

Commande du module RN2483

Sommaire

I – Description.....	2
I.1 – Fonctionnement.....	2
I.2 – Prérequis.....	2
II – Schéma électronique.....	3
III – Routage.....	4
IV – Communication.....	5
IV.1 – Génération des identifiants LoRaWAN.....	5
1 - Device Profile.....	5
2 – Application.....	6
IV.2 – Communication UART.....	7
V – Récupération et écriture des messages dans la base de données.....	8
V.1 – Connexion au MQTT Broker.....	8

I – Description

I.1 – Fonctionnement

Le module RN2483 est le premier module certifié LoRaWAN par LoRa Alliance. Il utilise la version LoRaWAN mac 1.0.0 .

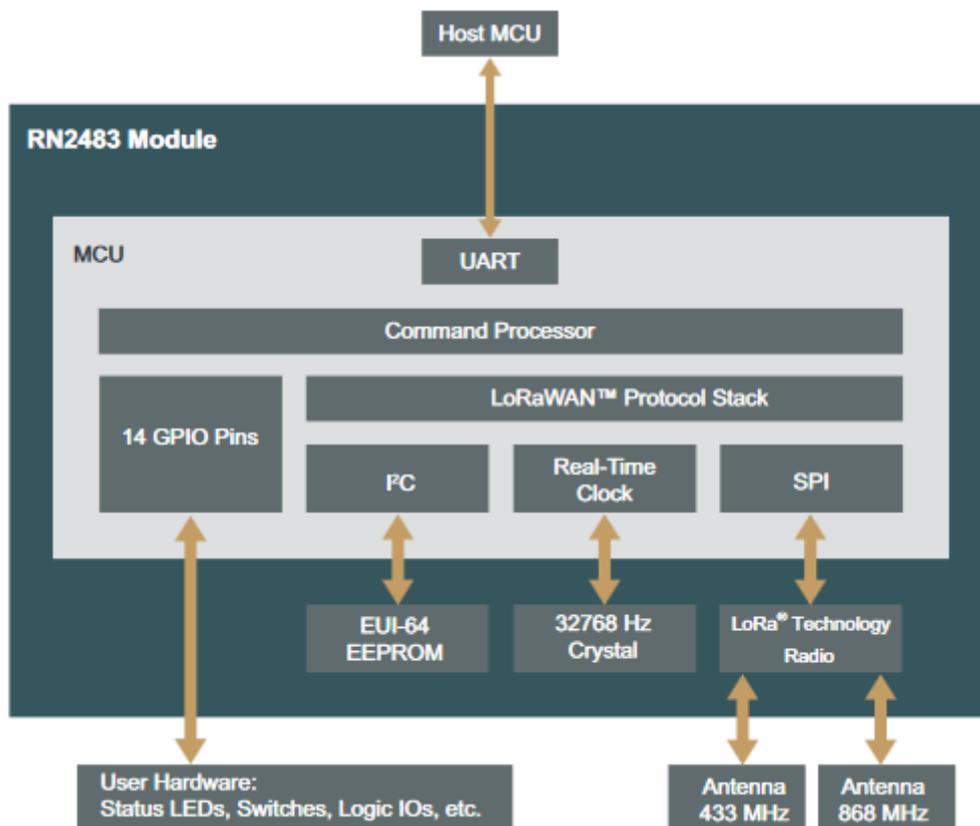


Figure 1: Diagramme de bloc

La communication entre le micro contrôleur et le module se fait par UART sous formes de commandes décrites dans cette fiche PDF : <https://ww1.microchip.com/downloads/en/DeviceDoc/40001784B.pdf> .

Pour des communications hors réseau LoRaWAN en Peer To Peer (P2P), les commandes radios transitent par modulation FSK.

I.2 – Prérequis

Pour permettre la communication, le module nécessite une alimentation continue comprise entre 2,1V et 3,6V ainsi qu'une antenne réglé pour 868Mhz et une communication en UART en niveau TTL.

III – Routage

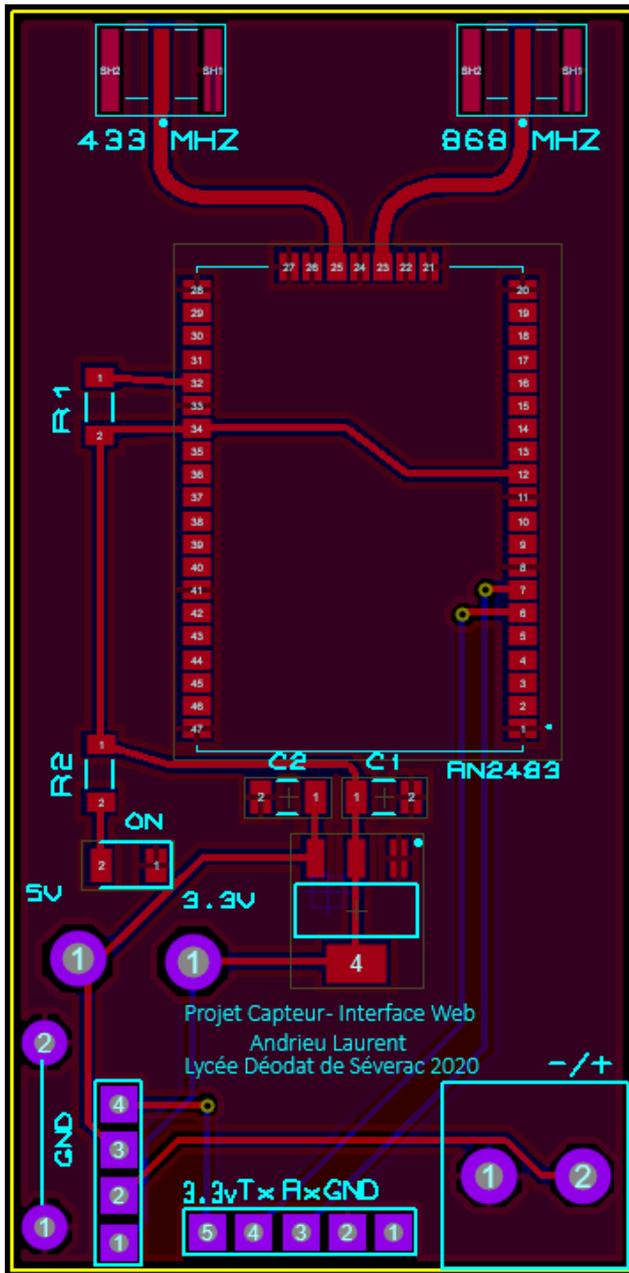


Figure 3: Routage

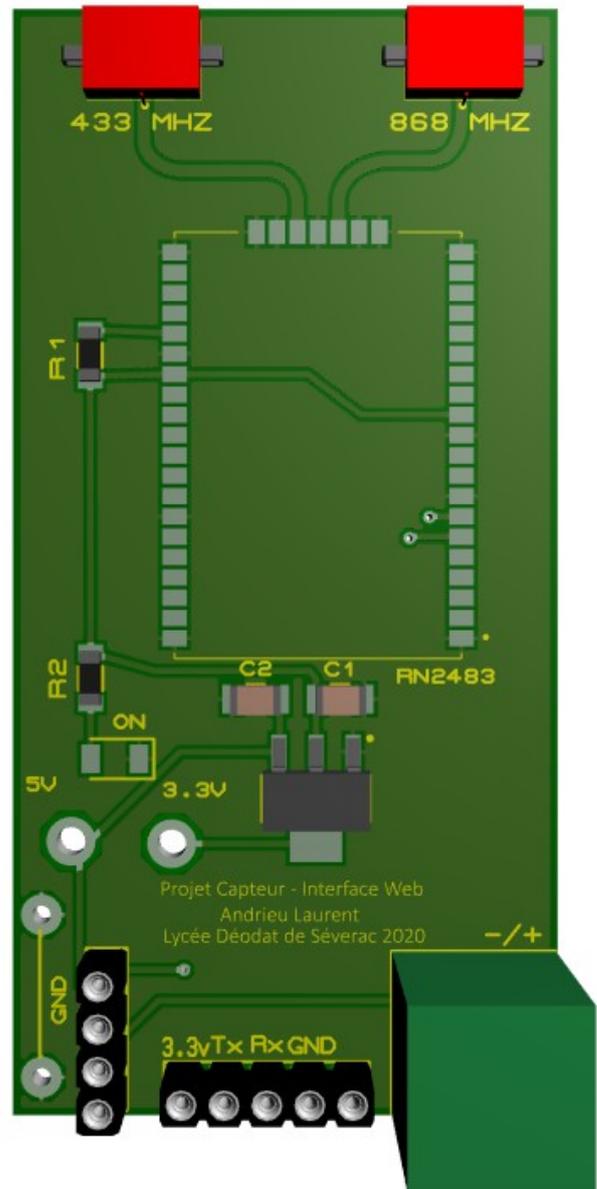


Figure 4: Vue 3D

IV – Communication

IV.1 – Génération des identifiants LoRaWAN

1 - Device Profile

Connectez vous sur <https://loraserver.tetaneutral.net> puis rendez vous dans la rubrique « Device profile » afin de créer des identifiants de connexion au réseau LoRaWAN.

Créez un profile par module en appuyant sur « + CREATE ».

Choisissez un nom et conservez les paramètres montrés ci-dessous.

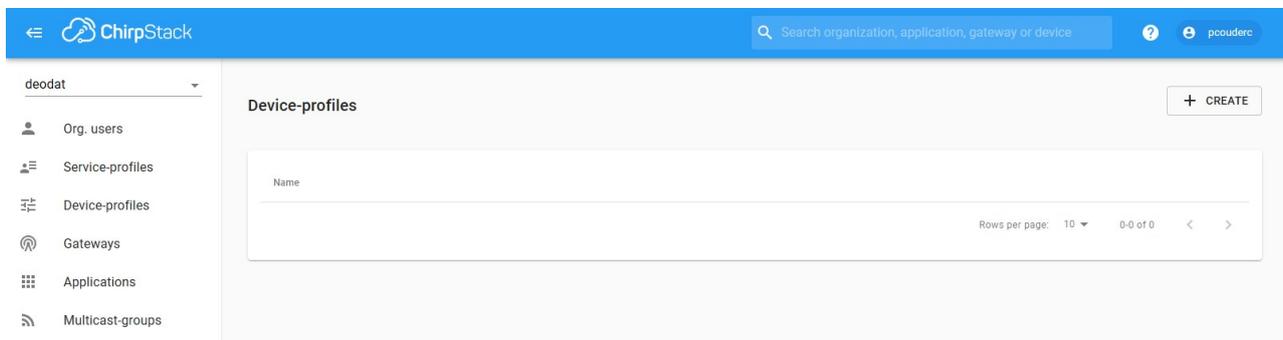


Figure 5: Device Profile

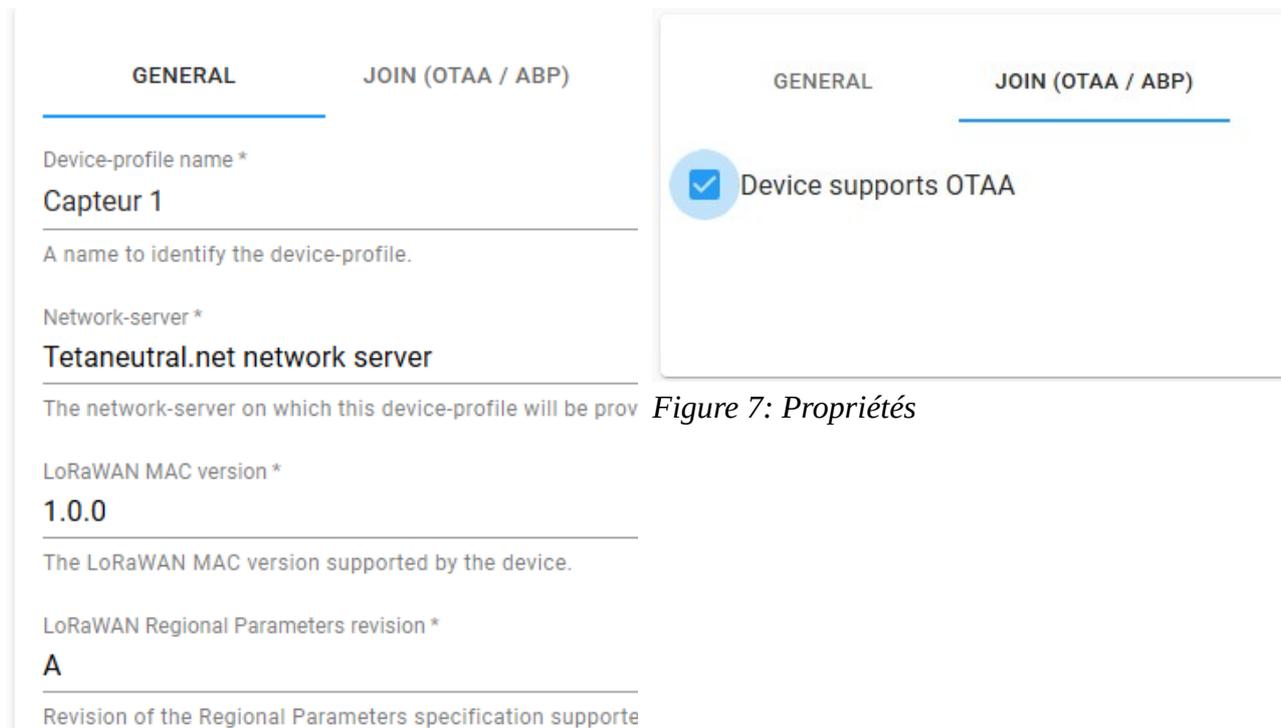


Figure 6: Propriétés

Figure 7: Propriétés

2 – Application

Créez ensuite une application dans la rubrique « Applications ».

Choisissez un nom et une description puis sélectionnez le Service-profile ‘default’.

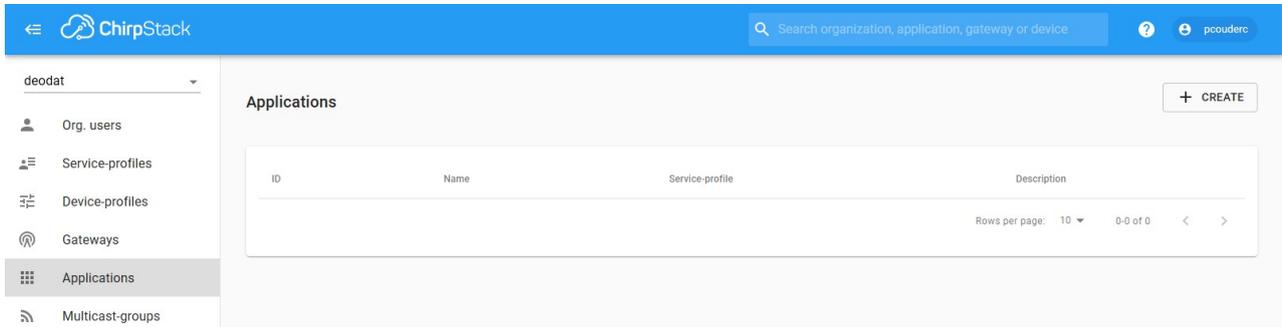
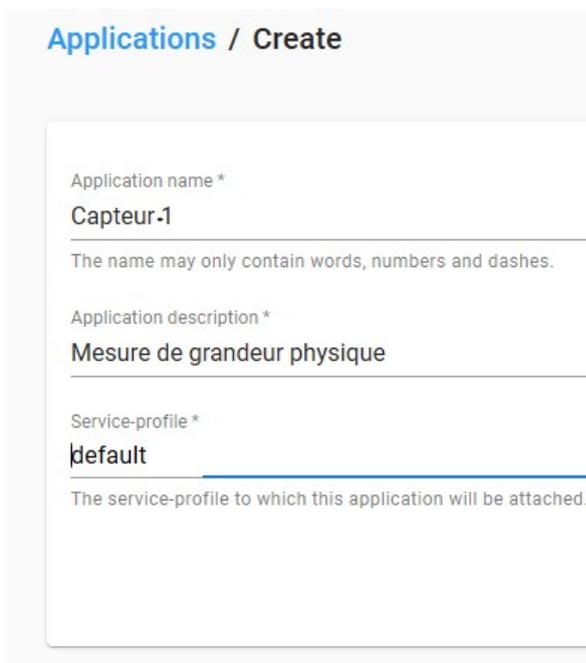


Figure 8: Applications



The screenshot shows the 'Applications / Create' form. The form has three main sections, each with a label and a text input field:

- Application name ***: The input field contains 'Capteur-1'. Below the field is a note: 'The name may only contain words, numbers and dashes.'
- Application description ***: The input field contains 'Mesure de grandeur physique'.
- Service-profile ***: The input field contains 'default'. Below the field is a note: 'The service-profile to which this application will be attached.'

Figure 9: Paramètres d'application

Commencez ensuite par ajouter les capteurs de votre choix dans cette application en choisissant le ‘Device-profile’ correspondant puis générez le DevEUI ou écrivez le vous-même selon votre choix.

The screenshot shows the 'Create Device' form with the following fields and options:

- GENERAL** (selected tab)
- Device name ***: Capteur-1
- Device description ***: Recuperation de donnees physiques
- Device EUI ***: e9 27 e5 c3 67 d8 90 7d (with MSB and refresh icons)
- Device-profile ***: Capteur 1 (dropdown menu)
- Disable frame-counter validation**
- Note: Note that disabling the frame-counter validation will compromise security as it enables people to perform replay-attacks.
- CREATE DEVICE** button

Figure 10: Ajout du 1er capteur

Générez ensuite la clé d’application.

The screenshot shows the 'Keys (OTAA)' page with the following fields and options:

- Application key ***: f4 3a fc a0 b3 8f 8f 5f 0e c7 59 b9 6e 90 bb 05 (with MSB, refresh, copy, and eye icons)
- Note: For LoRaWAN 1.0 devices. In case your device supports LoRaWAN 1.1, update the device-profile first.
- Gen Application key**: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 (with MSB, refresh, copy, and eye icons)
- Note: For LoRaWAN 1.0 devices. This key must only be set when the device implements the remote multicast setup specification / firmware updates over the air (FUOTA). Else leave this field blank.
- SET DEVICE-KEYS** button

Figure 11: Clé d'application

IV.2 – Communication UART

Pour connaître la liste des commandes, se référez au lien précédent :

- I.1 – Fonctionnement.

Connectez vous avec n’importe quel logiciel de communication série en 57600/8/N/1/None soit 8bits de données, pas de parité, 1 bit de stop, 57600bauds et pas de contrôle de flux.

Suite à une mise à jour, la connexion par OTAA en version LoRaWAN 1.0 ne nécessite plus que 2 clés à générer contre 3 :

- DevEUI : Identifiant d'appareil unique
- AppKey : Clé de chiffrement

Liste des paquets à envoyer au RN2483 (remplacez les 0 par vos valeurs):

1. mac set deveui 0000000000000000 (remplacer par vos valeurs)
2. mac set appeui 0000000000000000 (laisser à 0)
3. mac set appkey 00000000000000000000000000000000 (remplacer par vos valeurs)
4. mac save
5. mac join otaa

Attendez la réponse 'ok accepted' puis :

1. mac save

la commande 'mac save' permet la sauvegarde des données dans sa mémoire, il faut l'exécuter au moins une fois après avoir rejoint le réseau pour enregistrer les clés, les identifiants réseau et autres données que nous a fournis le réseau LoRaWAN.

À partir de là les données peuvent être envoyés de la manière suivante :

1. mac tx cnf 1 <données hexadécimales>

Référence constructeur : 2.4.2 mac tx <type> <portno> <data> page 18.

V – Récupération et écriture des messages dans la base de données

V.1 – Connexion au MQTT Broker

Le broker MQTT est le service permettant de faire communiquer des appareils et des applications via le protocole MQTT sur la couche TCP/IP.

Une connexion entraîne une session sur la quelle on vient s'abonner à un 'topic' avec un MQTT client.

Ici le broker est tetaneutral, il suffit de renseigner l'adresse. Par choix de tetaneutral, il n'est pas nécessaire de se connecter avec des identifiants, les données sont en libre accès en clair.

Connectez vous à votre service Node-Red puis créez un « flow ».

Recherchez et installez la bibliothèque mqtt et mysql dans « manage palette » / « Install »



Figure 12: Bibliothèque MQTT

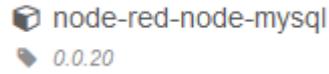


Figure 13: Bibliothèque mysql

Insérez les noeux (nodes) suivants :

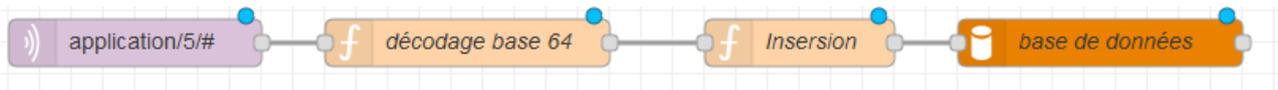


Figure 14: Flow

Configuration du client mqtt :

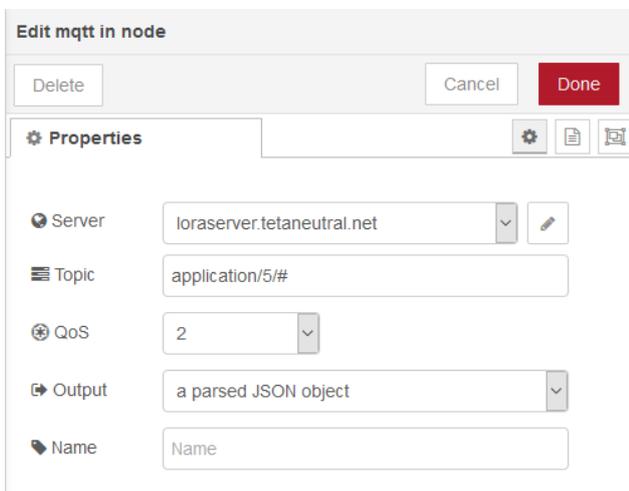


Figure 15: Node mqtt

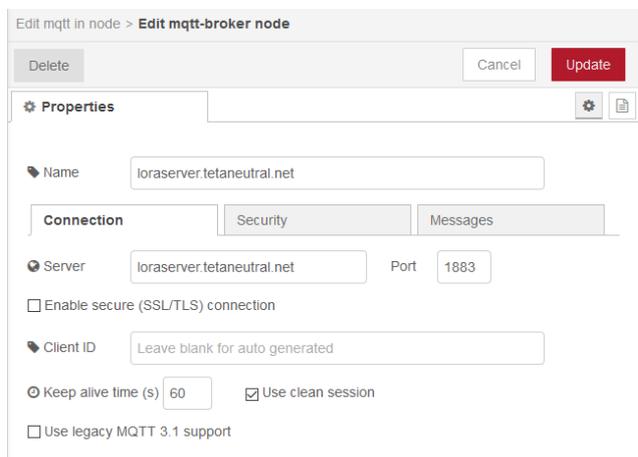


Figure 16: Propriétés de la node mqtt

Configurez ensuite votre node de connexion à la base de données avec vos paramètres :

The image shows a 'Properties' dialog box with the following fields and values:

- Host: 127.0.0.1
- Port: 3306
- User: acmp
- Password: [masked]
- Database: capteur
- Timezone: [empty]
- Name: Name

Figure

17: Propriétés connexion BDD

Dans la fonction de décodage, copiez le code permettant le décodage en base64 :http://www.webtoolkit.info/javascript_base64.html#.XqmwUJk6-M8

Ensuite, créez un objet contenant les informations souhaitées dans l'objet payload avec les données entouré par la fonction 'Base64.decode()'.

```
123 ▾ msg = {  
124 ▾   payload: {  
125     devEUI: msg.payload.devEUI,  
126     rssi: msg.payload.rxInfo[0].rssi,  
127     data: Base64.decode(msg.payload.data)  
128 ^   }  
129 ^ };  
130  
131 return msg;
```

Figure 18: Fonction Décodage

Enfin, créer votre requête SQL contenant les messages :

```
1 var id = msg.payload[0].devEUI; // Identité du capteur
2 var rssi = msg.payload[0].rssi; // Force du signal
3 var data = msg.payload[0].data.split(","); //données séparés par ','
4
5 var insersion = {};
6 requete = `
7 INSERT INTO table (id, rssi, lat, lon, idMeasure)
8     VALUES (${id}, ${rssi}, ${data[2]}, ${data[3]});
9 `;
10 insersion.topic = requete;
11
12 return insersion;
```

Figure 19: Fonction SQL