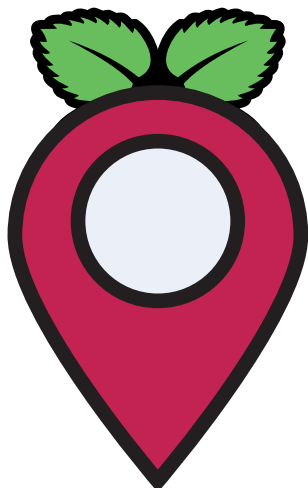




LoRa GPS-tracker update

ontvang en toon de locatie met een Raspberry Pi



Hans Schneider (Duitsland)

In het artikel “LoRa GPS-tracker” in Elektor van november/december 2020 wordt beschreven hoe u met Node-RED de trackinggegevens van een TTN-server kunt verzamelen en vervolgens op een kaart kunt visualiseren. In principe zou dit niet alleen op een PC, maar ook met een Raspberry Pi moeten werken. Dit is inderdaad mogelijk, maar er moeten een paar hindernissen worden overwonnen. Een van onze lezers heeft een goede oplossing gevonden, en beschrijft hoe hij de Node-RED datastroom heeft geport naar de single-board computer.

In het artikel “LoRa GPS-tracker” (Elektor november/december 2020) beschrijft Mathias Claussen de hard- en software waarmee bewegende objecten kunnen worden gevolgd. Een compacte trackermodule stuurt de gegevens van een GPS-sensor via LoRa naar een gateway, die ze doorstuurt naar een The Things Network-server, waar de ruwe gegevens handmatig kunnen worden opgevraagd via internet, mits de nodige autorisatie is verleend. Het is natuurlijk handiger als die gegevens automatisch wordt opgevraagd via een framework als Node-RED, dat ook de weergave op een (wereld)kaart kan aansturen. De in het artikel beschreven Node-RED-Flow werkt heel goed op een PC, maar dit heeft één groot nadeel: een energieslurpende PC moet voor dit doel ingeschakeld zijn en we moeten de Node-RED server draaien, anders krijgen we geen gegevens.

Waarschijnlijk hebben veel huishoudens met gevoel voor techniek al minstens één Raspberry Pi waarop software zoals Pi-Hole [2], Homebridge [3] of Flightradar24 [4] draait en die continu wacht om gegevens te ontvangen en te verzenden. Waarom zouden we geen andere taak aan deze Raspberry Pi toevertrouwen? Een waarschuwing: de door Node-RED gebruikte resources kunnen soms botsen

met een Homebridge-installatie. Met dat in het achterhoofd is de installatieprocedure ook niet geheel probleemloos.

Node-RED installeren op de Raspberry Pi

Eerst moeten we Node-RED installeren op de Raspberry Pi. We nemen aan dat de lezer bekend is met het invoeren van commando's via de Raspberry Pi Terminal-app. Vervolgens moeten we inloggen op de terminal via SSH, met behulp van de volgende opdrachtregel: `bash <(curl -sL https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered)`. Misschien moet u ook `sudo apt install build-essential git` invoeren, zodat npm de binaire modules kan construeren die geïnstalleerd moeten worden. Het duurt een tijdje om het script uit te voeren; het zal ongeveer een half uur duren voordat het voltooid is.

Wanneer het script klaar is, kan het commando `sudo systemctl enable nodered.service` ingevoerd worden via de Terminal zodat Node-RED automatisch start wanneer de Raspberry Pi (al dan niet opnieuw) wordt opgestart. Dit commando kan uitgeschakeld worden door `sudo systemctl disable nodered.service` in te voeren.

De Node-RED server kan gestart worden met `node-red-start` en gestopt met `node-red-stop`. Om de server te stoppen en te herstarten gebruikt u `node-red-restart`. De terminal kan worden afgesloten na het starten van Node-RED; de server zal blijven draaien. Wanneer de terminal opnieuw wordt gestart, kunt u de output-service van de server starten met `node-red-log` en de uitvoer in de terminal bekijken. Dat is handig als u wilt controleren of alles nog werkt zonder een nieuwe instantie van de server te hoeven starten.

Installatie van de vereiste modules in Node-RED

De modules kunnen worden geïnstalleerd in Node-RED door ofwel de Palette Manager in het browservenster te gebruiken of door de npm package manager te gebruiken vanuit de terminal. Mijn aanbeveling is om npm te gebruiken, omdat dit in mijn ervaring betrouwbaarder is. In dat geval moet Node-RED echter van tevoren worden afgesloten. Als de Node-RED service daarentegen op de Raspberry Pi draait, kunt u zich met een browser op de PC aanmelden bij de gebruikersinterface. Hiervoor moet u de adresregel invoeren: `http://<IP-adres van Raspberry Pi>:1880`, waarmee u in de Flow-Editor komt. Hier worden eerst de benodigde modules geïnstalleerd; in de Flow-Editor gebeurt dit via de Palette Manager. Dit zijn de modules die u nodig



Figuur 1. MQTT-node in Node-RED.

Figuur 2. MQTT-node: instelling server-eigenschappen.

Figuur 3. Een nieuwe server toevoegen.

Figuur 4. TLS-configuratie van de nieuwe server.

hebt (sommige zijn al geïnstalleerd bij de installatie van Node-RED):

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

Hier is enig geduld vereist bij installatie via de Palette Manager; als de activity monitor onzichtbaar is, duurt het even voordat de succesmelding bovenin het venster verschijnt.

Als de installatie via de Palette Manager niet lukt, sluit de Editor dan opnieuw af, en stop in de terminal de Node-RED service op de Raspberry Pi en doe het handmatig. Bijvoorbeeld, het statement `npm i --unsafe-perm node-red-node-sqlite` installeert de *node-red-node-sqlite* module; voor de andere modules gaat u op dezelfde manier te werk. Als *node-red-node-sqlite* de *node-red-sqlitedb* niet installeert, zal deze achteraf geïnstalleerd moeten worden. Dit is alleen mogelijk via npm, omdat de Palette Manager in dit geval een conflict detecteert met de *node-red-node-sqlite* module.

De ontbrekende TTN-module

Tot zover alles goed. Op dit punt is het u misschien opgevallen dat de TTN-module (The Things Network) die in het oorspronkelijke artikel werd besproken, ontbreekt. De reden hiervoor is dat deze niet wordt ondersteund door de Raspberry Pi-omgeving. Als *node-red-contrib-ttn* eenmaal is geladen, is het bijvoorbeeld niet meer mogelijk om Node-RED-Service op de Raspberry Pi te starten. Het kostte me een tijdje om dit uit te zoeken, maar ik ben blij dat ik dit nu kan doorgeven om u de stress van het oplossen van dat specifieke probleem te besparen. De oplossing ligt in de *mqtt*-module die in Node-RED is ingebouwd. Deze is opgenomen in de origin flow in plaats van de TTN uplink-module. U kunt mijn flow, die speciaal is aangepast voor de Raspberry Pi, downloaden van de LoRa GPS Tracker pagina [5] bij Elektor Labs. De flow wordt geïmporteerd door simpelweg het gedownloade JSON-bestand naar het Flow Editor-venster te slepen. Deze voor de Raspberry Pi aangepaste flow draait overigens ook op de PC in plaats van de flows met de speciale TTN-nodes.

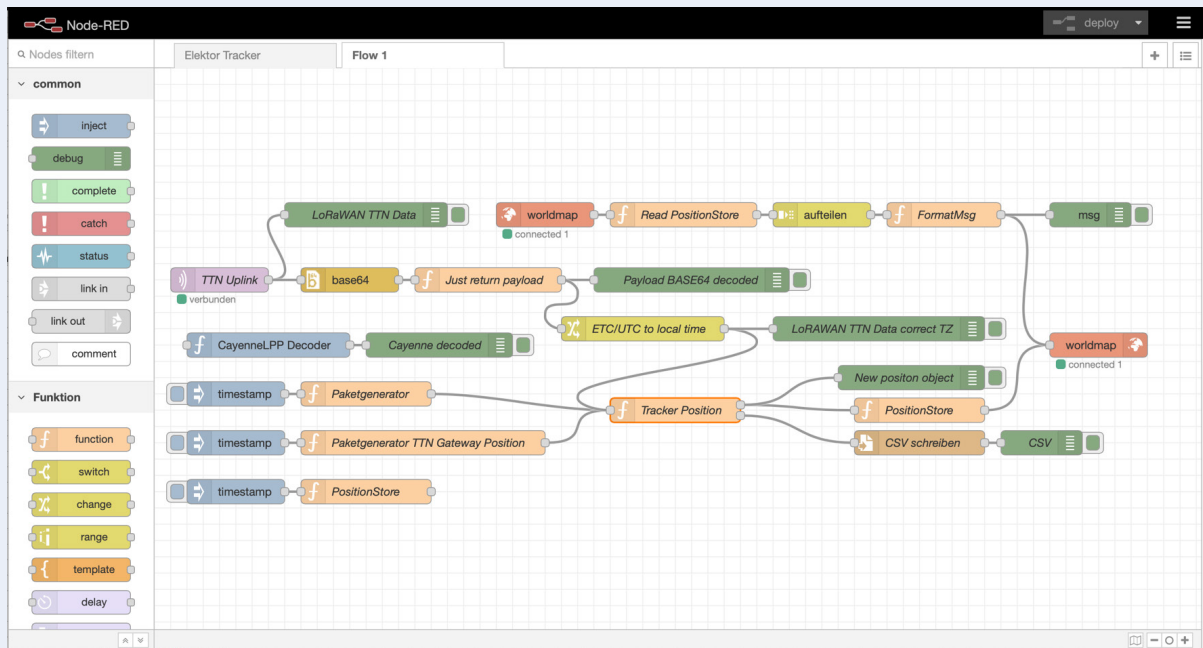
In de *Flow Editor* kunt u nu dubbelklikken op het *mqtt*-symbool (figuur 1) en de instellingen bewerken in het venster dat wordt geopend (figuur 2). Nu wordt "v3/+devices/+up" ingevuld onder *Topic* zoals in de figuur is aangegeven, om de gegevens van alle apparaten in de TTN-applicatie te ontvangen. Als *Output* kan de standaardconfiguratie "a parsed JSON object" blijven staan. Een nieuwe server kan worden ingevoerd in het *Server*-veld door op het potlood-symbool te klikken, waarna het venster van figuur 3 wordt geopend.

De Application ID uit de TTN Console wordt gebruikt als *Name*, en *eu1.cloud.thethings.network* is de *Server*. De standaardpoort is 1833 (MQTT zonder TLS) die moet worden gewijzigd in 8883. Wat ten slotte ontbreekt is het vinkje naast *Enable secure (SSL/TLS) connection*. Een klik op het potloodsymbool van het *TLS Configuration*-veld opent het volgende venster (figuur 4).

Hier worden de instellingen geaccepteerd met een klik op *Add*. Terug in het *mqtt-broker config node*-venster selecteert u het *Security*-tabblad



Figuur 5. Beheer van de Access Key van de MQTT-nodes.



Figuur 6. Het voltooid stroomdiagram in Node-RED.

(figuur 5). De *Username* is de Application ID uit de TTN-console waar *@ttn* aan is toegevoegd. Een Access Key met ten minste "Messages Right" moet worden gegenereerd als wachtwoord in de TTN-console. De uit de TTN-console gekopieerde Access Key wordt vervolgens bij *Password* geplakt en bevestigd door op *Add* te klikken. Terug in het *mqtt in* element kunt u het element een naam geven (bijvoorbeeld *TTN uplink*); met een klik op *Done* is de taak voltooid.

Figuur 6 toont het bijgewerkte Node-RED flowdiagram. De CayenneLPP-decoder die in de flow te zien is, is daar opgenomen voor toekomstige systeemuitbreiding

210120-03

Een bijdrage van

Idee en tekst: Hans Schneider

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

Vertaling: Eric Bogers

Layout: Harmen Heida

Vragen of opmerkingen?

Hebt u technische vragen of opmerkingen naar aanleiding van dit artikel? Stuur een e-mail naar de auteur via hans.schneider@belgacom.net of naar de redactie van Elektor via redactie@elektor.com.



GERELATEERDE PRODUCTEN

- **Elektor LoRa Node – Partly Assembled Module (SKU 19175)**
www.elektor.nl/19175
- **SeeedStudio RFM95 Ultra-long range LoRa Transceiver Module (868 MHz) (SKU 18715)**
www.elektor.nl/18715
- **Dragino PG1301 LoRaWAN GPS Concentrator for Raspberry Pi (868 MHz) (SKU 19367)**
www.elektor.nl/19367

WEBLINKS

- [1] M. Claußen, "LoRa GPS-tracker", Elektor Magazine november/december 2020: www.elektormagazine.nl/magazine/elektor-161/59164
- [2] Pi-hole: <https://pi-hole.net>
- [3] Homebridge: <https://homebridge.io>
- [4] Flightradar24: www.flightradar24.com/
- [5] "LoRa GPS Tracker" op Elektor Labs: www.elektormagazine.com/labs/lora-gps-tracker

