Connexion au serveur LoRaWAN

Tutorial: How to connect to the LoRaWAN server for sending and receiving data

Difficulté **Moyen** | Durée **10 minute(s)** | Catégories **Électronique, Machines & Outils**

Coût **5 USD ($)**

Sommaire

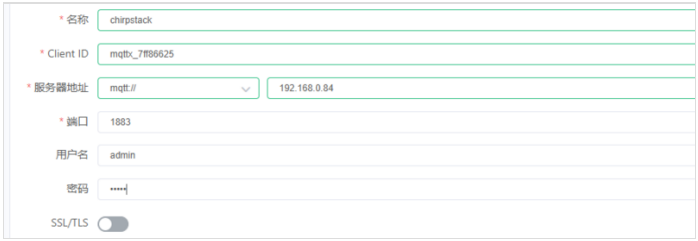
Matériaux

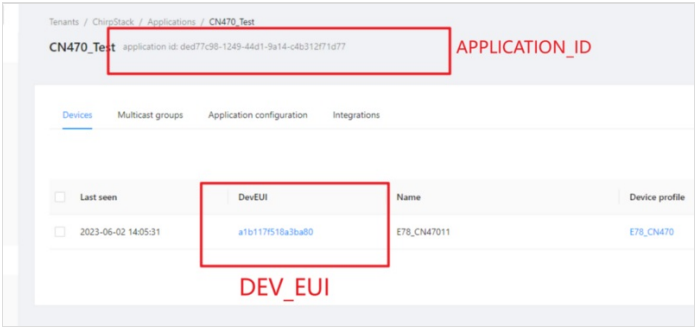
Outils

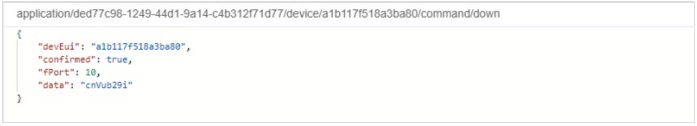
Étape 1 - Préparer

Préparer :
Utilisez le logiciel MQTTX pour vous abonner au serveur cible. Voici le serveur chirpstack construit par moi-même. L'IP est 192.168.0.84. Le nom d'utilisateur et le mot de passe sont tous deux admin, qui peuvent être écrits ou non.
Abonnez-vous à TOPIC via le serveur d'applications pour accepter les informations publiées par le serveur de l'appareil.
L'emplacement des informations sur l'appareil est indiqué dans la figure
Grammaire :
// SUJET téléchargé par le serveur de l'appareil
// affiche tout pour l'APPLICATION_ID donné
application/ID_APPLICATION/#
// affiche uniquement les charges utiles de liaison montante pour l'APPLICATION_ID donné
application/APPLICATION_ID/device/+/event/up
// Le serveur d'applications envoie TOPIC
application/APPLICATION_ID/device/DEV_EUI/command/down
Remarque : « # » et « + » sont des caractères génériques dans le protocole MQTT
Wildcard à un seul niveau (Wildcard à un seul niveau) : représenté par le symbole "+". Lorsqu'un niveau dans une rubrique utilise le caractère générique "+", il correspond à n'importe quel nom de niveau. Par exemple, « maison/+ » peut correspondre à des sujets tels que « maison/chambre », « maison/salon », etc., mais pas à plus d'un niveau de sujets tels que « maison/chambre/température ».
Caractère générique multi-niveaux (Multi-level wildcard) : représenté par le symbole "#". Lorsqu'un niveau d'un thème utilise le caractère générique "#", il peut correspondre à n'importe quel nom à plusieurs niveaux. "#" doit être le dernier niveau d'un sujet, qui correspond au niveau actuel ainsi qu'à tous les sujets plus profonds. Par exemple, « maison/# » peut correspondre à « maison/chambre », « maison/salon » et « maison/chambre/température » à n'importe quel niveau de thème.
Informations push sur l'appareil
//Recevoir le SUJET :
//Abonnez-vous au SUJET de téléchargement de données d'un seul appareil
application/ded77c98-1249-44d1-9a14-c4b312f71d77/device/a1b117f518a3ba80/event/up
//Abonnez-vous à tous les appareils sous l'application actuelle
demande/ded77c98-1249-44d1-9a14-c4b312f71d77/#
/* Commande AT pour que le nœud terminal télécharge les données
1 : Besoin de confirmer la trame // 0 n'a pas besoin de confirmer
2 : Le nombre maximum de retransmissions est de 2 fois

10 : le nombre d'octets dans le package actuel
xx:données */
AT+DTRX=1,2,10,3435363738
Les informations reçues par le serveur d'applications sont affichées dans la figure
Le serveur d'applications envoie des informations
//Envoyer le SUJET :
application/ded77c98-1249-44d1-9a14-c4b312f71d77/device/a1b117f518a3ba80/command/down
//Envoyer le format des données
{
"devEui": "a1b117f518a3ba80", #ID du périphérique
"confirmed": true, #Si une confirmation est requise
"fPort": 10, #Port cible de la couche application
"data": "cnVub29i" #data, remarque : nécessité d'utiliser le format d'encodage base64, par exemple : cnVub29i == 72756E6F6F62(runoob)
}
//Le terminal lit les données du tampon de réception et efface le tampon
AT+DRX ?
Les informations reçues par l'appareil sont affichées sur la figure :
Avis :
Les caractères génériques MQTT ne peuvent être utilisés que lors de l'abonnement, pas lors de l'envoi
Site Web d'outils :
ASCII en chaîne
https://www.asciim.cn/m/tools/convert_ascii_to_string.html
cryptage et déchiffrement base64
https://c.runoob.com/front-end/693/
Interagissez avec les données du serveur TTN
Dans l'article précédent, nous avons principalement expliqué comment enregistrer des passerelles, créer des applications, créer des appareils, etc. sur thethingsnetwork.org. thethingsnetwork.org (ci-après dénommé TTN) n'est qu'un serveur réseau (serveur réseau) et n'enregistrera pas d'application. données. Par conséquent, dans le projet lui-même, un serveur d'applications est également requis. thethingsnetwork.org propose diverses méthodes permettant à la plate-forme d'application d'obtenir des données et de gérer les appareils.
Principalement divisé en 3 catégories :
API : elle est divisée en API de données et API de gestion d'applications. L'API de données utilise principalement MQTT pour recevoir et envoyer des données, et l'API de gestion d'applications utilise principalement HTTP pour gérer les appareils enregistrés.
SDK : Différents langages tels que Go, Java, Node.js sont fournis.
Intégrations : ThingSpeak, AWS IOT, etc.
Dans la plupart des cas, vous n'avez qu'à prêter attention au reporting et à l'envoi de données, donc cet article explique principalement comment utiliser MQTT pour obtenir et envoyer des données, la description officielle
https://www.thethingsnetwork.org/docs/applications/mqtt /api.html
Le client MQTT.fx est utilisé ici pour démontrer que d'autres clients MQTT en langage de haut niveau peuvent être utilisés dans des applications pratiques.



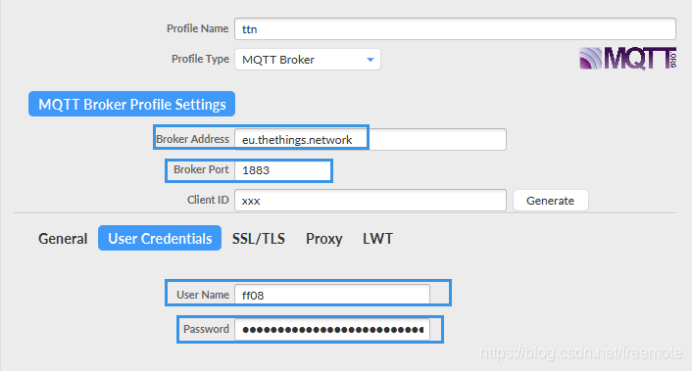




```
[2023-06-01 15:40:33.936]
RX:
OK+RECV:01,0A,06,72756E6F6F62

[2023-06-01 15:40:40.088]
TX: AT+DRX?

[2023-06-01 15:40:40.270]
RX:
+DRX:6,72756E6F6F62
OK
```



Étape 2 - Connecter

Connecter

Principalement les quatre paramètres du schéma fonctionnel ci-dessus, parmi lesquels :

Adresse du courtier : <Region>.thethings.network, où <Region> est la région sélectionnée.

Ce paramètre est également le Handler que nous avons choisi lors de l'enregistrement de l'application, qui est l'adresse à droite dans la figure ci-dessous :

Port du courtier : 1883 (non crypté)

ID client : il suffit d'en donner un

Nom d'utilisateur : ID d'application, qui est personnalisé lors de l'enregistrement de l'application, voici ff08

Mot de passe : clé d'accès à l'application, au format base64, celle-ci est générée par le système, bien sûr vous pouvez également l'ajouter vous-même, voici celle générée par le système :

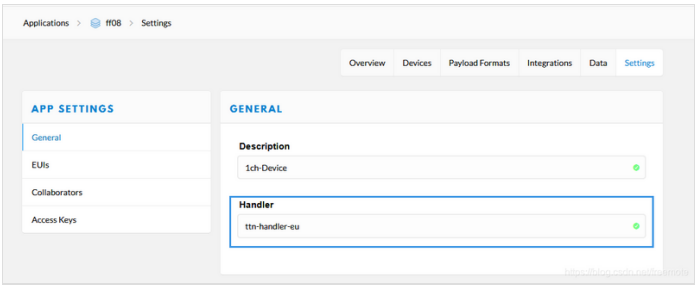
Données de liaison montante du nœud S'abonner (S'abonner)

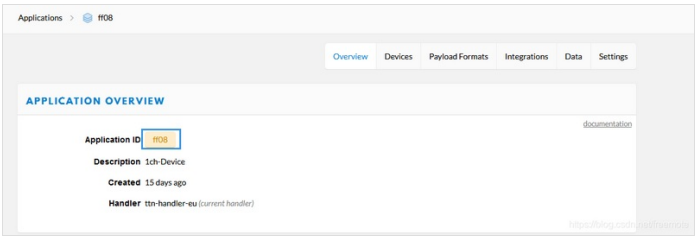
TTN fournit un sujet Topic : <AppID>/devices/<DevID>/up, où <AppID> et <DevID> sont tous deux définis lors de l'enregistrement, et vous pouvez le voir lorsque vous entrez l'appareil correspondant, comme suit :

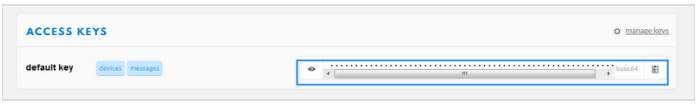
Vous pouvez voir l'ID du nœud, le numéro de port signalé, le compteur, le point de fréquence, l'horodatage, etc., où le champ payload_raw correspond aux données que nous avons téléchargées, qui sont affichées au format base64.

Ebyte est une entreprise nationale de haute technologie spécialisée dans la recherche et le développement de modules sans fil et de terminaux IoT industriels. Les produits développés et fabriqués de manière indépendante comprennent des modules sans fil LoRa / WiFi / Bluetooth / ZigBee , des modules Ethernet, des terminaux de transmission de données NB-IoT , de l'IoT industriel.

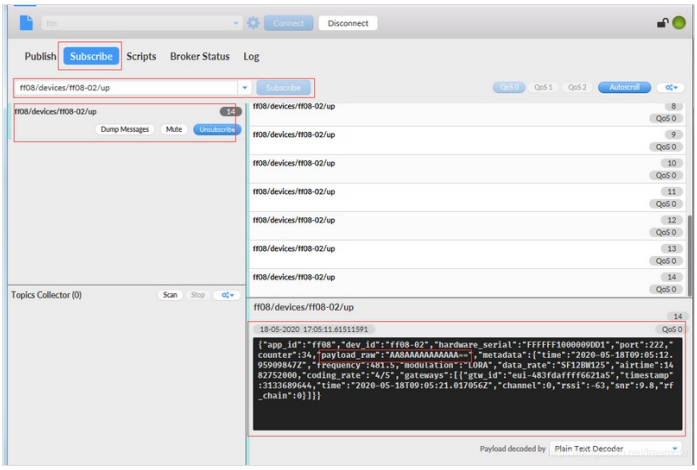

digitalcatapult-uk-router public ttn.thingsconnected.net
meshed-router public thethings.meshed.com.au
switch-router public ttn.opennetworkinfrastructure.org
ttn-router-asia-se public asia-se.thethings.network
ttn-router-brazil public brazil.thethings.network
ttn-router-eu public eu.thethings.network
ttn-router-jp public asia-se.thethings.network
ttn-router-us-west public us-west.thethings.network


Applications ff08 Settings
Overview Devices Payload Formats Integrations Data Settings
APP SETTINGS
General
EUIs
Collaborators
Access Keys
GENERAL
Description
1ch-Device
Handler
ttn-handler-eu


Applications ff08
Overview Devices Payload Formats Integrations Data Settings
APPLICATION OVERVIEW
documentation
Application ID ff08
Description 1ch-Device
Created 15 days ago
Handler ttn-handler-eu (current handler)


ACCESS KEYS
manage keys
default key
devices
messages


Applications ff08 Devices ff08-02
DEVICE OVERVIEW
Application ID ff08
Device ID ff08-02


Connect
Disconnect
Publish
Subscribe
Scripts
Broker Status
Log
ff08/devices/ff08-02/up
Dump Messages
Mute
Unsubscribe
Topics Collector (0)
Scan
Stop
Payload decoded by
Plain Text Decoder

Étape 3 - Publier (Publier) des données de liaison descendante

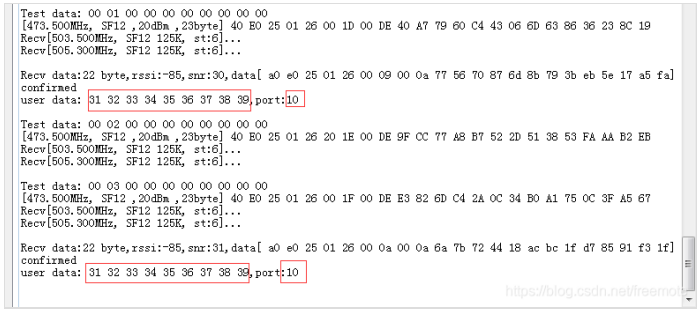
Publier (Publier) des données de liaison descendante

TTN fournit un sujet Topic : <AppID>/devices/<DevID>/down, où <AppID> et <DevID> sont tous deux définis lors de l'enregistrement et peuvent être vus lors de la saisie de l'appareil correspondant.

Les données sont au format json, les trois champs ci-dessus sont obligatoires et le champ payload_raw correspond aux données que nous souhaitons envoyer, qui sont au format base64.

Le texte en clair correspondant à « MTIzNDU2Nzg5 » est « 123456789 ».

Affichez les données d'application fournies sur le nœud :



Notes et références

Les données reçues par le nœud sont imprimées au format hexadécimal, qui correspond simplement à la chaîne « 123456789 ».