



# SigFox : un renard sur l'internet des objets (4)

## Composition d'un tableau de bord

Frank Schleking & Bernd vom Berg (Allemagne)

Notre courte série d'articles se terminera par la création d'un tableau de bord personnalisé pour afficher sous forme de graphiques et de chiffres les mesures envoyées sur le réseau Sigfox. Il existe pour cela un outil gratuit de tableau de bord basé sur la navigation HTML. Il permet à l'utilisateur de créer et d'utiliser une interface de visualisation et de surveillance personnalisée sur n'importe quel PC, ordinateur portable, tablette ou téléphone tactile.

Dans les précédents épisodes de cette série, nous avons examiné le matériel Sigfox de base pour acquérir et transmettre des données mesurées, puis les concepts qui sous-tendent la transmission de données dans le réseau Sigfox. Nos mesures se trouvent maintenant dans l'infrastructure Sigfox et attendent d'être collectées et traitées. Il existe un outil gratuit de tableau de bord basé sur la navigation HTML. Il permet à l'utilisateur de créer et d'utiliser une interface de visualisation et de surveillance personnalisée sur n'importe quel PC, ordinateur portable, tablette ou téléphone tactile. Et, s'il est correctement activé, ce tableau de bord peut être appelé et visualisé par n'importe quel utilisateur autorisé sur l'internet, partout dans le monde. Le point de départ des développements décrits ici est le croquis Arduino Sigfox-3\_0.ino pour la carte MKR FOX 1200. Si vous choisissez le quatrième point du menu Send three readings (*envoyer trois relevés*), les trois grandeurs mesurées (pression atmosphérique, température de l'air et intensité lumineuse captée par la LDR) seront acquises en continu, les résultats étant affichés sur l'ePaper de la carte mère et transmis à l'IDE

Arduino par le moniteur série. Lorsque le bouton est pressé, ces relevés sont automatiquement envoyés sur le réseau Sigfox, reçus par l'infrastructure Sigfox et stockés. Et à partir de ce moment, nous pouvons procéder à la visualisation des données sur un tableau de bord.

### Plateforme du tableau de bord *thinger.io*

Nous utiliserons *thinger.io*, une plateforme de visualisation de l'IdO à code source ouvert qui permet la construction rapide et simple d'affichages de données clairs et attrayants. Elle peut même être utilisée gratuitement pour des projets plus simples. Pour l'instant, le système ne prend pas en charge les opérations mathématiques sur les données reçues, ce qui signifie que les relevés doivent lui être fournis sous une forme traitée et prête à être affichée. Ce traitement préalable peut être effectué facilement par notre logiciel Arduino sur la carte MKR FOX 1200. Le modèle de visualisation des valeurs des données est basé sur un tableau de bord à la disposition de toute personne disposant d'un navigateur web.

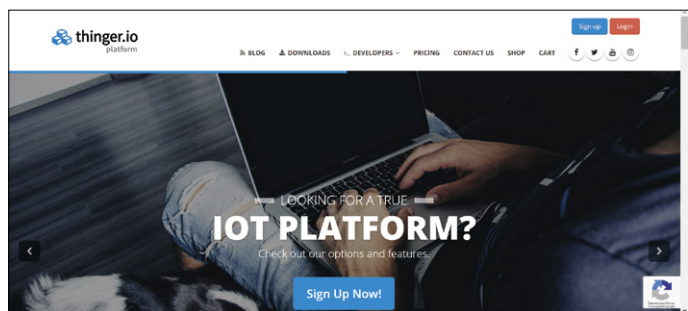


Figure 1. Page d'accueil de *thinger.io*.

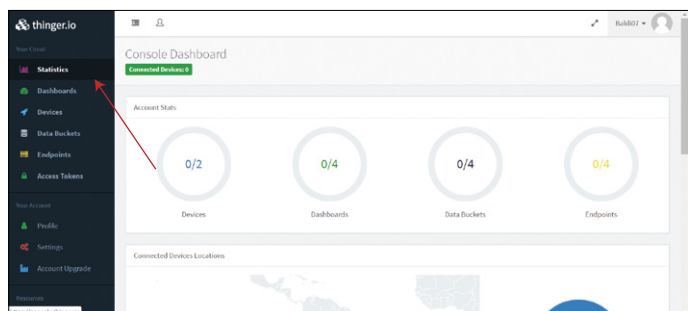


Figure 3. Écran principal du compte affichant les statistiques.

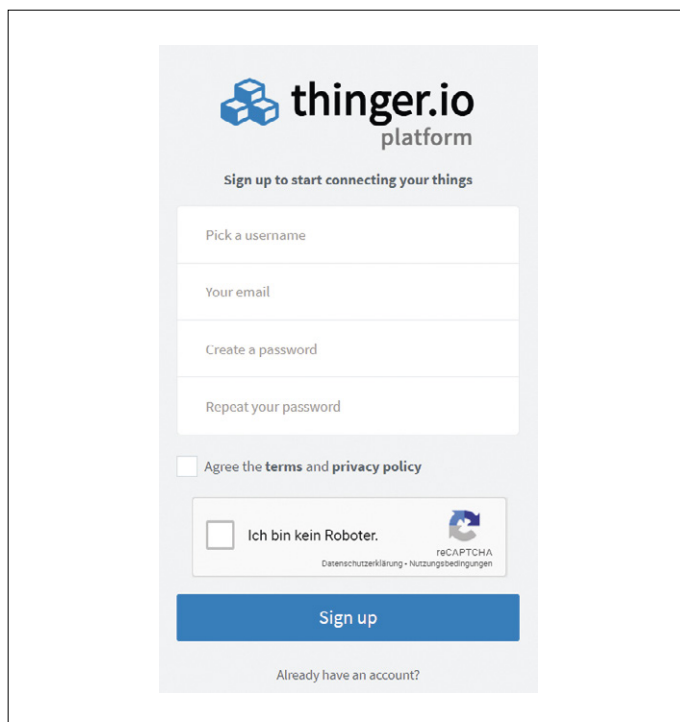


Figure 2. Création d'un nouveau compte.

### Pas à pas

Quelques étapes simples sont nécessaires pour créer un tableau de bord personnalisé comme suit.

- Créez un compte d'utilisateur gratuit sur *thinger.io*.
- Créez un «bac de données» avec *thinger.io* pour accumuler les relevés envoyés par l'infrastructure Sigfox.
- Créez un point d'accès et un jeton avec *thinger.io*, et avec celui-ci, configurez l'infrastructure Sigfox de sorte qu'elle dispose des identifiants nécessaires pour transférer des données à *thinger.io*.
- Créez un système de rappel sur l'infrastructure de Sigfox pour envoyer des données sur l'internet depuis l'infrastructure vers le bac de données de *thinger.io*.
- Dessinez sur le site *thinger.io* un tableau de bord graphique adapté pour visualiser les données.

### Création d'un compte utilisateur sur *thinger.io*

Un compte d'utilisateur (gratuit) doit être créé sur *thinger.io* pour pouvoir utiliser ses services. Pour ce faire, rendez-vous sur la page d'accueil de *thinger.io* [1] (**fig. 1**) et cliquez sur *Sign up* en haut à droite. Vous accéderez alors au formulaire d'enregistrement d'un nouvel utilisateur (**fig. 2**). Remplissez les champs du

formulaire et cliquez sur *Sign up*.

Les services offerts aux utilisateurs disposant d'un compte gratuit sont quelque peu limités, mais néanmoins plus que suffisants pour la plupart des petits projets expérimentaux. Vous trouverez tous les détails sous l'onglet *Pricing*.

Une fois votre adresse électronique confirmée, vous pouvez

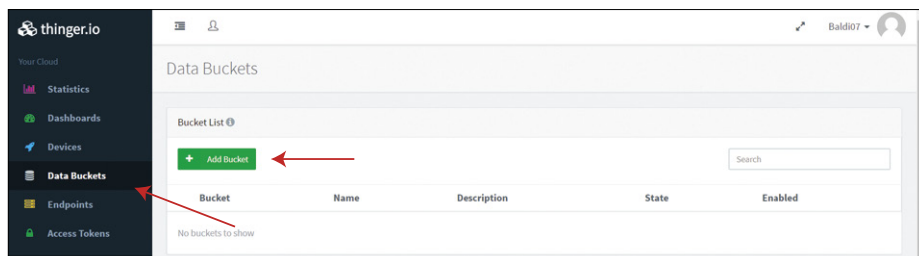


Figure 4. Mise en place d'un nouveau bac de données.

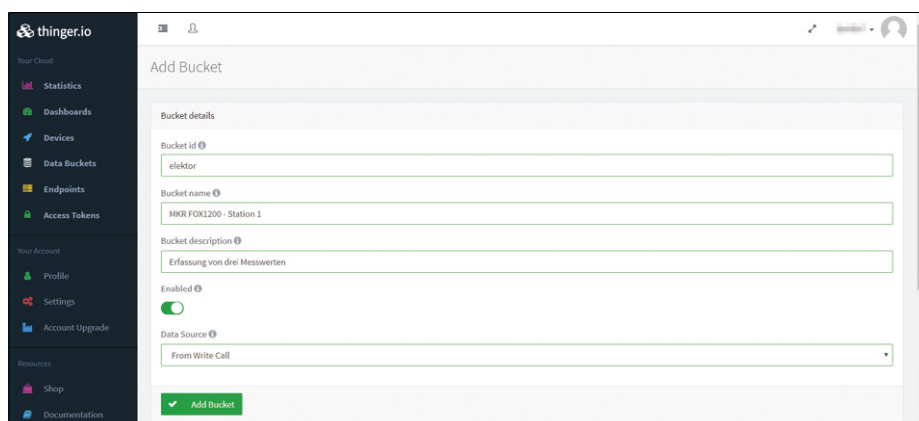


Figure 5. Spécification des propriétés du bac de données.

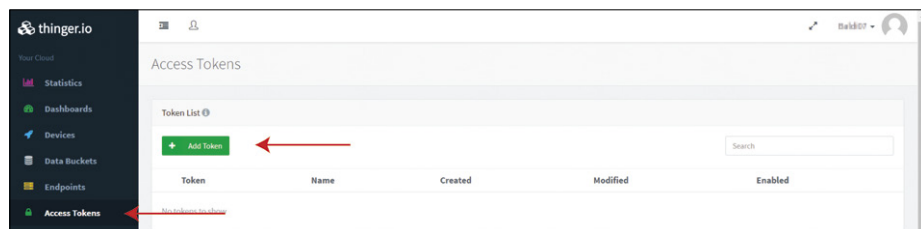


Figure 6. Mise en place d'un nouveau jeton (point d'accès).

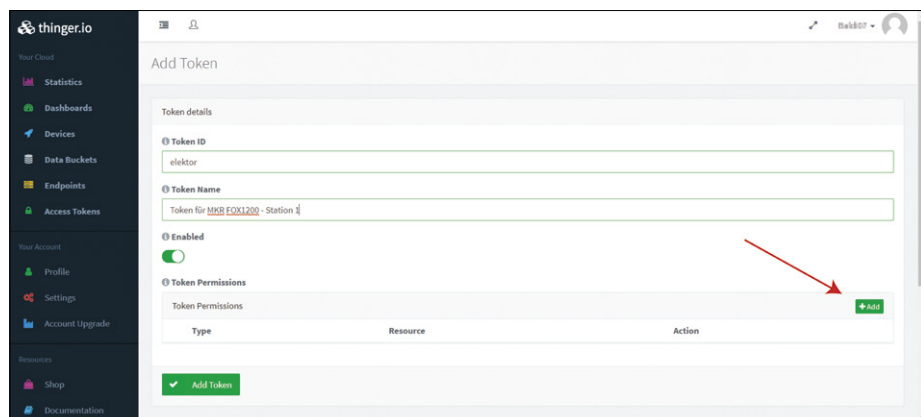


Figure 7. Spécification des propriétés du jeton.

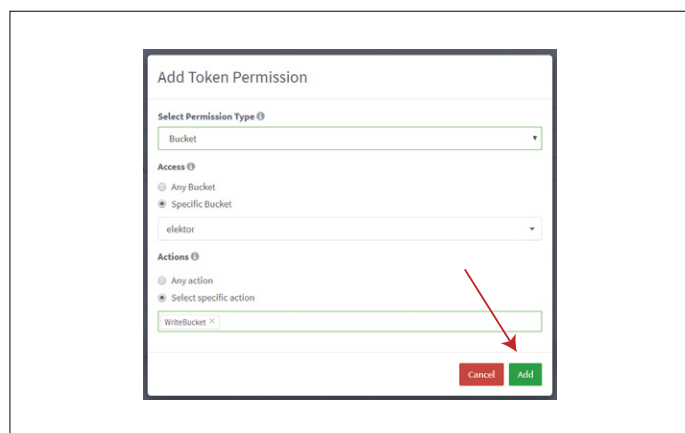


Figure 8. Configuration des autorisations de jetons.

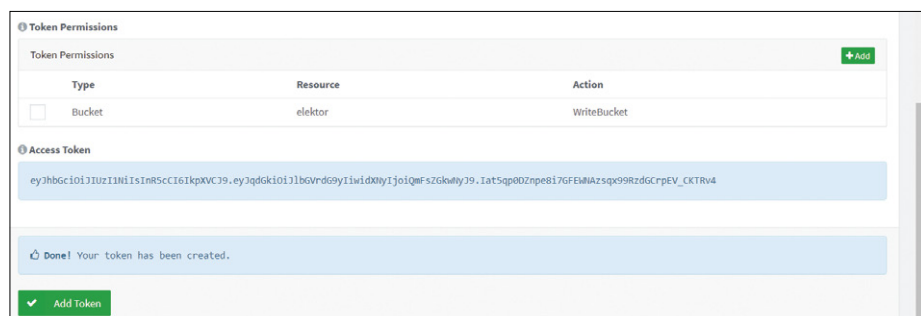


Figure 9. Le jeton est créé et activé avec les autorisations appropriées.

vous rendre sur la page de connexion et vous connecter. L'écran principal de votre compte s'appelle *Statistics* et c'est le point de départ (fig. 3). Le menu principal se trouve sur le côté gauche et contient toutes les fonctions nécessaires pour créer un tableau de bord pour votre projet IdO.

### Création d'un bac de données chez *thinger.io*

Ensuite nous avons créé un **bac** (ou panier) **de données**, c'est-à-dire une base de données dans laquelle les relevés peuvent être stockés, et à partir de laquelle elles peuvent être relues. L'infrastructure Sigfox écrira automatiquement les valeurs mesurées dans cette base de données en utilisant le rappel (*callback*) Sigfox. Sur l'écran principal, cliquez sur *Data Buckets* sur la gauche, puis sur *Add Bucket* (ajouter un bac) (fig. 4). Une fenêtre apparaîtra pour vous permettre de configurer les caractéristiques du bac de données (fig. 5). L'identifiant *Bucket id* doit être unique pour l'ensemble du compte et sera utilisé ultérieurement lors de la configuration du bac pour fonctionner avec

Sigfox. Dans cet exemple, nous entrons «elektor». Les champs *Bucket name* et *Bucket description* peuvent contenir n'importe quel texte utile à l'utilisateur. Utilisez le menu déroulant dans le champ de sélection *Data source* (source de données) pour définir sa configuration sur *From Write Call* (à partir de l'appel d'écriture). Cela signifie que les données seront envoyées à *thinger.io* sur Internet à partir de l'infrastructure Sigfox en utilisant le rappel Sigfox. Dès que le rappel envoie les valeurs des données, elles seront stockées ici dans le bac. Enfin, cliquez sur *Add bucket* et le bac sera créé.

### Mise en place d'un point d'accès chez *thinger.io* pour recevoir les données de l'infrastructure

L'infrastructure Sigfox doit bien sûr savoir où les mesures qu'il a acquises doivent être envoyées grâce à son rappel : en d'autres termes, nous devons indiquer à l'infrastructure Sigfox l'adresse du destinataire sur *thinger.io*. Cette étape se fait en créant un jeton avec les informations d'identification appropriées pour permettre l'accès au bac de données. Dans la fenêtre principale, cliquez sur *Access Tokens*, puis sur *Add Token* (fig. 6). Une fenêtre apparaîtra dans laquelle vous pourrez entrer les propriétés du jeton (fig. 7). Nous commençons par définir l'ID du jeton : pour cet exemple, nous choisissons «elektor». Sous la rubrique *Token Name*, vous pouvez ajouter une description du jeton en texte libre ou simplement un nom.

Cliquez ensuite sur *Add* (le petit bouton vert en bas à droite) et une autorisation de jeton (c'est-à-dire les informations d'identification pour accéder au bac de données) sera ajoutée. Dans l'étape suivante, vous pourrez configurer cela en détail (fig. 8).

Dans le champ *Select Permission Type*, choisissez *Bucket*. Dans le champ *Access*, cliquez sur *Specific Bucket* et choisissez ensuite «elektor» (l'identifiant que nous avons configuré) dans le menu déroulant. Sous *Actions*, choisissez *Select specific action* et dans le champ en dessous, choisissez *WriteBucket* dans le menu déroulant. Enfin, cliquez sur le bouton *Add* : cela va générer un jeton pour le bac cible que vous avez sélectionné (avec l'identifiant «elektor») et l'ajouter à la liste des permissions (fig. 7). Cliquez maintenant sur *Add Token* (fig. 7, en bas à gauche) et ce point d'accès, qui sera l'adresse de réception des messages de l'infrastructure Sigfox, sera créé dans *thinger.io* (fig. 9).

Notez que le champ bleu *Access Token*, qui contient les informations d'authentification générées par *thinger.io*, sera nécessaire pour l'infrastructure Sigfox. Celle-ci devra être configurée de manière à envoyer ce code d'accès à l'adresse de destination de *thinger.io* en même temps que les données utiles : cela garantira que les données utiles envoyées seront acceptées par *thinger.io* et correctement dirigées vers le bac de données «elektor». Il est donc nécessaire de copier-coller ce code d'accès et de le stocker en lieu sûr. Fin de la configuration des paramètres sur *thinger.io*. Retournons à l'infrastructure Sigfox.

## Configuration du 'rappel' dans l'infrastructure Sigfox

Nous voulons faire en sorte que lorsque le MKR FOX 1200 envoie des données sur l'infrastructure Sigfox, les informations soient automatiquement transmises à *thinger.io* pour traitement et visualisation ultérieurs. Pour ce faire, nous utilisons une fonction de l'infrastructure Sigfox déjà mentionnée, sous le nom de *callback*. Un rappel n'est rien d'autre qu'un transfert automatique de données exécuté chaque fois que l'infrastructure Sigfox reçoit des données d'une station Sigfox telle que notre MKR FOX 1200. Pour permettre ce transfert, l'infrastructure doit bien sûr être correctement configurée, en s'assurant que les données sont envoyées au destinataire souhaité.

Afin de transférer les données utiles dans notre bac de données «elektor» sur *thinger.io*, nous devons utiliser notre jeton et le code d'accès que *thinger.io* a générés pour nous lors de l'étape

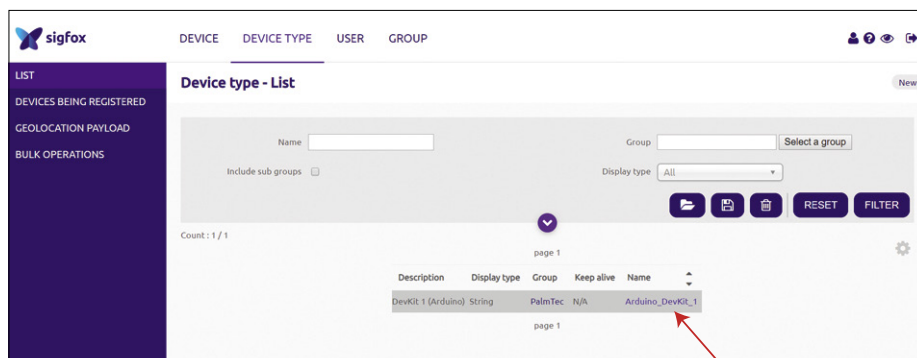


Figure 10. Choix du type d'appareil souhaité.

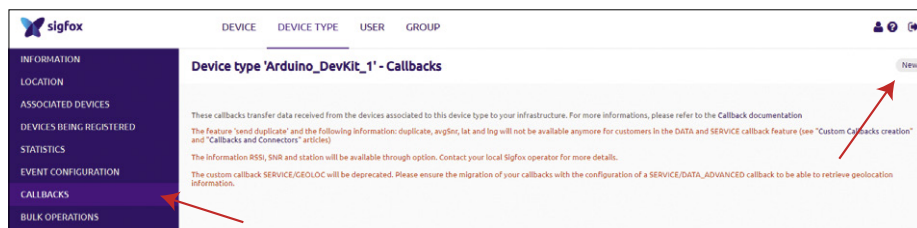


Figure 11. La fenêtre de rappel.

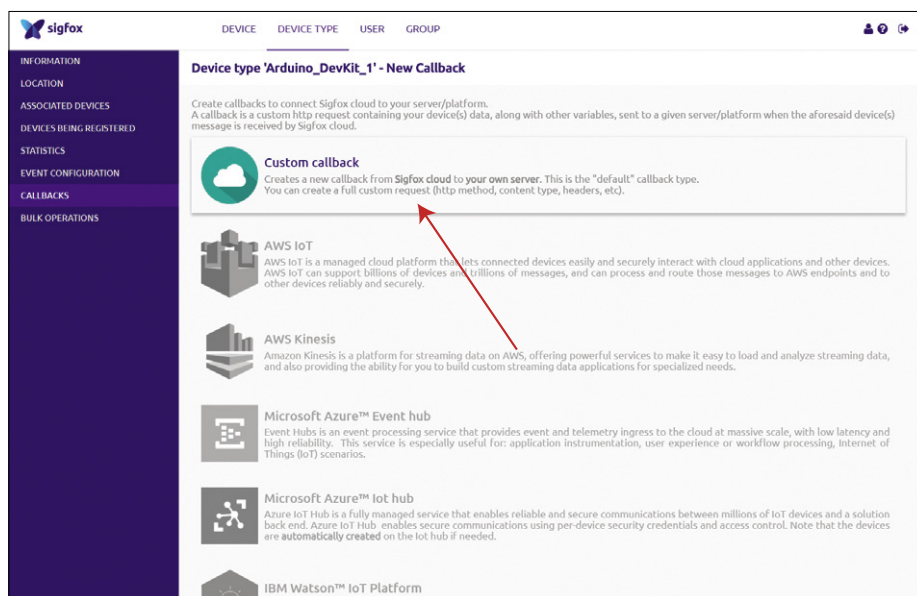


Figure 12. Il existe un vaste choix de types de rappel.

précédente. En d'autres termes, l'ordinateur central établira une connexion HTTP sur Internet pour accéder au jeton, puis utilisera le code du jeton d'accès pour s'authentifier. Une fois authentifié, il peut écrire des données dans le bac. Pour configurer le rappel, il faut d'abord se connecter à l'infrastructure Sigfox [2]. Sur la page principale, cliquez sur l'onglet *Device Type* (type d'appareil) et dans la liste qui apparaît alors (elle ne comportera qu'une seule entrée dans ce cas), cliquez sur le nom du type d'appareil *Arduino\_DevKit\_1* (fig. 10). La page d'aperçu de cet appareil apparaîtra alors. Dans la liste de gauche, cliquez sur *Callbacks* (fig. 11). Si vous cliquez maintenant sur le petit bouton *New* en haut à droite de



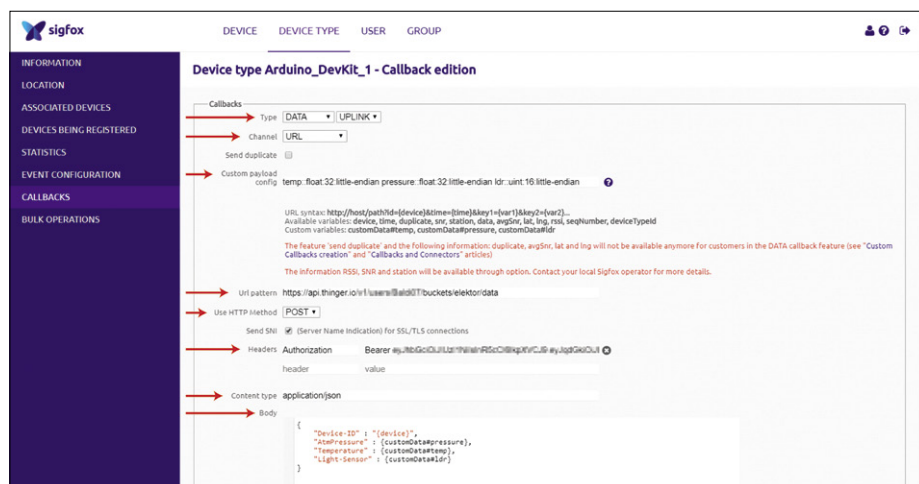


Figure 13. Configuration du rappel.

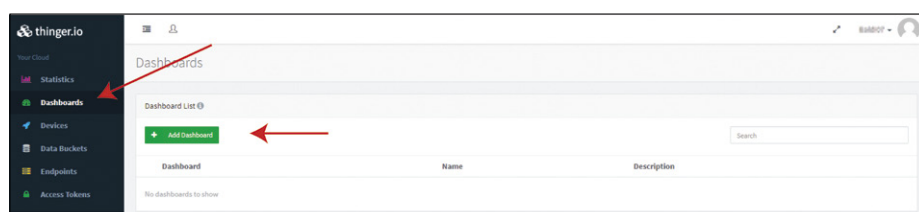


Figure 14. Début de la création d'un tableau de bord.

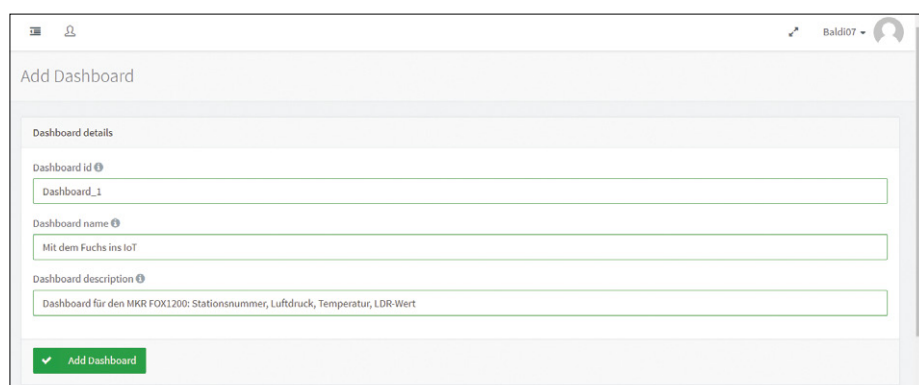


Figure 15. Paramètres de base du tableau de bord.

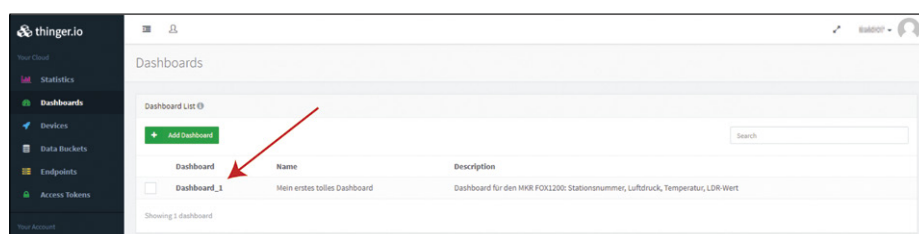


Figure 16. Liste des tableaux de bord disponibles.

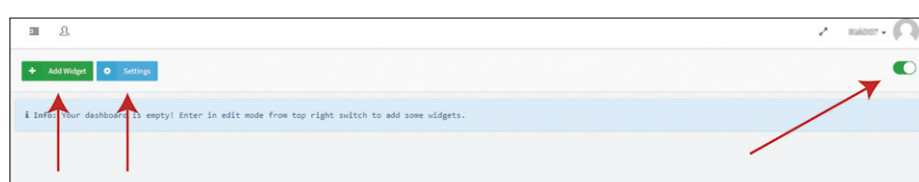


Figure 17. Le tableau de bord en mode édition.

cette fenêtre, une liste apparaîtra avec tous les types de rappel disponibles qui fonctionnent avec les programmes de tableau de bord et de cloud courants (**fig. 12**). Parmi ceux-ci, choisissez *Custom callback* (rappel personnalisé). Dans la fenêtre qui apparaît maintenant, vous pourrez configurer le rappel et la manière dont les données sont envoyées de l'infrastructure Sigfox à *thinger.io* (**fig. 13**).

Veillez à ce que tout ce que vous saisissez sur cette page apparaisse **exactement** comme décrit ci-dessous : la moindre co(q)uille entraînera des messages d'erreur et le système entier cafouillera.

Une description détaillée du fonctionnement des différents cadres dépasse le cadre de cet article – voir [3], [4] et [5]. Commençons par le haut de la figure 13. Sous *Type*, les paramètres corrects sont DATA et UPLINK, et pour *Channel*, choisissez URL, car c'est par une connexion HTTP que nous allons parler à *thinger.io*.

Maintenant, les choses se compliquent un peu. Les valeurs de données envoyées par le MKR FOX 1200 dans la charge utile vers l'infrastructure Sigfox doivent être décomposées en valeurs individuelles au sein de l'infrastructure Sigfox. En interne, un nom de variable librement choisi est attribué à chaque valeur. Ces noms sont utilisés pour spécifier le traitement ultérieur que l'infrastructure effectue sur les valeurs des variables. La manière dont la charge utile Sigfox est déballée est spécifiée dans le champ *Custom payload* (charge utile personnalisée) et doit refléter la manière dont les valeurs des données sont incorporées dans la charge utile sur le MKR FOX 1200. Pour que la charge utile Sigfox soit décomposée correctement en variables, il faut en spécifier précisément la séquence des données, les types de données concernés et le format de stockage.

Dans notre exemple, la *Custom payload config* devrait contenir trois éléments, chacun apparaissant **précisément** comme suit.

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

a première donnée de la charge utile envoyée par le MKR FOX 1200 est donc la température. Dans l'infrastructure

Sigfox, cette valeur porte le nom de «temp» et le type de données est «float», sur 4 octets (32 bits).

La deuxième valeur des données de la charge utile est la pression atmosphérique, également une variable flottante codée sur 32 bits.

La troisième et dernière valeur est la tension de la LDR. Dans l'infrastructure, elle porte le nom de «ldr» et est une variable de type «uint» stockée sur 2 octets (16 bits).

Le terme *little-endian* précise l'ordre des octets utilisés pour stocker une valeur donnée : *little-endian* est l'ordre utilisé par notre µC Atmel [6]. Vous devez entrer les trois éléments ci-dessus, concaténés sur une seule ligne et séparés par des espaces, dans le champ de configuration de la charge utile personnalisée. Vous pouvez maintenant configurer l'infrastructure Sigfox pour traiter ces 3 variables, ou bien vous pouvez les envoyer directement à *thinger.io* sous cette forme.

Sous *URL pattern*, vous spécifiez l'URL de destination sur *thinger.io*. L'URL est construite de la manière suivante.

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

Dans l'adresse, remplacez *user\_id* par le nom d'utilisateur de votre compte *thinger.io* (fig. 2) et *bucket\_id* par l'identifiant du bac, ici «elektor» (fig. 5), dans les deux cas sans parenthèses ni guillemets.

Comme méthode HTTP, choisissez *POST*.

L'étape suivante consiste à mettre en place une authentification pour le transfert de données vers *thinger.io*. Cela se fait dans les deux champs situés à côté des *headers* (en-têtes). Dans le champ *header* entrez le mot *Authorization* et, à côté, dans le champ *value*, entrez *Bearer* suivi d'un espace et du jeton d'accès que nous avons créé (fig. 9).

```
Bearer eyJhbGciOiJI1NiIsInR5cCI6IkpXZWQ... etc.
```

Dans le champ *Content type*, entrez «application/json», ce qui indique que le corps du message est au format JSON.

Dans la dernière étape, nous précisons exactement ce qui sera envoyé à *thinger.io*. La saisie (au format JSON) dans le champ *Body* doit être la suivante. Là encore, il est essentiel que le texte apparaisse **exactement** comme indiqué.

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

Le *Body* (corps) est entouré de parenthèses («{», «}»). Mais que signifie le texte dans le corps ?

Chaque ligne entre les accolades commence par des guillemets ; ensuite apparaît un nom de variable ad libitum (mais unique), suivi de nouveaux guillemets. Suivent un espace, un double-point et un autre espace. C'est là que la valeur de la variable sera insérée par l'infrastructure Sigfox. Le nom de la variable et sa valeur seront ensuite envoyés sous ce format à *thinger.io*, où ils seront stockés dans le bac de données approprié.

Nous avons donc

- "Device-ID" : "{device}" qui donnera le numéro d'appareil unique de la station Sigfox (notre MKR FOX 1200), qui est connu de l'infrastructure. Cette information est envoyée avec les données mesurées afin que la station

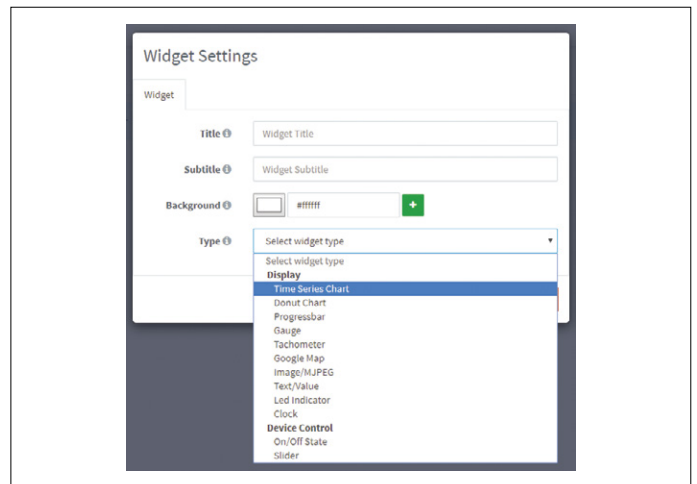


Figure 18. Première fenêtre pour la création d'un widget.

d'origine puisse être identifiée avec certitude par la suite.

- La ligne «AtmPressure» : {customData#pressure} envoie une variable avec le nom *AtmPressure*. La valeur de cette variable correspond à la valeur de la variable d'infrastructure appelée *pression*, que nous avons précédemment spécifiée dans le champ *Custom payload config*. Le mot-clé *customData#* indique que la valeur d'une variable déballée à l'aide du champ *Custom payload config* doit être insérée. C'est ainsi que les noms et les valeurs des variables sont reformatés pour être transmis à *thinger.io*.
- Les deux dernières lignes sont traitées de manière analogue. Chaque ligne, à l'exception de la dernière, doit être terminée par une virgule.

Cela complète la configuration du rappel dans l'infrastructure de Sigfox pour lui permettre d'envoyer les données souhaitées au bac de données «elektor» de *thinger.io*. Lorsque vous avez correctement effectué toutes les étapes ci-dessus, vous pouvez cliquer sur OK.

Désormais, lorsque le MKR FOX 1200 envoie une charge utile à l'infrastructure Sigfox, et en supposant qu'aucune erreur n'est rencontrée à la réception, l'infrastructure de rappel Sigfox transmet immédiatement les données au format spécifié ci-dessus à *thinger.io* pour traitement ultérieur.

### Tableau de bord chez *thinger.io*

Nous voulons présenter les données sous forme graphique sur un tableau de bord. Pour le créer, allez dans la console principale de *thinger.io* (fig. 3) et cliquez à gauche sur *Dashboards* et dans l'écran suivant sur *Add Dashboard* (fig. 14).

Remplissez les champs de la fenêtre *Dashboard details* (fig. 15). *Dashboard id* est un identifiant unique pour l'ensemble du compte. Les champs *Dashboard name* et *Dashboard description* peuvent contenir n'importe quelle information, pourvu qu'elle soit lisible par l'utilisateur. Confirmez les entrées en cliquant sur *Add Dashboard*. Le nouveau tableau de bord sera ajouté à la liste des tableaux de bord que vous verrez chaque fois que vous cliquerez sur l'élément de menu *Dashboards* dans la fenêtre principale de la console (fig. 16).

Si vous cliquez sur le nom du tableau de bord, une fenêtre apparaîtra avec un tableau de bord vide (fig. 17). En haut à

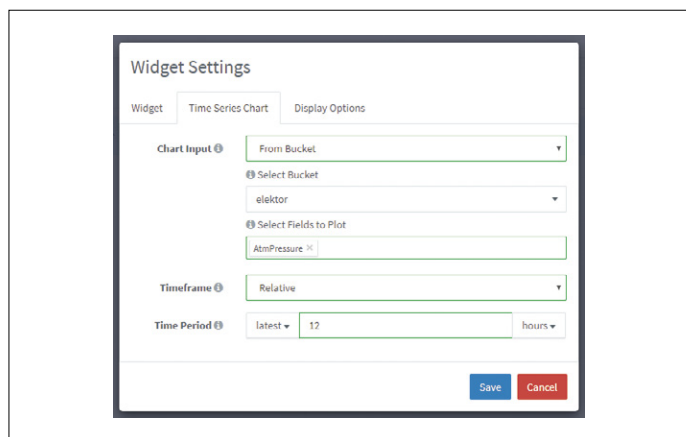


Figure 19. Configuration d'un graphique de séries chronologiques.

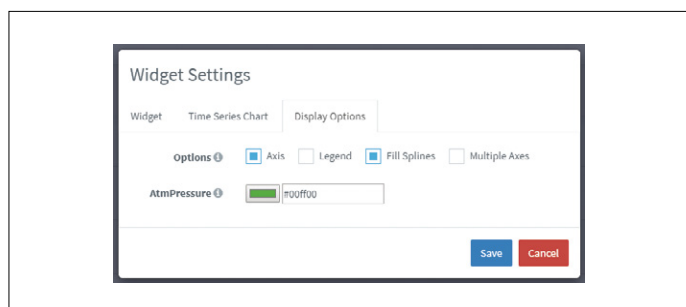


Figure 20. Réglage des caractéristiques de base de l'affichage.

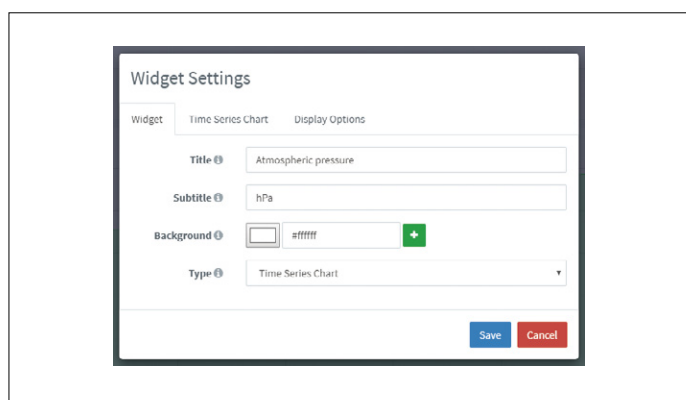


Figure 21. Paramètres dans l'onglet Widget.

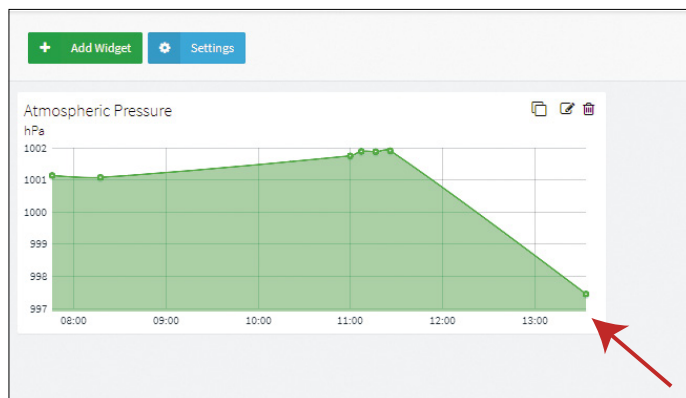


Figure 22. Le graphique des séries chronologiques ajouté au tableau de bord : quelques relevés sont déjà tracés.

droite, il y a un petit interrupteur : faites-le glisser vers le vert pour passer en mode édition. Il y a également deux boutons *Add Widget* et *Settings*. Dans *thinger.io*, les éléments d'affichage du tableau de bord qui font partie de l'interface utilisateur sont appelés *widgets*. Un large éventail de *widgets* est disponible, notamment des graphiques de séries chronologiques, des graphiques en beignet, des barres de progression, des affichages analogiques, des cartes Google avec des informations sur la position, des images, des affichages de texte, des DEL et une horloge. Dans notre application, nous souhaitons afficher un graphique chronologique pour chacune des trois valeurs mesurées et un affichage numérique pour chacune d'entre elles indiquant la valeur la plus récemment reçue.

Cliquez sur *Add Widget* et une fenêtre intitulée *Widget Settings* apparaîtra (**fig. 18**). Sous *Type*, vous pouvez choisir le type de *widget* souhaité. Dans ce cas, nous choisissons *Time Series Chart*. Deux autres onglets apparaîtront (**fig. 19**). Dans l'onglet *Time Series Chart*, sous *Chart Input*, choisissez l'option *From Bucket*. Au fur et à mesure que vous avancerez, plus de champs à remplir apparaîtront. Dans le champ *Select Bucket*, choisissez le bac «elektor». Dans le champ suivant, *Select Fields to Plot*, choisissez la valeur de la donnée à afficher.

Nous commençons par *AtmPressure*. Dans la rubrique *Timeframe* (cycle), choisissez *Relative* et réglez la fenêtre de temps à afficher sur 12 h. Le graphique affichera alors les données des 12 h les plus récentes ; ajustez ce cycle comme vous le souhaitez. Cliquez maintenant sur l'onglet *Display Options* (**fig. 20**) et désactivez l'option *Multiple Axes* : nous allons établir un graphique distinct pour chacune de nos variables, et nous n'avons donc pas besoin de plusieurs axes Y sur le même graphique. L'option *Axes* (déjà activée) permet de mettre automatiquement à l'échelle les axes X et Y du graphique. Dans le champ voisin de l'étiquette *AtmPressure*, nous pouvons choisir la couleur du graphique : un vert apaisant. Si vous activez l'option *Fill Splines*, la région entre le graphique et l'axe des X aura la couleur sélectionnée à côté de l'étiquette *Pression atmosphérique*. L'option *Legend* permet d'étiqueter séparément des séries chronologiques différentes tracées sur le même graphique.

Enfin, nous revenons à l'onglet *Widget* (**fig. 21**), où nous pouvons maintenant définir le titre et le sous-titre du graphique. Le titre est *Atmospheric pressure* et le sous-titre est l'unité de mesure : hPa. Le paramètre *Background* permet d'ajuster la couleur de fond du graphique. Nous terminons le paramétrage du graphique en sauvegardant : le graphique sera ajouté au tableau de bord (**fig. 22**). Vous pouvez agrandir ou réduire le graphique en cliquant dans le coin inférieur droit. Vous pouvez également faire glisser le graphique dans la zone du tableau de bord. Le changement de taille graphique ne se passe pas toujours comme on veut.

Nous voulons également afficher la lecture actuelle de la pression atmosphérique sous forme numérique (**fig. 23**). Le bouton *Add Widget* ouvre à nouveau le menu permettant d'ajouter un nouvel élément à l'affichage ; dans ce cas, dans le champ *Type*, nous sélectionnons *Text/Value*.

Dans le champ *Widget Value*, nous sélectionnons à nouveau *From Bucket*, choisissons le bac «elektor» et précisons à nouveau la valeur que nous voulons afficher (ici «AtmPressure»). Pour la *Bucket Value*, nous définissons *Latest Value*. L'onglet *Display Options* nous permet d'effectuer divers réglages de configuration. Par exemple, nous pouvons définir la précision

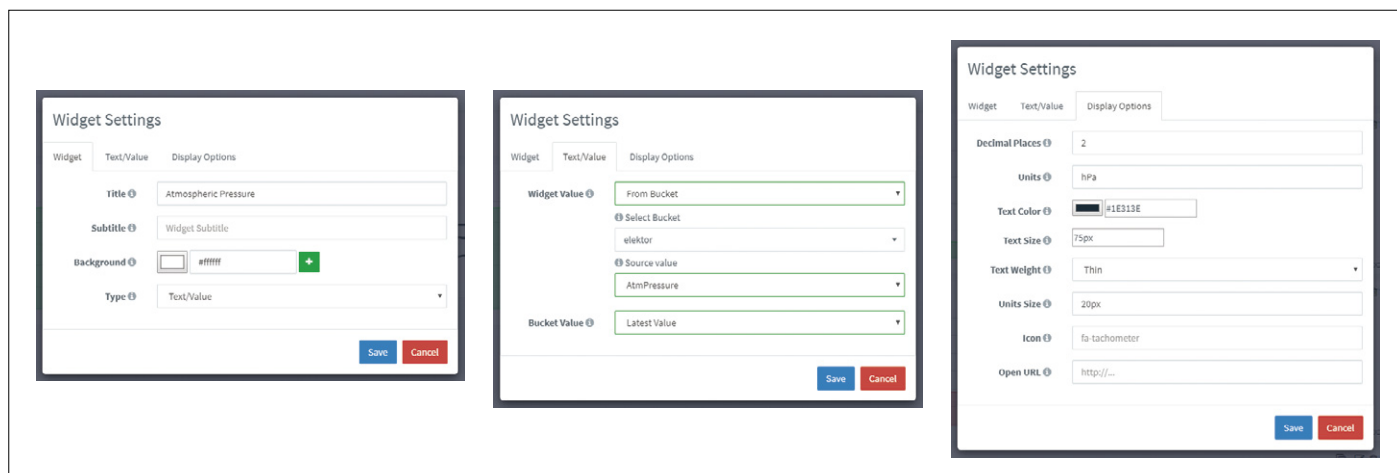


Figure 23. Configuration de l'affichage numérique pour la lecture de la pression atmosphérique.

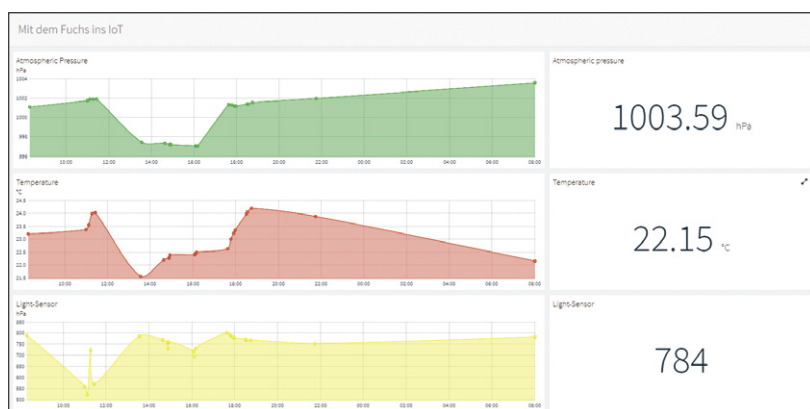


Figure 24. Le tableau de bord achevé.

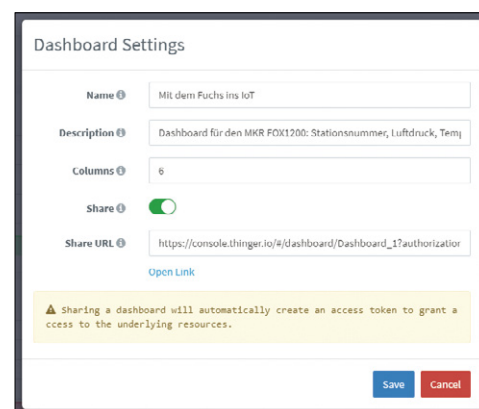


Figure 25. Paramètres généraux du tableau de bord.

avec laquelle la valeur est affichée (nombre de décimales) et nous pouvons définir 'hPa' pour le champ *Unités*. Cette chaîne sera affichée dans le *widget* à côté de la valeur mesurée. Il existe encore d'autres options. Dans l'onglet *Widget*, nous avons à nouveau défini *Atmospheric pressure* pour le champ *Title*. Nous pouvons laisser vide le sous-titre.

L'affichage des deux autres relevés est configuré de manière

similaire (**fig. 24**). Pour être sûr que tout est bien enregistré dans *thinger.io*, quittez le mode édition et retournez-y... D'autres d'informations sur les tableaux de bord sur *thinger.io* se trouvent ici [7].

Il ne nous reste plus qu'à permettre l'accès au tableau de bord «depuis l'extérieur». De retour dans le menu du tableau de bord, cliquez sur le bouton bleu intitulé *Settings*, qui vous amènera

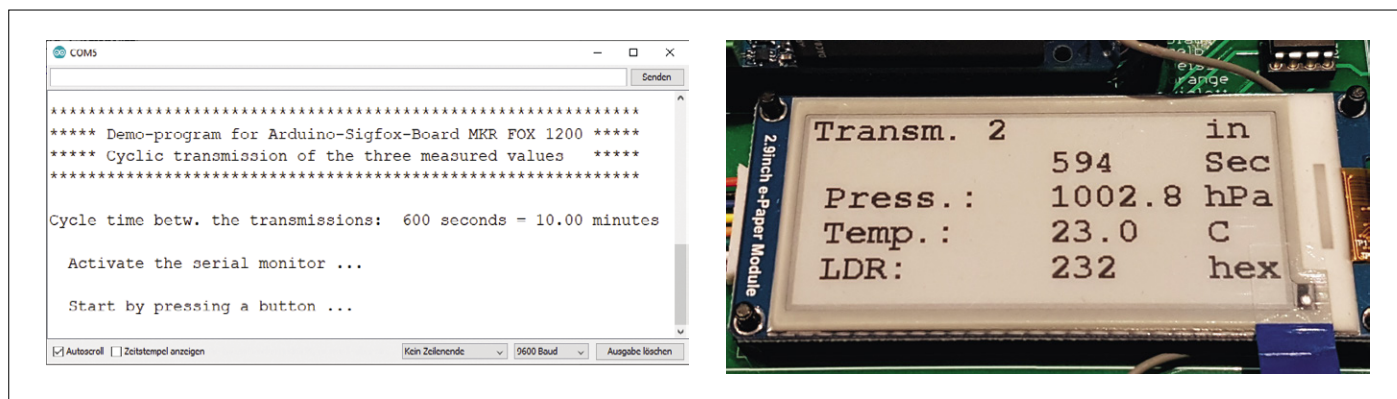


Figure 26. Sortie de Sigfox-Demo.ino sur le moniteur série et sur l'afficheur ePaper.





Figure 27. Le tableau de bord présente bien sur un téléphone !

à la page des paramètres du tableau de bord (**fig. 25**). Vous pouvez y donner un nom et une description à votre tableau de bord (ce n'est pas nécessaire dans ce cas, nous l'avons déjà configuré). Le bouton *Share* permet d'activer le tableau de bord pour un accès externe : le lien (assez long) nécessaire pour y accéder apparaîtra dans le champ «Partager l'URL». Grâce à ce lien, il est désormais possible d'accéder au tableau de bord depuis n'importe quel endroit dans le monde et d'examiner les données qu'il affiche. En raison de la longueur de l'URL, il est recommandé d'utiliser un raccourci d'URL (bitly.com) pour l'abrégé. L'URL raccourcie peut ensuite être convertie en un code graphique QR qui peut être scanné par un téléphone : cela évite la saisie de l'URL longue.

Lorsque sur *thinger.io* vous autorisez l'accès public au tableau de bord, celui-ci est automatiquement protégé en écriture. Les tiers ne peuvent pas modifier les paramètres ou effectuer d'autres ajustements : cela n'est possible qu'après avoir utilisé les informations d'identification appropriées pour se connecter au portail *thinger.io*.

Nous souhaitons tester le système dans son ensemble, mais pas avec le sketch *Sigfox-3\_0.ino*, car il faut appuyer sur un bouton chaque fois que nous voulons envoyer un message. Il existe un autre sketch, appelé *Sigfox-Demo.ino* [8]. Toutes les dix minutes, ce programme acquiert automatiquement les trois relevés et les transmet (**fig. 26**). L'intervalle entre les transmissions est déterminé par la variable globale *w\_zeit*. Réglez-la comme vous le souhaitez pour acquérir et afficher les lectures sur une longue période en n'oubliant pas que vous n'êtes autorisé à envoyer que 140 messages par jour sur le

réseau Sigfox, soit environ un message toutes les 10 min. Il n'est pas si critique qu'un compte gratuit chez *thinger.io* ne vous permette d'écrire des données dans un bac plus souvent qu'une fois par minute. Que dites-vous de la **fig. 27** avec le tableau de bord sur l'écran d'un téléphone ?

### En conclusion

Voici donc la fin de cette série d'articles de présentation du réseau Sigfox 0G. La programmation très flexible et polyvalente de l'Arduino MKR FOX 1200 facilite permet l'acquisition de types de données très variées, puis leur transmission sur le réseau Sigfox dans le monde entier. Parallèlement, grâce aux possibilités d'affichage simples, mais puissantes offertes par *thinger.io*, la programmation de tableaux de bord personnalisés pour la communication d'informations aux utilisateurs finaux est rapide et efficace.

Bonne collecte de données dans le monde passionnant de l'Internet des objets ! ◀

190281-D-03

**@ WWW.ELEKTOR.FR**

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

→ Arduino Antenne 868 MHz  
www.elektor.fr/19095

### Liens

- [1] Thinger: <http://www.thinger.io/>
- [2] Infrastructure Sigfox : <https://backend.sigfox.com/>
- [3] Infos sur la configuration de Callback: <https://support.sigfox.com/docs/callbacks-documentation>
- [4] Infos sur la configuration de Callback: <https://support.sigfox.com/docs/custom-callback-creation>
- [5] Infos sur la configuration de Callback: <https://backend.sigfox.com/apidocs/callback>
- [6] Ordre des octets dans le codage des nombres : <https://de.wikipedia.org/wiki/Byte-Reihenfolge>
- [7] Infos sur la création du tableau de bord : <https://github.com/thinger-io/Docs/blob/gh-pages/console/dashboards.md>
- [8] La page de ce projet : <https://www.elektor.fr/190281-D-03>