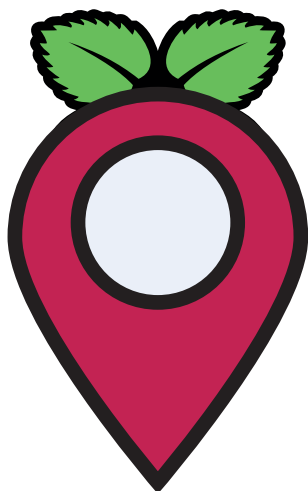




mise à jour de la balise GPS LoRa

Recevoir et afficher la localisation
à l'aide d'un Raspberry Pi



Hans Schneider (Allemagne)

L'article intitulé « balise GPS LoRa » publié dans *Elektor* (11-12/2020) décrit comment récupérer les données de suivi sur un serveur TTN à l'aide de Node-RED, puis les visualiser sur une carte. En principe, cela devrait fonctionner, non seulement avec un PC, mais aussi avec un Raspberry Pi. C'est effectivement possible, mais il y a un ou deux obstacles à franchir. Un de nos lecteurs a trouvé une bonne solution et décrit comment il a porté le flux Node-RED sur l'ordinateur monocarte.

Dans l'article « balise GPS LoRa » (*Elektor*, 11-12/2020), Mathias Claussen décrit le matériel et le logiciel utilisables pour suivre des objets en mouvement. Un module de suivi compact envoie les données d'un capteur GPS via LoRa à une passerelle, qui les transmet à un serveur *The Things Network*, d'où les données brutes peuvent être lues manuellement via l'internet, sous réserve d'avoir le droit d'accès. Bien sûr, c'est plus pratique d'y accéder automatiquement via un *framework* tel que Node-RED, qui peut également gérer l'affichage sur une carte (du monde). Le flux Node-RED décrit dans l'article convient très bien pour un PC, mais, pour un fonctionnement automatique, il a le grave inconvénient de nécessiter que le serveur Node-RED et un PC assoiffé d'énergie soient tous deux actifs en permanence.

Il est probable que de nombreux particuliers férus de technologie possèdent déjà au moins un Raspberry Pi sur lequel tourne un logiciel tel que Pi-Hole [1], Homebridge [2] ou Flightradar24 [3], en attente permanente d'échange de données. Pourquoi ne pas lui confier une tâche de plus ? Mais attention : les ressources nécessitées par Node-RED peuvent parfois entrer en conflit avec une installation de Homebridge. En outre, la procédure d'installation n'est pas non plus totalement exempte de difficultés.

Installation de Node-RED sur le RPi

Pour commencer, nous devons installer Node-RED sur le Raspberry Pi. Le lecteur est supposé savoir saisir des commandes dans la fenêtre de terminal du RPi. Ensuite, nous devons nous connecter au terminal via SSH, en utilisant la ligne de commande suivante : `bash <(curl -sL https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered)`. Vous devrez peut-être entrer `sudo apt install build-essential git` pour permettre à `npm` de compiler les modules binaires qui doivent être installés. L'exécution du script prend un certain temps, de l'ordre de la demi-heure.

Lorsque le script est terminé, la commande `sudo systemctl enable nodered.service` provoquera le démarrage automatique de Node-RED à la mise sous tension ou au redémarrage du RPi. Cette commande peut être désactivée en entrant `sudo systemctl disable nodered.service`.

Le serveur Node-RED peut être démarré avec `node-red-start` et arrêté avec `node-red-stop`. Pour arrêter et redémarrer le serveur, utilisez `node-red-restart`. Le terminal peut être quitté après avoir démarré Node-RED, et le serveur continuera à fonctionner. Lorsque le terminal est redémarré, vous pouvez démarrer le service de sortie du serveur en utilisant `node-red-log` et voir la sortie dans le terminal. Ceci est utile si vous voulez vérifier si tout fonctionne toujours sans démarrer une nouvelle instance du serveur.

Installation des modules requis dans Node-RED

Les modules peuvent être installés dans Node-RED en utilisant soit le « gestionnaire de palette » dans la fenêtre du navigateur, soit le gestionnaire de paquets `npm` depuis le terminal. Je recommande d'utiliser `npm` dont le fonctionnement me paraît plus fiable, auquel cas Node-RED doit être arrêté au préalable.

Par contre, si le service Node-RED est actif sur le RPi, vous pouvez vous connecter à l'interface utilisateur en utilisant un navigateur sur le PC et en saisissant le lien : `http://<adresse IP du RPi>:1880`, ce qui vous connectera à l'« éditeur de flux ». Les modules requis sont installés en premier ; dans l'« éditeur de flux », cela se fait via le « gestionnaire de palette ». Les modules dont vous aurez besoin



Figure 1. Nœud MQTT dans Node-RED.

Figure 2. Paramétrage des propriétés du serveur du nœud MQTT.

Figure 3. Ajouter un nouveau serveur.

Figure 4. Configuration TLS du nouveau serveur.

(dont certains ont déjà été installés lors de l'installation de Node-RED) sont les suivants :

- > *node-red*
- > *node-red-contrib-worldmap*
- > *node-red-node-rbe*
- > *node-red-node-sqlite*
- > *node-red-node-base64*
- > *node-red-tail*

L'installation via le « gestionnaire de palette » demande un peu de patience ; si le moniteur d'activité a été masqué, il faut un certain temps avant que le message de succès n'apparaisse en haut de la fenêtre. Si l'installation via le « gestionnaire de palette » échoue, quittez l'éditeur et, dans le terminal, arrêtez le service Node-RED sur le RPi et procédez manuellement. Par exemple, la commande `npm i --unsafe-perm node-red-node-sqlite` installe le module *node-red-node-sqlite*, et vous pouvez faire de même pour les autres modules.

Si *node-red-node-sqlite* n'installe pas le module *node-red-sqlitedb*, il faudra l'installer par la suite. Ceci n'est possible que via *npm*, car dans ce cas le « gestionnaire de palette » détecte un conflit avec le module *node-red-node-sqlite*.

Le module TTN manquant

Tout devrait s'être bien passé jusqu'ici, sauf que vous avez peut-être remarqué l'absence du module TTN (*The Things Network*) présenté dans l'article original. En effet il n'est pas pris en charge par l'environnement Raspberry Pi. Par exemple, une fois que *node-red-contrib-ttn* a été chargé, il n'est plus possible de démarrer le service Node-RED sur le RPi. Il m'a fallu un moment pour le comprendre, alors il m'est agréable de penser au temps que cette information va vous faire gagner !

La solution réside dans le module *mqtt* intégré à Node-RED. Il est inclus dans le flux d'origine plutôt que dans le module de liaison montante TTN. Vous pouvez utiliser mon flux, spécialement adapté au RPi et téléchargeable sur la page de la balise GPS LoRa [5] d'Elektor Labs. Le flux est importé en faisant simplement glisser le fichier téléchargé JSON dans la fenêtre de l'« éditeur de flux ». D'ailleurs, ce flux adapté au RPi fonctionne également sur le PC à la place des flux avec les nœuds TTN spéciaux.

Avec l'« éditeur de flux », vous pouvez maintenant double-cliquer sur le symbole *mqtt* (fig. 1) et modifier les paramètres dans la fenêtre qui s'ouvre (fig. 2). Maintenant, « v3/+/devices/+/up » est saisi sous *Topic* comme indiqué sur la figure afin de recevoir les données de tous les appareils de l'application TTN. Pour « output » (sortie), vous pouvez conserver la configuration par défaut « a parsed JSON object ». Un nouveau serveur peut être saisi dans le champ « Server » à l'aide du symbole du crayon ce qui ouvre la fenêtre illustrée à la figure 3. L'ID d'application de la console TTN est utilisé comme « name » (nom), et *eu1.cloud.thethings.network* est le « server » (serveur). Le port par défaut est 1833 (MQTT sans TLS) qui doit être changé en 8883. Enfin, il manque la coche à côté de « Enable secure (SSL/TLS) connection ». Un clic sur le symbole du crayon du champ « Configuration TLS » ouvre la fenêtre suivante (fig. 4).

Maintenant, les paramètres peuvent être acceptés en cliquant sur



Figure 5. Gestion des clés d'accès des nœuds MQTT.

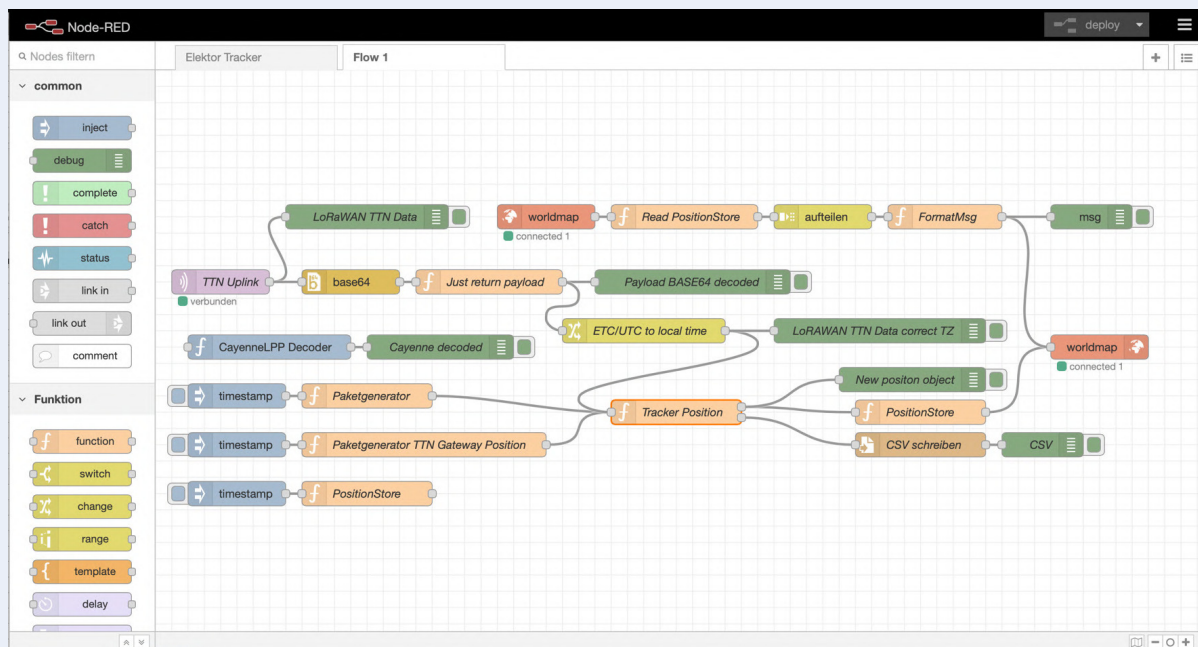


Figure 6. Le diagramme de flux terminé dans Node-RED.

« Add ». De retour dans la fenêtre « mqtt-broker config node », sélectionnez l'onglet « Security » (fig. 5). Le « Username » est l'ID d'application de la console TTN suivi de @ttn. Une clé d'accès avec au moins « Messages Rights » doit être produite comme mot de passe dans la console TTN. La clé d'accès copiée de la console TTN est ensuite collée sous « Password » et confirmée en cliquant sur « Add ». De retour dans l'élément *mqtt in*, vous pouvez donner un nom à cet élément (par ex. *TTN uplink*) ; enfin, un clic sur « Done » termine la tâche. Le **figure 6** montre le diagramme de flux Node-RED mis à jour. Le décodeur CayenneLPP présenté dans le schéma est inclus pour une extension future du système. ◀

210120-04



PRODUITS

- **Elektor LoRa Node** - module partiellement assemblé
www.elektor.fr/19175
- **RFM95** - module émetteur-récepteur LoRa à très longue portée (868 MHz) de SeeedStudio
www.elektor.fr/18715
- **PG1301** - concentrateur GPS LoRaWAN pour Raspberry Pi (868 MHz) de Dragino
www.elektor.fr/19367

Contributeurs

Idée et texte : Hans Schneider

Rédaction : Mathias Claußen, Jens Nickel, C.J. Abate

Traduction : Helmut Müller

Mise en page : Harmen Heida

Des questions, des commentaires ?

Envoyez un courriel à l'auteur (hans.schneider@belgacom.net) ou contactez Elektor (redaction@elektor.fr).

LIENS

- [1] « balise GPS LoRa », M. Claußen, Elektor 11-12/2020 : www.elektormagazine.fr/200096-02
- [2] Pi-hole : <https://pi-hole.net>
- [3] Homebridge : <https://homebridge.io>
- [4] Flightradar24 : www.flightradar24.com/
- [5] « LoRa GPS Tracker », Elektor Labs : www.elektormagazine.fr/labs/lora-gps-tracker

