaison, le sketch se bloque en attendant le signal.

Dans le cas d'une carte rév. 1.2, le plus simple est de faire un pont de soudure entre IO1 et la broche 1 qui sont face à face (comme on le voit sur la photo). Pour du LoRa (ce qui est notre cas) on ne se préoccupe pas de IO2 ni IO3.
Si la carte est vraiment comme sur le schéma, on relie DIO1 à la broche 2 avec un fil électrique.

Étape 3 - Petit tour rapide d'une application The Things Network



Pour utiliser TTN, il faut avoir un compte sur https://www.thethingsnetwork.org/.
C'est facile, c'est gratuit, et très bien expliqué dans ce tutoriel
Lorsqu'on est loggé avec son compte, il faut aller dans Console et choisir Applications.
Cliquer sur add application
Sur la page, renseigner le champ Application ID en donnant un nom significatif en minuscules sans espaces, mais on peut utiliser - et _
Vérifier que Handler registration soit bien sur ttn-handler-eu
Et voilà, cliquez sur le bouton vert Add application
La page qui s'affiche résume la création d'application, rappelle son nom et donne l'Application EUIs généré par TTN.
Ce que l'on vient de créer, c'est l'application du coté du serveur. Il faut encore déclarer que cette application est en réalité alimentée en données par une carte (un "device") qui transmet les données de ses capteurs ou autres.
Sur cette meême page, on voit qu'il y a justement une partie Device, avec un lien register device.
Cliquez dessus et vous arrivez sur une nouvelle page
Renseignez Device ID en donnant un nom un nom significatif en minuscules sans espaces, mais on peut utiliser - et _
Normalement, Device EUI et App Key sont déterminés par TTN. Si pour un des deux champs vous avez l'impression qu'il faut rentrer quelque chose, cliquez à gauche du champ sur les deux flèches croisées et le champ devient "this field will be generated", ce qui est exactement ce qu'on veut.
En bas de la page, App EUI est rappelé, pas besoin d'y faire attention
On clique sur Register et on arrive sur la page qu'il nous faut.
Sur cette page, on a dans l'ordre

- Device EUI
- Application EUI
- App Key

qui sont les 3 informations dont on aura besoin pour renseigner les champs correspondants dans le croquis.
On peut revenir sur cette page quand on veut ultérieurement.

> Souvenez vous siimplement que la page est la page DEVICE OVERVIEW si jamais vous vous perdez dans les pages sur le site TTN et qu'on y va en cliquant sur Applications > le nom de l'application > DEVICES > le nom du device

Voilà, coté TTN, tout est prêt, et il ne reste plus qu'a envoyer des données.

Étape 4 - Paramétrage du sketch

Il y a 2 paramétrages à faire :
1 - les données relatives à l'application TTN.
Ce sont les lignes suivantes qui doivent être complétées avec dans l'ordre

1. Device EUI
2. Application EUI
3. Application KEY

```
static const u1_t DEVEUI[8]  = {  }; //lsb
```

```
static const u1_t APPEUI[8] = {  }; //lsb
```

```
// This key should be in big endian format (or, since it is not really a
```

```
// number but a block of memory, endianness does not really apply). In
```

```
// practice, a key taken from ttnctl can be copied as-is.
```

```
// The key shown here is the semtech default key.
```

```
static const u1_t APPKEY[16] = {  }; //msb
```

Puis les lignes suivantes doivent être renseignées

```
const lmic_pinmap lmic_pins = {
```

```
  .nss = 8,
```

```
  .rxtx = LMIC_UNUSED_PIN,
```

```
  .rst = 4, // Needed on RFM92/RFM95? (probably not) D0/GPIO16
```

```
  .dio = {7, 1, LMIC_UNUSED_PIN}, // Specify pin numbers for DIO0, 1, 2
```

```
  // connected to D7, D1, -
```

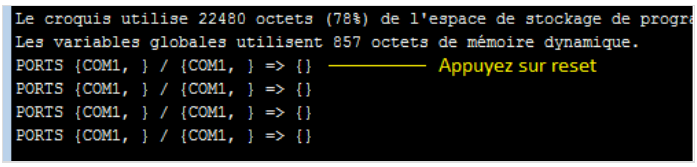
```
};
```

Bon normalement, pour les cartes Lora32u4,

- nss est câble en interne sur la broche 8
- DIO0 est câblé en interne sur la 7
- DIO1 : si vous avez fait le pont de soudure sur la rev 1.2, dans ce cas le deuxième chiffre est bien un 1, sinon, si vous avez câble DIO1 sur la broche 2, la séquence devient :

```
.dio = {7, 2, LMIC_UNUSED_PIN}, // Specify pin numbers for DIO0, 1, 2
```

Étape 5 - Téléversement du sketch



Normalement, il ne devrait pas y avoir d'étape spécifique pour le téléversement puisque c'est une opération qui se passe sans encombre.
Dans notre cas, il y a une petite condition supplémentaire qui provient de la spécificité de la puce ATMega32u4.
Cette puce dispose de sa propre liaison USB intégré (donc pas de FTDI sur la carte pour assurer la liaison USB<>UART comme pour les cartes arduino).
Cette particularité rend la liaison USB assez délicate car si le microcontrôleur est planté, s'il est un mode deep sleep, et dans d'autres cas encore, il n'y a plus de liaison USB (c'est pourquoi le port n'apparait pas toujours et qu'il faut faire un reset de la carte qui active pendant quelques secondes la liaison USB en attendant le chargement éventuel d'un sketch.
Donc lorsque vous activez le téléversement depuis l'environnement Arduino, il faut aussi appuyer sur reset pour forcer la carte à activer sa liaison USB.
Mais si la compilation dure un peu trop longtemps, la phase post-reset sera peut-être déja terminée.
Pour pallier ce problème, le mieux est d'aller dans fichier > préférence de l'environnement Arduino et de cocher Afficher les résultats détaillés pendant le téléversement.
De cette façon, vous voyez apparaitre dans la console du bas de l'environnement Arduino le moment où avrdude (la partie du logiciel qui s'occupe du transfert vers la carte) cherche le port COM correspondant. la console affiche en boucle des lignes similaires à celles de la photo ci-contre. Par défaut, avrdude ne voit que le port COM1 qui n'est pas le bon et il essaie plusieurs fois. Si vous appuyez à ce moment là sur reset, avrdude trouvera la carte et commencera le transfert.