



Met de vos in het IoT (4)

inrichten van het dashboard

Frank Schleking en Bernd vom Berg (Duitsland)

We sluiten deze korte serie af met het inrichten van een op maat gesneden dashboard voor grafische en numerieke visualisatie van de geregistreerde en via Sigfox doorgestuurde meetwaarden.

Nadat we in de vorige delen hebben kennism gemaakt met de Sigfox-basishardware voor data-acquisitie en -transmissie en het basisconcept van de datatransmissie van Sigfox hebben besproken, zijn onze meetwaarden nu beschikbaar op het Sigfox-backend voor verdere verwerking.

We gaan een gratis dashboard-tool op het web gebruiken, waarmee gebruikers een eigen visualisatie- en evaluatie-interface kunnen ontwerpen en gebruiken op een PC, laptop, tablet en/of smartphone. Na vrijgave is dit dashboard wereldwijd toegankelijk en kan het door iedereen, of door elke geautoriseerde persoon, in het internet worden bekeken.

We bespreken dit aan de hand van de Arduino-sketch *Sigfox-3_0.ino* voor het MKR FOX 1200-board. Als we daar menupunt 4 *Send out three readings* kiezen, dan worden drie meetwaarden (luchtdruk, luchttemperatuur en LDR-helderheid) continu geregistreerd en weergegeven op het ePaper-display van het moederboard en op de seriële monitor van de Arduino-IDE. Met een druk op de knop worden deze waarden automatisch via het Sigfox-netwerk verstuurd, door het Sigfox-backend ontvangen en opgeslagen. En van daaruit gaan de gegevens door naar de dashboard-visualisatie.

Het dashboard-platform **thinger.io**

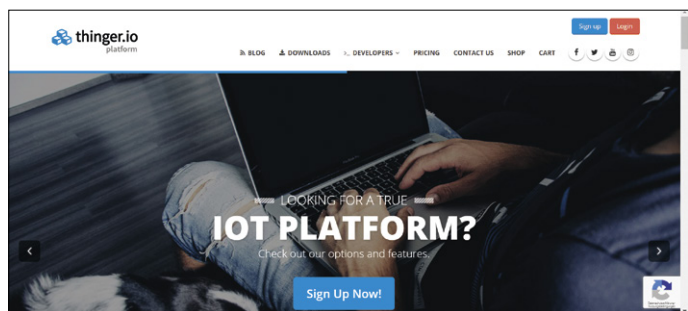
thinger.io is een open source IoT-visualisatieplatform dat het mogelijk maakt om snel en eenvoudig duidelijke, fraaie visualisaties van gegevens te maken. Het is voor kleine projecten zelfs gratis.

Het systeem ondersteunt helaas (nog) geen wiskundige bewerkingen met de ontvangen gegevens, dus de meetwaarden moeten in kant-en-klare vorm worden aangeleverd. Maar dat is geen probleem voor onze Arduino-software op de MKR FOX1200. De waarden worden gevisualiseerd op zogenaamde dashboards, die ook via een webbrowser openbaar kunnen worden gemaakt.

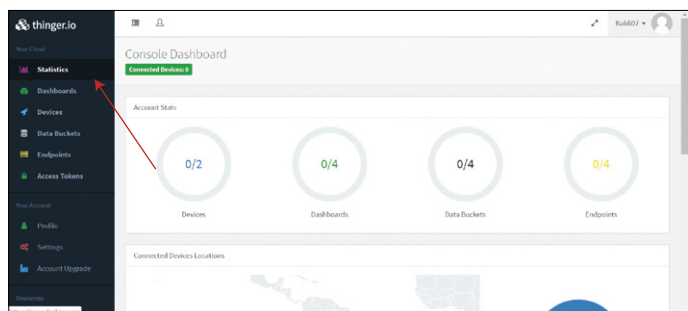
De noodzakelijke stappen

Het ontwikkelen van een eigen PC-dashboard gaat in een paar eenvoudige stappen:

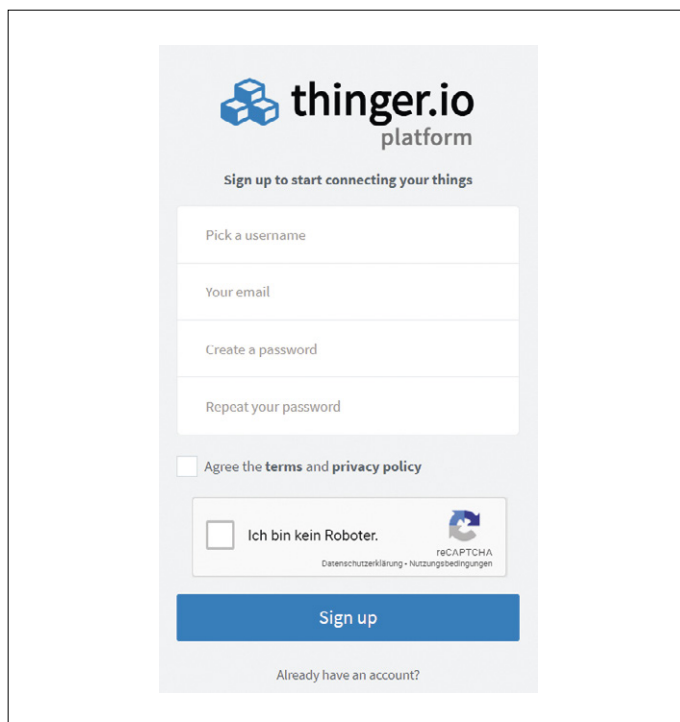
- Maak een gratis gebruikersaccount aan bij thinger.io.
- Maak een datacontainer (*data bucket*) op thinger.io die de door het Sigfox-backend verzonden meetwaarden opneemt.
- Leg een *token* voor het Sigfox-backend vast en maak een



Figuur 1. Homepage van thinger.io



Figuur 3. Hoofdscherm van het account (Statistics).



Figuur 2. Invoervenster voor het aanmaken van een eerste account.

ontvangstauthenticatie (*access token*) op thinger.io, zodat het Sigfox-backend toestemming krijgt om gegevens over te dragen naar thinger.io.

- Configureer een *callback* op het Sigfox-backend voor het overdragen van data van het Sigfox-backend naar de *data bucket* in thinger.io.
- Ontwerp van een grafisch aantrekkelijk dashboard op de thinger.io-pagina om de gegevens te presenteren.

Een gebruikersaccount aanmaken bij thinger.io

Voor het gebruik van thinger.io moeten we eerst een (gratis) gebruikersaccount aanmaken. Ga daarvoor naar de homepage van thinger.io [1] (**figuur 1**) en klik op *Sign up* in de rechterbovenhoek om naar het invulformulier voor nieuwe gebruikers te gaan (**figuur 2**). Vul de velden in en klik op *Sign up*.

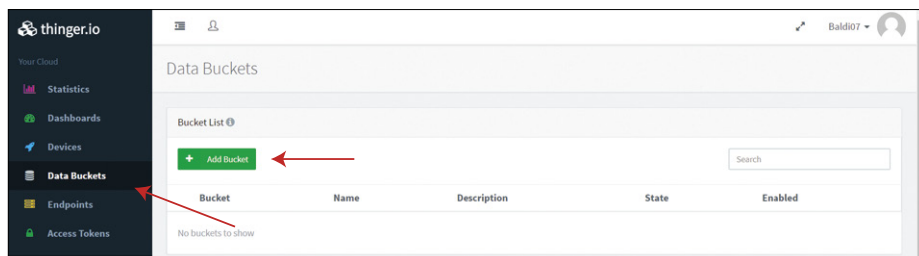
De mogelijkheden met een gratis account zijn weliswaar beperkt, maar ruim voldoende voor de meeste kleine Maker-projecten. Een gedetailleerd overzicht van de mogelijkheden vindt u op het tabblad *Pricing*.

Na bevestiging van uw email-adres wordt u naar de loginpagina gebracht en kunt u zich registreren. U komt dan op het hoofdscherm *Statistics* van uw account, van waaruit u alle verdere acties kunt gaan uitvoeren (**figuur 3**).

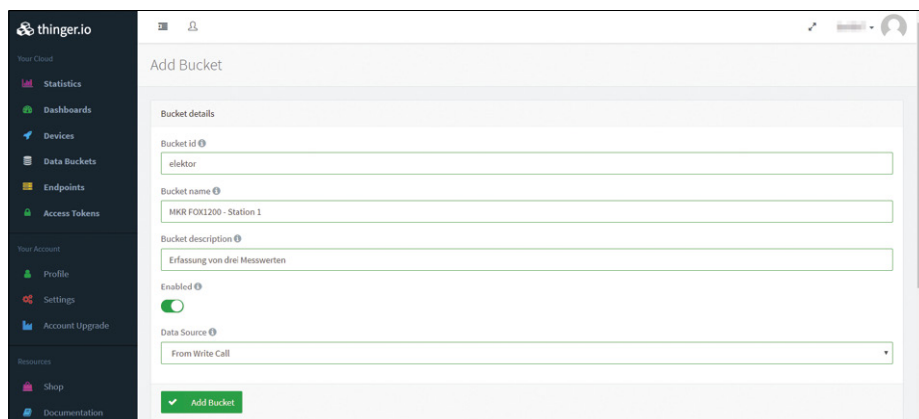
Aan de linkerkant ziet u het hoofdmenu met alle functies die nodig zijn om een visualisatie voor IoT-projecten te maken.

Aanmaken van een datacontainer bij thinger.io

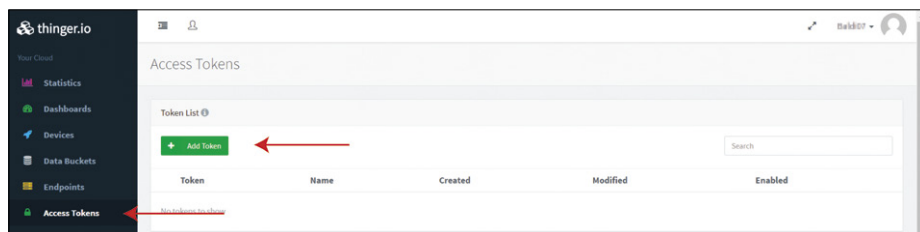
De eerste stap is het maken van een *Data Bucket* (bucket = emmer), een database (datacontainer) waar de meetwaarden



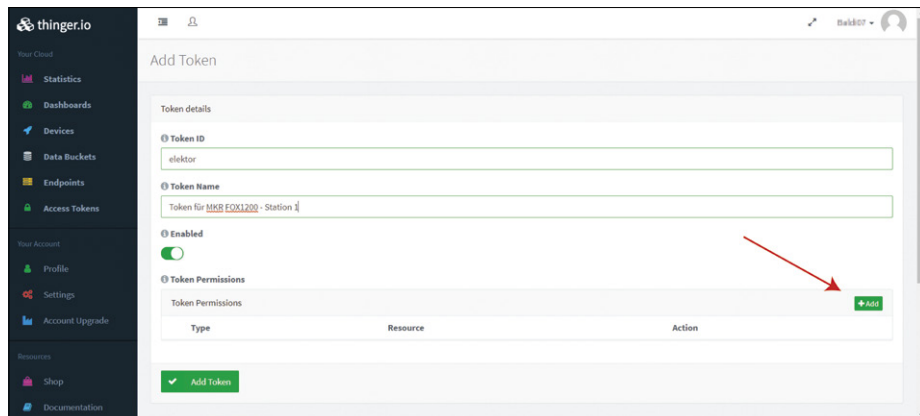
Figuur 4. Het aanmaken van een nieuwe Data Bucket.



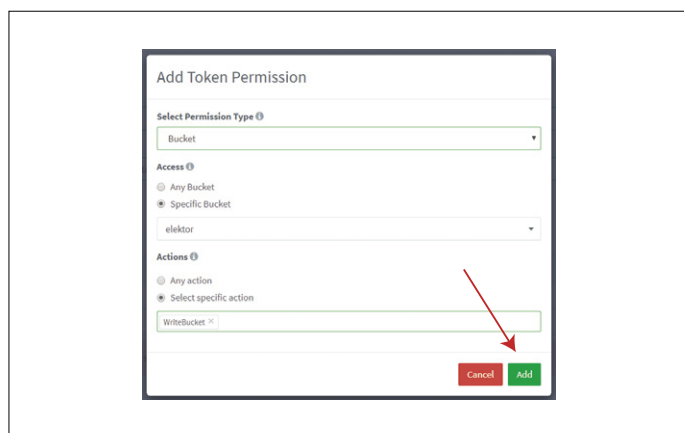
Figuur 5. Definieren van de eigenschappen van de Data Bucket.



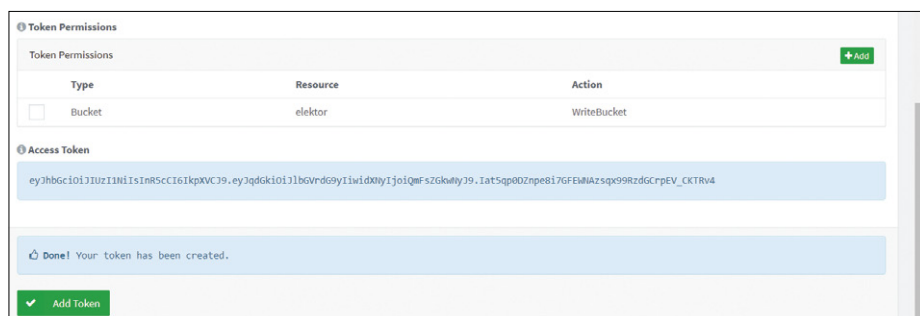
Figuur 6. Aanmaken van een nieuw token (toegangspunt).



Figuur 7. Definitie van de eigenschappen van het token.



Figuur 8. Configuratie van de Token Permission (toegangsrechten).



Figuur 9. Het token is aangemaakt, geactiveerd en de toegangsrechten zijn toegekend.

kunnen worden opgeslagen en uitgelezen. Het Sigfox-backend schrijft de meetwaarden automatisch naar deze database via de Sigfox-callback. Klik in het hoofdscherm op *Data Buckets* aan de linkerkant en vervolgens op *Add Bucket* (figuur 4). Het venster voor het invoeren van de eigenschappen van de Data Bucket verschijnt (figuur 5). De *Bucket id* is een unieke identificatie binnen het gehele account, die later nodig is voor de inrichting met Sigfox. We voeren hier bijvoorbeeld "elektor" in. In de velden *Bucket name* en *Bucket description* kunnen we gebruikersvriendelijke informatie invoeren. Kies als gegevensbron in het keuzeveld *Data Source* via het pull-downmenu *From Write Call*. Dat betekent dat de gegevens van het Sigfox-backend via internet naar thinger.io worden gestuurd door een Sigfox-callback. Dus zodra de Sigfox-callback data verstuurt, wordt die hier weggeschreven. Klik ter afsluiting op *Add Bucket*.

Toegangspunt op thinger.io voor de gegevens van het Sigfox-backend

Natuurlijk moet het Sigfox-backend weten waar het de geregistreerde meetdata naartoe moet sturen, met andere woorden: u moet het Sigfox-backend het ontvangstadres op thinger.io geven. Dat is de tweede stap. We maken een *token* (toegangspunt) met de juiste autorisatie voor de *Data Bucket*. Klik in het hoofdscherm aan de linkerkant op *Access Tokens* en vervolgens op *Add Token* (figuur 6). Er verschijnt dan een venster voor het invoeren van de eigenschappen van het token (figuur 7). Hier wijst u eerst een uniek *Token ID* toe, bijvoorbeeld "elektor". Onder *Token Name* kan weer elke beschrijving of naam worden ingevoerd.

Door op *Add*, het kleine groene veld rechtsonder, te klikken wordt een *Token Permission* toegevoegd, die heel specifiek geconfigureerd moet worden (figuur 8).

In het veld *Select Permission Type* selecteren we *Bucket*. Klik in het veld *Access* op *Specific Bucket* en selecteer dan de eerder aangemaakte bucket "elektor" in het dropdown-menu.

Onder *Actions* selecteren we *Select specific action* en in het (dropdown-) veld daaronder voegen we de optie *WriteBucket* toe. Gebruik tenslotte de knop *Add* om een token (schrijfrechten) toe te wijzen voor de gewenste doel-bucket "elektor". Dit token wordt nu vermeld in de lijst met *Token Permissions* (figuur 7). Als u nu op de knop *Add Token* (figuur 7, links-onder) klikt, wordt dit toegangspunt (ontvangstadres voor de gegevens van het Sigfox-backend) in thinger.io aangemaakt (figuur 9).

Belangrijk: het blauwe veld *Access Token* toont de toegangsautorisatie (authenticatie) voor het Sigfox-backend die automatisch door thinger.io wordt gegenereerd.

We moeten het Sigfox-backend straks zó configureren dat het deze toegangscode meestuurt naar het thinger.io-bestemmingsadres, zodat thinger.io de verzonden gegevens ook accepteert en opslaat in de bijbehorende *Data Bucket* "elektor". U moet deze toegangscode ergens veilig bewaren (knippen&plakken).

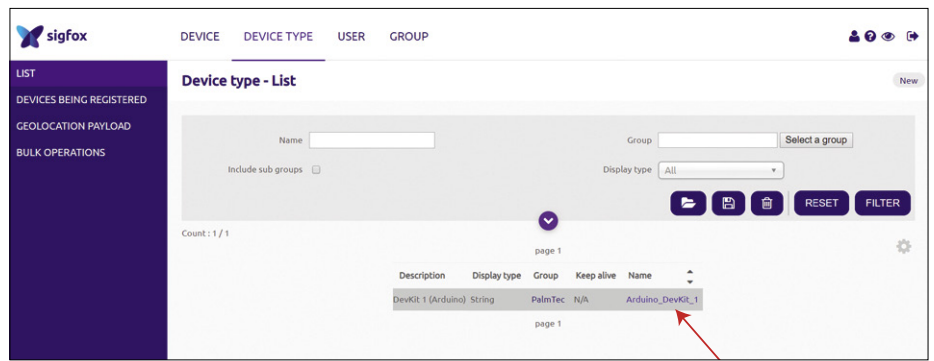
Hiermee zijn de instellingen bij thinger.io voorlopig afgerond, zodat we terug kunnen keren naar het Sigfox-backend.

Het aanmaken van de callback in het Sigfox-backend

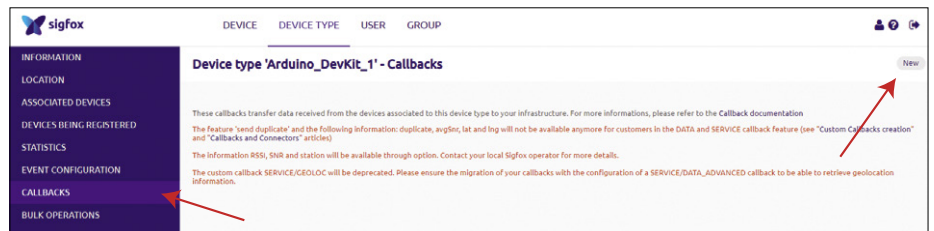
Als de MKR FOX 1200 data naar het Sigfox-backend stuurt, moet die automatisch worden doorgestuurd naar thinger.io voor verdere verwerking (visualisatie). Het Sigfox-backend heeft daarvoor de al eerder genoemde *callbacks*. Zo'n *callback* is niets anders dan een automatische doorstuur-service die altijd wordt uitgevoerd als het Sigfox-backend data heeft ontvangen van een Sigfox-station zoals onze MKR FOX 1200. Natuurlijk moet het backend correct worden geconfigureerd om ervoor te zorgen dat de gegevens de gewenste ontvanger bereiken.

Om de verzonden gegevens over te brengen naar onze thinger.io-Data Bucket "elektor", maken we gebruik van het token resp. van de *access token*-code die we bij thinger.io hebben gegenereerd. Met andere woorden: de backend heeft via het internet toegang tot het token via een http-verbinding en schrijft de gegevens naar de data bucket na succesvolle identificatie met het access token.

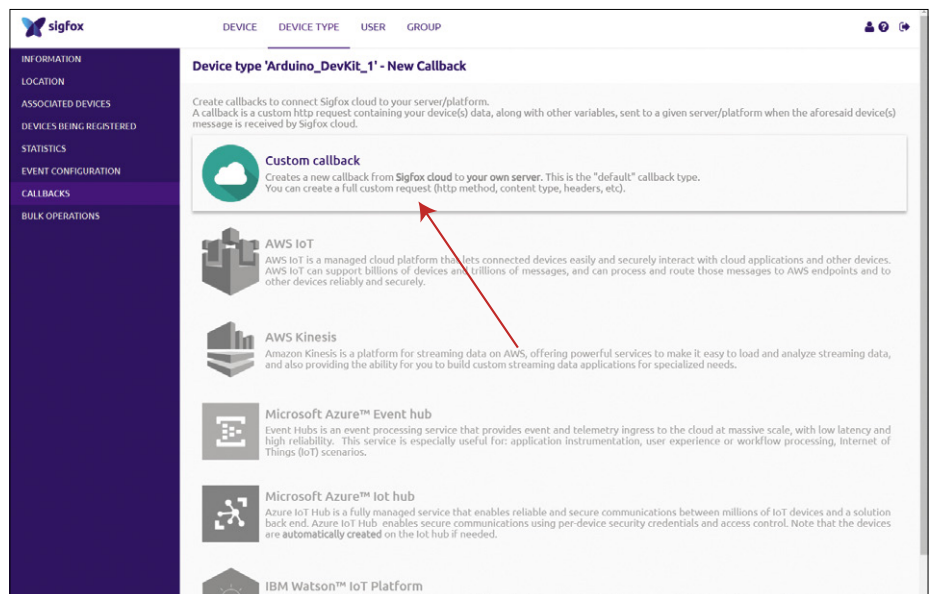
Om de *callback* aan te maken, logt u in op het Sigfox-backend [2]. In het hoofdscherm klikt u op het tabblad *Device Type* en in de lijst die verschijnt (met slechts één vermelding) klikt u op de naam van het apparaattype *Arduino_DevKit_1* (figuur 10). Dan verschijnt de overzichtspagina van dit apparaat. Klik op *Callbacks* in de lijst aan de linkerkant (figuur 11). Als u op de kleine knop *New* in de rechterbovenhoek van dit venster klikt, verschijnt er een lijst met alle mogelijke soorten callback-acties voor de meest voorkomende typen dashboard/cloud-programma's (figuur 12). Kies hier voor *Custom Callback*. In het venster dat dan verschijnt, kunt u de callback en het verzenden



Figuur 10. Selectie van het gewenste apparaattype.



Figuur 11: Het callback-venster.

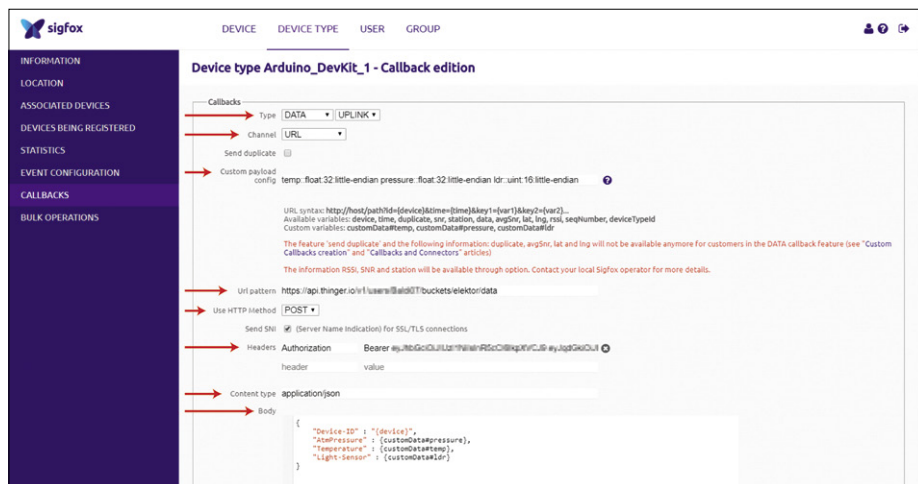


Figuur 12. Het brede scala aan mogelijke callback-types.

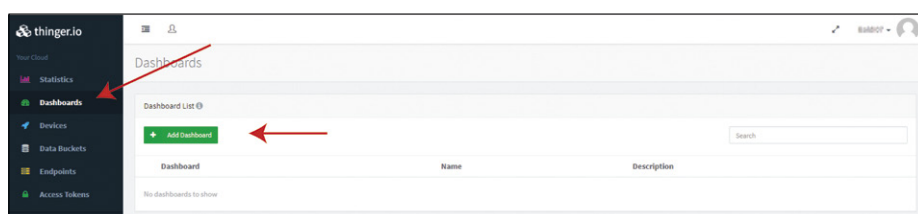
van gegevens van het Sigfox-backend naar thinger.io configureren (figuur 13).

Let op: hier moet u zeer zorgvuldig te werk gaan en alle gegevens precies zo invoeren als hieronder aangegeven, want zelfs de kleinste afwijkingen leiden tot foutmeldingen of niet-functioneren!

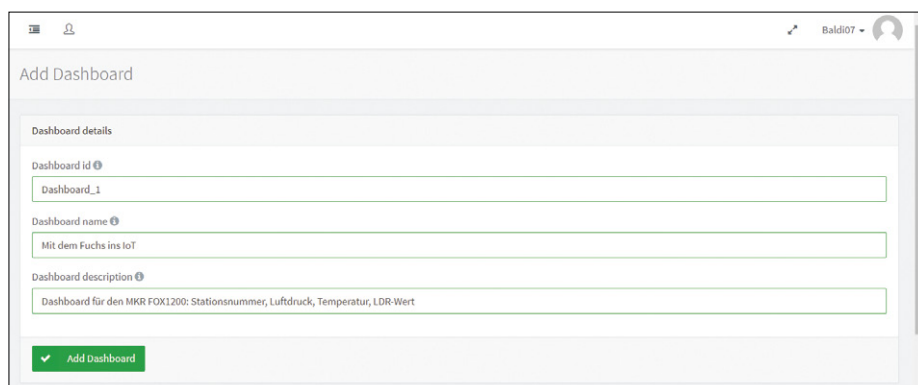
Een diepgaande beschrijving van de afzonderlijke specificaties zou buiten het bestek van dit artikel vallen, maar u vindt zeer gedetailleerde informatie hierover onder [3], [4] en [5]. Laten we bovenaan figuur 13 beginnen. Stel bij *Type DATA* en *UPLINK* in, kies bij *Channel* voor *URL* omdat we ons target, het thinger.io-token, via een URL aanspreken.



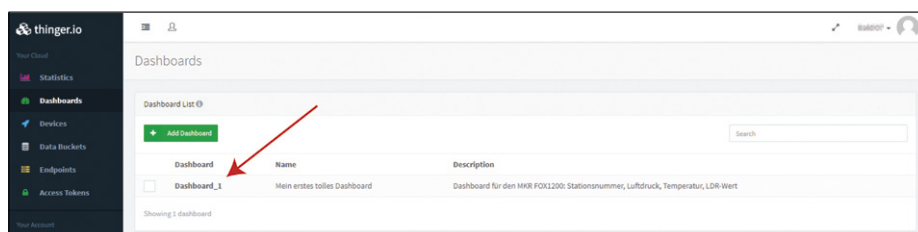
Figuur 13. Configuratie van de callback.



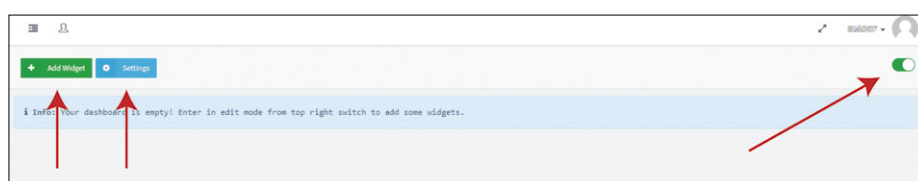
Figuur 14. Op weg naar het dashboard.



Figuur 15. Basisgegevens van het dashboard.



Figuur 16. Lijst van bestaande dashboards.



Figuur 17. In de Dashboard-bewerkingsmodus.

Nu wordt het wat ingewikkelder: de meetwaarden (data) die de MKR FOX 1200 als payload naar het Sigfox-backend stuurt, worden bij het Sigfox-backend eerst opgesplitst in individuele waarden, waarbij elke waarde in het backend een (willekeurige) interne variabelenaam krijgt. Deze namen (en de bijbehorende waarden) kunnen dan in het backend worden gebruikt voor verdere verwerking. Deze ontbinding van de Sigfox-payload is gedefinieerd in het veld *Custom payload config* en is uiteraard afhankelijk van de structuur van de MKR-FOX 1200-payload: de volgorde van de data in de Sigfox-payload, het type data en het type dataopslag bepalen de opsplitsing in individuele variabelen.

Voor ons voorbeeld ziet de *Custom payload config*-opsplitsing er daarom **exact** als volgt uit:

```
temp::float:32:little-endian
pressure::float:32:little-endian
ldr::uint:16:little-endian
```

De eerste waarde die door de MKR FOX 1200 in de payload wordt verzonden, is de gemeten temperatuur. Deze waarde wordt *temp* genoemd in het Sigfox-backend en heeft het datatype *float*, dat wordt weergegeven door vier bytes (32 bits).

De tweede waarde in de payload is de luchtdruk (*pressure*), ook dat is een *float* van 32 bits.

De derde en laatste gemeten waarde in de payload is de spanning over de LDR. Die wordt nu in de backend opgeslagen onder de naam *ldr* als een *uint* van 16 bits (2 bytes).

De term "little-endian" geeft aan in welke volgorde de afzonderlijke bytes van de numerieke waarden worden opgeslagen. Deze komt overeen met de volgorde die de Atmel-controller gebruikt. Voor meer informatie zie [6]. Voer deze drie specificaties, gescheiden door spaties, in op één regel in het corresponderende veld. U kunt met deze drie variabelen in het Sigfox-backend werken. Ze kunnen ook naar thinger.io gestuurd worden in precies deze vorm.

Onder *Url-pattern* wordt de doel-URL op thinger.io gespecificeerd. Die heeft de volgende structuur:

`https://api.thinger.io/v1/users/{user_id}/buckets/
{bucket_id}/data`

In dit adres is `user_id` de gebruikersnaam van het thinger.io-account (figuur 2) en `bucket_id` de bucket ID, d.w.z. "elektor" (figuur 5) (beide zonder accolades invoeren).

Kies bij *Use http Method* voor *POST*.

De volgende stap is het specificeren van een authenticatie voor de gegevensoverdracht naar thinger.io. Dit gebeurt in de twee velden naast *Headers*. In het veld *header* wordt het woord "Authorization" ingevoerd, en daarnaast in het veld *value* het woord "Bearer", een spatie en daarachter het gegenereerde *Access Token* uit figuur 9:

Bearer eyJhbGciOiJI1NiIsIn ... enz.

In het veld *Content type* voert u "application/json" in, aangezien de gegevens in JSON-formaat zijn geformatteerd.

In deze laatste stap legt u vast wat er precies naar thinger.io gestuurd moet worden. De gegevens (in JSON-formaat) in het veld *Body* moeten **precies** zo worden ingevoerd als beschreven:

```
{  
  "Device-ID" : "{device}",  
  "AtmPressure" : {customData#pressure},  
  "Temperature" : {customData#temp},  
  "Light-Sensor" : {customData#ldr}  
}
```

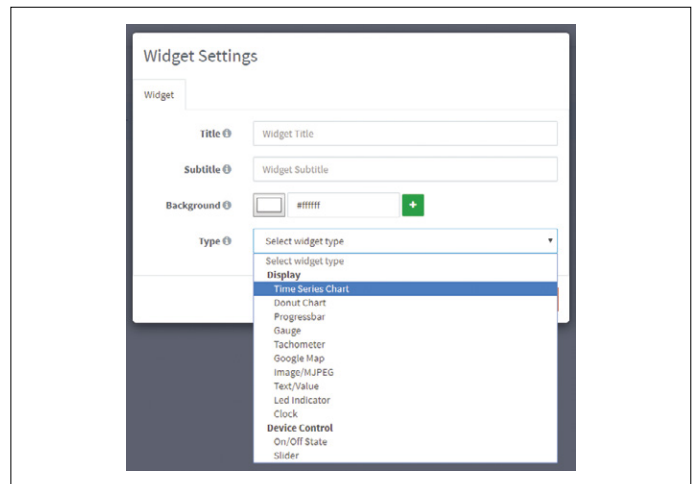
De *Body* staat tussen accolades {...}. Wat betekent deze informatie?

Elke regel tussen de accolades begint met een aanhalingsteken ", gevolgd door een willekeurige maar unieke variabeernaam, die weer wordt afgesloten met een aanhalingsteken. Voer daarna een spatie, een dubbele punt en nog een spatie in en daarna de waarde voor de variabele (de waarde uit het Sigfox-backend). De naam en de waarde van de variabele worden dan later op deze manier naar thinger.io gestuurd en opgeslagen in de bijbehorende Data Bucket.

De betekenis is als volgt:

- "Device-ID" : "{device}"
is het unieke apparaatnummer van het Sigfox-station (onze MKR FOX 1200), dat beschikbaar is in het backend. We sturen deze informatie met de meetwaarden mee, zodat het station later duidelijk kan worden geïdentificeerd.
- "Atmpressure" : {customData#pressure}
stuurt een variabele met de naam *AtmPressure*. De waarde van deze variabele komt overeen met de waarde van de backend-variabele *pressure*, die we eerder hebben gedefinieerd in de *Custom payload config*. Het sleutelwoord *customData#* geeft aan dat de waarde van een variabele uit het *Custom payload config*-veld moet worden gebruikt. Met andere woorden: de naam van deze variabele en de bijbehorende waarde worden overgedragen naar thinger.io.
- De laatste twee regels werken op dezelfde manier. Elke regel (behalve de laatste) moet eindigen met een komma.

Hiermee is de inrichting van de callback op het Sigfox-backend klaar om de gewenste waarden over te brengen naar de Data



Figuur 18. Basisvenster voor het maken van een widget.

Bucket "elektor" op thinger.io. Als alles goed is ingevuld, klikt u op *OK*.

Als nu onze MKR FOX1200 zijn payload naar het Sigfox-backend stuurt en die daar zonder fouten is ontvangen, stuurt het Sigfox-backend deze gegevens **direct en onmiddellijk** naar thinger.io voor verdere verwerking in de vorm die we zojuist hebben geconfigureerd.

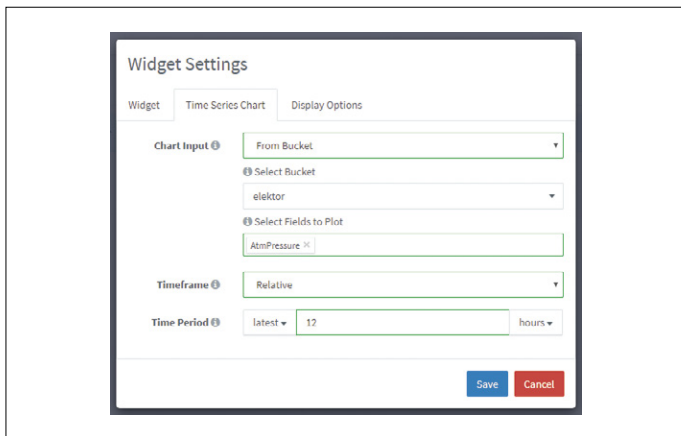
Het dashboard bij thinger.io

Nu komen we bij de laatste stap, de afwerking van onze applicatie. De gegevens moeten in een grafisch aantrekkelijke vorm op het dashboard worden gepresenteerd. Om een dashboard aan te maken, klikt u op het item *Dashboards* aan de linkerkant in de hoofdconsole van thinger.io (figuur 3) en in het volgende venster op de knop *Add Dashboard* (figuur 14).

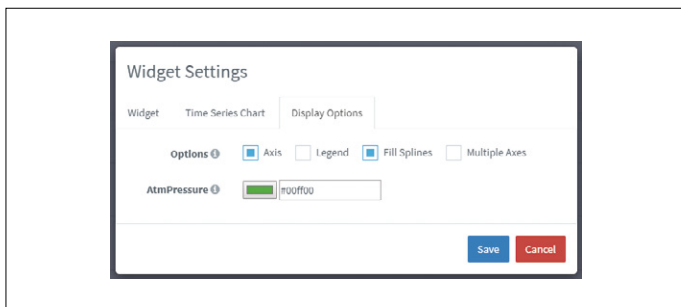
In het venster *Dashboard details* (figuur 15) vult u de velden in: de *Dashboard id* is een unieke identifier binnen het gehele account. In de velden *Dashboard name* en *Dashboard description* kunt u namen invoeren die voor de gebruiker gemakkelijk te begrijpen zijn. Sluit dan de invoer af met *AddDashboard*. Het dashboard is dan toegevoegd aan de zogenaamde *Dashboard list*, die u kunt zien als u weer op het menu-item *Dashboards* in het hoofdconsole klikt (figuur 16).

Als u op de naam van het dashboard klikt, wordt de ontwerp-interface geopend en ziet u een leeg dashboard (figuur 17). In de rechterbovenhoek zit een kleine knop die u naar "groen" schakelt om de bewerkingsmodus te openen. Er zijn ook twee knoppen genaamd *Add Widget* en *Settings*. De weergave-elementen (onderdelen van de gebruikersinterface) op een dashboard worden in thinger.io *widgets* genoemd. Er zijn veel widgets beschikbaar, bijvoorbeeld tijdgrafieken, donutgrafieken, staafdiagrammen, analoge displays, Google Maps voor positiegegevens, afbeeldingen, tekstvelden, LED's en een klok. Voor onze toepassing willen we voor elk van de drie gemeten waarden een tijddiagram maken en een numerieke weergave van de laatst verzonden meetwaarde.

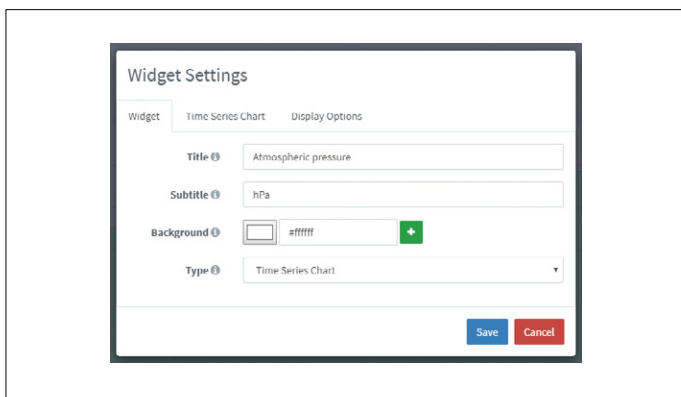
Als u op *Add Widget* klikt, verschijnt er een keuzevenster met instellingen: *Widget Settings* (figuur 18). Onderaan, bij *Type*, kunt u aangeven welk type widget u wilt aanmaken. Hier kiezen we voor *Time-Series-Chart*. Er verschijnen dan twee extra tabbladen (figuur 19). Op het tabblad *Time Series Chart*



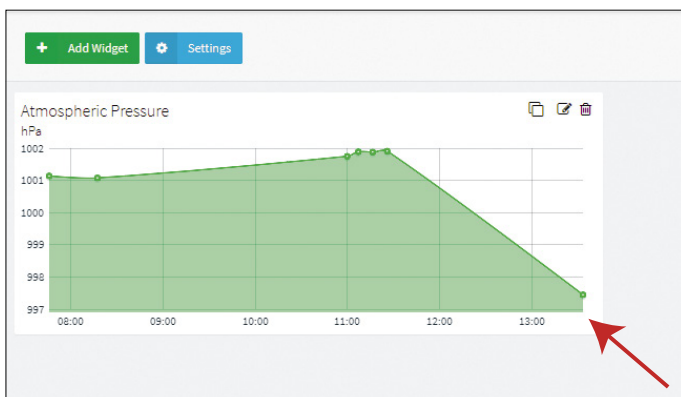
Figuur 19. Configuratie van het tijddiagram.



Figuur 20. Definieren van de basiseigenschappen van het display.



Figuur 21. Definities in het tabblad Widget.



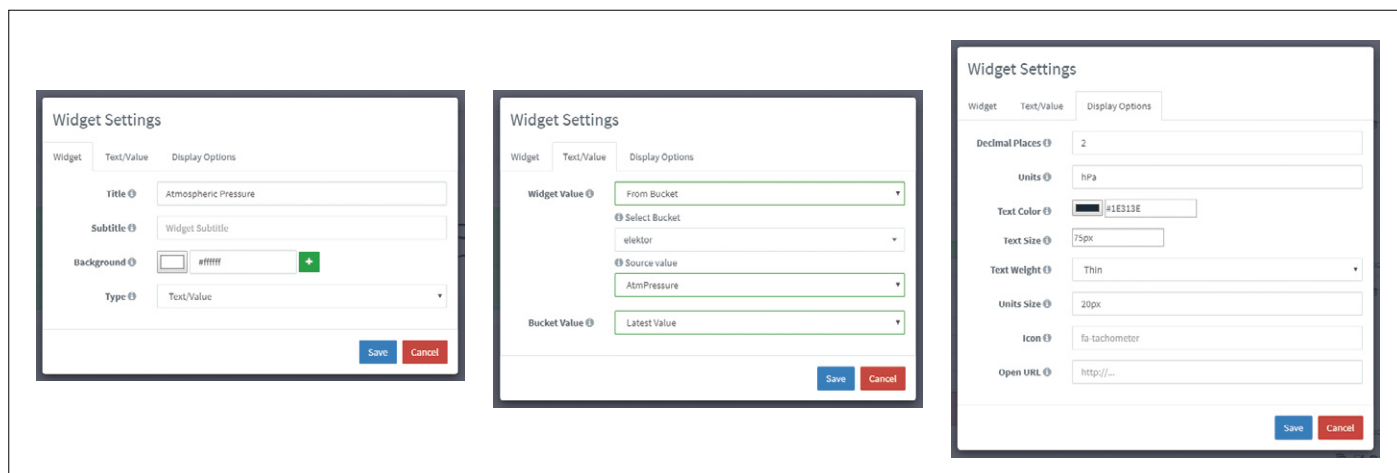
Figuur 22. Het tijddiagram op het dashboard (met enkele meetwaarden).

selecteren we in het veld *Chart Input* de optie *From Bucket*. Er verschijnen nu achtereenvolgens verschillende velden, die we moeten invullen. In het veld *Select Bucket* kiezen we onze bucket "elektor". In het volgende veld *Select Fields to Plot* kiezen we de meetwaarde die we willen weergegeven.

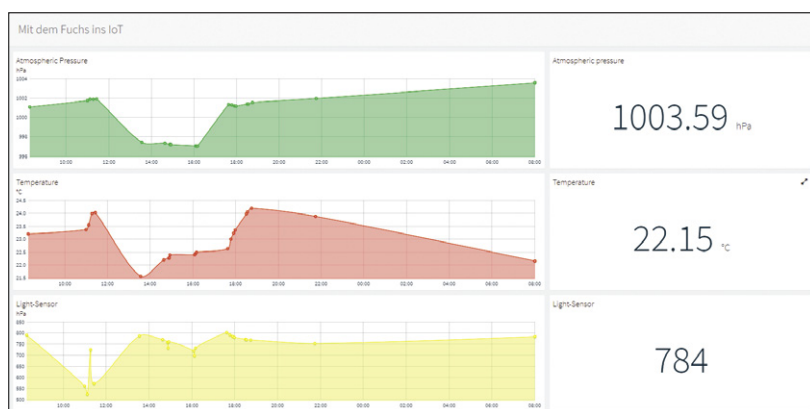
Laten we beginnen met *AtmPressure*. Onder *Timeframe* selecteren we *Relative* en stellen we het tijdvenster in op *12 hours*. Dan worden de waarden van de laatste 12 uur weergegeven in het diagram. Natuurlijk kunt u, als u dat wilt, ook een andere periode kiezen. Klik nu op het tabblad *Display Options* (figuur 20) en schakel *Multiple Axes* uit: we willen voor elke gemeten waarde een apart diagram maken en hebben dus niet meerdere Y-assen in één grafiek nodig. Door de al geactiveerde optie *Axis* worden de X- en Y-as in het diagram automatisch geschaald. In het veld bij *AtmPressure* kunnen we de kleur van de lijn in de grafiek kiezen. Wij hebben gekozen voor een zacht groen. Als u nu *Fill Splines* activeert, wordt het gebied (het oppervlak) tussen de curve en de X-as opgevuld met een kleur die u kiest in het veld *AtmPressure*. Met de optie *Legend* kunnen de afzonderlijke curven in een grafiek worden gelabeld. Dat is natuurlijk alleen zinvol als er meerdere curven in één diagram staan.

We komen nu bij het laatste tabblad *Widget* (figuur 21), waar u een titel en een subtitel voor het diagram kunt invoeren. In het veld *Title* voeren we *Atmospheric pressure* in en bij *Subtitle* de eenheid *hPa*. Indien gewenst kunt u ook de achtergrondkleur van het hele diagram aanpassen onder *Background*, maar we willen niet overdrijven en blijven bij de standaardkleur wit. Door op *Save* te klikken beëindigt u de configuratie van de grafiek en wordt die toegevoegd aan het dashboard (figuur 22). Rechtsonder kan het diagram groter of kleiner worden gesleept en u kunt het ook met de muis over het dashboard verplaatsen. De instelling van de grootte is niet continu, zodat het diagram 'heen en weer' springt als het wordt gesleept. De momenteel gemeten luchtdrukwaarde moet ook numeriek worden weergegeven (figuur 23). Het tabblad *Add Widget* opent opnieuw het menu voor het toevoegen van widgets. Dit keer selecteren we bij *Type* de optie *Text/Value*.

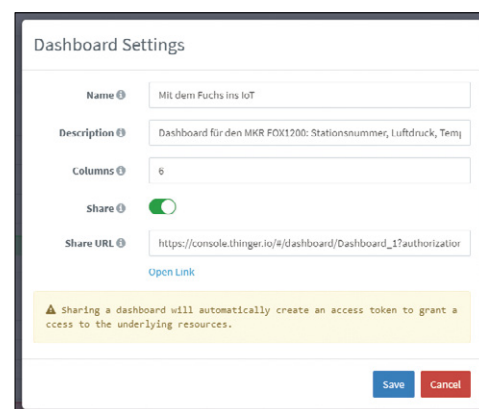
In het veld *Widget Value* voeren we weer *From Bucket* in, selecteren we onze bucket "elektor" en geven we aan welke waarde we willen weergegeven (hier: "AtmPressure"). Voor *Bucket Value* stellen we *Latest Value* in. In het tabblad *Display Options* kunnen verschillende instellingen worden gedaan. We stellen hier bijvoorbeeld twee cijfers achter de komma in (*Decimal Places*) en de eenheid "hPa" in het veld *Units*. Dit wordt later in de widget weergegeven achter de waarde. Indien nodig kunnen hier nog meer instellingen worden gedaan, maar dat doen we nu niet. In het tabblad *Widget* voeren we weer *Atmospheric pressure* in in het veld *Title*. Het veld *Subtitle* laten we hier leeg, want de eenheid wordt al weergegeven achter de waarde. Hiermee zijn de twee dashboarddisplays voor de luchtdrukmeting compleet. De displays voor de andere twee metingen worden op dezelfde manier aangemaakt. Het resultaat kan en moet er uitzien zoals in figuur 24. **Belangrijk:** om alle invoer en wijzigingen op te slaan in thinger.io, moet u de bewerkingsmodus even uitschakelen en vervolgens weer inschakelen! Meer informatie over het ontwerpen van dashboards in thinger.io is te vinden op [7].



Figuur 23. Definitie van de numerieke weergave van de gemeten luchtdrukwaarde.



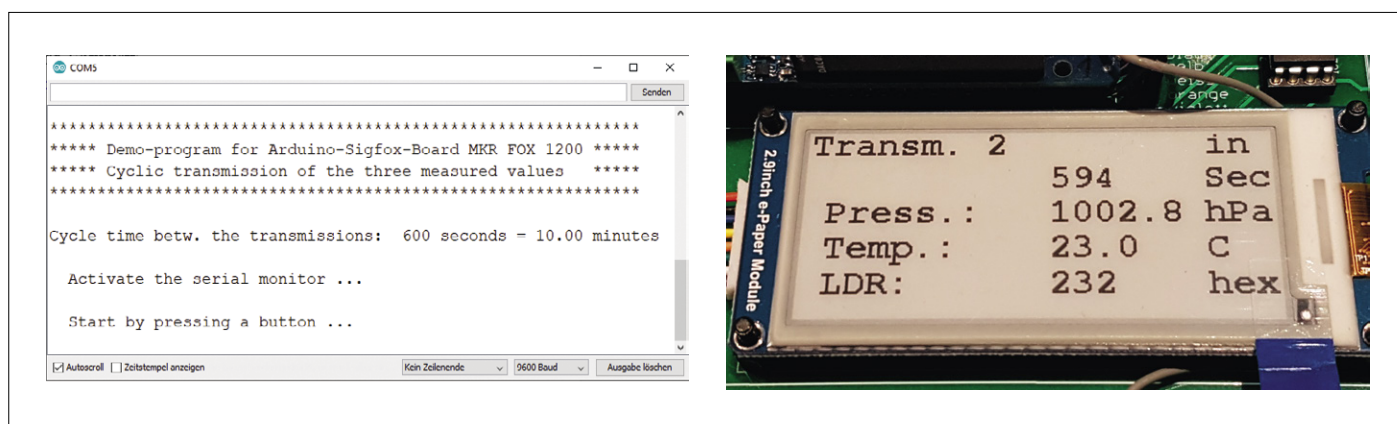
Figuur 24. Het complete dashboard.



Figuur 25. De dashboardinstellingen.

Wat nu nog ontbreekt, is het vrijgeven van het dashboard voor toegang 'van buitenaf'. Klik terug in het dashboard-menu op de blauwe knop *Settings* om toegang te krijgen tot de dashboardinstellingen (**figuur 25**). Hier kunt u een naam en een beschrijving voor het dashboard invoeren (maar dat is niet meer nodig, want dat hebben we al gedaan). Met de knop

Share wordt het dashboard vrijgegeven. De (extreem lange) link naar het dashboard wordt weergegeven in het veld *Share URL*. Via deze link heeft u nu overal ter wereld toegang tot het dashboard en kunt u de waarden bekijken. Omdat de URL erg lang is, is het aan te raden om die in te korten met behulp van een URL-shortener zoals bitly.com. Deze verkorte URL kan



Figuur 26. De uitvoer van Sigfox-Demo.ino in de seriële monitor en op het ePaper-display.



Figuur 27. Maakt ook op de smartphone een goede indruk!

worden omgezet in een QR-code die door smartphones kan worden gescand. Zo hoeft u niets in te typen op uw smartphone. Het is ook belangrijk dat dit gedeelde dashboard van thinger.io automatisch alleen-lezen is voor het publiek. Er kunnen dus geen instellingen en wijzigingen worden aangebracht door derden. Dat is nog steeds alleen mogelijk na inloggen via het thinger.io-portaal.

Als we het hele systeem uitvoerig willen testen, is het een beetje omslachtig om telkens opnieuw met een druk op de knop één Sigfox-telegram te versturen vanuit onze Arduino-sketch *Sigfox-3_0.ino*. Daarom is er nog een andere sketch, *Sigfox-Demo.ino*, in het softwarepakket opgenomen [8]. Deze registreert automatisch elke tien minuten de drie meetwaarden en stuurt ze door (**figuur 26**). Het interval is in de programmatekst van de sketch gedefinieerd met de globale variabele `w_zeit` en kan naar believen worden gewijzigd. Dit maakt metingen over een lange termijn heel gemakkelijk. Maar vergeet niet dat we in het Sigfox-netwerk maximaal 140 telegrammen per dag mogen verzenden, wat overeenkomt met ongeveer één telegram per 10 minuten. Het maakt dan ook niet meer zoveel uit, dat u met het gratis account van thinger.io maar één keer per minuut naar de Bucket mag schrijven. **Figuur 27** laat zien hoe mooi ons zojuist gemaakte dashboard eruit ziet op het scherm van een smartphone.

Tot slot

We hebben nu het einde bereikt van onze kleine serie over de wereld van het 0G-netwerk Sigfox. Dankzij de flexibiliteit en veelzijdigheid van de Arduino MKR FOX 1200 kunnen we een breed scala aan gegevens verzamelen en deze wereldwijd verzenden via het Sigfox-netwerk. Met de ook zeer eenvoudige en uitgebreide ontwerp mogelijkheden van thinger.io kunnen we snel en effectief dashboards ontwerpen voor de communicatie met de gebruiker.

We wensen u veel plezier en succes met Sigfox bij uw uitstapjes op het Internet of Things! ◀

190281-D-04



IN DE STORE

→ Arduino MKR FOX 1200
www.elektor.nl/19096

→ Arduino-antenne 868 MHz
www.elektor.nl/19095

Weblinks

- [1] Thingy: <http://www.thinger.io/>
- [2] Sigfox-backend: <https://backend.sigfox.com/>
- [3] Informatie over de callback-configuratie van Sigfox: <https://support.sigfox.com/docs/callbacks-documentation>
- [4] Informatie over de callback-configuratie van Sigfox: <https://support.sigfox.com/docs/custom-callback-creation>
- [5] Informatie over de callback-configuratie van Sigfox: <https://backend.sigfox.com/apidocs/callback>
- [6] Bytevolgorde voor het opslaan van numerieke waarden: <https://nl.wikipedia.org/wiki/Endianness>
- [7] Informatie over het ontwerpen van dashboards:
<https://github.com/thinger-io/Docs/blob/gh-pages/console/dashboards.md>
- [8] Projectpagina bij dit artikel: <https://www.elektor.nl/190281-D-04>